

# **Altova Authentic 2023 Desktop**



**Manual del usuario y referencia**

## **Altova Authentic 2023 Desktop 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

<b>1</b>	<b>Acerca de Authentic Desktop y su documentación</b>	<b>11</b>
1.1	Guía del usuario y de referencia.....	12
1.2	About This Documentation.....	13
<b>2</b>	<b>Interfaz y entorno</b>	<b>14</b>
2.1	La interfaz gráfica de usuario (IGU).....	15
2.1.1	Ventana principal.....	16
2.1.2	Ventana Proyecto.....	17
2.1.3	Ventana Información.....	19
2.1.4	Ayudantes de entrada.....	19
2.1.5	Ventana de resultados: Mensajes.....	20
2.1.6	Barra de menú, barras de herramientas y barra de estado.....	20
2.2	El entorno de aplicación.....	22
2.2.1	Configuración y personalización.....	22
2.2.2	Tutoriales, proyectos y ejemplos.....	22
2.2.3	Características y archivos de ayuda.....	23
<b>3</b>	<b>Interfaz y entorno</b>	<b>24</b>
3.1	La interfaz gráfica de usuario (IGU).....	25
3.1.1	Ventana principal.....	26
3.1.2	Ventana Proyecto.....	27
3.1.3	Ventana Información.....	29
3.1.4	Ayudantes de entrada.....	29
3.1.5	Ventana de resultados: Mensajes.....	30
3.1.6	Barra de menú, barras de herramientas y barra de estado.....	30
3.2	El entorno de aplicación.....	32
3.2.1	Configuración y personalización.....	32
3.2.2	Tutoriales, proyectos y ejemplos.....	33

---

3.2.3	Características y archivos de ayuda.....	33
<b>4</b>	<b>Tutorial de la vista Authentic</b>	<b>35</b>
4.1	Abrir un documento en la vista Authentic.....	37
4.2	La interfaz de la vista Authentic.....	39
4.3	Operaciones con nodos.....	42
4.4	Introducir datos en la vista Authentic.....	45
4.5	Introducir valores de atributo.....	48
4.6	Añadir entidades.....	49
4.7	Imprimir el documento.....	50
<b>5</b>	<b>Interfaz de la vista Authentic</b>	<b>51</b>
5.1	La interfaz gráfica del usuario (IGU).....	52
5.2	Iconos de la barra de herramientas de la vista Authentic.....	54
5.3	Ventana principal de la vista Authentic.....	57
5.4	Ayudantes de entrada de la vista Authentic.....	60
5.5	Menús contextuales de la vista Authentic.....	65
<b>6</b>	<b>Edición en la vista Authentic</b>	<b>68</b>
6.1	Copia de seguridad automática de archivos.....	69
6.2	Edición básica.....	71
6.3	Tablas en la vista Authentic.....	76
6.3.1	Tablas SPS.....	76
6.3.2	Tablas CALS/HTML.....	77
6.3.3	Iconos de edición para tablas CALS/HTML.....	81
6.4	Editar una BD.....	84
6.4.1	Navegar por una tabla de BD.....	84
6.4.2	Consultas de BD.....	85
6.4.3	Modificar una tabla de BD.....	89
6.5	Trabajar con fechas.....	91
6.5.1	Selector de fecha.....	91
6.5.2	Entrada de texto.....	92
6.6	Definir entidades.....	94

---

6.7	Firmas XML.....	96
6.8	Imágenes en la Authentic View.....	98
6.9	Teclas de acceso rápido en la Authentic View.....	99
<b>7</b>	<b>Scripting en Authentic</b>	<b>100</b>
<b>8</b>	<b>Vista Explorador</b>	<b>103</b>
<b>9</b>	<b>Recursos globales de Altova</b>	<b>105</b>
9.1	Definir recursos globales.....	106
9.1.1	Archivos.....	108
9.1.2	Carpetas.....	113
9.1.3	Bases de datos.....	114
9.2	Usar recursos globales.....	117
9.2.1	Asignar archivos y carpetas.....	117
9.2.2	Cambiar de configuración.....	120
<b>10</b>	<b>Control de código fuente</b>	<b>121</b>
10.1	Configurar el control de código fuente.....	123
10.2	Sistemas de control de código fuente compatibles.....	124
10.3	Carpeta de trabajo local.....	126
10.4	Proyecto de la aplicación.....	127
10.5	Agregar al control de código fuente.....	129
10.6	Trabajar con control de código fuente.....	132
10.6.1	Agregar o quitar del control de código fuente.....	132
10.6.2	Proteger, desproteger.....	133
10.6.3	Obtener archivos como archivos de solo lectura.....	136
10.6.4	Copiar y compartir desde el control de código fuente.....	137
10.6.5	Cambiar control de código fuente.....	140
10.7	Control de código fuente con Git.....	142
10.7.1	Habilitar Git con el complemento de control de código fuente.....	143
10.7.2	Agregar un proyecto al control de código fuente de Git.....	143

---

10.7.3	Clonar un proyecto desde el control de código fuente de Git.....	145
<b>11</b>	<b>Gestor de esquemas</b>	<b>147</b>
11.1	Ejecutar el gestor de esquemas.....	151
11.2	Categorías de estado.....	154
11.3	Aplicar parches o instalar un esquema.....	156
11.4	Desinstalar o restaurar esquemas.....	158
11.5	Interfaz de la línea de comandos (ILC).....	159
11.5.1	help .....	159
11.5.2	info .....	160
11.5.3	initialize.....	160
11.5.4	install .....	161
11.5.5	list .....	161
11.5.6	reset .....	162
11.5.7	uninstall.....	163
11.5.8	update.....	164
11.5.9	upgrade.....	164
<b>12</b>	<b>Authentic Desktop en Visual Studio</b>	<b>166</b>
12.1	Instalar el complemento de Authentic Desktop para Visual Studio.....	167
12.2	Diferencias entre la versión independiente y la versión para Visual Studio.....	168
<b>13</b>	<b>Authentic Desktop en Eclipse</b>	<b>169</b>
13.1	Instalar el complemento de Authentic Desktop para Eclipse.....	170
13.2	Perspectiva de Authentic Desktop en Eclipse.....	172
13.3	Otros puntos de entrada de Authentic Desktop en Eclipse.....	175
<b>14</b>	<b>Comandos de menú</b>	<b>176</b>
14.1	Menú Archivo.....	177
14.1.1	Nuevo .....	177
14.1.2	Abrir .....	178
14.1.3	Volver a cargar.....	183

---

14.1.4	Codificación.....	184
14.1.5	Cerrar, Cerrar todos, Cerrar documentos inactivos.....	184
14.1.6	Guardar, Guardar como, Guardar todos.....	185
14.1.7	Enviar por correo electrónico.....	190
14.1.8	Imprimir.....	191
14.1.9	Vista previa de impresión, Configurar impresión.....	191
14.1.10	Archivos recientes, Salir.....	192
14.2	Menú Edición.....	193
14.2.1	Deshacer, Rehacer.....	193
14.2.2	Cortar, Copiar, Pegar, Eliminar.....	193
14.2.3	Seleccionar todo.....	194
14.2.4	Buscar, Buscar siguiente.....	194
14.2.5	Reemplazar.....	195
14.3	Menú Proyecto.....	196
14.3.1	Proyecto nuevo.....	199
14.3.2	Abrir proyecto.....	199
14.3.3	Volver a cargar el proyecto.....	199
14.3.4	Cerrar el proyecto.....	199
14.3.5	Guardar el proyecto, Guardar el proyecto como.....	200
14.3.6	Control de código fuente.....	200
14.3.7	Agregar archivos al proyecto.....	215
14.3.8	Agregar recurso global al proyecto.....	215
14.3.9	Agregar URL al proyecto.....	215
14.3.10	Agregar archivo activo al proyecto.....	216
14.3.11	Agregar archivo activo y relacionados al proyecto.....	216
14.3.12	Agregar carpeta de proyecto al proyecto.....	216
14.3.13	Agregar carpeta externa al proyecto.....	217
14.3.14	Agregar carpeta web externa al proyecto.....	220
14.3.15	Configuración de script.....	226
14.3.16	Propiedades.....	226
14.3.17	Proyectos usados recientemente.....	229
14.4	Menú XML.....	230
14.4.1	Comprobar formato XML.....	230
14.4.2	Validar el documento XML.....	230

---

14.4.3	Validar al editar.....	231
14.5	Menú XSL/XQuery.....	232
14.5.1	Transformación XSL.....	232
14.5.2	Transformación XSL-FO.....	233
14.5.3	Parámetros de XSL / Variables de XQuery.....	234
14.6	Menú Authentic.....	239
14.6.1	Documento nuevo.....	240
14.6.2	Editar datos de una base de datos.....	240
14.6.3	Editar una hoja de estilos de StyleVision.....	241
14.6.4	Seleccionar y editar una fila nueva con datos XML.....	241
14.6.5	Firma XML.....	242
14.6.6	Definir entidades XML.....	244
14.6.7	Ver marcado.....	246
14.6.8	RichEdit.....	246
14.6.9	Anexar/insertar/duplicar/eliminar fila.....	247
14.6.10	Mover fila, Eliminar fila.....	247
14.6.11	Generar un documento HTML, RTF, PDF o Word 2007+.....	247
14.6.12	Ubicaciones de confianza.....	248
14.7	Menú Vista.....	250
14.7.1	Vista Authentic.....	250
14.7.2	Vista Explorador.....	250
14.8	Menú Explorador.....	251
14.9	Menú Herramientas.....	252
14.9.1	Ortografía.....	252
14.9.2	Opciones de ortografía.....	255
14.9.3	Editor de script.....	258
14.9.4	Macros.....	258
14.9.5	Herramientas definidas por el usuario.....	258
14.9.6	Recursos globales.....	258
14.9.7	Configuración activa.....	259
14.9.8	Personalizar.....	260
14.9.9	Restaurar barras de herramientas y ventanas.....	278
14.9.10	Opciones.....	278
14.10	Menú Ventanas.....	295

14.11	Menú Ayuda.....	297
14.11.1	Contenido, Índice, Buscar.....	297
14.11.2	Mapa de teclado.....	298
14.11.3	Activación, formulario de pedido, registro, actualizaciones.....	298
14.11.4	Otros comandos.....	302
14.12	Línea de comandos.....	303

## **15 Referencia del programador 304**

15.1	Editor de scripts.....	306
15.1.1	Crear un proyecto de scripting.....	307
15.1.2	Comandos integrados.....	321
15.1.3	Habilitar scripts y macros.....	331
15.2	Complementos de Authentic Desktop para entornos IDE.....	334
15.2.1	Cómo registrar complementos para entornos IDE.....	334
15.2.2	Controles ActiveX.....	335
15.2.3	Configuración XML.....	335
15.2.4	Archivos de muestra ATL.....	338
15.2.5	IXMLSpyPlugIn.....	344
15.3	Application API.....	351
15.3.1	Overview.....	352
15.3.2	Interfaces.....	381
15.3.3	Enumerations.....	604
15.4	ActiveX Integration.....	618
15.4.1	Prerequisites.....	618
15.4.2	Adding the ActiveX Controls to the Toolbox.....	619
15.4.3	Integration at Application Level.....	621
15.4.4	Integration at Document Level.....	623
15.4.5	ActiveX Integration Examples.....	626
15.4.6	Command Reference.....	639
15.4.7	Object Reference.....	646

## **16 Anexos 667**

16.1	Datos técnicos.....	668
------	---------------------	-----

---

16.1.1	Requisitos de SO y memoria.....	668
16.1.2	Motores XSLT y XQuery de Altova.....	668
16.1.3	Compatibilidad con Unicode.....	669
16.1.4	Uso de Internet.....	669
16.2	Información sobre licencias.....	671
16.2.1	Distribución electrónica de software.....	671
16.2.2	Activación del software y medición de licencias.....	672
16.2.3	Altova Authentic: contrato de licencia para el usuario final.....	673

## Índice

**674**

# 1 Acerca de Authentic Desktop y su documentación

**Altova Authentic 2023 Desktop** es una innovadora herramienta visual para publicar documentos XML sin necesidad de preocuparse por los aspectos técnicos del lenguaje XML. Authentic Desktop 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. Authentic Desktop Enterprise Edition se puede usar en equipos de 64 y 32 bits.



*Última actualización: 17 October 2022*

## 1.1 Guía del usuario y de referencia

### 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:** Authentic Desktop también es compatible con Windows Server 2008 R2 SP1 con actualización de la plataforma o superior.

## 1.2 About This Documentation

Esta guía del usuario y de referencia contiene un tutorial y un manual que explica las características básicas de la Authentic View para que pueda empezar a trabajar con la aplicación. También incluye una extensa referencia que describe todas las características de la interfaz.

La presente documentación está dividida en estas secciones:

- La [introducción](#)<sup>24</sup> describe la interfaz gráfica del usuario y el entorno de Authentic Desktop.
- El [tutorial](#)<sup>35</sup> le ayudará a empezar a trabajar con Authentic Desktop.
- La sección sobre la [vista Authentic](#)<sup>51</sup> describe esta vista WYSIWYG para trabajar con documentos XML. La Authentic View permite al usuario escribir y editar documentos XML como si fueran sencillos documentos de texto o formularios interactivos. El marcado XML se oculta para que el usuario pueda concentrarse en el contenido del documento. La Authentic View es la vista principal de Authentic Desktop.
- La sección [Vista Explorador](#)<sup>103</sup> describe la vista en la que puede transformar documentos XML instantáneamente y verlos en una ventana del explorador web.
- La sección [Recursos globales](#)<sup>105</sup> describe esta característica de los productos de Altova que permite cambiar de recursos de forma rápida y sencilla.
- Dos apartados sobre cómo integrar Authentic Desktop en [Visual Studio](#)<sup>166</sup> y [Eclipse](#)<sup>169</sup>.
- La sección [Comandos de menú](#)<sup>176</sup> describe todas las ventanas y comandos de menú de Authentic Desktop.

## 2 Interfaz y entorno

Esta introducción describe:

- [la interfaz gráfica del usuario \(IGU\)](#)<sup>15</sup> y
- [el entorno de aplicación](#)<sup>22</sup>.

La [sección dedicada a la IGU](#)<sup>15</sup> empieza con una introducción sobre la IGU y después describe en detalle las diferentes ventanas de la interfaz. También explica como cambiar el tamaño de las ventanas, moverlas y cómo trabajar con ellas en general.

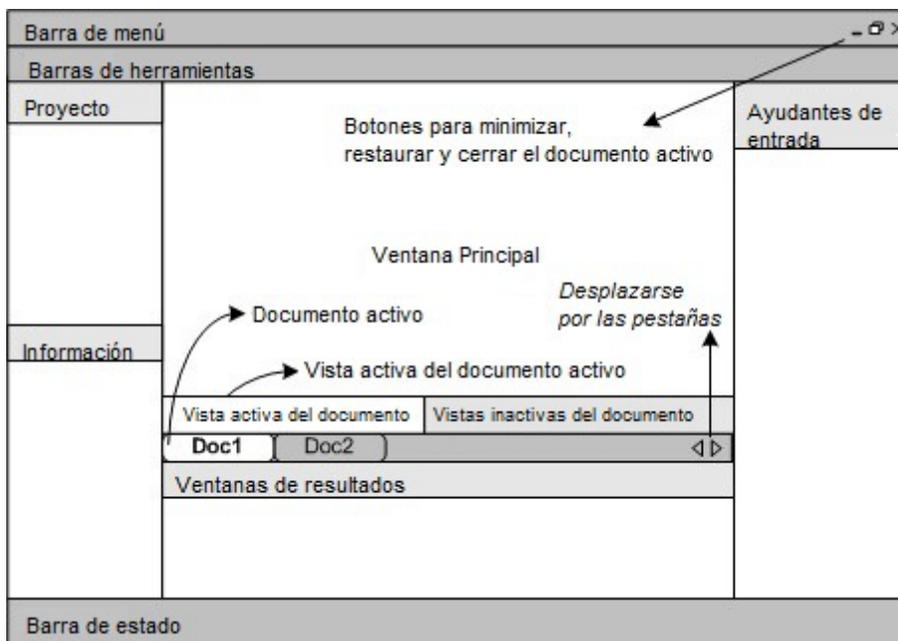
La [sección dedicada al entorno de aplicación](#)<sup>22</sup> describe las opciones de configuración con las que puede controlar cómo se visualizan y editan los diferentes tipos de archivo. También explica cómo se puede personalizar la aplicación. Además aprenderá dónde se guardan los archivos de ejemplo importantes y el tutorial. Al final de la sección, encontrará enlaces al [sitio web de Altova](#), donde podrá consultar una lista completa con todas las funciones y características de la aplicación. En el sitio web también encontrará el *Manual del usuario* en diferentes formatos, información sobre las opciones de soporte técnico disponibles y detalles sobre los demás productos de Altova.

## 2.1 La interfaz gráfica de usuario (IGU)

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

- Ventana Proyecto
- Ventana Información
- Ayudantes de entrada: Elementos, Atributos, Entidades, etc. (dependiendo del tipo de documento que esté activo)
- Ventanas de resultados: Mensajes

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



La IGU también incluye una barra de menú, una barra de estado y barras de herramientas, todas ellas descritas en esta sección.

### Activar o desactivar las barras laterales

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

### Desacoplar y acoplar las barras laterales

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

Hay varios métodos para acoplar o desacoplar una ventana:

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

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

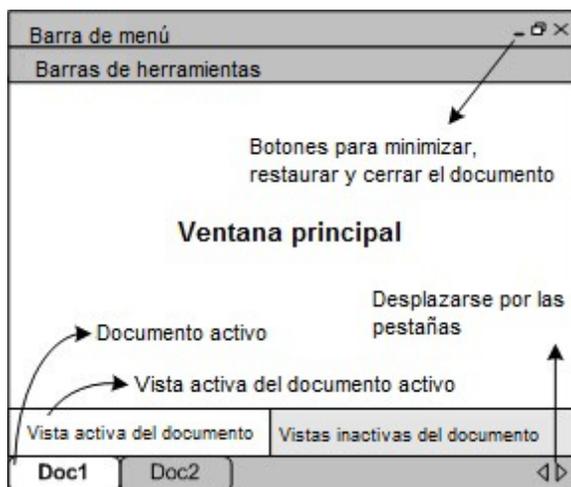
### Ocultar automáticamente las barras laterales

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

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

## 2.1.1 Ventana principal

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



### Archivos en la ventana principal

- Se pueden abrir y editar a la vez cualquier cantidad de archivos
- Cada documento que esté abierto tiene su propia ventana y pestaña (que tiene el nombre de archivo del documento) en la parte inferior de la ventana principal. Para activar un documento, haga clic en su pestaña.
- Si hay varios archivos abiertos, puede que algunas pestañas no sean visibles por falta de espacio en la barra de pestañas. Para ver todas las pestañas: (i) utilice los botones de desplazamiento situados a la derecha de la barra de pestañas o (ii) seleccione el documento pertinente en el menú [Ventanas](#)<sup>295</sup>.
- Cuando se maximiza el documento activo, los botones **Minimizar**, **Restaurar** y **Cerrar** están situados en el lado derecho de la barra de menú. Cuando un documento está en cascada, en mosaico o minimizado, los botones **Maximizar**, **Restaurar** y **Cerrar** están situados en la barra de título de la ventana del documento.
- Cuando se maximiza un archivo, se maximizan también todos los archivos que estén abiertos.
- Puede organizar los archivos abiertos en cascada o en mosaico con los comandos del menú [Ventanas](#)<sup>295</sup>.
- También puede activar los archivos abiertos en la misma secuencia en la que se abrieron si pulsa las teclas **Ctrl+Tabulador** o **Ctrl+F6**.
- Al hacer clic con el botón derecho en la pestaña de un documento aparece un menú contextual con varios comandos del menú **Archivo**, como **Imprimir** y **Cerrar**.

## Vistas en la ventana principal

El documento activo se puede visualizar y editar en varias vistas. Las vistas disponibles aparecen en una barra situada sobre las pestañas de los documentos (*ver imagen anterior*) y la vista activa aparece resaltada. Para seleccionar una vista, haga clic en el botón de la vista o seleccione la vista pertinente en el menú [Vista](#)<sup>250</sup>.

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

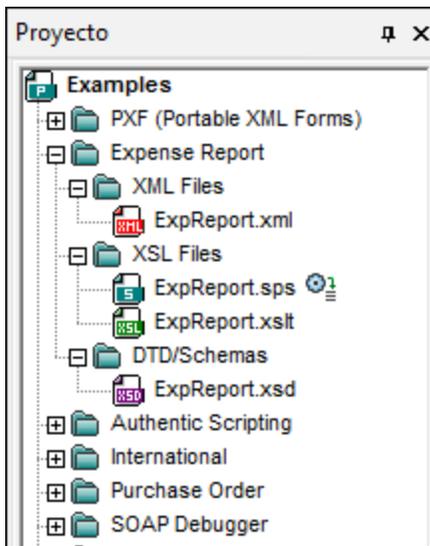
- [Vista Authentic](#)<sup>51</sup>: para editar documentos XML basados en hojas de estilos StyleVision Power Stylesheet en una interfaz gráfica.
- [Vista Explorador](#)<sup>103</sup>: una vista de explorador integrada compatible con CSS y hojas de estilos XSL

**Nota:** la vista predeterminada para cada tipo de archivo se puede personalizar en el cuadro de diálogo [Herramientas | Opciones](#)<sup>278</sup> (en el panel *Vista predeterminada* de la sección *Tipos de archivo*).

## 2.1.2 Ventana Proyecto

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

**Nota:** el proyecto de ejemplos `Examples` se abre por defecto en la ventana Proyecto. Si lo ha cerrado y necesita abrirlo vaya a la carpeta `Examples` de la aplicación, en la [carpeta \(Mis\) Documentos](#)<sup>12</sup> y haga doble clic en el archivo `Examples.spp`.



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

## Operaciones de proyecto

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



Los comandos de la barra de herramientas son, de izquierda a derecha: *Proyecto nuevo*, *Abrir proyecto*, *Volver a cargar el proyecto*, *Guardar el proyecto*, *Agregar archivo activo al proyecto*, *Seleccionar archivo activo*, *Expandir todo*, *Contraer todo*, *Buscar*. Los nombres de estos comandos son autoexplicativos pero también se describen en el [menú Proyecto](#) <sup>196</sup>.

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

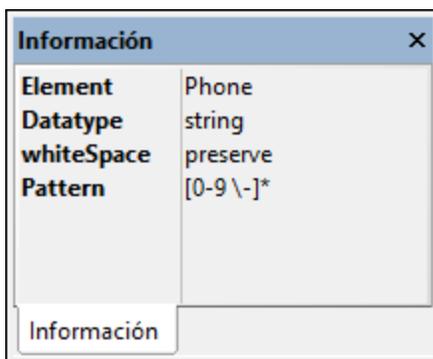
- En la ventana Proyecto se puede abrir sólo un proyecto. Cuando se crea un proyecto nuevo o se abre un proyecto ya existente, el proyecto de la ventana Proyecto se sustituye con el proyecto nuevo.
- Después de realizar cambios en un proyecto, es necesario guardar los cambios (clic en el comando **Proyecto | Guardar proyecto**). Si en un proyecto hay cambios sin guardar, junto al nombre del proyecto aparece un asterisco.
- El proyecto se representa en forma de árbol, con carpetas, archivos y otros recursos. Los recursos se pueden agregar en cualquier nivel y en cualquier subcarpeta.
- Las carpetas de proyecto son carpetas *semánticas* que representan una agrupación lógica de archivos. **No se corresponden** con la organización jerárquica de los archivos en su disco duro.
- Las carpetas pueden coincidir y estar relacionadas con directorios de su sistema de archivo. A esas carpetas las llamamos *carpetas externas* y en la ventana Proyecto se representan con un icono de carpeta amarilla (en contraste con las carpetas normales, que son verdes). Las carpetas externas de proyecto deben sincronizarse expresamente con el comando **Actualizar**.

- Cada carpeta puede contener varios tipos de archivo diferentes. También puede definir extensiones de archivo para cada carpeta (en el cuadro de diálogo "Propiedades" de la carpeta) para guardar los archivos más comunes en varias carpetas. Cuando se añade un archivo a la carpeta primaria, el archivo se añade automáticamente a la subcarpeta a la que se asignaron archivos con ese tipo de extensión.
- Al pasar el cursor del ratón sobre un archivo de imagen de una de las carpetas del proyecto aparece una vista previa de esa imagen (formatos `.png`, `.jpeg`, `.gif`, `.bmp`, `.tiff` y `.ico`). Haga doble clic en la imagen para abrirla en el visor/editor de imágenes predeterminado del sistema.
- En la ventana Proyecto puede arrastrar carpetas y colocarlas en otra carpeta o en otra posición de la misma carpeta. Los archivos, sin embargo, sólo se pueden arrastrar a otra carpeta, pero no se pueden mover a otra posición de la misma carpeta (en la que los archivos están organizados alfabéticamente). Además, puede arrastrar archivos y carpetas desde el explorador de archivos de Windows hasta la ventana Proyecto.
- Cada carpeta tiene un conjunto de propiedades que se definen en el cuadro de diálogo "Propiedades" de cada carpeta. Estas propiedades incluyen extensiones de archivos asignadas a la carpeta, el esquema con el que validar los archivos XML, el archivo XSLT con el que transformar los archivos XML, etc.
- Para procesar por lotes los archivos de una carpeta, haga clic con el botón derecho en la carpeta y seleccione el comando pertinente en el menú contextual (p. ej. **Validar el documento XML** o **Comprobar formato XML**).

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

### 2.1.3 Ventana Información

La ventana Información (*imagen siguiente*) muestra información sobre el elemento o el atributo donde está situado el cursor.



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

### 2.1.4 Ayudantes de entrada

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

editando un documento XML, los elementos, atributos y entidades que se pueden insertar en la posición actual del cursor aparecerán en las ventanas de los ayudantes de entrada.

**Nota:** puede habilitar o deshabilitar los ayudantes de entrada con la opción de menú **Ventanas | Ayudantes de entrada**.

## 2.1.5 Ventana de resultados: Mensajes

La ventana Mensajes muestra mensajes (como errores y otros resultados) sobre acciones llevadas a cabo en Authentic Desktop. Por ejemplo, si se valida un documento XML, esquema XML, DTD o XQuery y el documento es válido, aparece un mensaje de validación correcta (*ver imagen siguiente*) en la ventana Mensajes. De lo contrario, aparece un mensaje que describe el error (*ver imagen siguiente*). Observe que el mensaje de la imagen anterior incluye hipervínculos a los nodos del documento XML (texto subrayado en negro), así como hipervínculos a las secciones de la especificación pertinente en Internet que describe la regla en cuestión (texto subrayado en azul).

### Validar carpetas y archivos en la ventana Proyecto

El comando **Validar** (del menú **XML**) se suele aplicar al documento activo. Sin embargo, también puede aplicar este comando a un archivo, una carpeta o un grupo de archivos del proyecto activo. Seleccione el archivo o la carpeta que desea validar en la ventana Proyecto y después haga clic en [XML | Validar el documento XML](#)<sup>230</sup> o pulse la tecla **F8**. Los archivos del proyecto que no sean válidos se abren en la ventana principal y aparece un mensaje de error de tipo *El archivo no es válido*.

**Nota:** también puede realizar comprobaciones de formato XML (haciendo clic en el comando [Comprobar formato XML](#)<sup>230</sup> o pulsando **F7**) en la ventana Proyecto.

## 2.1.6 Barra de menú, barras de herramientas y barra de estado

### Barra de menú

La barra de menú (*ver ilustración*<sup>15</sup>) contiene los menús de la aplicación. Tenga en cuenta estas normas generales:

- Si los comandos de un menú **no son pertinentes** en una vista o en una posición determinada del documento, entonces no están disponibles.
- Algunos comandos abren un submenú con una lista de opciones adicionales. Los comandos de menú que tienen submenús se identifican con una punta de flecha apuntando hacia la derecha, situada después del nombre del comando.
- Algunos comandos abren un cuadro de diálogo que solicita más información al usuario. Esta información es necesaria para llevar a cabo el comando seleccionado. Estos comandos se identifican con unos puntos suspensivos (...) situados después del nombre del comando.
- Para acceder a un comando de menú, haga clic en el nombre del menú y después en el comando. Si un comando incluye un submenú, el submenú se abre al pasar el puntero del ratón sobre el comando. A continuación haga clic en el comando correspondiente del submenú.
- Los menús se pueden abrir si se pulsa la combinación de teclas correspondiente. La combinación de teclas de cada menú es **Alt+TECLA**, siendo **TECLA** la letra que aparece subrayada en el nombre del menú. Por ejemplo, para abrir el menú **A**rchivo la combinación de teclas es **Alt+A**.

- Los comandos de menú (es decir, los comandos de cada menú) se pueden ejecutar si selecciona primero (i) el menú con su combinación de teclas (ver el punto anterior) y después (ii) la combinación de teclas del comando en sí (**Alt+TECLA**, siendo **TECLA** la letra que aparece subrayada en el nombre del comando). Por ejemplo, para crear un archivo nuevo (**A**rchivo | **N**uevo), pulse **Alt+A** y después **Alt+N**.
- Algunos comandos de menú se pueden seleccionar directamente si presiona una tecla de **acceso rápido** o combinación de teclas (**Ctrl+TECLA**). Puede identificar los comandos con accesos rápidos porque la tecla de acceso rápido o la combinación de teclas aparece después del nombre del comando. Por ejemplo, puede usar la combinación de teclas **Ctrl+N** para crear un archivo nuevo o la tecla de acceso rápido **F8** para validar un archivo XML. También puede [crear sus propias teclas de acceso rápido](#)<sup>265</sup> en la pestaña Teclado del cuadro de diálogo "Personalizar" (**Herramientas | Personalizar**).

## Barras de herramientas

Las barras de herramientas ([ver ilustración](#)<sup>15</sup>) contienen iconos que funcionan como accesos rápidos para seleccionar comandos de menú. El nombre del comando aparece al pasar el puntero del ratón sobre el icono. Para ejecutar el comando, haga clic en el icono.

Los botones de las barras de herramientas se organizan en grupos. En el cuadro de diálogo [Herramientas | Personalizar | Barras de herramientas](#)<sup>262</sup> puede indicar qué barras de herramientas desea activar. Estas opciones de configuración afectan a la vista actual. Para crear una configuración para otra vista, cambie a esa vista y vuelva a seleccionar las opciones correspondientes en el cuadro de diálogo [Herramientas | Personalizar | Barras de herramientas](#)<sup>262</sup>. En la IGU también puede arrastrar barras de herramientas por sus controladores (o barras de título) y colocarlas en otra posición de la pantalla. Al hacer doble clic en el controlador la barra de herramientas se desacopla. Si hace doble clic otra vez en la barra de título, la barra de herramientas se acopla en la posición donde estuvo acoplada por última vez.

## Barra de estado

La barra de estado está situada en la parte inferior de la ventana de la aplicación ([ver ilustración](#)<sup>15</sup>) y muestra (i) información de estado sobre la carga de archivos e (ii) información sobre los comandos de menú y los accesos rápidos a comandos de las barras de herramientas al pasar sobre estos el puntero del ratón. Si usa la versión de 64 bits de Authentic Desktop, la barra de estado incluye el sufijo (x64) después del nombre de la aplicación. Para la versión de 32 bits no se incluye ningún sufijo.

## 2.2 El entorno de aplicación

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

Esta sección se divide en tres partes:

- [Configuración y personalización](#)<sup>22</sup>: describe cómo definir importantes opciones de configuración y personalización.
- [Tutoriales, proyectos y ejemplos](#)<sup>22</sup> repasa la ubicación de los archivos incluidos con la aplicación que no pertenecen al programa.
- [Características y archivos de ayuda](#)<sup>23</sup>: ofrece enlaces al [sitio web de Altova](#), donde encontrará información sobre las funciones del producto, recursos de ayuda en varios formatos e información sobre otros productos de Altova.

### 2.2.1 Configuración y personalización

Esta sección explica las opciones de configuración y personalización que se pueden elegir en Authentic Desktop.

#### Configuración

El cuadro de diálogo "Opciones" incluye varias secciones donde se definen importantes opciones de configuración de Authentic Desktop. Consulte las secciones de este cuadro de diálogo y las diferentes opciones para familiarizarse con ellas.

#### Personalización

También puede personalizar varios aspectos de Authentic Desktop, incluido el aspecto de la IGU. Estas opciones de personalización están disponibles en el cuadro de diálogo "Personalizar" (comando de menú [Herramientas | Personalizar](#)<sup>260</sup>). Para una descripción más detallada de las opciones de personalización, consulte la sección [Referencia del usuario](#)<sup>176</sup>.

### 2.2.2 Tutoriales, proyectos y ejemplos

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

#### Ubicación de los tutoriales, proyectos y archivos de ejemplo

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

```
C:\Documents and Settings\\Mis
Documentos\Altova\Authentic2023\AuthenticExamples\
```

La carpeta `Mis Documentos\Altova\Authentic2023` se instalará para cada usuario registrado en un equipo dentro de la carpeta `<usuario>` del usuario. Por tanto, según este sistema de instalación, cada usuario tendrá su propia carpeta `AuthenticExamples` en un área de trabajo distinta.

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

El tutorial, el proyecto y todos los archivos de ejemplo se encuentran en la carpeta `AuthenticExamples`.

## 2.2.3 Características y archivos de ayuda

El sitio web de Altova [www.altova.com](http://www.altova.com) contiene abundante información y gran cantidad de recursos relacionados con Authentic Desktop. A continuación mencionamos algunos ejemplos.

### Authentic Desktop feature listing

El sitio web de Altova incluye [una lista actualizada de las características de Authentic Desktop](#).

### Ayuda de Authentic Desktop

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

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

### Opciones de soporte técnico

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

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

### Productos de Altova

El [sitio web de Altova](#) incluye una lista con todos los productos de Altova.

## 3 Interfaz y entorno

Esta introducción describe:

- [la interfaz gráfica del usuario \(IGU\)](#)<sup>25</sup> y
- [el entorno de aplicación](#)<sup>32</sup>.

La [sección dedicada a la IGU](#)<sup>25</sup> empieza con una introducción sobre la IGU y después describe en detalle las diferentes ventanas de la interfaz. También explica como cambiar el tamaño de las ventanas, moverlas y cómo trabajar con ellas en general.

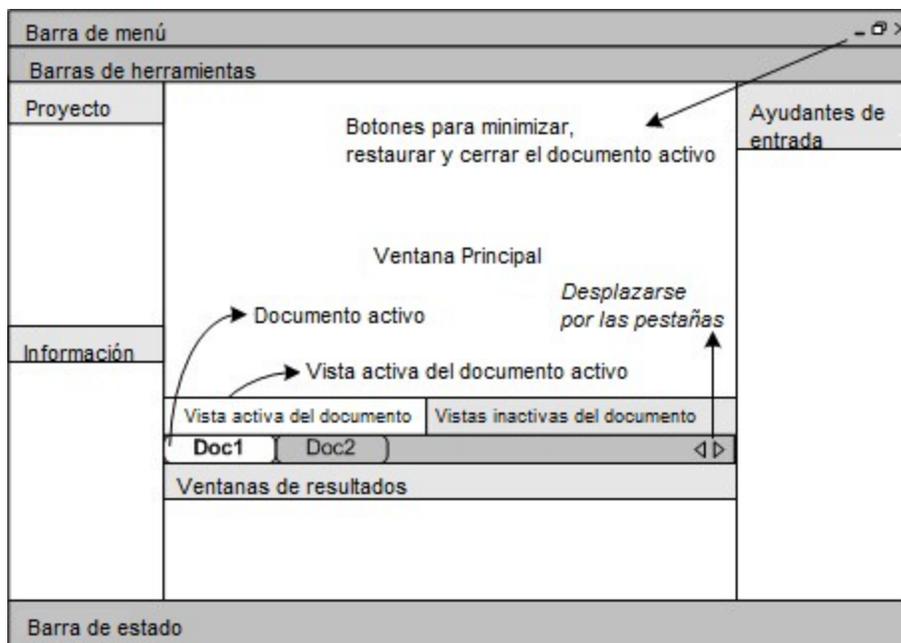
La [sección dedicada al entorno de aplicación](#)<sup>32</sup> describe las opciones de configuración con las que puede controlar cómo se visualizan y editan los diferentes tipos de archivo. También explica cómo se puede personalizar la aplicación. Además aprenderá dónde se guardan los archivos de ejemplo importantes y el tutorial. Al final de la sección, encontrará enlaces al [sitio web de Altova](#), donde podrá consultar una lista completa con todas las funciones y características de la aplicación. En el sitio web también encontrará el *Manual del usuario* en diferentes formatos, información sobre las opciones de soporte técnico disponibles y detalles sobre los demás productos de Altova.

### 3.1 La interfaz gráfica de usuario (IGU)

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

- Ventana Proyecto
- Ventana Información
- Ayudantes de entrada: Elementos, Atributos, Entidades, etc. (dependiendo del tipo de documento que esté activo)
- Ventanas de resultados: Mensajes

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



La IGU también incluye una barra de menú, una barra de estado y barras de herramientas, todas ellas descritas en esta sección.

#### Activar o desactivar las barras laterales

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

#### Desacoplar y acoplar las barras laterales

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

Hay varios métodos para acoplar o desacoplar una ventana:

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

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

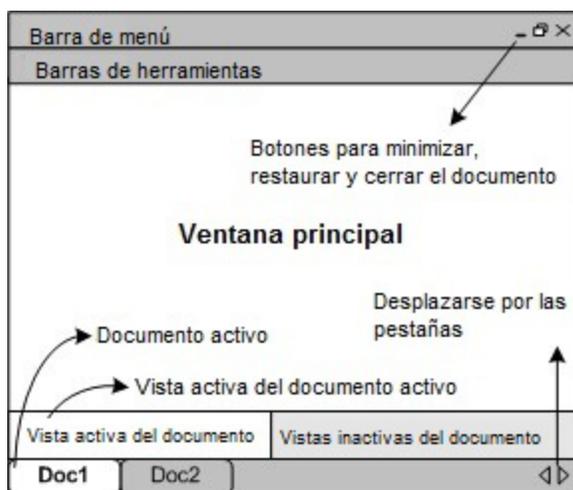
### Ocultar automáticamente las barras laterales

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

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

## 3.1.1 Ventana principal

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



### Archivos en la ventana principal

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

### Vistas en la ventana principal

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

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

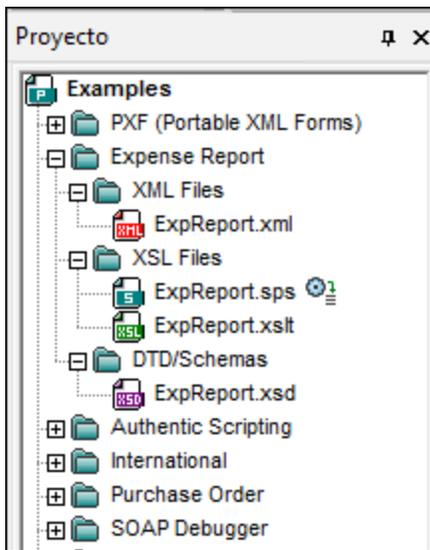
- [Vista Authentic](#)<sup>51</sup>: para editar documentos XML basados en hojas de estilos StyleVision Power Stylesheet en una interfaz gráfica.
- [Vista Explorador](#)<sup>103</sup>: una vista de explorador integrada compatible con CSS y hojas de estilos XSL.

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

## 3.1.2 Ventana Proyecto

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

**Nota:** el proyecto de ejemplos `Examples` se abre por defecto en la ventna Proyecto. Si lo ha cerrado y necesita abrirlo vaya a la carpeta `Examples` de la aplicación, en la [carpeta \(Mis\) Documentos](#)<sup>12</sup>, y haga doble clic en el archivo `Examples.spp`.



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

## Operaciones de proyecto

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



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

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

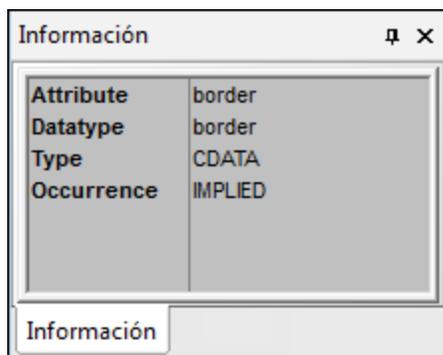
- En la ventana Proyecto se puede abrir sólo un proyecto. Cuando se crea un proyecto nuevo o se abre un proyecto ya existente, el proyecto de la ventana Proyecto se sustituye con el proyecto nuevo.
- Después de realizar cambios en un proyecto, es necesario guardar los cambios (clic en el comando **Proyecto | Guardar proyecto**). Si en un proyecto hay cambios sin guardar, junto al nombre del proyecto aparece un asterisco.
- El proyecto se representa en forma de árbol, con carpetas, archivos y otros recursos. Los recursos se pueden agregar en cualquier nivel y en cualquier subcarpeta.
- Las carpetas de proyecto son carpetas *semánticas* que representan una agrupación lógica de archivos. **No se corresponden** con la organización jerárquica de los archivos en su disco duro.
- Las carpetas pueden coincidir y estar relacionadas con directorios de su sistema de archivo. A esas carpetas las llamamos *carpetas externas* y en la ventana Proyecto se representan con un icono de carpeta amarilla (en contraste con las carpetas normales, que son verdes). Las carpetas externas de proyecto deben sincronizarse expresamente con el comando **Actualizar**.

- Cada carpeta puede contener varios tipos de archivo diferentes. También puede definir extensiones de archivo para cada carpeta (en el cuadro de diálogo "Propiedades" de la carpeta) para guardar los archivos más comunes en varias carpetas. Cuando se añade un archivo a la carpeta primaria, el archivo se añade automáticamente a la subcarpeta a la que se asignaron archivos con ese tipo de extensión.
- Al pasar el cursor del ratón sobre un archivo de imagen de una de las carpetas del proyecto aparece una vista previa de esa imagen (formatos `.png`, `.jpeg`, `.gif`, `.bmp`, `.tiff` y `.ico`). Haga doble clic en la imagen para abrirla en el visor/editor de imágenes predeterminado del sistema.
- En la ventana Proyecto puede arrastrar carpetas y colocarlas en otra carpeta o en otra posición de la misma carpeta. Los archivos, sin embargo, sólo se pueden arrastrar a otra carpeta, pero no se pueden mover a otra posición de la misma carpeta (en la que los archivos están organizados alfabéticamente). Además, puede arrastrar archivos y carpetas desde el explorador de archivos de Windows hasta la ventana Proyecto.
- Cada carpeta tiene un conjunto de propiedades que se definen en el cuadro de diálogo "Propiedades" de cada carpeta. Estas propiedades incluyen extensiones de archivos asignadas a la carpeta, el esquema con el que validar los archivos XML, el archivo XSLT con el que transformar los archivos XML, etc.
- Para procesar por lotes los archivos de una carpeta, haga clic con el botón derecho en la carpeta y seleccione el comando pertinente en el menú contextual (p. ej. **Validar el documento XML** o **Comprobar formato XML**).

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

### 3.1.3 Ventana Información

La ventana Información (*imagen siguiente*) muestra información sobre el elemento o el atributo donde está situado el cursor.



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

### 3.1.4 Ayudantes de entrada

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

editando un documento XML, los elementos, atributos y entidades que se pueden insertar en la posición actual del cursor aparecerán en las ventanas de los ayudantes de entrada.

Tenga en cuenta que puede habilitar o deshabilitar los ayudantes de entrada con la opción de menú **Ventanas | Ayudantes de entrada**.

### 3.1.5 Ventana de resultados: Mensajes

La ventana Mensajes muestra información sobre las acciones realizadas en Authentic Desktop así como sobre errores y otros resultados. Por ejemplo, si se valida un documento XML y este resulta ser válido, aparece un mensaje de validación correcta en la ventana de mensajes. De lo contrario, aparece un mensaje que describe el error de validación. Recuerde que el mensaje de validación contiene hipervínculos (texto de vínculo negro) a los nodos y el contenido de los nodos del documento XML, así como hipervínculos (texto de vínculo azul) a la sección pertinente de la especificación que describe la regla en cuestión.

#### Validar carpetas y archivos en la ventana Proyecto

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

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

### 3.1.6 Barra de menú, barras de herramientas y barra de estado

#### Barra de menú

La barra de menú ([ver ilustración](#)<sup>25</sup>) contiene los menús de la aplicación. Tenga en cuenta estas normas generales:

- Si los comandos de un menú **no son pertinentes** en una vista o en una posición determinada del documento, entonces no están disponibles.
- Algunos comandos abren un submenú con una lista de opciones adicionales. Los comandos de menú que tienen submenús se identifican con una punta de flecha apuntando hacia la derecha, situada después del nombre del comando.
- Algunos comandos abren un cuadro de diálogo que solicita más información al usuario. Esta información es necesaria para llevar a cabo el comando seleccionado. Estos comandos se identifican con unos puntos suspensivos (...) situados después del nombre del comando.
- Para acceder a un comando de menú, haga clic en el nombre del menú y después en el comando. Si un comando incluye un submenú, el submenú se abre al pasar el puntero del ratón sobre el comando. A continuación haga clic en el comando correspondiente del submenú.
- Los menús se pueden abrir si se pulsa la combinación de teclas correspondiente. La combinación de teclas de cada menú es **Alt+TECLA**, siendo **TECLA** la letra que aparece subrayada en el nombre del menú. Por ejemplo, para abrir el menú **A**rchivo la combinación de teclas es **Alt+A**.

- Los comandos de menú (es decir, los comandos de cada menú) se pueden ejecutar si selecciona primero (i) el menú con su combinación de teclas (ver el punto anterior) y después (ii) la combinación de teclas del comando en sí (**Alt+TECLA**, siendo **TECLA** la letra que aparece subrayada en el nombre del comando). Por ejemplo, para crear un archivo nuevo (**A**rchivo | **N**uevo), pulse **Alt+A** y después **Alt+N**.
- Algunos comandos de menú se pueden seleccionar directamente si presiona una tecla de **acceso rápido** o combinación de teclas (**Ctrl+TECLA**). Puede identificar los comandos con accesos rápidos porque la tecla de acceso rápido o la combinación de teclas aparece después del nombre del comando. Por ejemplo, puede usar la combinación de teclas **Ctrl+N** para crear un archivo nuevo o la tecla de acceso rápido **F8** para validar un archivo XML. También puede crear sus propias teclas de acceso rápido en la pestaña Teclado del cuadro de diálogo "Personalizar" (**H**erramientas | **P**ersonalizar).

## Barras de herramientas

Las barras de herramientas ([ver ilustración](#)<sup>25</sup>) contienen iconos que funcionan como accesos rápidos para seleccionar comandos de menú. El nombre del comando aparece al pasar el puntero del ratón sobre el icono. Para ejecutar el comando, haga clic en el icono.

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

## Barra de estado

La barra de estado está situada en la parte inferior de la ventana de la aplicación ([ver ilustración](#)<sup>25</sup>) y muestra (i) información de estado sobre la carga de archivos e (ii) información sobre los comandos de menú y los accesos rápidos a comandos de las barras de herramientas al pasar sobre estos el puntero del ratón. Si usa la versión de 64 bits de Authentic Desktop, la barra de estado incluye el sufijo (x64) después del nombre de la aplicación. Para la versión de 32 bits no se incluye ningún sufijo.

## 3.2 El entorno de aplicación

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

Esta sección se divide en tres partes:

- [Configuración y personalización](#)<sup>32</sup>: describe cómo definir importantes opciones de configuración y personalización.
- [Tutoriales, proyectos y ejemplos](#)<sup>33</sup>: repasa la ubicación de los archivos incluidos con la aplicación que no pertenecen al programa.
- [Características y archivos de ayuda](#)<sup>33</sup>: ofrece enlaces al [sitio web de Altova](#), donde encontrará información sobre las funciones del producto, recursos de ayuda en varios formatos e información sobre otros productos de Altova.

### 3.2.1 Configuración y personalización

Esta sección explica las opciones de configuración y personalización que se pueden elegir en Authentic Desktop.

#### Configuración

El cuadro de diálogo "Opciones" incluye varias secciones donde se definen importantes opciones de configuración de Authentic Desktop. Consulte las secciones de este cuadro de diálogo y las diferentes opciones para familiarizarse con ellas.

#### Personalización

También puede personalizar varios aspectos de Authentic Desktop, incluido el aspecto de la IGU. Estas opciones de personalización están disponibles en el cuadro de diálogo "Personalizar" (comando de menú **Herramientas | Personalizar**).

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

## 3.2.2 Tutoriales, proyectos y ejemplos

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

### Ubicación de los tutoriales, proyectos y archivos de ejemplo

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

```
C:\Documents and Settings\<<usuario>\Mis
Documentos\Altova\Authentic2023\AuthenticExamples\
```

La carpeta `Mis Documentos\Altova\Authentic2023` se instalará para cada usuario registrado en un equipo dentro de la carpeta `<usuario>` del usuario. Por tanto, según este sistema de instalación, cada usuario tendrá su propia carpeta `AuthenticExamples` en un área de trabajo distinta.

### Nota sobre la carpeta maestra de Authentic Desktop

Cuando se instala Authentic Desktop en un equipo, se crea la carpeta maestra `Altova\Authentic2023` en esta ubicación:

```
C:\Documents and Settings\All Users\Datos de programa\
```

Cuando un usuario inicia Authentic Desktop por primera vez en ese equipo, Authentic Desktop crea una copia de esa carpeta maestra en la carpeta `<nombreDeUsuario>\Mis Documentos\` del usuario. Por tanto, es importante no usar la carpeta maestra mientras trabaja con el tutorial o los archivos de ejemplo. De lo contrario, los archivos editados se copiarán en la carpeta de usuario del siguiente usuario que utilice Authentic Desktop por primera vez en el mismo equipo.

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

El tutorial, el proyecto y todos los archivos de ejemplo se encuentran en la carpeta `AuthenticExamples`.

## 3.2.3 Características y archivos de ayuda

El sitio web de Altova [www.altova.com](http://www.altova.com) contiene abundante información y gran cantidad de recursos relacionados con Authentic Desktop. A continuación mencionamos algunos ejemplos.

### Lista de características de Authentic Desktop

El sitio web de Altova incluye [una lista actualizada de las características de Authentic Desktop](#).

### Ayuda de Authentic Desktop

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

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

### Opciones de soporte técnico

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

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

### Productos de Altova

El [sitio web de Altova](#) incluye una lista con todos los productos de Altova.

## 4 Tutorial de la vista Authentic

En la Authentic View puede editar documentos XML en una interfaz gráfica WYSIWYG (*ver imagen siguiente*) como si se tratase de una aplicación de procesador de textos como Microsoft Word. De hecho, lo único que tiene que hacer es introducir los datos. No tiene que preocuparse por el formato del documento, puesto que este ya está definido en la hoja de estilos que controla la Authentic View del documento XML. La hoja de estilos StyleVision Power Stylesheet (en adelante SPS) la crea un diseñador de hojas de estilos con Altova StyleVision.

<b>Nanonull, Inc.</b>	
Location: <input type="text" value="US"/>	
<b>Street:</b> 119 Oakstreet, Suite 4876	<b>Phone:</b> +1 (321) 555 5155 0
<b>City:</b> Vereno	<b>Fax:</b> +1 (321) 555 5155 4
<b>State &amp; Zip:</b> <input type="text" value="DC"/> <input type="text" value="29213"/>	<b>E-mail:</b> <a href="mailto:office@nanonull.com">office@nanonull.com</a>

**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.

Para editar un documento XML en la Authentic View el usuario puede hacer dos cosas: (i) editar la estructura del documento (por ejemplo, agregar/eliminar párrafos y títulos) e (ii) introducir datos (el contenido de los párrafos y títulos, por ejemplo).

Este tutorial describe:

- Cómo se abre un documento XML en la Authentic View. El principal requisito para editar documentos XML en la Authentic View es que los documentos estén asociados a un archivo SPS.
- La interfaz de la Authentic View y cómo funcionan los mecanismos de edición en general.
- Cómo se edita la estructura de un documento insertando y eliminando nodos.
- Cómo se introducen datos en un documento XML.
- Cómo se introducen valores de atributo desde el ayudantes de entrada Atributos y valores de entidades.
- Cómo se imprime el documento.

Recuerde que este tutorial es solo una introducción y, por tanto, es bastante simple y directo. Si desea más información consulte el material de referencia de la sección [Interfaz de la Authentic View](#)<sup>51</sup>.

### Material necesario para este tutorial

**Todos los archivos** necesarios para este tutorial se encuentran en la carpeta `...\Examples` de la carpeta de aplicación. Los archivos necesarios para este tutorial son:

- `NanonullOrg.xml` (el documento XML)
- `NanonullOrg.sps` (la hoja de estilos SPS a la que está vinculado el documento XML)
- `NanonullOrg.xsd` (el esquema XML en el que se basan y al que están vinculados el documento XML y la hoja de estilos SPS)
- `nanonull.gif` y `Altova_right_300.gif` (dos archivos de imagen usados en el tutorial)

**Nota:** en algunas secciones del tutorial le pediremos que mire el texto XML del documento XML (a diferencia de la Authentic View del documento). Si la edición del producto de Altova que utiliza no incluye la vista Texto (si utiliza por ejemplo Authentic Desktop o Authentic Browser), entonces puede utilizar un editor de texto corriente como Wordpad o Notepad para ver el texto del documento XML.

**Advertencia:** le recomendamos que haga una copia del archivo `NanonullOrg.xml` y la utilice para este tutorial. Así podrá consultar siempre que quiera el documento original.

## 4.1 Abrir un documento en la vista Authentic

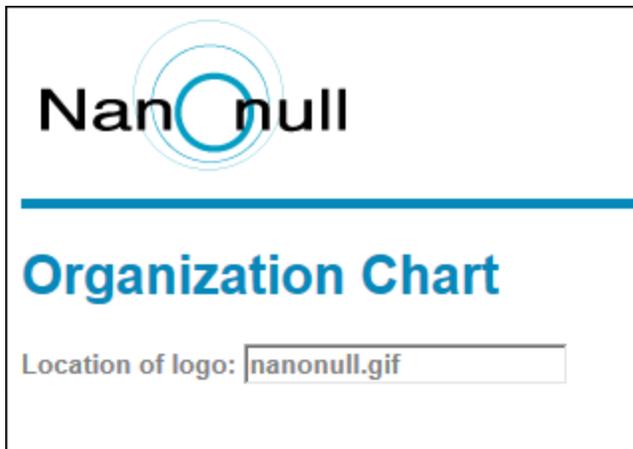
En la Authentic View puede editar un documento XML ya existente o crear uno nuevo y editarlo. En esta sección del tutorial abrirá un documento XML en la Authentic View y en las secciones siguientes aprenderá a editarlo. Esta sección también explica cómo se crea un documento XML nuevo para editarlo en la Authentic View.

### Abrir un documento XML ya existente

El archivo que abrirá es `NanonullOrg.xml`. Se encuentra en la carpeta `Examples` de su aplicación de Altova. Hay dos maneras de abrir el archivo:

- Haga clic en **Archivo | Abrir**, busque el archivo `NanonullOrg.xml` en el cuadro de diálogo que aparece y haga clic en **Abrir**.
- Utilice el explorador de Windows para localizar el archivo, haga clic con el botón derecho en el archivo y seleccione su producto de Altova en el submenú **Abrir con**.

El archivo `NanonullOrg.xml` se abre directamente en la Authentic View (*imagen siguiente*).



**Nota:** el archivo que define y controla cómo se presenta el documento XML en la Authentic View es el archivo SPS. Si no existe un SPS, el documento no se puede mostrar en la Authentic View.

### Crear un documento XML basado en una SPS

También puede crear un documento XML nuevo basado en una SPS. Hay dos maneras de hacerlo: con el comando de menú **Archivo | Nuevo** o con el comando de menú **Authentic | Documento nuevo**. En ambos casos es necesario seleccionar un archivo SPS.

#### Opción 1: Archivo | Nuevo

1. Seleccione **Archivo | Nuevo** y en el cuadro de diálogo "Crear documento nuevo" seleccione el tipo de archivo XML.
2. Haga clic en el botón **Seleccione una hoja de estilos de StyleVision** y busque el archivo SPS.

Si se ha asignado un archivo de plantilla XML al archivo SPS, los datos de ese archivo de plantilla XML se usan como base para crear la plantilla del documento XML en Authentic View.

### Opción 2: Authentic | Documento nuevo

1. Seleccione **Authentic | Documento nuevo**.
2. En el diálogo "Crear documento nuevo" busque el archivo SPS.

Si se asignó un archivo de plantilla XML al SPS, entonces los datos de la plantilla XML se usarán como datos de partida del documento XML que se crea en la Authentic View.

## 4.2 La interfaz de la vista Authentic

La interfaz de edición de la Authentic View está compuesta por una ventana principal, en la que se introducen y editan los datos del documento, y tres ayudantes de entrada. Editar documentos en la vista Authentic es fácil. Si desea ver el marcado del documento, active las etiquetas de marcado. Después puede empezar a escribir en el contenido del documento. Para cambiar la estructura del documento puede usar el menú contextual o el ayudante de entrada Elementos.

### Mostrar las etiquetas de los nodos XML (marcado del documento)

Un documento XML es básicamente una jerarquía de nodos. Por ejemplo:

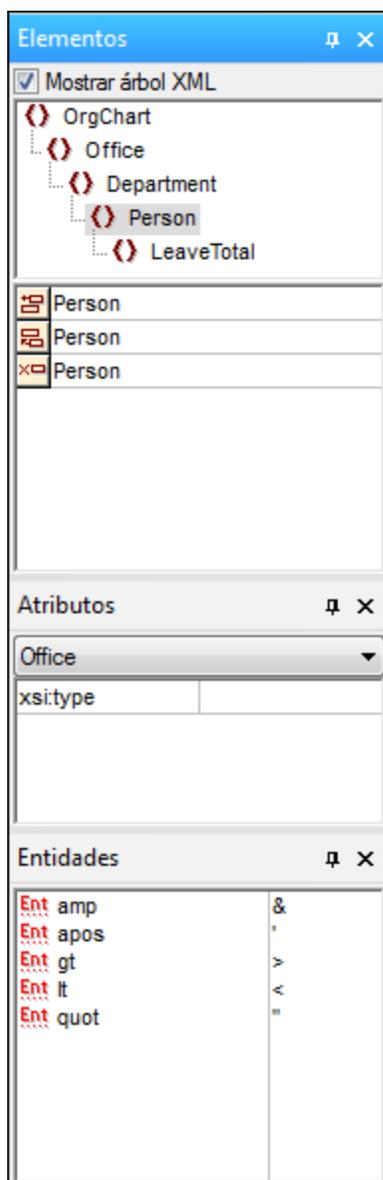
```
<DocumentRoot>
  <Person id="ABC001">
    <Name>Alpha Beta</Name>
    <Address>Some Address</Address>
    <Tel>1234567</Tel>
  </Person>
</DocumentRoot>
```

La opción predeterminada de la Authentic View es no mostrar las etiquetas de los nodos. Para activar las etiquetas de los nodos, seleccione el comando de menú **Authentic | Mostrar marcado en tamaño grande** (o haga clic en el icono  de la barra de herramientas). Las etiquetas en tamaño grande muestran el nombre del nodo. También puede ver el marcado en tamaño pequeño (las etiquetas no incluyen el nombre del nodo) o como marcado mixto (definido por el diseñador de la hoja de estilos. En el marcado mixto predeterminado todas las etiquetas están desactivadas).

Puede ver el texto del documento XML en la vista Texto de su producto de Altova o en un editor de texto.

### Ayudantes de entrada

Hay tres ayudantes de entrada en la interfaz de la vista Authentic (*imagen siguiente*), situados a la derecha de la ventana de la aplicación. Son los ayudantes de entrada Elementos, Atributos y Entidades.



#### Ayudante de entrada Elementos

Este ayudante de entrada muestra los elementos que se pueden insertar y quitar dependiendo de la posición del cursor o del texto que está seleccionado en la ventana principal. Tenga en cuenta que el ayudante de entrada es contextual y su contenido cambia dependiendo de la posición del cursor o del texto que está seleccionado. Además, cuando se selecciona un nodo del árbol XML del ayudante de entrada, los elementos relacionados con ese nodo aparecen en el ayudante de entrada. Para ver el árbol XML en el ayudante de entrada Elementos, marque la casilla "Mostrar árbol XML" situada en la parte superior del ayudante (ver *imagen anterior*). El árbol XML muestra la jerarquía de nodos a partir del elemento de nivel superior y hasta llegar al nodo que está seleccionado en la ventana principal.

#### Ayudante de entrada Atributos

Este ayudante de entrada muestra los atributos del elemento que está seleccionado en la ventana principal. Los valores de los atributos se pueden introducir y editar en este ayudante. El ayudante también incluye un cuadro combinado donde puede seleccionar todos los elementos de la jerarquía, desde el elemento de nivel superior hasta el elemento seleccionado. Al seleccionar un elemento en el cuadro combinado aparecen los atributos de ese elemento en el ayudante de entrada, donde pueden editarse.

### Ayudante de entrada Entidades

Este ayudante de entrada no es contextual y muestra todas las entidades que se declararon para el documento. Al hacer doble clic en una entidad, esta se inserta en la posición del cursor. Para más información sobre cómo añadir entidades en un documento, consulte la sección [Interfaz de la Authentic View](#)<sup>51</sup>.

### Menú contextual

Al hacer clic con el botón derecho en un documento abierto en la Authentic View aparece un menú contextual con opciones relacionadas con la posición (el nodo) del documento donde se hizo clic. El menú contextual incluye comandos para:

- Insertar nodos en la posición del nodo, antes o después. Los submenús incluyen todos los nodos permitidos en la posición donde se hizo clic.
- Quitar el nodo seleccionado (si el esquema lo permite) o los elementos antecesores que se puedan quitar. Los nodos que se pueden quitar (de acuerdo con el esquema) aparecen en un submenú.
- Insertar entidades y secciones CDATA. Las entidades declaradas para el documento aparecen en un submenú. Las secciones CDATA sólo se pueden insertar dentro de texto.
- Cortar, copiar, pegar (como texto o como XML) y eliminar contenido del documento.

**Nota:** para más información sobre la interfaz, consulte la sección [Interfaz de la Vista Authentic](#)<sup>51</sup>

## 4.3 Operaciones con nodos

En la Authentic View de un documento XML existen dos tipos principales de nodos: **nodos de elemento** y **nodos de atributo**. Estos nodos están marcados con etiquetas, que puede [activar](#)<sup>39</sup> si lo desea. El documento también puede incluir otros nodos, como nodos de texto (que no están marcados) y nodos de sección CDATA (que están marcados para separarlos del texto que los rodea).

Las operaciones de nodo descritas en este apartado se refieren únicamente a los nodos de elemento y de atributo. Recomendamos que tenga [activado el marcado en tamaño grande](#)<sup>39</sup> mientras prueba las operaciones descritas en este apartado.

**Nota:** es importante recordar que antes y después del elemento seleccionado sólo se pueden insertar **elementos del mismo nivel o de nivel superior**. Por ejemplo, los **elementos del mismo nivel** de un elemento párrafo serían otros elementos párrafos, pero también listas, una tabla, una imagen, etc. Los elementos del mismo nivel pueden aparecer antes o después de un elemento. Los elementos de nivel superior son los elementos **antecesores** y los elementos del mismo nivel de los antecesores. Por ejemplo, los elementos antecesores de un elemento párrafo serían un elemento sección, capítulo, artículo, etc. Un párrafo de un archivo XML válido ya tendría elementos antecesores. Por tanto, al añadir un elemento de nivel superior en la Authentic View, el elemento nuevo que se añade se crea al mismo nivel que el antecesor correspondiente. Por ejemplo, si se inserta una sección después de un párrafo, se crea como elemento del mismo nivel que la sección en la que se encuentra dicho párrafo.

### Realizar operaciones con nodos

Para realizar operaciones con nodos seleccione un comando del [menú contextual](#)<sup>41</sup> o haga clic en la opción correspondiente del [ayudante de entrada Elementos](#)<sup>39</sup>. En algunos casos se pueden añadir elementos o atributos haciendo clic en el [hipervínculo Agregar nodo](#)<sup>43</sup> de la Authentic View del documento. Hay casos especiales en los que los elementos se definen como párrafos o elementos de lista. Si coloca el cursor dentro de dichos elementos y pulsa la [tecla Entrar](#)<sup>43</sup>, se crea un elemento del mismo nivel de ese tipo. Este apartado también describe cómo se crean y eliminan nodos con los mecanismos [Aplicar elemento](#)<sup>43</sup>, [Quitar nodo](#)<sup>44</sup> y [Borrar elemento](#)<sup>44</sup>.

### Insertar elementos

Puede insertar elementos en estas posiciones:

- Dentro del nodo de elemento, en la posición donde está situado el cursor. Los elementos que se pueden insertar en la posición seleccionada aparecen en un submenú del comando **Insertar** del menú contextual. En el ayudante de entrada Elementos los elementos que se pueden insertar en la posición seleccionada llevan el icono . En el documento `NanonullOrg.xml` coloque el cursor dentro del elemento `para` y cree los elementos `bold` e `italic` usando tanto el menú contextual como el ayudante de entrada Elementos.
- Antes o después del elemento seleccionado o de sus antecesores, si el esquema lo permite. Seleccione el elemento que desea insertar en los submenús del menú contextual. En el ayudante de entrada Elementos los elementos que se pueden insertar antes o después del elemento seleccionado llevan los iconos y respectivamente. Recuerde que desde el ayudante de entrada Elementos no se pueden insertar elementos antes ni después del elemento antecesor, sino antes o después del elemento seleccionado. Pruebe este comando primero dentro del elemento `para` y después dentro de la tabla que incluye los empleados de la compañía.

## Hipervínculo Agregar nodo

Si un elemento o atributo está incluido en el diseño pero no está presente en el documento XML, entonces aparece el hipervínculo Agregar nodo en la posición del documento donde se especifica dicho nodo. Para ver este hipervínculo, vaya a la línea que tiene el texto *Location of logo*, seleccione el nodo @href del elemento CompanyLogo y elimínelo (pulsando la tecla **Suprimir**). Paso seguido aparece el hipervínculo [agregar @href](#) dentro del elemento CompanyLogo que acaba de cambiar (*ver imagen siguiente*). Al hacer clic en el hipervínculo se añade el nodo @href al documento XML. El cuadro de texto situado dentro de las etiquetas de @href aparece porque el diseño indica que el nodo @href se debe añadir de esta manera. Es decir, todavía tiene que introducir el valor (o el contenido) del nodo @href. Escriba `nanonull.gif` en el cuadro de texto.



Si el modelo de contenido de un elemento es ambiguo, por ejemplo, si especifica que una secuencia de elementos secundarios puede aparecer en cualquier orden, entonces aparece el hipervínculo [agregar...](#) (sin el nombre del nodo). Al hacer clic en este hipervínculo aparece una lista de elementos que se pueden insertar.

**Nota:** el hipervínculo [agregar](#) Nodo aparece directamente en la plantilla del documento. No hay una opción equivalente ni en el menú contextual ni en el ayudante de entrada Elementos.

## Crear elementos nuevos con la tecla Entrar

Hay casos en los que el diseñador de la hoja de estilos dio formato de párrafo o elemento de lista a un nodo. Si sitúa el cursor dentro de un nodo de este tipo y pulsa la tecla **Entrar**, entonces se inserta otro nodo del mismo tipo después del nodo seleccionado. Pruebe este mecanismo en el documento `NanonullOrg.xml`. Para ello sitúe el cursor al final de un nodo `para` (es decir, justo antes de su etiqueta final) y pulse la tecla **Entrar**.

## Aplicar elementos

En los elementos de contenido mixto (es decir los que incluyen tanto texto como elementos secundarios) puede seleccionar parte del texto y aplicarle un elemento secundario. Como resultado el texto pasa a ser el contenido del elemento que se aplicó. Para aplicar elementos, seleccione **Aplicar** en el menú contextual y después seleccione uno de los elementos disponibles. (Si no se puede aplicar ningún elemento al texto seleccionado, entonces el comando **Aplicar** no aparece en el menú contextual.) En el ayudante de entrada Elementos los elementos que se pueden aplicar al texto seleccionado llevan el icono . En el documento `NanonullOrg.xml` seleccione parte del texto del contenido del elemento `para` y pruebe esta operación, aplicando los elementos `bold` e `italic`.

El diseñador de la hoja de estilos también puede crear un icono en la barra de herramientas para aplicar un elemento. Es el caso del documento `NanonullOrg.xml`, cuyos elementos `bold` e `italic` se pueden aplicar con solo hacer clic en los iconos **B** e **I** de la barra de herramientas de Authentic.

## Quitar nodos

Puede quitar un nodo siempre y cuando al quitarlo el documento siga siendo válido. Cuando se quita un nodo se elimina también su contenido. Para quitar un nodo seleccione el comando **Quitar** del menú contextual. Cuando el comando **Quitar** está resaltado, aparece un submenú que contiene todos los nodos que se pueden quitar, empezando por el nodo seleccionado hasta llegar al nodo de nivel superior del documento. Para seleccionar el nodo que desea quitar, coloque el cursor dentro del nodo o resalte el nodo (o parte del nodo). En el ayudante de entrada Elementos los nodos que se pueden quitar llevan el icono . También puede hacer clic en un nodo y pulsar la tecla **Suprimir**. Pruebe esta operación en el documento `NanonullOrg.xml` y quite varios nodos usando los diferentes mecanismos descritos en este párrafo. Para deshacer los cambios pulse **Ctrl+Z**.

## Borrar elementos

Puede borrar los elementos secundarios de elementos de contenido mixto (elementos que tienen texto y elementos secundarios). Para borrar el elemento entero, seleccione el nodo o coloque el cursor dentro del nodo. Para borrar un fragmento de texto, resalte el fragmento de texto. Cuando termine de seleccionar el nodo o fragmento de texto que desea borrar, seleccione la opción **Borrar** del menú contextual y después seleccione el elemento que desea borrar. En el ayudante de entrada Elementos los elementos que se pueden borrar en la posición seleccionada llevan el icono  (si la selección se hizo colocando el cursor en un punto de inserción) o el icono  (si seleccionó un intervalo). Pruebe este tipo de operación en el documento `NanonullOrg.xml`. Por ejemplo, intente borrar los elementos secundarios `bold` e `italic` del nodo `para` (que tiene contenido mixto).

## Tablas y estructura de las tablas

En la Authentic View hay dos tipos de tablas:

- *Tablas SPS (estáticas y dinámicas)*. La estructura general de una tabla SPS la define el diseñador de la hoja de estilos. Dentro de la estructura general de la tabla, los únicos cambios que se pueden realizar son cambios de contenido. Por ejemplo, puede añadir filas nuevas en una tabla SPS dinámica.
- *Tablas XML*. Puede decidir si presenta el contenido de un nodo en forma de tabla. Si el diseñador de la hoja de estilos habilitó la posibilidad de crear este nodo en forma de tabla, entonces puede definir su estructura y editar su contenido. Las tablas XML se explican en detalle en el apartado [Tablas en la Authentic View](#) <sup>76</sup>.

## 4.4 Introducir datos en la vista Authentic

En la Authentic View los datos se introducen en el documento XML desde la ventana principal. Los valores de los atributos también se pueden [introducir en el ayudante de entrada Atributos](#)<sup>48</sup>. Los datos se pueden introducir de dos maneras: (i) escribiendo el texto o (ii) mediante un dispositivo de entrada de datos, que se asigna a una entrada de texto predefinida.

### Agregar contenido de texto

En la ventana principal de la Authentic View puede introducir el contenido de los elementos y el valor de los atributos directamente en forma de texto. Para insertar contenido, sitúe el cursor en la posición deseada y teclee el texto. También puede copiar texto y pegarlo en el documento. También puede usar las teclas **Mayús** y **Suprimir**. Por ejemplo, puede resaltar el texto que desea editar y teclear el texto nuevo con la tecla **Mayús** activada.

Por ejemplo, para cambiar el nombre de la compañía, en el campo `Name` de `Office`, sitúe el cursor después de `Nanonull` y teclee `USA` para cambiar el nombre de la compañía a `Nanonull USA, Inc.`



The screenshot shows a form titled "Nanonull USA, Inc." with the following fields:

- Location: US (dropdown menu)
- Street: 119 Oakstreet, Suite 4876
- City: Vereno
- State & Zip: DC (dropdown menu) 29213

Si el texto se puede editar, entonces podrá situar el cursor en el texto y resaltarlo. De lo contrario no podrá resaltar el texto con el cursor. Por ejemplo, intente cambiar el **nombre de los campos** (no su valor) `Street`, `City` o `State/Zip` del bloque `Address`. Verá que no puede colocar el cursor en este texto porque no es contenido XML, sino que viene dado por la hoja de estilos `StyleVision Power Stylesheet`.

### Insertar caracteres especiales y entidades

En la vista Authentic los caracteres especiales y las entidades se gestionan de manera especial:

- *Caracteres especiales que se usan para marcado XML (&, ', >, <, ")*. Estos caracteres son [entidades integradas](#)<sup>49</sup> y se puede introducir en el documento con solo hacer doble clic en ellos en el ayudante de entrada Entidades. Si estos caracteres aparecen con frecuencia (en fragmentos de código, por ejemplo) puede incluirlos en secciones CDATA. Para introducir una sección CDATA, haga clic con el botón derecho en la posición elegida y seleccione **Insertar sección CDATA** en el menú contextual. El procesador XML ignora todos los caracteres de marcado que están dentro de secciones CDATA. Si desea introducir un carácter especial dentro de una sección CDATA, introduzca el carácter, no su referencia de entidad.
- *Caracteres especiales que no se pueden introducir con el teclado*. Para introducir estos caracteres cópielos y péguelos en la posición deseada.
- Puede [definir como entidades](#)<sup>94</sup> las *cadenas de texto que se usan frecuentemente*. Las entidades aparecen en el ayudante de entrada Entidades. Para [insertar una entidad](#)<sup>49</sup> sitúe el cursor en la

posición elegida y haga doble clic en la entidad en el ayudante de entrada. Esta función es muy útil porque el valor de la cadena de texto se guarda en una sola ubicación y si necesita cambiar el valor de la cadena, basta con cambiar la definición de entidad.

**Nota:** en la Authentic View es fácil pasar por alto los elementos vacíos si el marcado XML está oculto. Por eso recomendamos que [active el marcado en tamaño grande o pequeño](#) <sup>39</sup>.

Pruebe a insertar caracteres especiales y entidades tal y como acabamos de explicar.

## Agregar contenido mediante dispositivos de entrada de datos

Hasta ahora aprendió a introducir datos tecleando el texto directamente en el documento. En la Authentic View también puede introducir **contenido en los elementos** (o valores de atributo) mediante dispositivos de entrada de datos.

A continuación mostramos una lista de los dispositivos de entrada de datos de la Authentic View, con una breve explicación de cada uno de ellos.

Dispositivo	Datos en el archivo XML
Campo de entrada (cuadro de texto)	Texto introducido por el usuario
Campo de entrada multilínea	Texto introducido por el usuario
Cuadro combinado	La opción elegida por el usuario tiene asignado un valor
Casilla de verificación	La opción elegida por el usuario tiene asignado un valor
Botón de opción	La opción elegida por el usuario tiene asignado un valor
Botón	La opción elegida por el usuario tiene asignado un valor

Por ejemplo, en la tabla estática que contiene los campos de dirección (*imagen siguiente*), hay dos dispositivos de entrada de datos: un campo de entrada para el campo *Zip* y un cuadro combinado para el campo *State*. Los valores que introduzca en el campo de entrada se introducen en el elemento correspondiente del documento XML. En el cuadro combinado cada opción tiene asignado ya un valor y en el documento XML se introduce el valor de la opción seleccionada por el usuario.

Nanonull, Inc.

Location: US

Street: 119 Oakstreet, Suite 4876

City: Vereno

State & Zip: DC 29213

[Vereno Office Summary:](#) Vereno Office Summary, 15 employees.

El texto XML de la Authentic View de la imagen anterior sería:

```
<Address>
  <ipo:street>119 Oakstreet, Suite 4876</ipo:street>
  <ipo:city>Vereno</ipo:city>
  <ipo:state>DC</ipo:state>
  <ipo:zip>29213</ipo:zip>
</Address>
```

Observe que la opción DC del cuadro combinado tiene asignado el valor DC. Por el contrario, el valor del campo zip lo introduce el usuario libremente.

## 4.5 Introducir valores de atributo

Un atributo es una propiedad de un elemento y un elemento puede tener varios atributos. Los atributos tienen valores. A veces puede ser necesario introducir datos XML como valor del atributo. En la Authentic View hay dos maneras de introducir valores de atributo:

- en la ventana principal, si así lo decidió el diseñador de la hoja de estilos y
- en el ayudante de entrada Atributos.

### Valores de atributo en la ventana principal

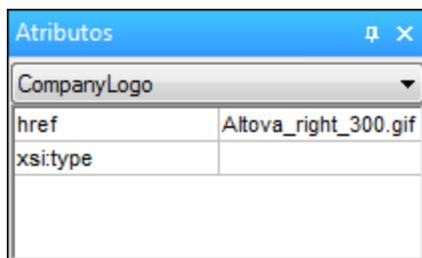
Para introducir el valor de un atributo puede escribir texto, escribir texto en un campo de entrada o seleccionar una opción de entrada que tiene asignado un valor XML. Para más información consulte el apartado [Introducir datos en la Authentic View](#)<sup>45</sup>. En estos casos la distinción entre el contenido del elemento y el valor del atributo se vale del archivo StyleVision Power Stylesheet

In such cases, the distinction between element content and attribute value is made by the StyleVision Power Stylesheet and the data is handled appropriately.

### Valores de atributo en el ayudante de entrada Atributos

También puede introducir un valor para un atributo o cambiar el valor de un atributo desde el ayudante de entrada Atributos. Primero seleccione el nodo de atributo en la Authentic View. Después introduzca o edite el valor del atributo en el ayudante de entrada Atributos. En el documento `NanonullOrg.xml` la ubicación del logotipo viene dada por el valor del atributo `href` del elemento `CompanyLogo`. Para cambiar el logotipo de la compañía utilizado en el documento siga estos pasos:

1. Haga clic en la etiqueta `CompanyLogo` para seleccionar el elemento `CompanyLogo`. Los atributos del elemento `CompanyLogo` aparecen en el ayudante de entrada Atributos.
2. En el ayudante de entrada Atributos cambie el valor `nanonull.gif` del atributo `href` por `Altova_right_300.gif` (una imagen de la carpeta `Examples`).

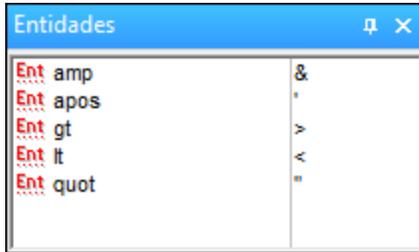


Como resultado el logotipo Nanonull desaparece y lo sustituye el logotipo de Altova.

**Nota:** en el ayudante de entrada Atributos no se pueden introducir entidades.

## 4.6 Añadir entidades

En la Authentic View las entidades suelen ser datos XML (aunque no siempre), como un solo carácter, una cadena de texto o incluso un fragmento de un documento XML. Las entidades también pueden ser archivos binarios, como archivos de imagen por ejemplo. Todas las entidades que se pueden usar en el documento aparecen en el ayudante de entrada Entidades (*imagen siguiente*). Para insertar una entidad, sitúe el cursor en la posición deseada y haga doble clic en la entidad en el ayudante de entrada Entidades. Recuerde que en el ayudante de entrada Atributos no se pueden insertar entidades.



El carácter et (&) es de especial importancia en XML (al igual que los caracteres ', >, < y "). Para insertar estos caracteres, se utilizan entidades y así no se confunden con caracteres XML. Estos caracteres son entidades predeterminadas en la Authentic View.

En el documento `NanonullOrg.xml` cambie el puesto de (*Title*) de Joe Martin (del departamento de Marketing) por *Marketing Manager Europe & Asia*. Para ello siga estos pasos:

1. Sitúe el cursor en la posición donde debe insertarse el carácter et (&).
2. En el ayudante de entrada Entidades, haga doble clic en la entidad `amp`. Como resultado se inserta el carácter et (&).

Marketing ( 2 )		
First	Last	Title
Joe	Martin	Marketing Manager Europe &
Susi	Sanna	Art Director
Employees: 2 (13% of Office, 5% of Company)		
Non-Shareholders: None.		

**Nota:** el ayudante de entrada Entidades no es contextual y muestra todas las entidades disponibles independientemente de la posición del cursor. Esto no significa que las entidades se puedan introducir en cualquier parte del documento. Por ello, después de insertar una entidad, es muy importante validar el documento: **XML | Validar (F8)**.

### Definir entidades propias

Si lo desea también puede editar sus propias entidades. Para aprender a hacerlo consulte el apartado [Definir entidades en Authentic View](#)<sup>94</sup>.

## 4.7 Imprimir el documento

Si lo desea puede imprimir la Authentic View de su documento.

Para imprimir la vista Authentic de `NanonullOrg.xml` siga estos pasos:

1. Cambie al modo **Ocultar marcado** (es necesario desactivar el marcado si no quiere que se imprima).
2. Seleccione **Archivo | Vista previa de impresión** para obtener una vista previa de todas las páginas. A continuación aparece parte de la vista previa, reducida un 50%. Observe que el formato de la página es idéntico al que tiene en la Authentic View.

Page 1 of 5

# ALTOVA

[www.altova.com](http://www.altova.com)

---

## Organization Chart

Location of logo:

---

### Nanonull, Inc.

Location:

<b>Street:</b>	119 Oakstreet, Suite 4876	<b>Phone:</b>	+1 (321) 555 5155 0
<b>City:</b>	Vereno	<b>Fax:</b>	+1 (321) 555 5155 4
<b>State &amp; Zip:</b>	<input type="text" value="DC"/> <input type="text" value="29213"/>	<b>E-mail:</b>	<a href="mailto:office@nanonull.com">office@nanonull.com</a>

**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.

3. Para imprimir el archivo, haga clic en **Archivo | Imprimir**.

Recuerde que también puede imprimir el documento con el marcado XML. Para ello, cambie la Authentic View al modo **Mostrar marcado en tamaño grande** o **Mostrar marcado en tamaño pequeño** y después imprima el documento.

## 5 Interfaz de la vista Authentic

La Authentic View se habilita haciendo clic en la pestaña Authentic del documento activo. Si no se asignó un archivo SPS al documento XML, se le pedirá que asigne uno. Para asignar un archivo SPS utilice el comando **Authentic | Asignar hoja de estilos de StyleVision**.

Esta sección incluye:

- Una presentación de la interfaz gráfica.
- Una descripción de los iconos de la barra de herramientas propios de la Authentic View.
- Una descripción de los modos de visualización disponibles en la ventana principal de la Authentic View.
- Una descripción de los ayudante de entrada e información sobre cómo usarlos.
- Una descripción de los menús contextuales disponibles en varios puntos de la Authentic View del documento XML

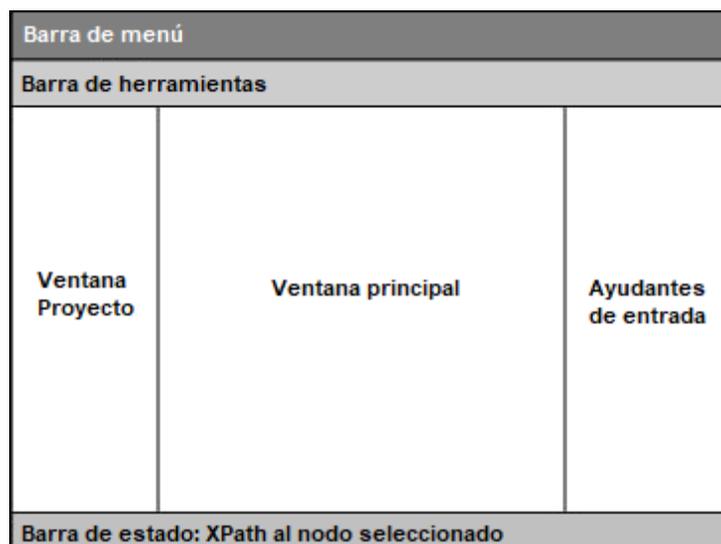
Para más información sobre la Authentic View consulte estos recursos:

- El tutorial de la Authentic View, que explica cómo usar la interfaz de la Authentic View. 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 Authentic View.

Sitio web de Altova:  [Edición de contenido XML](#), [Creación de contenido XML](#)

## 5.1 La interfaz gráfica del usuario (IGU)

La Authentic View tiene una barra de menú y una barra de herramientas situadas en la parte superior de la ventana y tres secciones: la ventana Proyecto, la ventana principal y la ventana de los ayudantes de entrada. La imagen que aparece a continuación muestra todos estos componentes.



### Barra de menú

Los menús de la barra de menú se describen en detalle en la *Referencia del usuario* del producto.

### Barra de herramientas

Los símbolos e iconos de la barra de herramientas se describen en la sección [iconos de la barra de herramientas de la Authentic View](#)<sup>54</sup>.

### Ventana Proyecto

Puede agrupar en un proyecto todos los archivos XML, XSL, XSD y de entidad. Para crear o modificar la lista de archivos del proyecto, use los comandos del menú Proyecto (descritos en la *Referencia del usuario* de su producto). La lista de archivos del proyecto aparece en la ventana Proyecto. Para tener acceso a los archivos de la ventana Proyecto, haga doble clic en el archivo que le interese.

### Ventana principal

Esta es la ventana donde se visualiza y edita el documento XML. Para más información, consulte la sección [ventana principal de la Authentic View](#)<sup>57</sup>.

### Ayudantes de entrada

La vista incluye tres ayudantes de entrada: Elementos, Atributos y Entidades. Las opciones de entrada de los ayudantes Elementos y Atributos son contextuales, es decir, dependen de la posición del cursor en el documento. Para introducir un elemento o entidad en el documento, haga doble clic en ellos en su ayudante

de entrada. El valor de un atributo se introduce en el campo de valor del atributo en el ayudante de entrada Atributos. Para más información consulte la sección [ayudantes de entrada de la Authentic View](#)<sup>60</sup>.

## Barra de estado

La barra de estado muestra la XPath al nodo que esté seleccionado en cada momento.

## Menús contextuales

Los menús contextuales son los menús que aparecen al hacer clic con el botón derecho en la ventana principal. Los comandos son comandos de edición contextuales, es decir, permiten manipular la estructura y el contenido del nodo seleccionado. Por ejemplo, puede insertar, anexar o eliminar un nodo, añadir entidades o cortar y pegar contenido.

## 5.2 Iconos de la barra de herramientas de la vista Authentic

Los iconos de la barra de herramientas de la Authentic View 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 Authentic View.

### Mostrar u ocultar marcado XML

En la Authentic View 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 Authentic View 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 Authentic View no puede configurar estas opciones.

### Editar estructuras de tablas dinámicas

Las filas de una **tabla dinámica SPS** son básicamente repeticiones de una estructura de datos. Cada fila representa una repetición de un único elemento. Cada fila, por tanto, tiene la misma subestructura XML que la fila siguiente.

Los comandos de edición de la tabla dinámica permiten manipular las filas de una tabla dinámica SPS. Es decir, puede cambiar el número de veces que aparecen los elementos y el orden en el que aparecen. Sin embargo, no puede editar las columnas de una tabla dinámica SPS porque eso supondría cambiar la subestructura de las diferentes repeticiones del elemento.

Los iconos de edición de tablas dinámicas están en la barra de herramientas y se corresponden con los comandos del menú **Authentic**.



*Anexar una fila a la tabla*



*Insertar una fila en la tabla*



*Duplicar la fila actual (es decir, el contenido de la celda se duplica)*



*Subir la fila actual*



*Bajar la fila actual*



*Eliminar la fila actual*

**Nota:** estos comandos corresponden sólo a las **tablas dinámicas SPS**. No se deben usar dentro de tablas SPS estáticas. Los distintos tipos de tablas utilizados en la Authentic View se describen en la sección [Usar tablas en la Authentic View](#)<sup>76</sup> de la presente documentación.

## Crear y editar tablas XML

Si desea presentar sus datos en forma de tabla, puede insertar sus propias tablas. Estas tablas se insertan como tablas XML. Puede modificar la estructura de la tabla XML y dar formato a la tabla. Los iconos para crear y editar tablas XML están disponibles en la barra de herramientas (*ver imagen siguiente*). Estos iconos se describen en detalle en la sección [Iconos de edición de tablas XML](#)<sup>81</sup>.



Los comandos de estos iconos **no están disponibles como opciones de menú**. Recuerde también que para poder usar tablas XML, esta función debe estar habilitada y configurada correctamente en la StyleVision Power Stylesheet.

Para más información sobre los tipos de tablas utilizados en la Authentic View y sobre cómo crear y editar tablas XML, consulte la sección [Usar tablas en la Authentic View](#)<sup>76</sup>.

## Iconos de formato de texto

Para dar formato al texto en la Authentic View es necesario aplicar el elemento o atributo XML que contiene el formato deseado. Si el formato deseado ya se definió, el diseñador de la StyleVision Power Stylesheet puede incluir iconos para dicho formato en la barra de herramientas de la Authentic View. Para dar formato al texto, seleccione el texto y pulse el icono correspondiente.

## Iconos de navegación por filas de base de datos



Estos iconos significan, de izquierda a derecha: *Ir al primer registro de la BD*, *Ir al registro anterior*, *Abrir el cuadro de diálogo Ir al registro número*, *Ir al registro siguiente* e *Ir al último registro*.



Este icono abre el cuadro de diálogo *Editar la consulta de base de datos*, donde puede escribir una consulta. La Authentic View 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 Authentic View, donde se puede editar. Después puede guardar la fila otra vez en la base de datos.

## Botones de la barra de herramientas para formularios XML portátiles (PXF)

La Authentic View de Authentic Desktop incluye estos botones en la barra de herramientas para archivos PXF:



Al hacer clic en cada uno de estos botones, se genera un archivo de salida en formato HTML, RTF, PDF y DocX respectivamente.

Estos botones se habilitan cuando se abre un archivo PXF en la Authentic View y de acuerdo con la configuración del archivo PXF. Por ejemplo, si el archivo PXF se configuró para contener una hoja de estilos XSLT para formatos de salida HTML y RTF, entonces se habilitarán los botones HTML y RTF, pero los botones PDF y DocX estarán deshabilitados.

## 5.3 Ventana principal de la vista Authentic

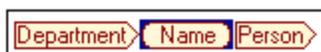
La Authentic View ofrece cuatro modos de visualización: marcado grande, marcado pequeño, marcado mixto y marcado oculto. Estos modos permiten visualizar la información de marcado del documento de diferentes maneras. Para cambiar de un modo a otro, utilice los comandos del menú Authentic o los iconos de la barra de herramientas (ver el apartado anterior [Iconos de la barra de herramientas de la Authentic View](#)<sup>54</sup>).

### 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:

**Nanonull, Inc.**

Location: **US**

<p><b>Street:</b> 119 Oakstreet, Suite 4876</p> <p><b>City:</b> Vereno</p> <p><b>State &amp; Zip:</b> DC 29213</p>	<p><b>Phone:</b> +1 (321) 555 5155 0</p> <p><b>Fax:</b> +1 (321) 555 5155 4</p> <p><b>E-mail:</b> <a href="mailto:office@nanonull.com">office@nanonull.com</a></p>
--	--

**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 Authentic View ve este marcado personalizado en el modo de marcado mixto.

## Ocultar marcado

Se oculta el marcado XML. Puesto que el formato visible en la Authentic View es el formato del documento impreso, este modo de visualización es una vista WYSIWYG del documento.

## Visualización de contenidos

En la Authentic View 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

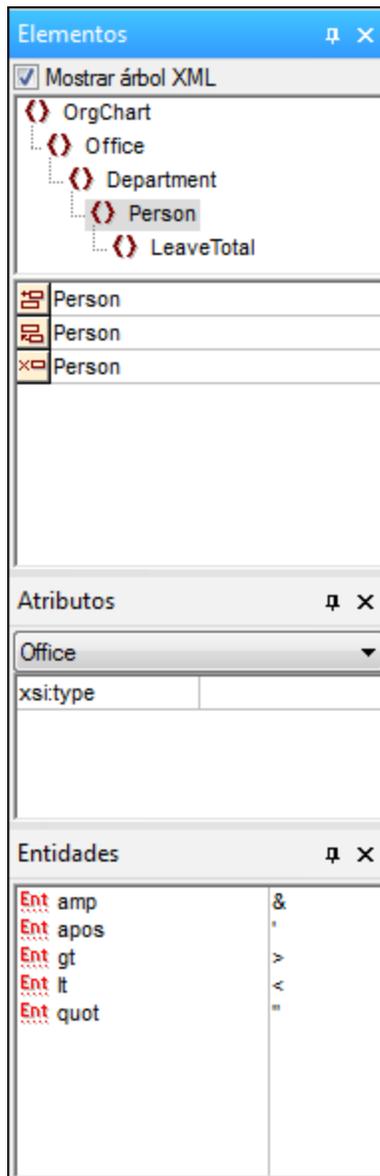
Cuando un elemento o atributo es opcional (de acuerdo con el esquema al que se hace referencia), aparece un aviso de tipo `agregar [nombre del elemento/atributo]`:

`agregar Shares`

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.

## 5.4 Ayudantes de entrada de la vista Authentic

La Authentic View 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 Authentic View (*imagen siguiente*).



Los ayudantes de entrada Elementos y Atributos son contextuales, es decir, sus opciones dependen de la posición del cursor en el documento. Las entidades del ayudante de entrada Entidades no son contextuales, es decir, el ayudante muestra todas las entidades permitidas en el documento, independientemente de la posición del cursor.

A continuación describimos los ayudantes de entrada uno por uno.

## Ayudante de entrada Elementos

El ayudante de entrada Elementos está dividido en dos paneles:

- Un panel superior con el árbol XML (que se puede activar o desactivar haciendo clic en la casilla **Mostrar árbol XML**). El árbol XML muestra los antecesores del elemento seleccionado hasta la altura del elemento raíz del documento. Al hacer clic en un elemento del árbol XML, los elementos relevantes 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 Authentic View. Los nodos de esta lista van acompañados por un icono que indica el tipo de acción que se puede realizar. A continuación describimos cada uno de estos iconos y su significado.

Para utilizar un nodo del ayudante de entrada Elementos, haga clic en su icono.



### Insertar después del elemento

El elemento del ayudante de entrada se inserta después del elemento seleccionado y se anexa en el nivel jerárquico correcto. Por ejemplo, si el cursor está dentro de un elemento `//sect1/para` y anexa un elemento `sect1`, el elemento `sect1` nuevo no se anexará como elemento del mismo nivel que `//sect1/para` sino como elemento del mismo nivel que `sect1`, que es el elemento primario del elemento `para`.



### Insertar antes del elemento

El elemento del ayudante de entrada se inserta antes del elemento seleccionado. Tenga en cuenta que, al igual que el comando **Insertar después del elemento**, el elemento se inserta en el nivel jerárquico correcto.



### Quitar elemento

Quita el elemento y su contenido.



### Insertar elemento

También puede insertar un elemento del ayudante de entrada dentro de otro elemento. Cuando el cursor está dentro de un elemento, entonces se pueden insertar los elementos secundarios permitidos de ese elemento. Recuerde que los elementos secundarios permitidos pueden formar parte de un modelo de contenido que sólo contiene elementos o de un modelo de contenido mixto (con elementos y texto).

Para insertar un elemento secundario permitido, seleccione un intervalo de texto o sitúe el cursor en una posición dentro del texto (el punto de inserción).

- Cuando se selecciona un intervalo de texto y se inserta un elemento, el intervalo de texto se convierte en el contenido del elemento insertado.
- Cuando se inserta un elemento en un punto de inserción, se inserta el elemento en ese punto.

Después de insertar el elemento, puede borrarlo con solo hacer clic en el icono **Borrar elemento** del ayudante de entrada Elementos, disponible para estos elementos insertados.

El icono **Borrar elemento** puede ser de dos tipos, dependiendo de si selecciona un intervalo de texto o sitúa el cursor en el texto para indicar un punto de inserción (véase la descripción que aparece más abajo).



### Aplicar elemento

Si selecciona un elemento del documento (active el modo de vista "Mostrar marcado en tamaño grande" y seleccione la etiqueta inicial o final del elemento) y ese elemento se puede sustituir por otro elemento (p. ej. en un elemento de contenido mixto como `para`, el elemento `italic` se puede sustituir por el elemento `bold`), este icono indica que el elemento del ayudante de entrada se puede aplicar al elemento seleccionado (el elemento original). El comando **Aplicar elemento** también se puede aplicar a un intervalo de texto de un elemento de contenido mixto. El intervalo de texto se convertirá en el contenido del elemento que se aplicó.

- Si el elemento que se aplicó tiene un **secundario con el mismo nombre** que el secundario del elemento original y existe una instancia de este secundario en el elemento original, entonces se mantiene el secundario del elemento original en el contenido del elemento nuevo.
- Si el elemento que se aplicó **no tiene un secundario con el mismo nombre** que el secundario del elemento original y existe una instancia del secundario en el elemento original, entonces el secundario del elemento original se anexa como elemento del mismo nivel de los secundarios del elemento nuevo.
- Si el elemento que se aplicó tiene un **secundario para el que no hay un equivalente** en el modelo de contenido del elemento original, entonces este elemento secundario no se crea automáticamente, aunque la Authentic View 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.



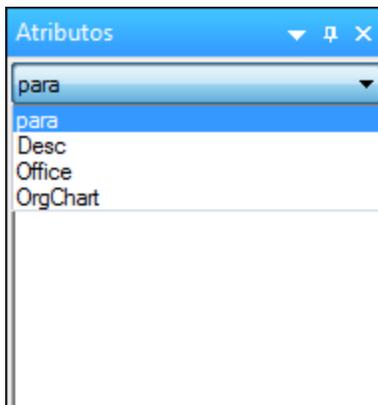
### Borrar elemento (cuando se selecciona un punto de inserción)

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.

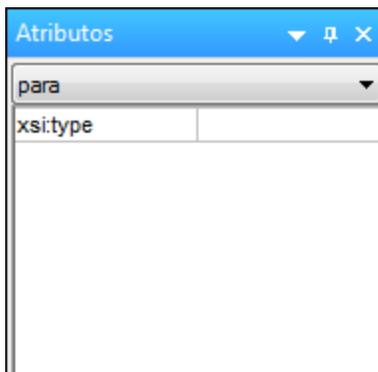
## Ayudante de entrada Atributos

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 al 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`.)



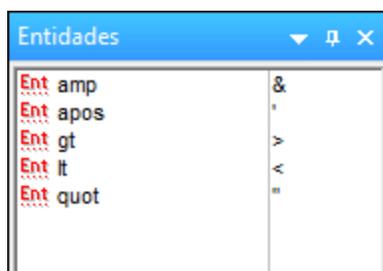
Para introducir el valor de un atributo, haga clic en el campo de valor del atributo y escriba el valor. Como resultado se crea el atributo y su valor en el documento XML.

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.



**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 Authentic View **muestra el valor de la entidad**. Si la entidad es externa, la Authentic View muestra la entidad, pero no su valor. Esto significa que, por ejemplo, en la Authentic View un archivo XML que sea una entidad externa se mostrará como entidad. Su contenido no sustituirá a la entidad en la Authentic View.

En la Authentic View también puede definir entidades nuevas, que aparecerán también en el ayudante de entrada. Para más información consulte el apartado [Definir entidades](#)<sup>94</sup> de la sección *Edición en la Authentic View*.

## 5.5 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 Authentic View](#)<sup>60</sup> 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](#)<sup>94</sup>.

### 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 Authentic View](#)<sup>60</sup>.

## 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 Authentic View (que se copió en el portapapeles) como XML o como texto. Si el fragmento copiado se pega como XML, se pega e incluye el marcado XML. Si se pega como texto, entonces se pega sólo el contenido de texto del fragmento copiado (sin el marcado XML). Se pueden dar estos dos casos:

- Se selecciona **todo un nodo y sus etiquetas de marcado** en la Authentic View 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 Authentic View 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.

## 6 Edición en la vista Authentic

En esta sección describimos con detalle algunas funciones de la Authentic View que consideramos importantes. Estas funciones son importantes bien porque se utilizan con mucha frecuencia o bien porque implican una cierta complejidad y deben ser explicadas en más detalle.

Los temas que describimos en esta sección son:

- En la Authentic View se pueden usar tres tipos de tablas diferentes. El apartado [Tablas en la Authentic View](#)<sup>76</sup> describe los tres tipos de tablas (tablas SPS estáticas, tablas SPS dinámicas y tablas XML) así como cuándo y cómo se utilizan. Este apartado empieza con una descripción general y después explica detalladamente el uso de las tablas.
- El selector de fechas es un calendario gráfico con el que puede introducir fechas en el formato XML correcto con solo hacer clic en una fecha. Consulte el apartado titulado [Selector de fecha](#)<sup>91</sup>.
- 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](#)<sup>94</sup>.
- En las ediciones Enterprise de los productos de Altova, el usuario de la Authentic View puede firmar los documentos XML con [firmas XML digitales](#)<sup>96</sup> y verificar las firmas.
- Qué [formatos de imagen](#)<sup>98</sup> pueden visualizarse en la Authentic View.

Sitio web de Altova: [🔗 Edición de contenido XML](#), [Creación de contenido XML](#)

## 6.1 Copia de seguridad automática de archivos

Si modifica archivos con **Authentic Desktop**, la aplicación guarda una copia de seguridad automáticamente a intervalos regulares. En la pestaña Archivo del cuadro de diálogo "Opciones" ([Herramientas | Opciones | Archivo](#)<sup>279</sup>) que se ve en la imagen siguiente, puede:

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



### Indicadores

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



#### Guardada/No guardada

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

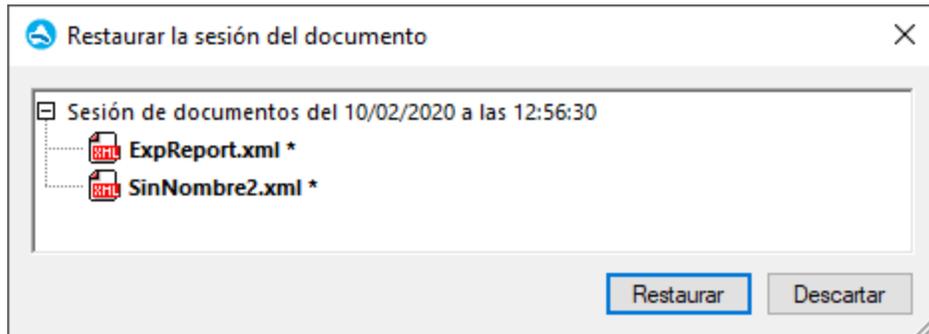
#### Estado de la copia de seguridad

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

- **Amarillo:** el archivo se ha modificado, pero no se ha guardado o no hay una copia de seguridad.
- **Verde:** se ha hecho una copia de seguridad del archivo y este no se ha modificado desde entonces pero tampoco se ha guardado (si se guarda el círculo desaparece).
- **Rojo:** no se puede hacer una copia de seguridad de este archivo o ha ocurrido un error al intentar hacerla.
- **Gris:** la función de copia de seguridad automática se ha deshabilitado (*desde el cuadro de diálogo "Opciones"*<sup>278</sup>; véase más arriba). La presencia del círculo indica que el archivo no se ha guardado desde la última modificación (si se hubiera guardado no habría ningún círculo).

### Restaurar una copia de seguridad

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



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

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

Desde este cuadro de diálogo puede:

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

## 6.2 Edición básica

En la Authentic View puede editar documentos XML. Sin embargo, la Authentic View 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 Authentic View sin marcado XML.**

Address ipo:city **Vereno** ipo:city Address **Office Summary:**  
**4 departments, 16 employees.** Desc para

The company was established **bold** in Vereno in 1995 **bold** 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 *italic* *NanoSoft Development Suite* *italic* 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.

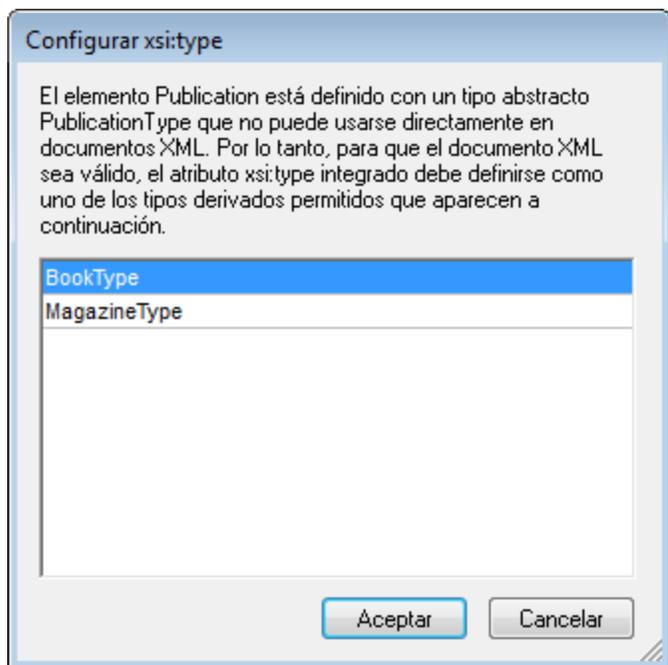
para para

**Un documento editable en la Authentic View 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 Authentic View 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 Authentic View viene con formato previo, el usuario no puede agregar líneas ni espacios entre los componentes. Por tanto, en la Authentic View la tecla **Entrar** sirve para anexar otra instancia del elemento que se está editando y sólo se puede usar para esto.

## Copiar como XML o como texto

Puede copiar y pegar texto como XML o como texto.

- Si el texto se pega como XML, entonces se pega el marcado XML y el contenido de texto de los nodos. El marcado XML se pega incluso si solo copió parte del contenido del nodo. Para poder pegar el marcado es necesario que el esquema permite insertar el nodo en la posición elegida.
- Si el texto se pega como texto, entonces no se pega el marcado XML.

Para pegar el texto como XML o como texto, antes debe copiar el texto (**Ctrl+C**), hacer clic con el botón derecho en la posición donde desea pegarlo y seleccionar **Pegar como | XML** o **Pegar como | Texto** en el menú contextual. Si utiliza **Ctrl+V** para pegar el texto, este se pega en el modo predeterminado de la hoja de estilos SPS. El modo predeterminado lo especifica el diseñador de la hoja de estilos SPS. Para más información, consulte el apartado [Menús contextuales](#) <sup>65</sup>.

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 Authentic View el documento XML se presenta por medio de la hoja de estilos. Esto significa que todo el formato que ve en la Authentic View 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 Authentic View, contiene únicamente el contenido, pero no formato. El formato está definido en la hoja de estilos. Esto significa que en la Authentic View 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 Authentic View, debe introducir la cadena de texto dentro del marcado adecuado. Para más información, consulte el comando **Insertar elemento** del apartado [Ayudante de entrada Elementos](#)<sup>61</sup>.

## Usar RichEdit en la Authentic View

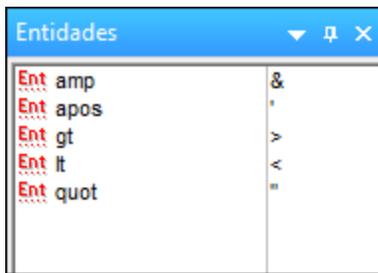
En la Authentic View los botones y controles de la barra de herramientas RichEdit (*imagen siguiente*) se habilitan cuando se coloca el cursor dentro de un elemento que se creó como componente RichEdit. De lo contrario, las opciones de esta barra de herramientas aparecen atenuadas.



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 Authentic View 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](#)<sup>63</sup> (*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](#)<sup>94</sup> en la Authentic View directamente. Después de definir las, estas entidades aparecen en el [ayudante de entrada Entidades](#)<sup>63</sup> 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 (`&gt;`). 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 Authentic View, [active la visualización del marcado](#)<sup>54</sup>. 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 Authentic View.

### 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 editelo. El destino del hipervínculo no se puede editar. (El destino del hipervínculo lo define el diseñador de la hoja de estilos, escribiendo la dirección de destino estática o derivando la dirección de destino de los datos que contiene el documento XML). Para ir al destino del hipervínculo desde la Authentic View 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.)

## 6.3 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](#)<sup>76</sup> que aparece a continuación explica las características de este tipo de tablas.

Lo que sí puede insertar el usuario de la Authentic View 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](#)<sup>77</sup> y de los [iconos de edición para tablas CALS/HTML](#)<sup>81</sup> se describen en los apartados siguientes.

### 6.3.1 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.	
<b>Street:</b>	119 Oakstreet, Suite 4876
<b>City:</b>	Vereno
<b>State &amp; Zip:</b>	DC 29213
<b>Phone:</b>	+1 (321) 555 5155
<b>Fax:</b>	+1 (321) 555 5155 - 9
<b>E-mail:</b>	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	<a href="#">add Shares</a>	25	7	18
<b>Employees: 3 (20% of Office, 9% of Company)</b>					<b>Shares: 1500 (13% of Office, 6% of Company)</b>			
<b>Non-Shareholders: Frank Further, Loby Matise.</b>								

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.

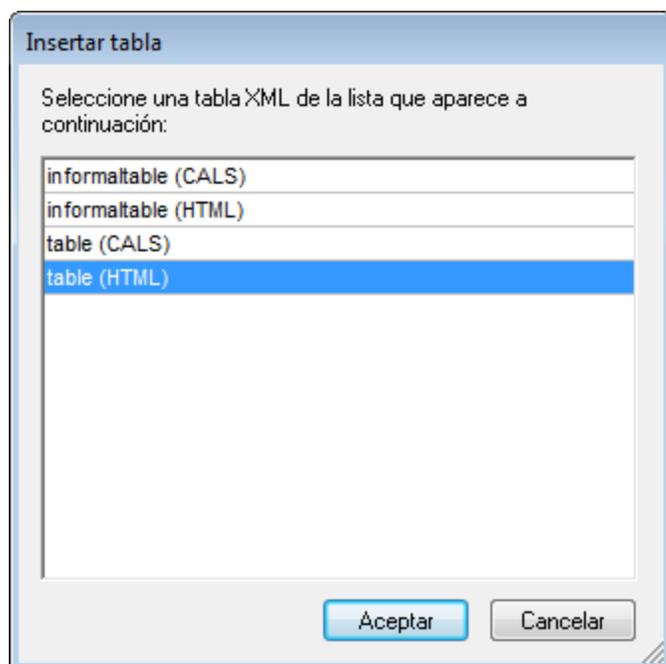
## 6.3.2 Tablas CALS/HTML

El usuario de la Authentic View 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](#) <sup>81</sup>).

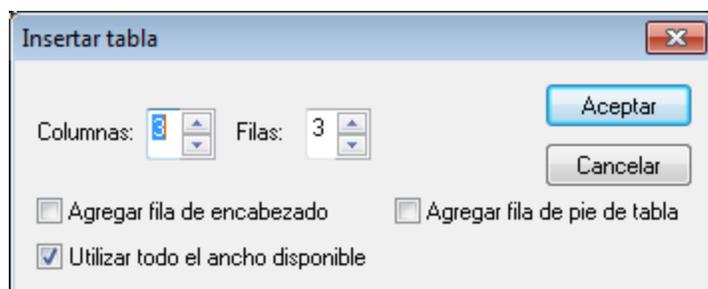
### Insertar tablas

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.

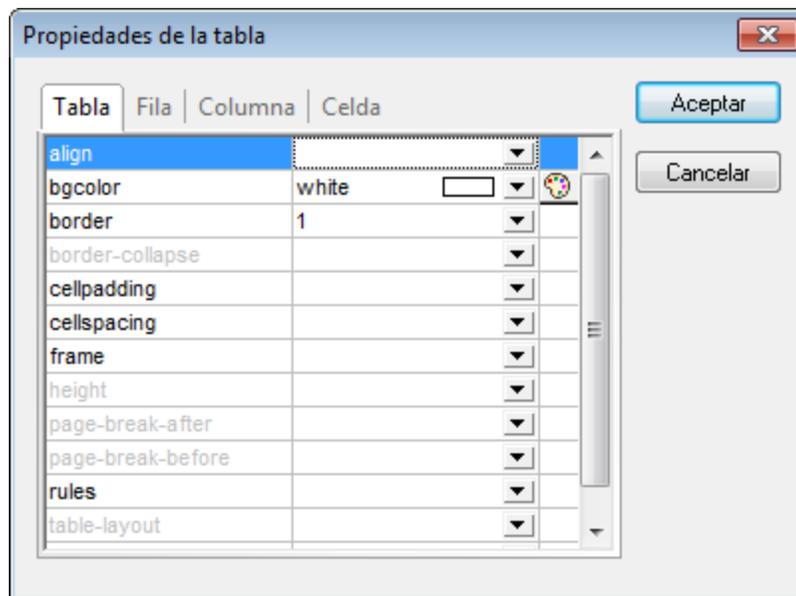
### Cómo dar formato a las tablas e introducir datos

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 Authentic View.
- 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 Authentic View.

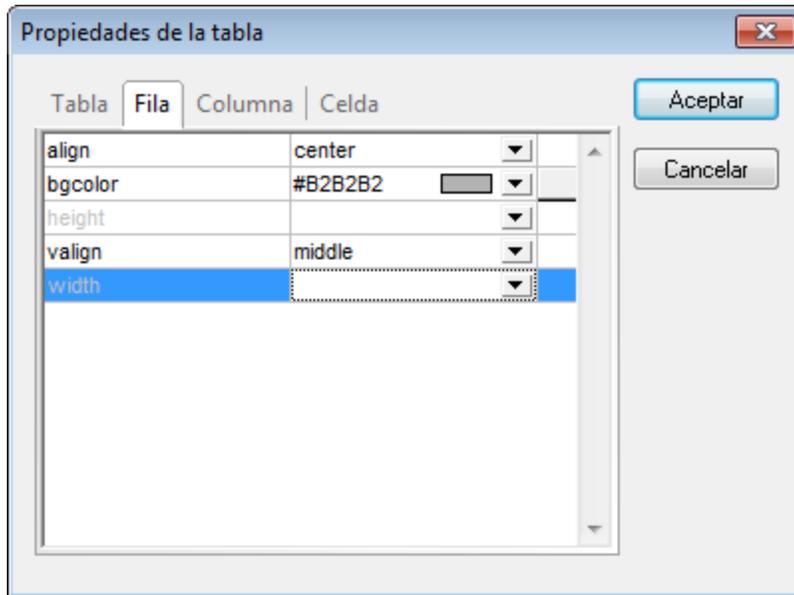
Siga estos pasos para dar formato a una tabla, fila, columna o celda:

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`.

- 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

- 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](#) <sup>81</sup>.

### 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 Authentic View, es necesario crear en StyleVision una plantilla global con formato de texto para el elemento en cuestión.

## 6.3.3 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](#) <sup>77</sup>.

#### Insertar tabla



Este comando inserta una **tabla CALS/HTML** en la posición del cursor.

#### Eliminar tabla



Este comando elimina la tabla activa.

#### Anexar fila



Este comando anexa una fila al final de la tabla activa.

#### Anexar columna



Este comando anexa una columna al final de la tabla activa.

### Insertar fila



Este comando inserta una fila encima de la posición del cursor en la tabla activa.

### Insertar columna



Este comando inserta una columna a la izquierda de la posición del cursor en la tabla activa.

### Combinar celda hacia la izquierda



Este comando une la celda actual (donde se encuentra el cursor) con la celda de la izquierda. La celda nueva incluye las etiquetas de ambas celdas, el contenido de las celdas se encadena y los encabezados siguen como estaban.

### Combinar celda hacia la derecha



Este comando une la celda actual (donde se encuentra el cursor) con la celda de la derecha. El contenido de las celdas se encadena en la celda nueva.

### Combinar celda hacia abajo



Este comando une la celda actual (donde se encuentra el cursor) con la celda situada justo debajo. El contenido de las celdas se encadena en la celda nueva.

### Combinar celda hacia arriba



Este comando une la celda actual (donde se encuentra el cursor) con la celda situada justo encima. El contenido de las celdas se encadena en la celda nueva.

### Dividir horizontalmente



Este comando crea una celda nueva a la derecha de la celda activa. El tamaño de las celdas resultantes es el mismo que el de la celda original.

### Dividir verticalmente



Este comando crea una celda nueva debajo de la celda activa.

### Alinear arriba



Este comando alinea el contenido en la parte superior de la celda.

### Centrar verticalmente



Este comando alinea el contenido en el centro de la celda.

### Alinear abajo

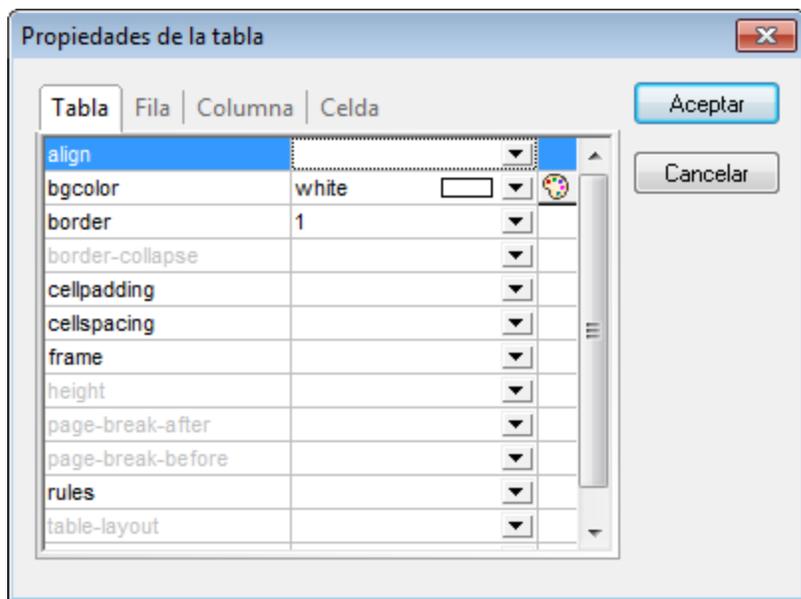


Este comando alinea el contenido en la parte inferior de la celda.

### Propiedades de la tabla



Este comando abre el cuadro de diálogo "Propiedades de la tabla". Este icono solo está habilitado para las tablas HTML, no para las tablas CALS.



## 6.4 Editar una BD

En la Authentic View 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 Authentic View para que el diseño tenga un aspecto más compacto. Si así fuera, en la Authentic View 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](#)<sup>84</sup>).
- Puede [consultar la BD](#)<sup>85</sup> 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](#)<sup>89</sup>.

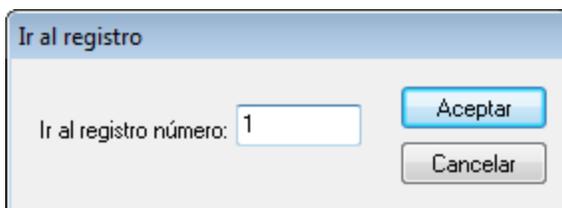
**Nota:** en la Authentic View los datos procedentes de bases de datos SQLite no se pueden editar. Si intenta guardar datos SQLite desde la Authentic View, aparece un cuadro de mensaje informando de esta restricción.

### 6.4.1 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 Authentic View. 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 Authentic View 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](#)<sup>241</sup>.

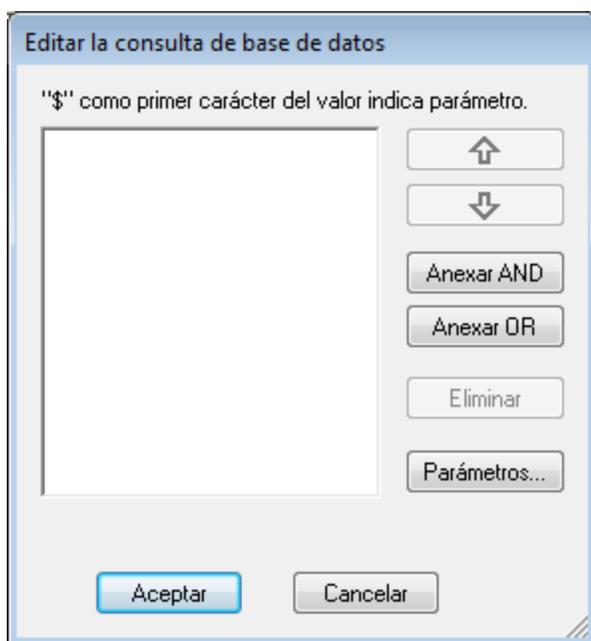
## 6.4.2 Consultas de BD

Las consultas de BD sirven para consultar los registros de una tabla de la Authentic View. 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 Authentic View, 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.

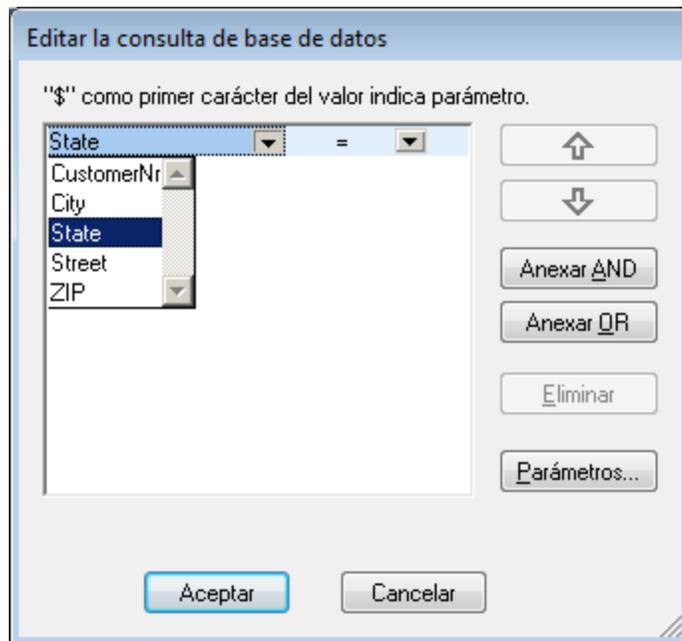
**Nota importante:** si recibe un mensaje diciendo que hay demasiadas tablas abiertas, puede reducir el número de tablas que están abiertas con ayuda de una consulta.

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](#) <sup>86</sup>.
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](#) <sup>87</sup>.

### 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

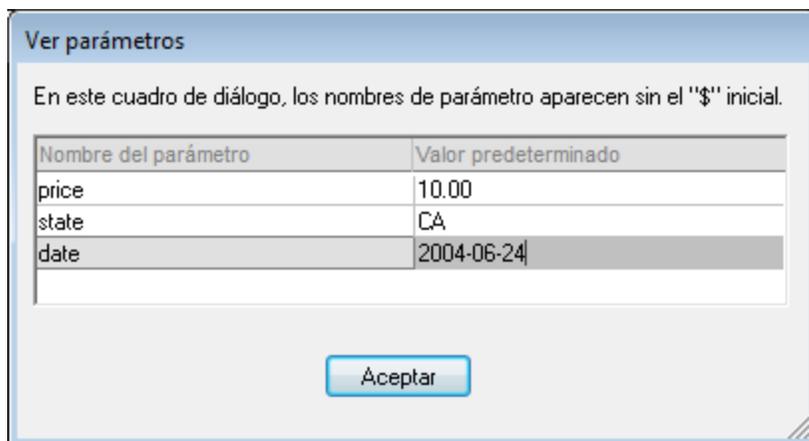
Si selecciona la opción IS NULL o NOT NULL, el campo Valor se deshabilita. Los **valores** deben introducirse sin comillas (ni ningún otro delimitador). También es necesario que los valores tengan el mismo formato que su

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).

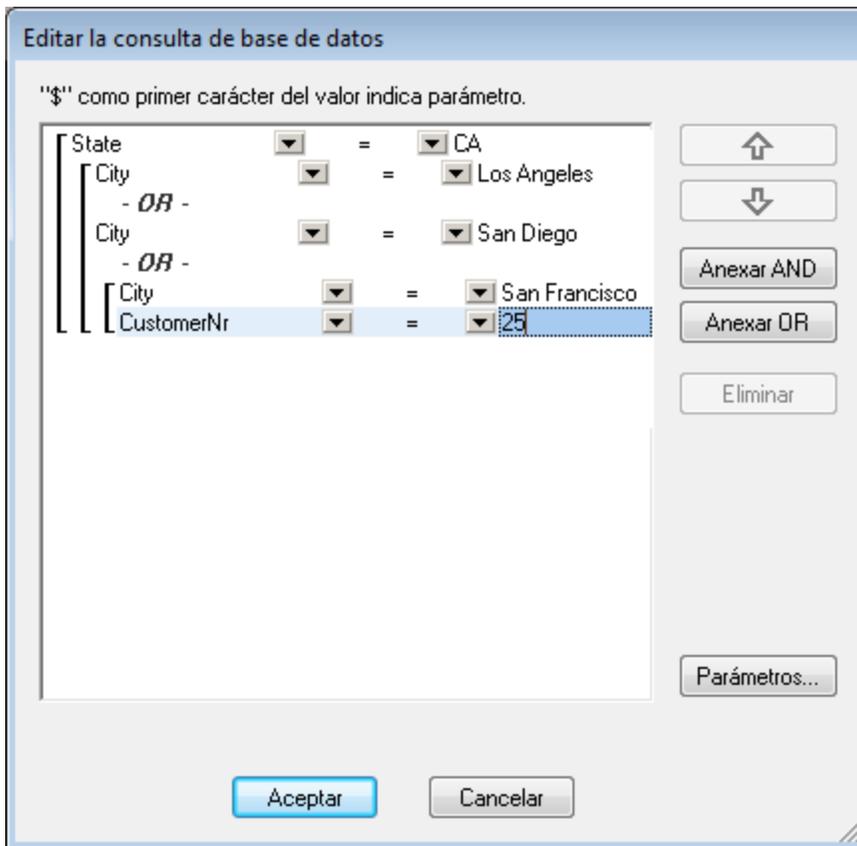
Para ver qué parámetros se definieron en el archivo SPS pulse el botón **Parámetros** del cuadro de diálogo "Editar la consulta de base de datos". Como resultado aparece el cuadro de diálogo "Ver parámetros" (*imagen siguiente*).



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.



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**. Los datos de la BD se recargan automáticamente en la vista Authentic, para reflejar los cambios realizados en la consulta de BD.

## 6.4.3 Modificar una tabla de BD

### Agregar un registro

Para agregar un registro a una tabla de BD:

1. Ponga el cursor en la fila de la tabla de BD y haga clic en el icono  (para anexar una fila) o en  (para insertar una fila). Como resultado se crea un registro nuevo en el archivo XML temporal.
2. Haga clic en el comando **Archivo | Guardar** para agregar el registro nuevo en la BD. En la vista Authentic se anexa una fila nueva para el registro nuevo en la tabla de BD. El valor del campo `AltovaRowStatus` de este registro pasa a ser `A` (de "Added", es decir, agregado).

Los datos que introduzca para el registro nuevo aparecen en negrita y subrayados. Así puede distinguir los registros agregados de los registros ya existentes (siempre que no se aplicase ese mismo formato a ningún registro ya existente). Los errores de tipos de datos aparecen en color rojo para que pueda detectarlos inmediatamente.

El nuevo registro se agrega a la BD nada más hacer clic en **Archivo | Guardar**. Una vez agregado el nuevo registro a la BD, su campo `AltovaRowStatus` se inicializa (se indica con ---) y el registro aparece en la vista Authentic como cualquier otro registro.

### Modificar un registro

Para modificar un registro, ponga el cursor en el registro de la tabla de BD y editelo. 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](#)<sup>84</sup>).

Cuando se modifica un registro, las entradas de todos los campos del registro se subrayan y el valor del campo `AltovaRowStatus` de las instancias principales de este registro pasa a ser `U` (de "Updated", es decir, actualizado). En las instancias secundarias de este registro el valor del campo `AltovaRowStatus` pasa a ser `u` (en minúsculas). Las instancias principales y secundarias de un registro están definidas por la estructura de la BD (y del esquema XML generado a partir de ella). Por ejemplo, si se incluye una tabla Dirección en una tabla Cliente, entonces la tabla Dirección puede aparecer en el documento de diseño en dos tipos de instancias: como tabla Dirección propiamente dicha y dentro de las instancias de la tabla Cliente. El tipo de instancia que se modifique es el tipo principal. Los demás tipos son los tipos secundarios. Los errores de tipo de datos aparecen en color rojo para que pueda detectarlos inmediatamente.

Los cambios se guardan en la BD al hacer clic en **Archivo | Guardar**. Después de guardar en la BD un registro en el que se realizaron cambios, su campo `AltovaRowStatus` se inicializa (se indica con ---) y el registro aparece en la vista Authentic como cualquier otro registro.

**Nota importante:**

- Si se modifica un solo campo del registro en la vista Authentic, al guardar los datos en la BD se actualiza el registro entero.
- El valor de fecha 0001-01-01 se define como valor NULL en algunas BD, lo cual podría dar lugar a errores.

## Eliminar un registro

Para eliminar un registro:

1. Ponga el cursor en la fila del registro que desea eliminar y haga clic en el icono . El registro que se va a eliminar aparece tachado. El campo `AltovaRowStatus` toma el valor `D` en las instancias principales del registro, `d` en las instancias secundarias y `X` en los registros eliminados de forma indirecta. Los registros eliminados de forma indirecta son aquellos campos del registro eliminado que se almacenan en otra tabla diferente. Por ejemplo, la tabla Dirección puede estar incluida en una tabla Cliente. Si se elimina un registro de Cliente, su registro Dirección correspondiente se elimina de forma indirecta. Si se elimina un registro Dirección de la tabla Cliente, entonces el registro Dirección de la tabla Cliente se elimina de forma principal, pero el mismo registro se elimina de forma secundaria en la tabla independiente Dirección si se creó una instancia de esta tabla.
2. Haga clic en **Archivo | Guardar** para guardar los cambios en la BD.

**Nota importante:** después de guardar los cambios en la BD el comando **Deshacer** se restablece. Es decir, ya no podrá deshacer las acciones llevadas a cabo antes de guardar los cambios.

## 6.5 Trabajar con fechas

En la vista Authentic las fechas se pueden editar de dos maneras diferentes:

- Con ayuda del [selector de fecha](#)<sup>91</sup>.
- O [escribiendo el valor](#)<sup>92</sup> 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.

### 6.5.1 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*).

Location of logo:

Last Updated: 2003-09-01

**Nanonull, Inc.**

Location:

septiembre							2003
l	m	m	j	v	s	d	
25	26	27	28	29	30	31	35
1	2	3	4	5	6	7	36
8	9	10	11	12	13	14	37
15	16	17	18	19	20	21	38
22	23	24	25	26	27	28	39
29	30	1	2	3	4	5	40

Hoy Sin zona horaria

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.

## 6.5.2 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.

Invoice Number: 001 2006-03-10 Customer: The ABC Company Invoice Amount: 40.00
---

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*).

<h1>Invoice</h1> <hr/> <p> <b>Invoice Number:</b> 45656    <b>Invoice Date:</b> 2007-13-14  <b>Customer Name:</b> John Smith         </p> <p>validación: Valor no válido para el tipo de datos date en el elemento InvoiceDate</p>
--

Si intenta cambiar el formato de la fecha, la fecha se vuelve de color rojo para informarle del error (*ver imagen*).

Invoice Number: 001  
2006/03/10  
Customer: The ABC Company  
Invoice Amount: 40.00

## 6.6 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.

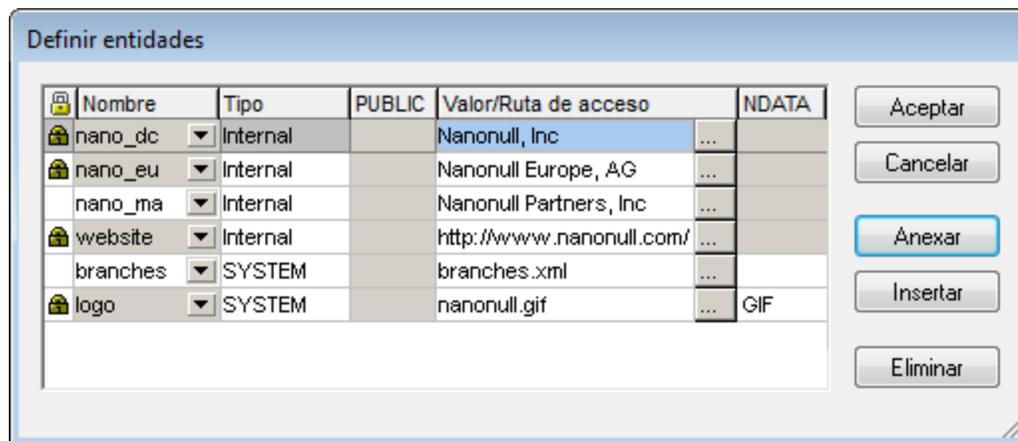
### Tipos de entidades

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 `&amp;`.
- 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).

## 6.7 Firmas XML

Los archivos SPS se pueden diseñar con una firma XML configurada para la vista Authentic. Cuando se habilita una firma XML en el SPS, el usuario de la vista Authentic puede firmar el archivo XML de Authentic digitalmente con la firma habilitada. Si se realizan cambios en el documento después de firmarlo, la verificación de la firma dará error. Cuando un documento XML de Authentic firmado se abre en la Authentic View, el proceso de verificación se ejecuta automáticamente en el documento y el resultado de la verificación aparece en una ventana.

**Nota:** las firmas XML se pueden usar y verificar en la vista Authentic de las ediciones Enterprise y Professional de estos productos de Altova: Authentic Desktop, Authentic Browser, XMLSpy y StyleVision.

### Operaciones con firmas XML

Estas son las operaciones con firmas XML que el usuario de la vista Authentic puede realizar:

- *Elegir el certificado o la contraseña:* las firmas se autentican con un certificado o con una contraseña. Es necesario indicar el objeto de autenticación (el certificado o la contraseña) tanto al crear la firma como al verificarla. Si el documento XML de Authentic tiene asignado un SPS en el que se habilitó la firma XML, el SPS puede especificar un certificado o una contraseña predeterminados para la firma. Se especifique o no el certificado o la contraseña predeterminados, la firma se puede configurar para que el usuario de la Authentic View pueda seleccionar su propio certificado o contraseña. El usuario de la Authentic View puede seleccionar su certificado o contraseña en el cuadro de diálogo "Firma XML" (*imagen siguiente*). Al seleccionar un certificado o contraseña, se anula el certificado o contraseña predeterminados. El certificado o la contraseña del usuario se almacena en memoria y se usa en la sesión actual. Si, después de seleccionar el certificado o la contraseña, el usuario de la Authentic View cierra el archivo o la aplicación, el SPS devuelve el certificado o la contraseña a su configuración predeterminada.
- *Firmar el documento:* el documento XML de Authentic se puede firmar automática o manualmente. La opción de firma automática se especifica en la configuración de la firma en el archivo SPS y hace que el documento XML de Authentic se firme automáticamente cada vez que se guarda. Si la opción de firma automática no está activada, entonces puede firmar el documento manualmente. Para ello, haga clic en el icono de firma XML  de la barra de herramientas o en el comando **Authentic | Firma XML**. Aparece el cuadro de diálogo "Firma XML" (*imagen anterior*). Haga clic en el botón **Firmar documento** de este cuadro de diálogo. Tenga en cuenta que si firma el documento con una firma incrustada, es necesario que el esquema XML autorice el uso del elemento `Signature` como último elemento secundario del elemento raíz. De lo contrario, el documento no será válido. Al firmar el documento, el objeto de autenticación y la ubicación de la firma vienen determinadas por la configuración de la firma. Asegúrese de que tiene la información necesaria sobre la firma y hable con el diseñador de su SPS.
- *Verificar el documento XML de Authentic:* si un SPS tiene firmas XML habilitadas, el proceso de verificación se ejecuta cada vez que se carga el documento XML en la Authentic View. Si la contraseña o la información de clave del certificado no está guardada con el SPS y con la firma respectivamente, el usuario de la Authentic View debe introducir la contraseña o seleccionar un certificado para poder verificar el documento. Recuerde que si se genera una firma incrustada, la firma se guarda en el archivo XML cuando este se guarda. La firma XML generada debe quitarse (utilice el botón **Eliminar firma** del cuadro de diálogo "Firma XML", *ver imagen anterior*) si no quiere guardarla

con el archivo XML. Asimismo, si se genera una firma separada y no desea guardarla con el archivo XML también debe quitarla.

## 6.8 Imágenes en la Authentic View

En la Authentic View 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 Authentic View 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.

## 6.9 Teclas de acceso rápido en la Authentic View

### Tecla Entrar

En la Authentic View 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**.

## 7 Scripting en Authentic

La característica Scripting de Authentic ofrece una mayor flexibilidad e interacción en los diseños SPS. Estos diseños se pueden crear o editar en las ediciones StyleVision Enterprise y Professional y se pueden ver en la vista Authentic de las ediciones Enterprise y Professional de los productos de Altova.

A continuación puede ver qué productos de Altova son compatibles con esta característica. No obstante, tenga en cuenta que en la versión de confianza del complemento Authentic Browser para exploradores la característica de scripting interna está desactivada por motivos de seguridad.

Producto	Creación de scripts de Authentic	Scripts de Authentic habilitados
StyleVision Enterprise	Sí	Sí
StyleVision Professional	Sí	Sí
StyleVision Basic *	No	No
XMLSpy Enterprise	No	Sí
XMLSpy Professional	No	Sí
AuthenticDesktop Enterprise	No	Sí
Authentic Browser Plug-in Enterprise Trusted **	No	Sí
Authentic Browser Plug-in Enterprise Untrusted	No	Sí

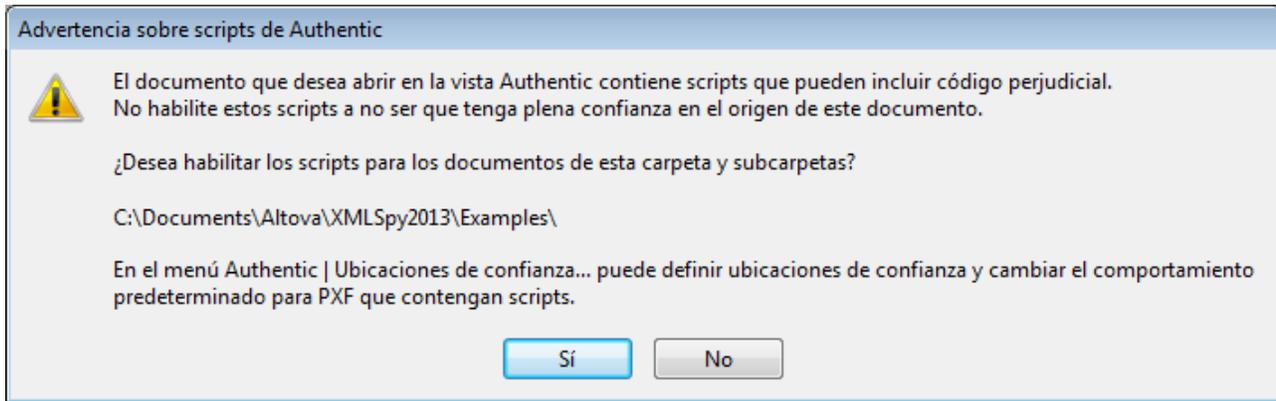
\* No tiene vista Authentic

\*\* Muestra diseños con scripts. No ofrece ejecución de macros ni control de eventos. Los eventos externos se desencadenan.

Los scripts de Authentic tienen el mismo comportamiento en todos los productos de Altova así que no es necesario ningún código ni configuración específicos para cada producto.

### Advertencia sobre scripts de Authentic

Si un archivo XML vinculado a un diseño SPS o un archivo PXF contiene un script y dicho archivo se abre o se consulta en la vista Authentic, entonces aparecerá una advertencia como esta:



Cuando esto ocurra, tendrá dos opciones:

- Hacer clic en **Sí** para agregar la carpeta que contiene el archivo a la lista de ubicaciones de confianza para scripts de Authentic. A partir de este momento todos los archivos de la carpeta de confianza se abrirán en la vista Authentic sin desencadenar esta advertencia. Puede acceder a la lista de ubicaciones de confianza para modificarla con el comando de menú [Authentic | Ubicaciones de confianza](#)<sup>248</sup>.
- Hacer clic en **No** para no agregar la carpeta que contiene el archivo a la lista de ubicaciones de confianza. El archivo se presentará en la vista Authentic pero tendrá deshabilitados todos los scripts. La advertencia sobre scripts de Authentic aparecerá cada vez que se abra el archivo en la vista Authentic. Si en algún momento cambia de opinión y se decide por agregar la carpeta del archivo a la lista de ubicaciones de confianza, abra el cuadro de diálogo "Ubicaciones de confianza" con el comando de menú [Authentic | Ubicaciones de confianza](#)<sup>248</sup> y añada la carpeta o modifique la lista, según corresponda.

Para más información sobre el cuadro de diálogo "Ubicaciones de confianza" consulte la descripción del comando [Authentic | Ubicaciones de confianza](#)<sup>248</sup> en la *Referencia del usuario*.

**Nota:** cuando acceda a Authentic Desktop por su interfaz de COM (véase la [Referencia del programador](#)<sup>304</sup>), la **comprobación de seguridad se omitirá** y la **advertencia sobre scripts de Authentic no aparecerá**.

## Funcionamiento de los scripts de Authentic

El usuario que diseña el diseño SPS puede usar la característica Scripting de Authentic de dos formas distintas para conseguir documentos de Authentic más interactivos:

- Asignando scripts para acciones definidas por el usuario (macros) a los elementos de diseño, botones de la barra de herramientas y comandos de los menús contextuales.
- Añadiendo agentes de control de eventos al diseño que reaccionen a los eventos de la vista Authentic.

Todos los scripts necesarios para crear documentos de Authentic interactivos se diseñan en la interfaz de StyleVision (en las ediciones Enterprise y Professional). Los formularios, las macros y los controladores de eventos se crean en el Editor de scripts de StyleVision y estos scripts se guardan junto con el diseño SPS. Más adelante, en la vista Diseño de StyleVision, el usuario asignará los scripts guardados a elementos de diseño, a botones de barras de herramientas y a menús contextuales. Cuando se abra un documento XML basado en un diseño SPS en un producto de Altova compatible con scripts de Authentic (*ver tabla más arriba*), el documento contará con una mayor flexibilidad y será más interactivo.

## Documentación de Scripting de Authentic

Puede consultar la documentación de Authentic Scripting en la documentación de StyleVision, que encontrará en el [sitio web de Altova](#).

## 8 Vista Explorador

La vista **Explorador** se suele usar para ver:

- Archivos XML que tienen un archivo XSLT asociado. Al cambiar a la vista **Explorador**, el archivo XML se transforma instantáneamente usando la hoja de estilos XSLT asociada y el resultado se presenta directamente en la vista **Explorador**.
- Archivos HTML que se crearon directamente como HTML o a partir de un archivo XML transformado con una hoja de estilos XSLT.

Para ver archivos XML y HTML en la vista **Explorador**, basta con hacer clic en la pestaña **Explorador**.

### Motores de navegación en la vista Explorador

Por defecto, la vista Explorador utiliza actualmente Internet Explorer de Microsoft como motor del explorador. Si desea usar el nuevo motor Edge WebView2 de Microsoft para la vista Explorador, puede seleccionar esta opción en la [sección Vista](#)<sup>286</sup> del [cuadro de diálogo Opciones](#)<sup>278</sup>.

**Nota:** Microsoft Edge WebView2 utiliza el proyecto de software Chromium, en el que se basa el explorador Chrome de Google, por lo que si usa WebView2 para la vista Explorador también podrá acceder a una buena vista previa del aspecto en Chrome de una página web.

### Nota sobre Microsoft Internet Explorer y XSLT

Para poder usar la vista Explorador necesita tener instalados Internet Explorer de Microsoft, versión 5.0 o superior, o bien Microsoft Edge WebView2 (*véase el punto anterior*).

Tenga en cuenta estos puntos sobre Internet Explorer en la vista Explorador:

- Si quiere usar la vista Explorador para visualizar archivos XML transformados por una hoja de estilos XSLT, recomendamos que use Internet Explorer 6.0 o superior, que usa MSXML 3.0, un analizador XML completamente compatible con el estándar XSLT 1.0. Puede que también quiera instalar MSXML 4.0.
- La compatibilidad de Internet Explorer 5 no se ajusta al 100% a la recomendación XSLT oficial, por lo que si está teniendo problemas en la vista Explorador con IE5, deberá actualizarlo a la versión 6 o superior.
- En general es recomendable que compruebe siempre la compatibilidad con XSLT de su versión de Internet Explorer.
- Si tiene problemas con la visualización de HTML en Internet Explorer, incluya esta etiqueta `meta` en el elemento `head` de su documento HTML:

```
<head>
... <meta http-equiv="X-UA-Compatible" content="ie=edge">...
</head>
```

### Herramientas de desarrollo en la vista Explorador

Puede usar las herramientas de desarrollo del explorador Edge para inspeccionar, depurar y hacer pruebas con su código HTML. Para abrir esas herramientas haga clic con el botón derecho en el panel vista Explorador y seleccione **Abrir herramientas de desarrollo**.

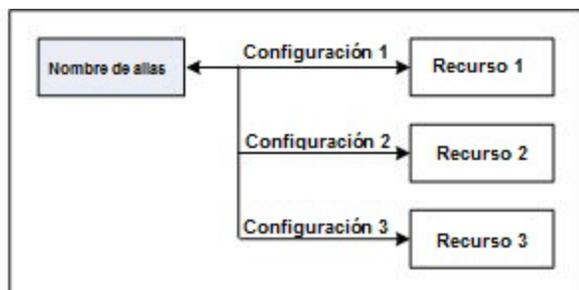
## Funciones de la vista Explorador

Estas son las funciones disponibles en la vista **Explorador**. Para tener acceso a estas funciones, utilice los menús **Explorador**, **Archivo** y **Edición**.

- **Abrir en una ventana separada:** cuando la vista **Explorador** se abre en una ventana separada, se puede colocar en paralelo a una de las vistas de edición del mismo documento. Este comando está en el menú **Explorador | Ventana separada**. Con él puede usar la vista Explorador en dos ventanas: (i) una ventana separada y (ii) una pestaña en la ventana principal. También puede acceder a este comando desde el menú desplegable del botón **Vista Explorador** (en la parte inferior de la ventana principal).
- **Adelante y Atrás:** los comandos típicos de un explorador para navegar por las páginas que se cargaron en la vista **Explorador**. Estos comandos están en el menú **Explorador**.
- **Tamaño de la fuente:** se puede ajustar con el comando **Tamaño de fuente** del menú **Explorador**.
- **Detener, Actualizar e Imprimir:** más comandos típicos de un explorador. Estos están en los menús **Explorador** y **Archivo**.
- **Buscar:** permite buscar cadenas de texto. Este comando está en el menú **Edición**.

## 9 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.



El uso de los recursos globales de Altova conlleva dos procesos:

- [Definir los recursos globales](#)<sup>106</sup>: 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](#)<sup>117</sup>: 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 Authentic Desktop.

### 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 Definir recursos globales

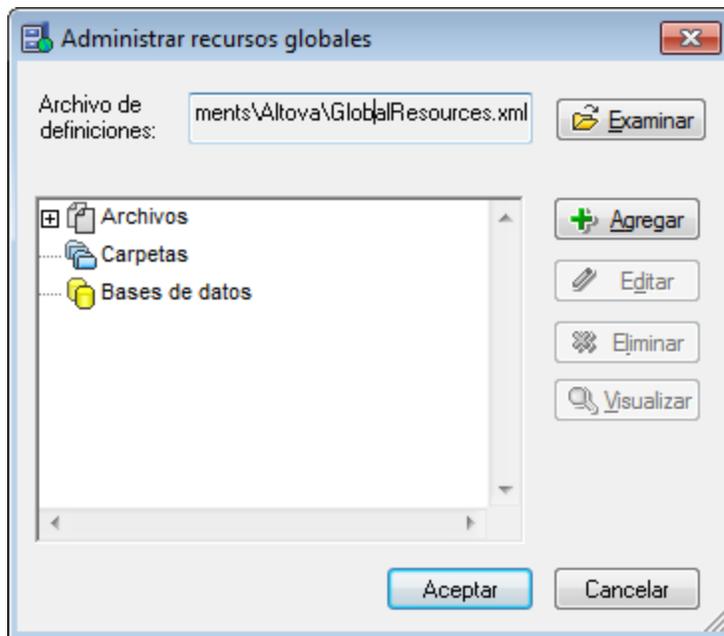
Los recursos globales de Altova se definen en el cuadro de diálogo "Administrar recursos globales". Hay dos maneras de abrir este cuadro de diálogo:

- Seleccionando el comando de menú **Herramientas | Recursos globales**.
- Haciendo clic en el icono **Administrar recursos globales** de la barra de herramientas Recursos globales (*imagen siguiente*).



### El archivo de definiciones de recursos globales

La información sobre los recursos globales se almacena en un archivo XML llamado archivo de definiciones de recursos globales. Este archivo se crea y guarda cuando se define el primer recurso global en el cuadro de diálogo "Administrar recursos globales" (*imagen siguiente*).



Cuando se abre el cuadro de diálogo "Administrar recursos globales" por primera vez, el nombre y la ubicación predeterminados del archivo de definiciones de recursos globales aparece en el cuadro de texto *Archivo de definiciones* (*imagen anterior*):

```
C:\Users\
```

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](#)<sup>108</sup>, [carpetas](#)<sup>113</sup> y [bases de datos](#)<sup>106</sup> 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.

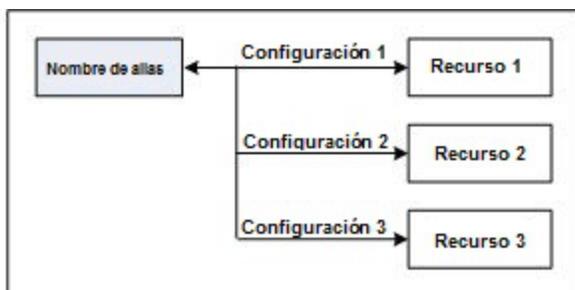
Para **editar un recurso global**, selecciónelo y haga clic en **Editar**. En el cuadro de diálogo que aparece puede realizar todos los cambios necesarios (*consulte los subapartados [archivos](#)<sup>108</sup>, [carpetas](#)<sup>113</sup> y [bases de datos](#)<sup>106</sup> para más información*).

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 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.

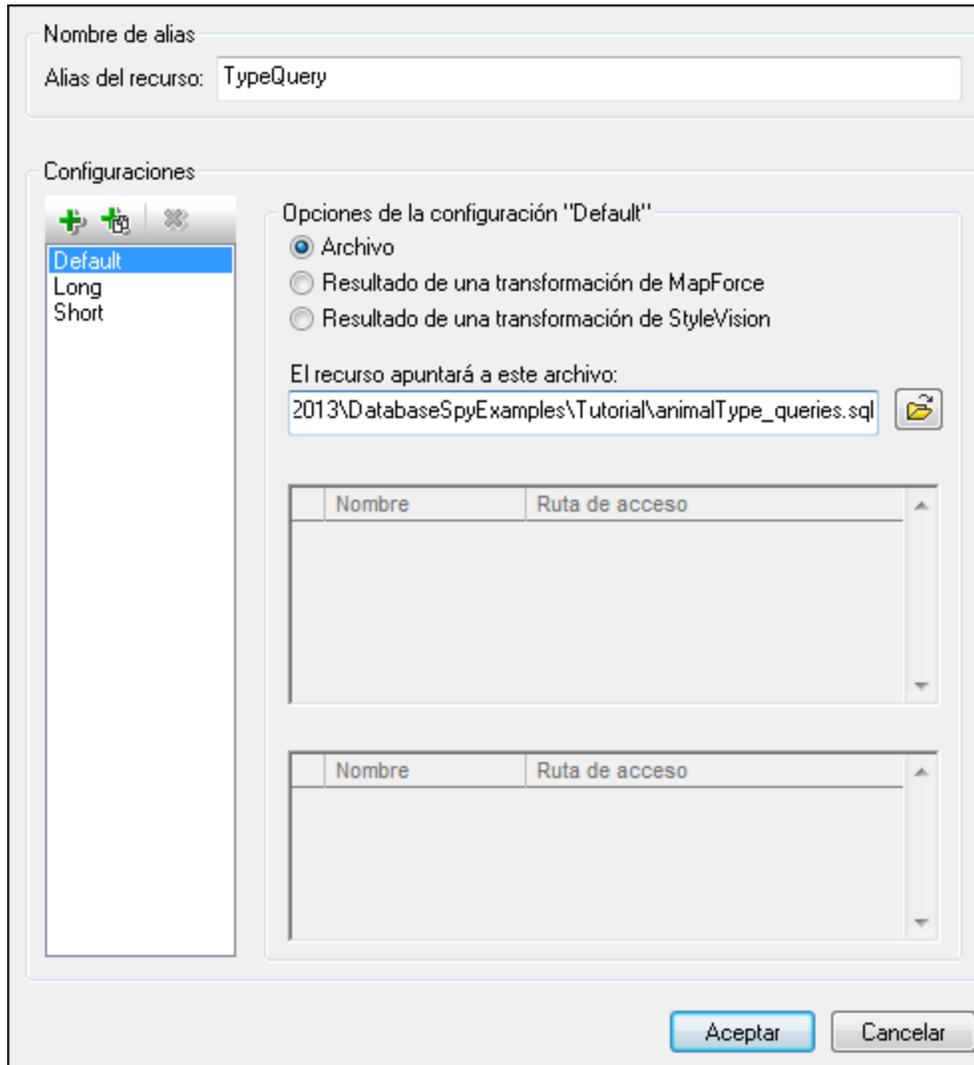
### 9.1.1 Archivos

El cuadro de diálogo "Recurso global" para archivos (*imagen siguiente*) se abre con el comando **Agregar | Archivos** del [cuadro de diálogo Administrar recursos globales](#)<sup>106</sup>. 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](#))<sup>106</sup>.

#### Cuadro de diálogo "Recursos globales"

Los alias se definen en el cuadro de diálogo de recursos globales (*imagen siguiente*).



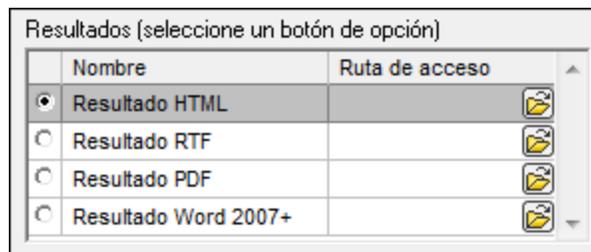
## Iconos del cuadro de diálogo de recursos globales

-  *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):

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 añadida 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).



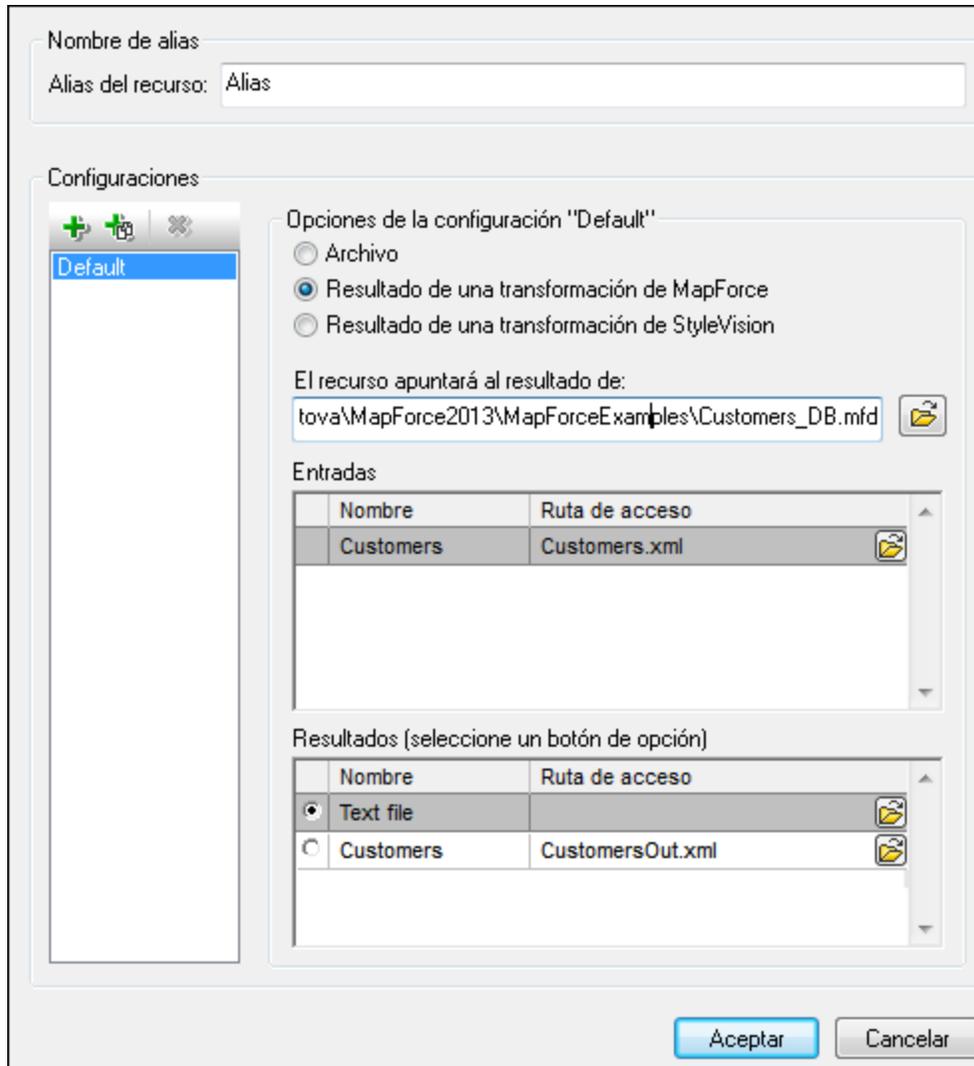
Seleccione una opción (por ejemplo, en la imagen anterior se seleccionó el resultado en formato HTML). Si el recurso es el resultado de una transformación, el resultado se puede guardar como archivo o como recurso global. Haga clic en el icono  y seleccione **Examinar...** (si desea guardar el resultado 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. *Guarda 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](#) <sup>106</sup>.

## 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

guardará en la ubicación de archivo del esquema de salida seleccionado. Este es el archivo que se usará como recurso global.

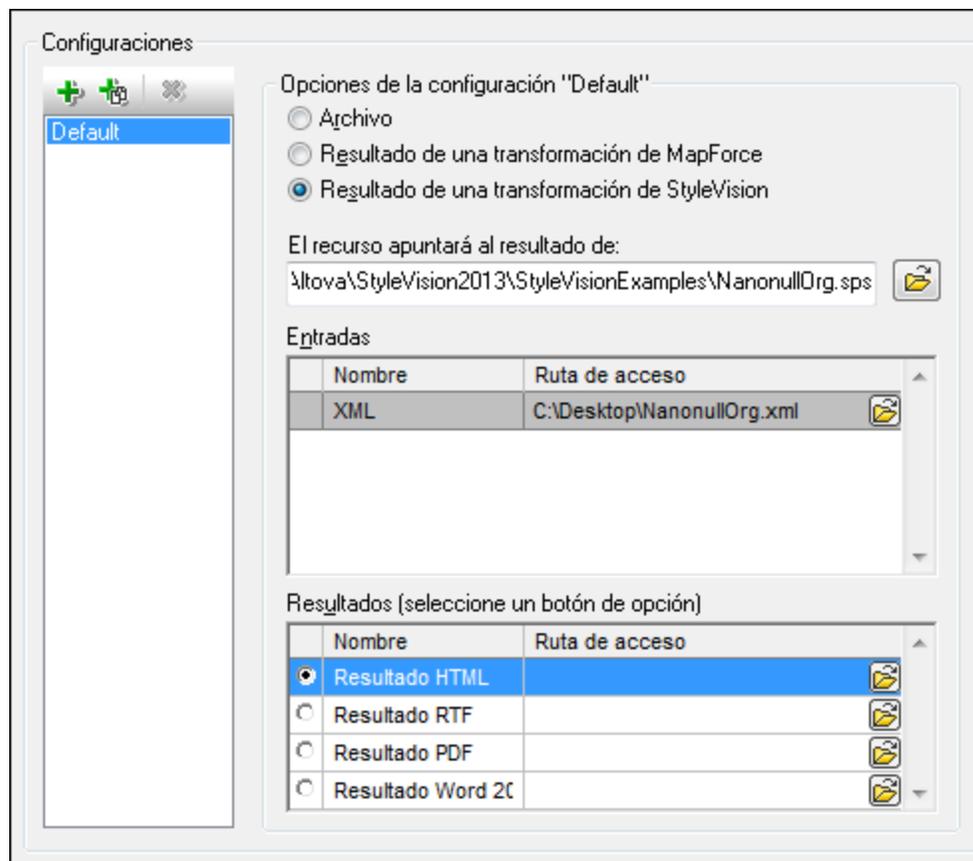
**Nota:** la ventaja de usar esta opción (resultado de una transformación de MapForce) es que la transformación se ejecuta al invocarse el recurso global. Esto significa que el recurso global contendrá los datos más recientes (de los archivos de entrada).

**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).

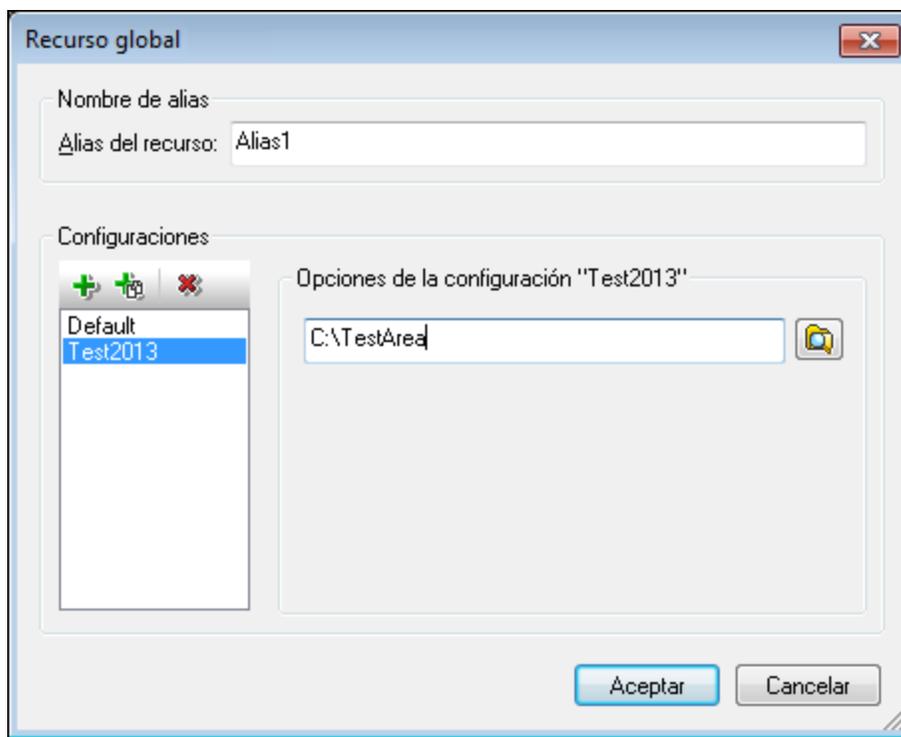
- **Archivos de entrada.** El archivo de entrada puede venir dado por el archivo SPS. Si es así, el archivo de entrada aparecerá automáticamente en el panel *Inputs*. En este panel puede elegir el archivo de entrada que quiera. Si el panel está vacío, entonces añada un archivo.
- **Archivos de resultados.** 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.

## 9.1.2 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.



*Eliminar:* elimina la configuración seleccionada.



*Abrir:* examina el sistema de archivos para buscar el archivo que se debe usar para crear el recurso global.

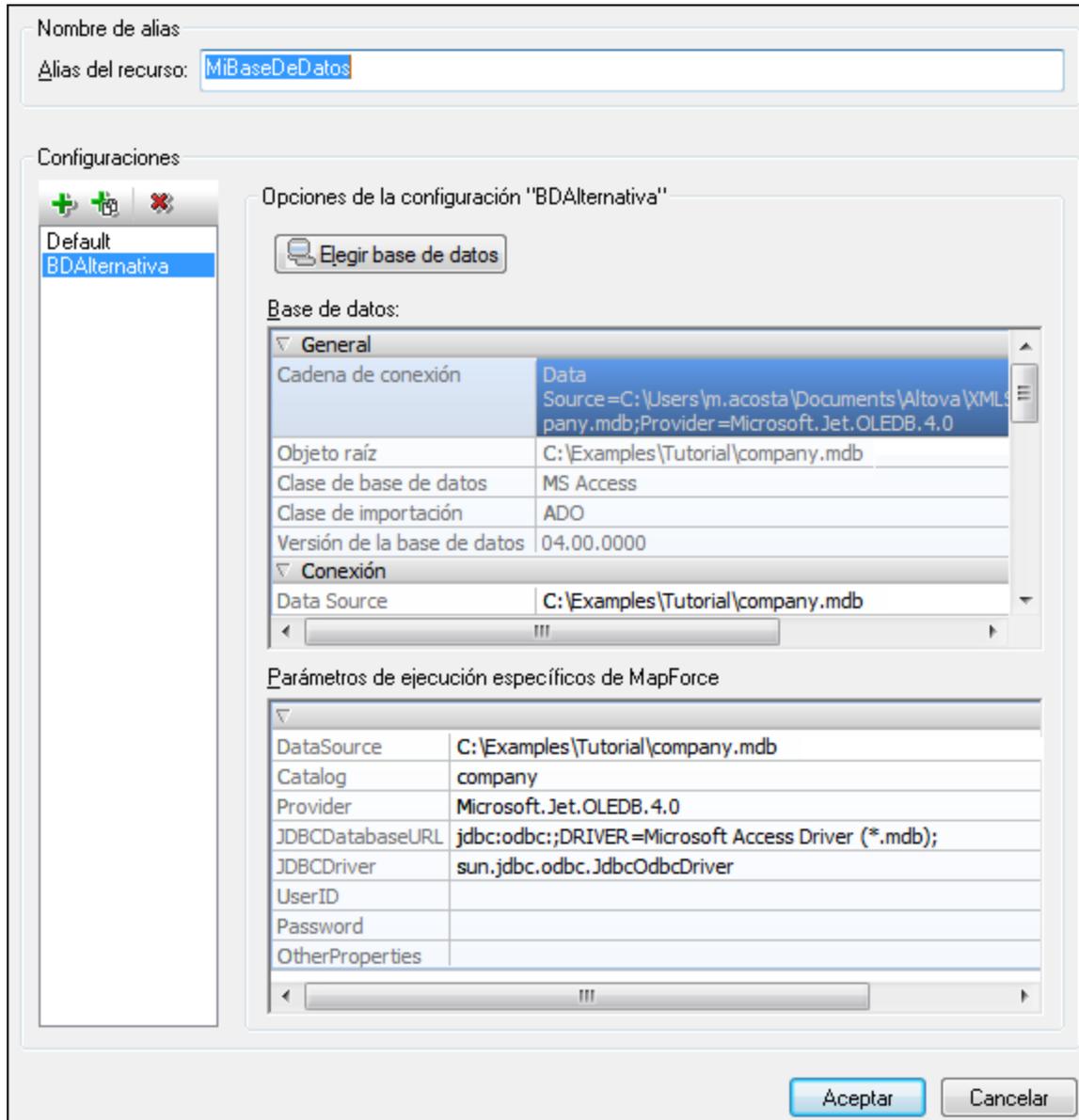
## Definir el alias

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*.

### 9.1.3 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.



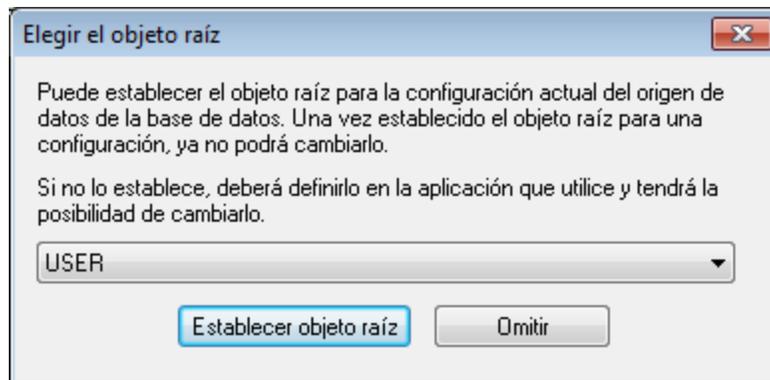
### Iconos del cuadro de diálogo de recursos globales

-  **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):

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. Si ya se estableció una conexión a una base de datos desde Authentic Desktop, 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".

## 9.2 Usar recursos globales

Hay varios tipos de recursos globales (de tipo archivo, carpeta y base de datos). Algunos casos de uso de los recursos globales en Authentic Desktop se describen aquí: [Archivos y carpetas](#)<sup>117</sup>.

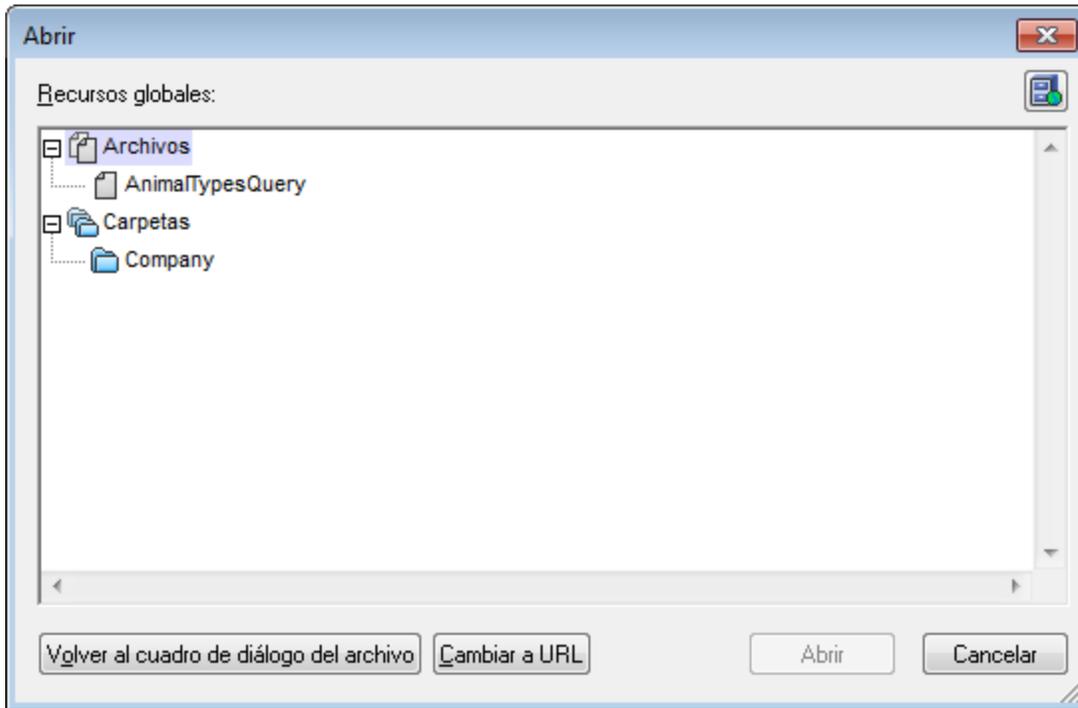
### Selecciones que determinan el recurso que se utiliza

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](#)<sup>106</sup>. 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 . 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.

### 9.2.1 Asignar archivos y carpetas

Los recursos globales de tipo archivo no se asignan igual que los recursos globales de tipo carpeta. En los [casos de uso](#)<sup>118</sup> 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*).



*Administrar recursos globales*: abre el cuadro de diálogo [Administrar recursos globales](#)<sup>106</sup>.

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](#)<sup>106</sup>.

## Casos de uso

Puede usar recursos globales de tipo archivo y carpeta en estos casos:

- [Al abrir recursos globales](#)<sup>118</sup>
- [Al guardar un archivo como recurso global](#)<sup>119</sup>
- [En transformaciones XSLT](#)<sup>119</sup>

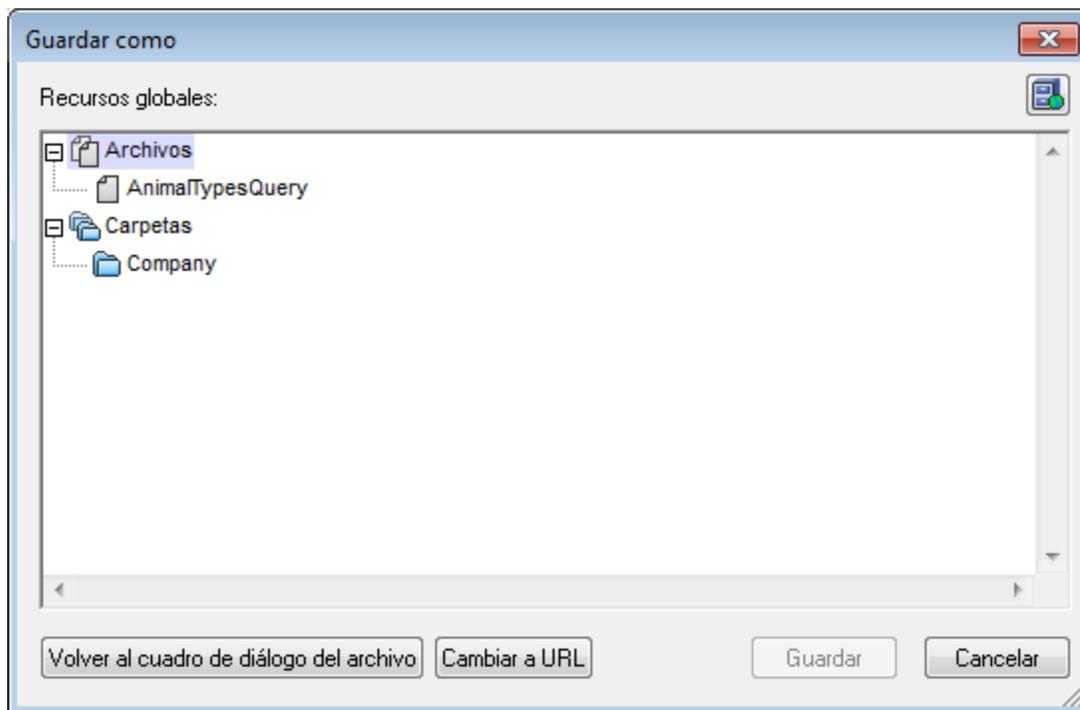
## Abrir recursos globales

Puede abrir recursos globales en Authentic Desktop con el comando [Archivo | Abrir \(Cambiar a recursos globales\)](#)<sup>178</sup>. Si se trata de un recurso global de tipo archivo, el archivo se abre directamente. Si se trata de un recurso global de tipo carpeta, entonces aparece un cuadro de diálogo "Abrir" donde puede examinar la

carpeta seleccionada y buscar el archivo deseado. Una ventaja de editar archivos a través de recursos globales es que los archivos relacionados se pueden guardar en configuraciones de un recurso global y para abrirlos basta con cambiar de configuración. Recuerde guardar los cambios realizados en el archivo antes de cambiar de configuración.

## Guardar un archivo como recurso global

Los archivos recién creados se pueden guardar como recurso global. También puede abrir un archivo ya existente y guardarlo como recurso global. Al hacer clic en los comandos **Archivo | Guardar** o **Archivo | Guardar como** aparece el cuadro de diálogo "Guardar". Pulse el botón **Cambiar a recursos globales** para acceder a los recursos globales disponibles (*imagen siguiente*), es decir, los alias definidos en el archivo XML de recursos globales actual.



Seleccione un alias y pulse el botón **Guardar**. Si el alias es un [alias de archivo](#)<sup>108</sup> el archivo se guarda directamente. Si el alias es un [alias de carpeta](#)<sup>113</sup>, 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](#)<sup>120</sup>.

**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 Authentic Desktop. El motivo es que, al abrir el archivo, Authentic Desktop asume el tipo de archivo indicado en la definición de la configuración.

## Transformaciones XSLT

Al hacer clic en los comandos [XSL/XQuery | Transformación XSL](#)<sup>232</sup> [XSL/XQuery | Transformación XSL-FO](#)<sup>233</sup> aparece un cuadro de diálogo en el que puede buscar el archivo XSLT y XML pertinentes. Haga

clic en el botón **Examinar** y después en el botón **Cambiar a recursos globales**. Aparece el cuadro de diálogo "Abrir" ([ver primera imagen de este apartado](#)<sup>117</sup>). Para la transformación se usa el archivo asociado a la configuración activa del recurso global seleccionado en este diálogo.

## 9.2.2 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.



De este manera, al cambiar de configuración activa, puede cambiar los archivos de origen que se asignaron a través de recursos globales.

## 10 Control de código fuente

La función de control de código fuente de Authentic Desktop funciona con la API del complemento Microsoft Source Control (antes conocido como MSSCCI), versiones 1.1, 1.2 y 1.3. Gracias a esta API podrá ejecutar comandos de control de código fuente como **Proteger** y **Desproteger** desde Authentic Desktop directamente con prácticamente cualquier control de código fuente que permita la conexión a clientes nativos o de terceros a través de la API del complemento Microsoft Source Control.

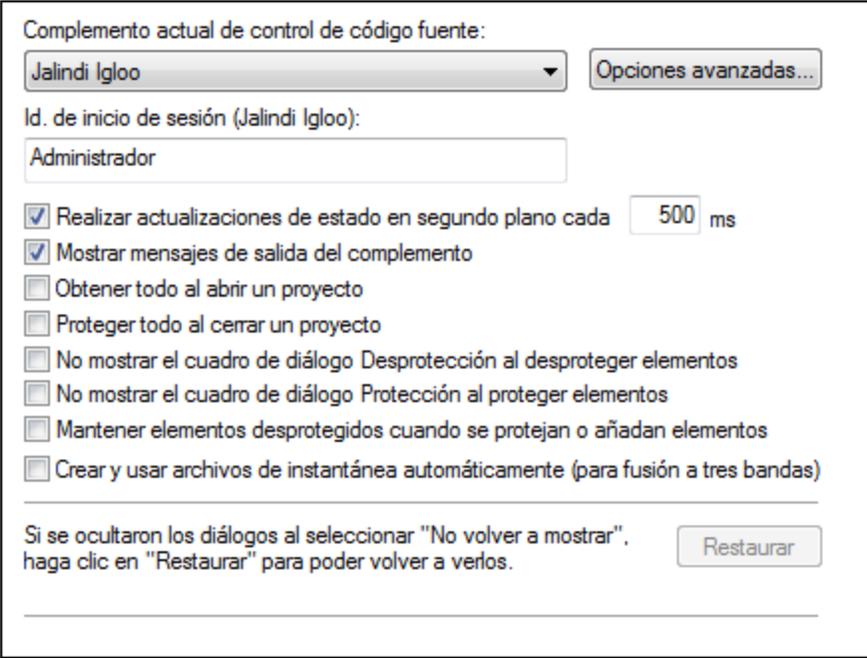
Puede usar cualquier complemento comercial o libre que sea compatible con la API del complemento Microsoft Source Control y puede conectarse a todos los sistemas de control de versiones compatibles ([ver lista de sistemas de control de código fuente compatibles](#)<sup>124</sup>).

### Instalar y configurar el proveedor de control de código fuente

Para ver los proveedores de control de código fuente que están disponibles en el sistema:

1. Haga clic en **Opciones** en el menú **Herramientas**.
2. Haga clic en la sección *Control de código fuente*.

Los complementos de control de código fuente que sean compatibles con la API del complemento Microsoft Source Control aparecerán en la lista desplegable *Complemento actual de control de código fuente*.



Complemento actual de control de código fuente:

Jalindi Igloo Opciones avanzadas...

Id. de inicio de sesión (Jalindi Igloo):

Administrador

Realizar actualizaciones de estado en segundo plano cada  ms

Mostrar mensajes de salida del complemento

Obtener todo al abrir un proyecto

Proteger todo al cerrar un proyecto

No mostrar el cuadro de diálogo Desprotección al desproteger elementos

No mostrar el cuadro de diálogo Protección al proteger elementos

Mantener elementos desprotegidos cuando se protejan o añadan elementos

Crear y usar archivos de instantánea automáticamente (para fusión a tres bandas)

---

Si se ocultaron los diálogos al seleccionar "No volver a mostrar", haga clic en "Restaurar" para poder volver a verlos. Restaurar

Si no se encuentra ningún complemento compatible en el sistema, aparece este mensaje:

"No se Registration of installed source control providers could not be found or is incomplete."

Algunos sistemas de control de código fuente no instalan el complemento de control de código fuente automáticamente. En este caso deberá instalar el complemento por separado. Authentic Desktop espera que

los complementos compatibles con la API del complemento Microsoft Source Code Control estén instalados bajo esta entrada del registro del sistema operativo:

```
HKEY_LOCAL_MACHINE\SOFTWARE\SourceCodeControlProvider\InstalledSCCProviders
```

Tras la instalación, el complemento aparecerá automáticamente en la lista de complementos disponibles de Authentic Desktop.

### Acceso a los comandos de control de código fuente

Los comandos para trabajar con el control de código fuente están en el menú **Proyecto | Control de código fuente**.

### Problemas de rendimiento y recursos

Algunas bases de datos de control de código fuente de gran tamaño pueden crear problemas con recursos y de rendimiento cuando realicen actualizaciones automáticas de estado en segundo plano.

Para aumentar la velocidad del sistema puede deshabilitar (o aumentar el intervalo de) la opción *Realizar actualizaciones de estado en segundo plano cada....segundos* de la sección *Control de código fuente* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**).

**Nota:** la versión de 64 bits de Authentic Desktop es compatible automáticamente con todos los programas de control de código fuente de 32 bits que se enumeran en esta documentación. Cuando usa una versión de 64 bits de Authentic Desktop con un programa de control de código fuente de 32 bits, la opción *Realizar actualizaciones de estado en segundo plano cada....segundos* se deshabilita automáticamente y no se puede seleccionar.

### Comparación de datos con Altova DiffDog

Muchos sistemas de control de código fuente (como Git y TortoiseSVN) se pueden configurar para usar la herramienta de comparación Altova DiffDog. Para más información consulte la [documentación de Altova DiffDog](#) y el [sitio web de Altova](#).

## 10.1 Configurar el control de código fuente

El mecanismo para configurar el control de código fuente y poner archivos de un proyecto de Authentic Desktop bajo control de código fuente se describe a continuación:

1. Instale un programa de control de código fuente compatible, si todavía no tiene uno instalado (puede consultar la lista de [Sistemas de control de código fuente compatibles](#)<sup>124</sup>). Configure la BD del control de código fuente (repositorio) donde desea guardar su trabajo.
2. Cree una carpeta local de la copia de trabajo donde se guardarán los archivos de trabajo que desea poner bajo control de código fuente. La carpeta que contiene todas sus carpetas y archivos de la copia de trabajo se llama *carpeta local*. La ruta de acceso de la carpeta local se denomina *ruta de acceso local*. La carpeta se enlazará a una carpeta determinada del repositorio.
3. En su aplicación de Altova cree una carpeta de proyecto de la aplicación donde debe añadir los archivos que desea poner bajo control de código fuente. El proyecto de la aplicación es una organización abstracta de los archivos. Los archivos del proyecto hacen referencia a archivos físicos guardados en la carpeta local..
4. En la BD del sistema de control de código fuente (conocida también como repositorio) se crea una carpeta que está enlazada a la carpeta local. Esta carpeta (denominada *carpeta enlazada*) es una réplica de la estructura de la carpeta local. De este modo los archivos que se ponen bajo control de código fuente se colocan en el puesto correcto dentro de la jerarquía de archivos. La carpeta enlazada se suele crear cuando se agrega por primera vez un archivo o un proyecto de la aplicación al control de código fuente. Para más información sobre la estructura de carpetas del repositorio, consulte el apartado [Proyecto de la aplicación](#)<sup>127</sup>.
5. Los archivos del proyecto se añaden al control de código fuente con el comando **Proyecto | Control de código fuente | Agregar al control de código fuente**. Cuando se agrega por primera vez un proyecto o un archivo del proyecto por primera vez, se crea la carpeta enlazada.
6. Las operaciones de control de código fuente, como la protección o desprotección de archivos, se realizan con los comandos del submenú **Proyecto | Control de código fuente**. Estos comandos se describen en el apartado [Menú Proyecto](#)<sup>200</sup> de la *Referencia del usuario*.

**Nota:** si desea cambiar de proveedor de control de código fuente, puede hacerlo de dos maneras diferentes: (i) en el cuadro de diálogo "Opciones" ([Herramientas | Opciones | Control de código fuente](#)<sup>291</sup>) o (ii) en el cuadro de diálogo "Cambiar control de código fuente" (**Proyecto | Control de código fuente | Cambiar control de código fuente**).

## 10.2 Sistemas de control de código fuente compatibles

A continuación puede ver una lista con todos los servidores de control de código fuente compatibles con Authentic Desktop, junto con sus correspondientes clientes de control de código fuente. La lista está ordenada alfabéticamente.

### Notas:

- Altova ha implementado la API del complemento Microsoft Source Control (versiones 1.1, 1.2 y 1.3) en Authentic Desktop y ha probado la compatibilidad con los controladores y sistemas de control de versiones de la lista que aparece a continuación. Altova seguirá ofreciendo compatibilidad con estos productos cuando se actualicen.
- Los clientes de control de código fuente que no aparecen en la lista pero que implementan la API del complemento Microsoft Source Control también deberían funcionar con Authentic Desktop.

Sistema de control de código fuente	Clientes de control de código fuente
AccuRev 4.7.0 Windows	AccuBridge for Microsoft SCC 2008.2
Bazaar 1.9 Windows	Aigenta Unified SCC 1.0.6
Borland StarTeam 2008	Borland StarTeam Cross-Platform Client 2008 R2
Codice Software Plastic SCM Professional 2.7.127.10 (Server)	Codice Software Plastic SCM Professional 2.7.127.10 (SCC Plugin)
Collabnet Subversion 1.5.4	<ul style="list-style-type: none"> <li>• Aigenta Unified SCC 1.0.6</li> <li>• PushOK SVN SCC 1.5.1.1</li> <li>• PushOK SVN SCC x64 versión 1.6.3.1</li> <li>• TamTam SVN SCC 1.2.24</li> </ul>
ComponentSoftware CS-RCS (PRO) 5.1	ComponentSoftware CS-RCS (PRO) 5.1
Dynamsoft SourceAnywhere for VSS 5.3.2 Standard/Professional Server	Dynamsoft SourceAnywhere for VSS 5.3.2 Client
Dynamsoft SourceAnywhere Hosted	Dynamsoft SourceAnywhere Hosted Client (22252)
Dynamsoft SourceAnywhere Standalone 2.2 Server	Dynamsoft SourceAnywhere Standalone 2.2 Client
Git	PushOK GIT SCC plug-in (consulte <a href="#">Control de código fuente con Git</a> <sup>142</sup> )
IBM Rational ClearCase 7.0.1 (LT)	IBM Rational ClearCase 7.0.1 (LT)
March-Hare CVSNT 2.5 (2.5.03.2382)	Aigenta Unified SCC 1.0.6
March-Hare CVS Suite 2008	<ul style="list-style-type: none"> <li>• Jalindi Igloo 1.0.3</li> <li>• March-Hare CVS Suite Client 2008 (3321)</li> <li>• PushOK CVS SCC NT 2.1.2.5</li> <li>• PushOK CVS SCC x64 versión 2.2.0.4</li> <li>• TamTam CVS SCC 1.2.40</li> </ul>

Sistema de control de código fuente	Clientes de control de código fuente
Mercurial 1.0.2 for Windows	Sergey Antonov HgSCC 1.0.1
Microsoft SourceSafe 2005 with CTP	Microsoft SourceSafe 2005 with CTP
Microsoft Visual Studio Team System 2008/2010 Team Foundation Server	Microsoft Team Foundation Server 2008/2010 MSSCCI Provider
Perforce 2008 P4S 2008.1	Perforce P4V 2008.1
PureCM Server 2008/3a	PureCM Client 2008/3a
QSC Team Coherence Server 7.2.1.35	QSC Team Coherence Client 7.2.1.35
Reliable Software Code Co-Op 5.1a	Reliable Software Code Co-Op 5.1a
Seapine Surround SCM Client/Server for Windows 2009.0.0	Seapine Surround SCM Client 2009.0.0
Serena Dimensions Express/CM 10.1.3 for Win32 Server	Serena Dimensions 10.1.3 for Win32 Client
Softimage Alienbrain Server 8.1.0.7300	Softimage Alienbrain Essentials/Advanced Client 8.1.0.7300
SourceGear Fortress 1.1.4 Server	SourceGear Fortress 1.1.4 Client
SourceGear SourceOffsite Server 4.2.0	SourceGear SourceOffsite Client 4.2.0 (Windows)
SourceGear Vault 4.1.4 Server	SourceGear Vault 4.1.4 Client
VisualSVN Server 1.6	<ul style="list-style-type: none"> <li>• Aigenta Unified SCC 1.0.6</li> <li>• PushOK SVN SCC 1.5.1.1</li> <li>• PushOK SVN SCC x64 versión 1.6.3.1</li> <li>• TamTam SVN SCC 1.2.24</li> </ul>

## 10.3 Carpeta de trabajo local

Los archivos con los que trabajará se deberían guardar en una jerarquía dentro de la carpeta local de la copia de trabajo:

### Carpeta local de la copia de trabajo

```
|
|-- MiProyecto.spp
|-- QuickStart
|   |-- QuickStart.css
|   |-- QuickStart.xml
|   |-- QuickStart.xsd
|-- Grouping
|   |-- Persons
|       |-- Persons.xml
```

El archivo de proyecto de la aplicación (archivo `.spp`) está por lo general dentro de la carpeta local de la copia de trabajo (*ver ejemplo anterior*).

Cuando uno de los archivos de esta carpeta (copia de trabajo) se ponen bajo control de código fuente, la estructura de la carpeta local de la copia de trabajo se reproduce por completo o parcialmente en el repositorio. Por ejemplo, si el archivo `Persons.xml` de la carpeta local del ejemplo anterior se pone bajo control de código fuente, entonces su ruta de acceso en el repositorio sería esta:

```
[CarpetaRepositorio]/MiProyecto/Grouping/Persons/Persons.xml
```

La carpeta `MiProyecto` de la carpeta del repositorio está enlazada a la carpeta local. Por lo general tendrá el nombre del proyecto, pero puede ponerle otro nombre si quiere.

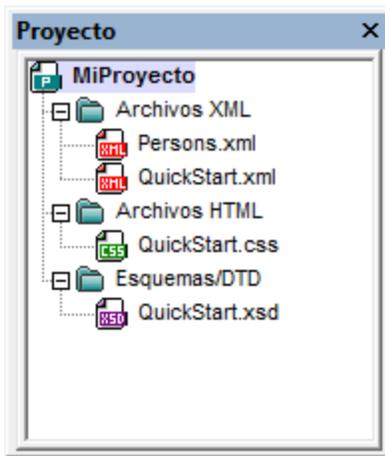
Si se pone bajo control de código fuente todo el proyecto de la aplicación (seleccione el nombre del proyecto en la ventana Proyecto y póngalo bajo control de código fuente), entonces se recrea toda la estructura de la carpeta local en el repositorio.

**Nota:** los archivos que están fuera de la carpeta local de la copia de trabajo se pueden añadir al proyecto de la aplicación. Pero dependiendo del sistema SCS que utilice podrá poner esos archivos bajo control de código fuente o no. Algunos sistemas de control de código fuente tienen problemas a la hora de poner en el repositorio archivos situados fuera de la carpeta local. Por tanto, recomendamos que guarde en la carpeta local todos los archivos que quiera poner bajo control de código fuente.

## 10.4 Proyecto de la aplicación

Cree o cargue el proyecto de su aplicación de Altova que quiere poner bajo control de código fuente. Si desea poner un solo archivo bajo control de código fuente, este archivo debe estar incluido en el proyecto (porque al control de código fuente solamente se puede acceder desde un proyecto).

Por ejemplo, imagine que trabaja con un proyecto de la aplicación XMLSpy. Las propiedades del proyecto están guardadas en un archivo `.spp`. En la aplicación, el proyecto se muestra en la ventana Proyecto (*imagen siguiente*). El proyecto de la imagen se llama `MiProyecto` y sus propiedades están guardadas en el archivo `MiProyecto.spp`.



Puede poner bajo control de código fuente el proyecto entero (todos sus archivos) o solamente algunos archivos del proyecto. **Solamente se pueden poner bajo control de código fuente los archivos que están incluidos en el proyecto.** Por tanto, para poner archivos bajo control de código fuente, antes debe agregarlos al proyecto. El archivo de proyecto (archivo `.spp`) se pone bajo control de código fuente automáticamente cuando uno de sus archivos se ponen bajo control de código fuente.

El proyecto y sus archivos se ponen bajo control de código fuente con el comando **Proyecto | Control de código fuente | Agregar al control de código fuente** (*ver el apartado siguiente*).

No obstante, recuerde que la estructura de la carpeta del repositorio no se corresponde a la estructura de la carpeta del proyecto (*imagen anterior*) sino a la estructura de la [carpeta local de la copia de trabajo](#) <sup>126</sup> (*ver más abajo*). En el ejemplo siguiente observe que la estructura de la carpeta `MiProyecto` del repositorio se corresponde con la de la carpeta local de la copia de trabajo. Observe también que la carpeta enlazada está dentro de la carpeta del repositorio.

Carpeta local de la copia de trabajo	Repositorio
-- MiProyecto.spp	-- <u>MiProyecto (enlazada a la copia local)</u>
-- QuickStart	-- MiProyecto.spp
-- QuickStart.css	-- QuickStart
-- QuickStart.xml	-- QuickStart.css
-- QuickStart.xsd	-- QuickStart.xml
-- Grouping	-- QuickStart.xsd

```
| | -- Persons | |-- Grouping
| | | -- Persons.xml | | | -- Persons
| | | | -- Persons.xml | | | | -- Persons.xml
```

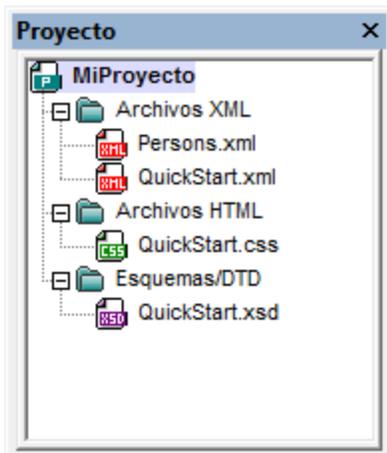
**Nota:** un proyecto de la aplicación puede contener carpetas de proyecto (verdes) o carpetas externas (amarillas). Solamente puede poner bajo control de código fuente archivos que estén dentro de carpetas de proyecto (verdes). Los archivos de las carpetas externas no se pueden poner bajo control de código fuente.

**Nota:** los archivos que están fuera de la carpeta local de la copia de trabajo se pueden añadir al proyecto de la aplicación. Pero dependiendo del sistema SCS que utilice podrá poner esos archivos bajo control de código fuente o no. Algunos sistemas de control de código fuente tienen problemas a la hora de poner en el repositorio archivos situados fuera de la carpeta local. Por tanto, recomendamos que guarde en la carpeta local todos los archivos que quiera poner bajo control de código fuente.

## 10.5 Agregar al control de código fuente

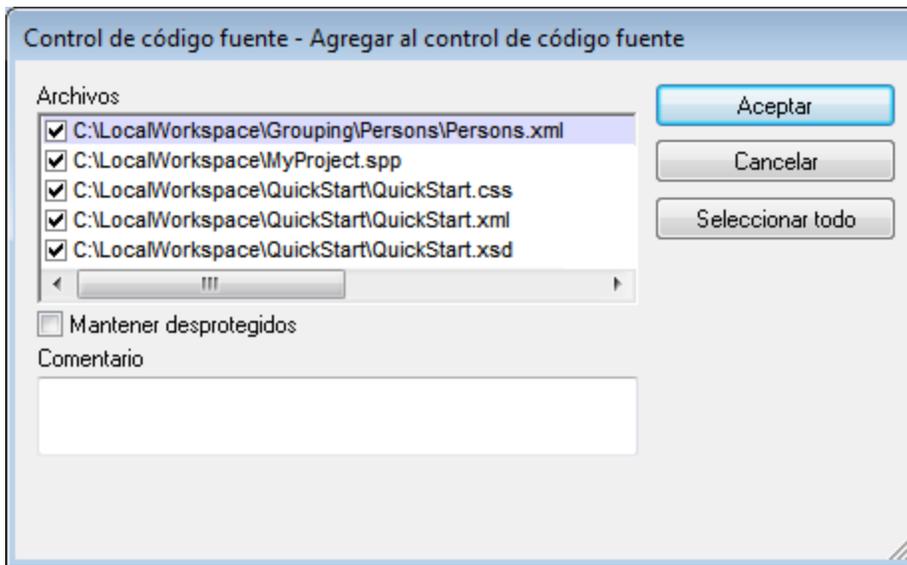
Cuando se agrega el proyecto al control de código fuente se crean automáticamente los enlaces correspondientes y la estructura del repositorio antes de poner el archivo de proyecto (archivo `.spp`) o los archivos bajo control de código fuente. A continuación explicamos cómo agregar el proyecto al control de código fuente.

Seleccione el proyecto en la ventana Proyecto (*imagen siguiente*) y asegúrese de que está resaltado el título del proyecto (como en la imagen). También puede seleccionar uno o varios archivos. Si añade un archivo al control de código fuente, se añade también el archivo de proyecto (archivo `.spp`).



Después seleccione el comando **Proyecto | Control de código fuente | Agregar al control de código fuente**. Aparecen los cuadros de diálogo de conexión y configuración de sus sistema de control de código fuente. (Para cambiar de sistema haga clic en **Proyecto | Control de código fuente | Cambiar control de código fuente**.)

Siga las instrucciones del sistema de control de código fuente para realizar la conexión y configurar las opciones. Cuando termine, los archivos que seleccionó y el archivo de proyecto (archivo `.spp`) aparecen en el cuadro de diálogo "Agregar al control de código fuente" (*imagen siguiente*). Seleccione los archivos que desea agregar y pulse el botón **Aceptar**.



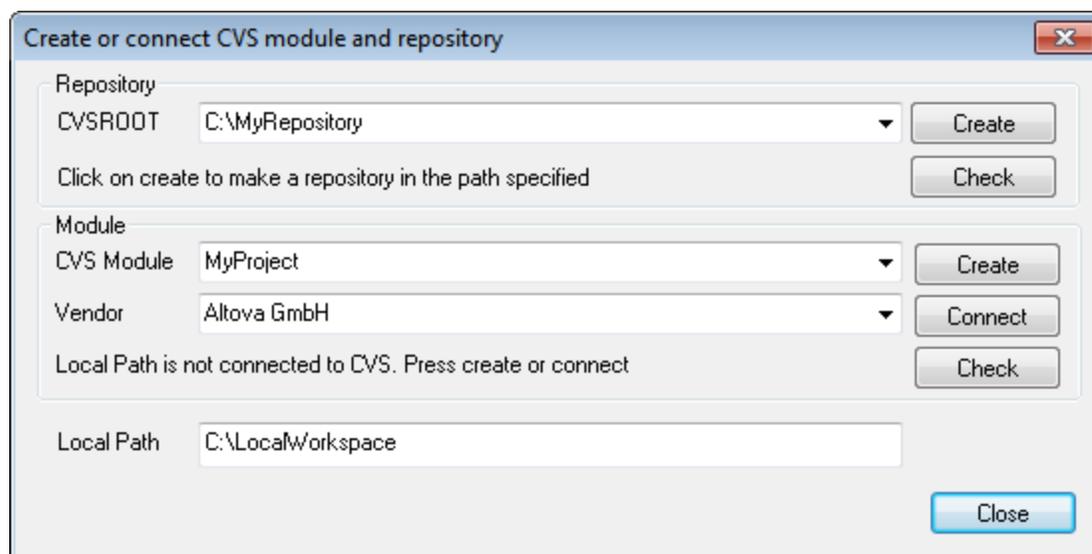
Los archivos se añadirán al repositorio y estarán protegidos o desprotegidos dependiendo de si marcó o no la casilla *Mantener desprotegidos*.

## Notas de configuración

Puede que se solicite crear una carpeta para el proyecto en el repositorio (si todavía no se creó). La [carpeta local de la copia de trabajo](#) <sup>126</sup> estará enlazada a esta carpeta del repositorio (*ver ejemplo siguiente*).

Carpeta local de la copia de trabajo	Repositorio
-- MiProyecto.spp	-- <b>MiProyecto (enlazada a la copia local)</b>
-- QuickStart	-- MiProyecto.spp
-- QuickStart.css	-- QuickStart
-- QuickStart.xml	-- QuickStart.css
-- QuickStart.xsd	-- QuickStart.xml
-- Grouping	-- QuickStart.xsd
-- Persons	-- Grouping
-- Persons.xml	-- Persons
	-- Persons.xml

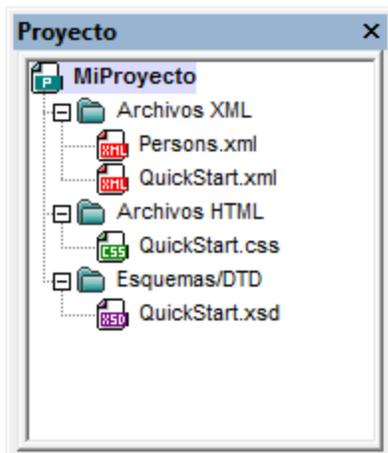
Por ejemplo, aquí puede ver el cuadro de diálogo de configuración de Jalindi Igloo. El campo *CVSROOT* es la ruta de acceso de la carpeta del repositorio.



En la imagen anterior, la ruta de acceso local se corresponde con la ruta de acceso del módulo CVS MiProyecto.

## 10.6 Trabajar con control de código fuente

Para trabajar con control de código fuente seleccione el proyecto, una carpeta de proyecto o un archivo del proyecto en la ventana Proyecto (imagen siguiente) y seleccione un comando del menú **Proyecto | Control de código fuente**. Los comandos **Proteger** y **Desproteger** también están disponibles en el menú contextual que aparece al hacer clic con el botón derecho en las entradas de la ventana Proyecto.



En esta sección describimos las características principales del control de código fuente:

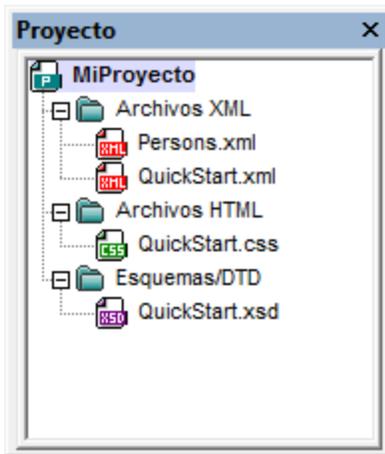
- [Agregar o quitar del control de código fuente](#) <sup>132</sup>
- [Proteger, Desproteger](#) <sup>133</sup>
- [Obtener archivos como archivos de solo lectura](#) <sup>136</sup>
- [Copiar y compartir desde el control de código fuente](#) <sup>137</sup>
- [Cambiar control de código fuente](#) <sup>140</sup>

Los demás comandos del menú **Proyecto | Control de código fuente** se describen en la [Referencia del usuario](#) <sup>200</sup>. Para más información consulte la documentación de su sistema de control de código fuente.

### 10.6.1 Agregar o quitar del control de código fuente

#### Agregar

Después de añadir un proyecto al control de código fuente, puede añadir archivos o grupos de archivos al sistema de control. Seleccione el archivo en la ventana Proyecto y haga clic en el comando **Proyecto | Control de código fuente | Agregar al control de código fuente**. Para seleccionar varios archivos mantenga pulsada la tecla **Ctrl** mientras hace clic en los archivos que desea añadir. Si ejecuta el comando en una carpeta de proyecto (de color verde), entonces se agregan al control de código fuente todos los archivos y subcarpetas de la carpeta.



Cuando se agregan archivos al control de código fuente, la [jerarquía de la carpeta local se copia al repositorio](#)<sup>127</sup> (no la jerarquía de la carpeta de proyecto). Es decir, si un archivo está en una subcarpeta de la carpeta local, entonces se crean en el repositorio la carpeta primaria y las demás carpetas antecesoras del archivo.

Cuando se añade el primer archivo de un proyecto al control de código fuente, se crean los enlaces correspondientes en el repositorio y se añade automáticamente el archivo de proyecto (archivo .spp). Para más información consulte el apartado [Agregar al control de código fuente](#)<sup>129</sup>.

## Símbolos de control de código fuente

Las carpetas y los archivos aparecen con un icono diferente dependiendo de su estado:

	Protegido. Se puede desproteger.
	Desprotegido por otro usuario. No se puede desproteger.
	Desprotegido localmente. Se puede editar y proteger.

## Quitar

Para quitar un archivo del control de código fuente, selecciónelo y haga clic en el comando **Proyecto | Control de código fuente | Quitar del control de código fuente**. También puede quitar: (i) archivos de una carpeta de proyecto ejecutando el comando sobre la carpeta y (ii) todo el proyecto, ejecutando el comando sobre el proyecto.

## 10.6.2 Proteger, desproteger

Tras poner un archivo de proyecto bajo control de código fuente, puede desprotegerlo o protegerlo. Para ello seleccione el archivo (en la ventana Proyecto) y seleccione el comando pertinente en el menú **Proyecto | Control de código fuente**, es decir **Desproteger** o **Proteger**.

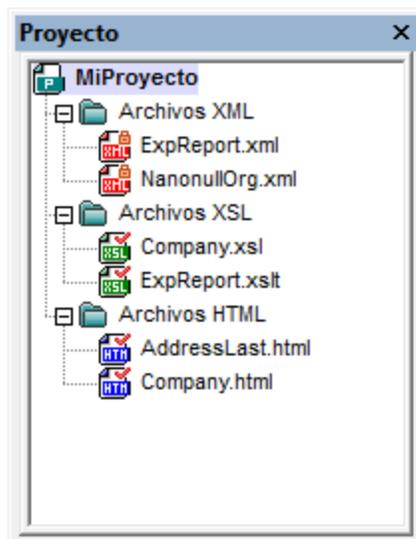
Cuando se desprotege un archivo, se pone una copia del archivo del repositorio en la carpeta local. Los archivos desprotegidos se pueden editar. Si un archivo que está bajo control de código fuente no se

desprotege, no se puede editar. Después de editar el archivo, los cambios se pueden guardar en el repositorio protegiendo el archivo. Incluso si no guarda el archivo, al protegerlo se guardan los cambios en el repositorio. El icono del archivo indica si está desprotegido o protegido.

Las carpetas y los archivos aparecen con un icono diferente dependiendo de su estado:

	Protegido. Se puede desproteger.
	Desprotegido por otro usuario. No se puede desproteger.
	Desprotegido localmente. Se puede editar y proteger.

Si selecciona el proyecto o una carpeta del proyecto, se seleccionan todos los archivos del objeto seleccionado. Para seleccionar varios objetos (archivos y carpetas) pulse la tecla **Ctrl** mientras hace clic en los objetos. La imagen siguiente muestra un proyecto que se desprotegió. Después se protegió el archivo `ExpReport.xml`.



[Obtener archivos como archivos de solo lectura](#) 136

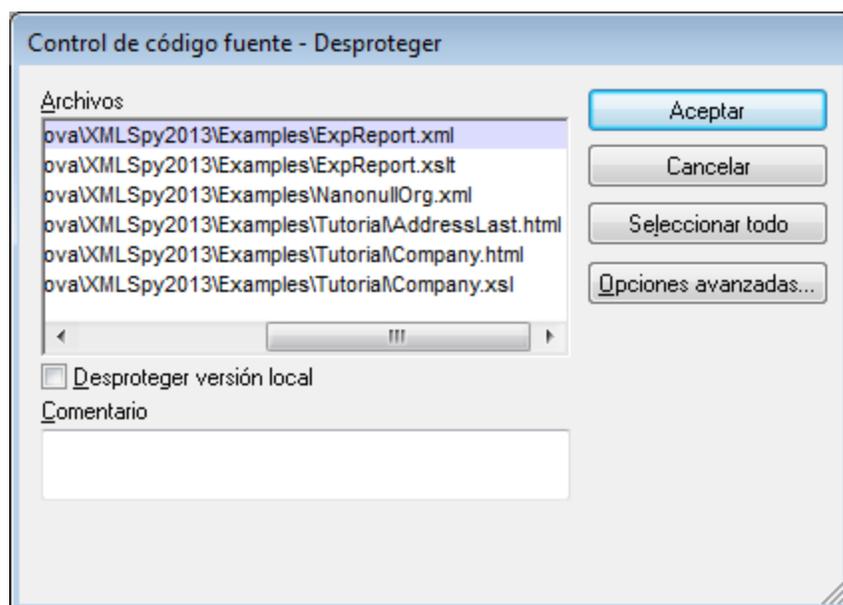
## Guardar cambios y rechazar cambios

Recuerde que al proteger un archivo puede elegir si se protegen los cambios o no. Es decir, puede guardar los cambios en el repositorio y seguir con el archivo desprotegido. Esto es muy útil si desea guardar cambios cada cierto tiempo y después seguir editando el archivo desprotegido.

Si desprotegió un archivo y realizó cambios pero quiere rechazarlos por cualquier motivo, puede volver a la versión del documento guardada en el repositorio con solo hacer clic en el comando **Proyecto | Control de código fuente | Deshacer desprotección**.

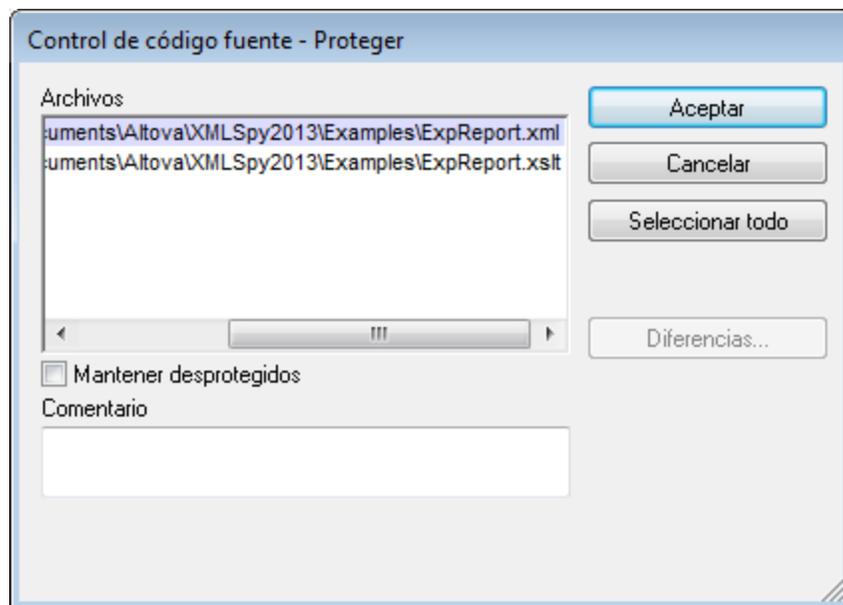
## Desproteger

El cuadro de diálogo "Desproteger" (*imagen siguiente*) sirve para: (i) seleccionar los archivos que desea desproteger y (ii) seleccionar si se desprotege la versión del repositorio o la versión local.



## Proteger

El cuadro de diálogo "Proteger" (*imagen siguiente*) sirve para: (i) seleccionar los archivos que desea proteger y (ii) mantener el archivo desprotegido.

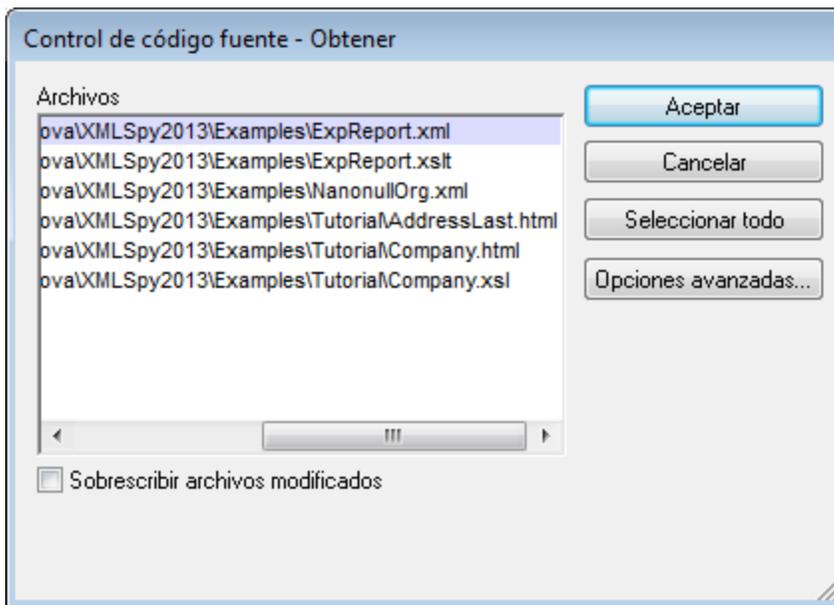


**Nota:** en estos dos cuadros de diálogo aparecen varios archivos si el objeto seleccionado (el proyecto o las carpetas de proyecto) contienen varios archivos.

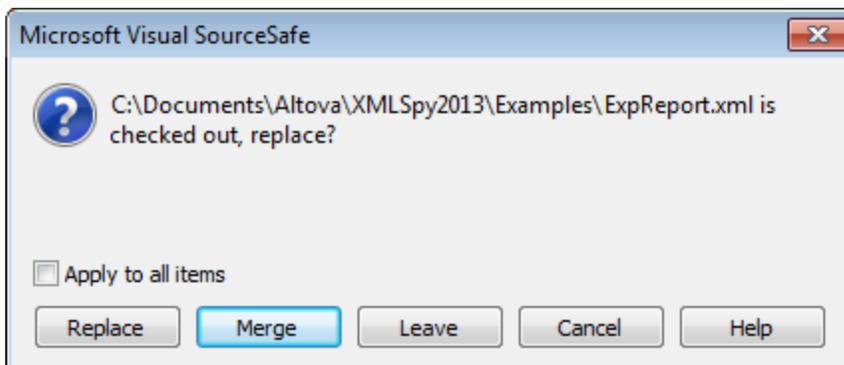
### 10.6.3 Obtener archivos como archivos de solo lectura

El comando **Obtener** (del menú **Proyecto | Control de código fuente**) recupera los archivos del repositorio como archivos de solo lectura. (Para poder editar un archivo, antes es necesario desprotegerlo.) El cuadro de diálogo "Obtener" enumera los archivos del objeto (proyecto o carpeta) sobre el que se ejecutó el comando **Obtener** (*imagen siguiente*). En el cuadro de diálogo puede marcar las casillas de los archivos que desea recuperar.

**Nota:** el comando **Obtener carpetas** sirve para seleccionar subcarpetas del repositorio (si su sistema de control de código fuente permite esta opción).

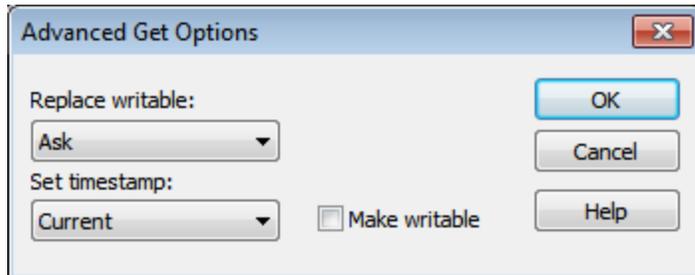


Si quiere, puede sobrescribir los archivos desprotegidos modificados marcando la opción *Sobrescribir archivos modificados* situada en la parte inferior del cuadro de diálogo. Al hacer clic en **Aceptar** los archivos se sobrescriben. Si alguno de los archivos sobrescritos está abierto, aparece un aviso preguntando si desea volver a cargar los archivos (botón **Volver a cargar**), cerrar los archivos (botón **Cerrar**) o conservar la vista actual del archivo (botón **Cancelar**).



## Opciones avanzadas

El cuadro de diálogo "Opciones avanzadas" (*imagen siguiente*) se abre con el botón **Opciones avanzadas** del cuadro de diálogo "Obtener" (*primera imagen de este apartado*).



Aquí puede seleccionar (i) si reemplaza los archivos que se pueden escribir y que están desprotegidos, (ii) la marca de tiempo y (iii) si la propiedad de solo lectura del archivo recuperado se cambia para que el archivo se pueda escribir.

## Obtener la versión más reciente

El comando **Obtener la versión más reciente** (del menú **Proyecto | Control de código fuente**) recupera y coloca en el directorio de trabajo la versión más reciente de los archivos seleccionados del control de código fuente. Los archivos se recuperan como archivos de solo lectura y no están desprotegidos. Este comando es igual que el comando **Obtener**, la única diferencia es que no aparece el cuadro de diálogo "Obtener".

Si los archivos seleccionados están desprotegidos, la acción que se toma dependerá de lo que haga su sistema de control en situaciones así. Por lo general, el sistema de control de código fuente le preguntará si desea reemplazar el archivo desprotegido, combinarlo o dejarlo tal y como está.

**Nota:** cuando seleccione este comando para una carpeta, el comando se aplica de forma recursiva. Esto significa que se aplica a todos los archivos situados dentro de la carpeta actual.

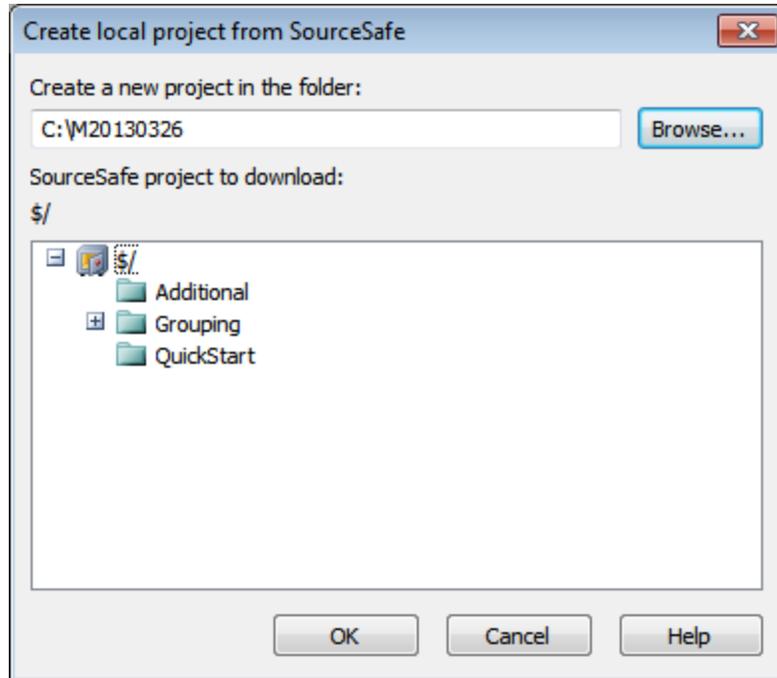
## 10.6.4 Copiar y compartir desde el control de código fuente

El comando **Abrir desde el control de código fuente** crea un proyecto de aplicación nuevo a partir de un proyecto bajo control de código fuente.

Estos son los pasos para crear el proyecto:

1. Dependiendo del sistema de control de código fuente que utilice, quizás sea necesario comprobar que ningún archivo del proyecto está desprotegido antes de crear un proyecto nuevo.
2. No es necesario tener abierto un proyecto en la aplicación.
3. Seleccione el comando **Proyecto | Control de código fuente | Abrir desde el control de código fuente**.
4. El sistema de control de código fuente que tiene instalado muestra sus cuadros de diálogo de verificación y conexión. Conéctese al repositorio pertinente (es decir, a la [carpeta enlazada del repositorio que equivale a la carpeta local](#)<sup>127</sup>).

5. En el cuadro de diálogo que aparece (*imagen siguiente*) busque la carpeta local en la que desea copiar el contenido de la carpeta del repositorio (a la que se acaba de conectar). En la imagen siguiente, por ejemplo, la carpeta asociada del repositorio se llama `MiProyecto` y se representa con el signo `$`. La carpeta local es `C:\M20130326`.

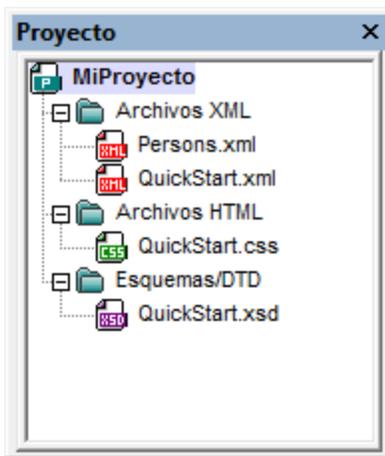


6. Haga clic en **OK**. El contenido de la carpeta asociada (`MiProyecto`) se copiará a la carpeta local `C:\M20130326`, y aparece un cuadro de diálogo pidiendo que seleccione el archivo de proyecto (archivo `.spp`) que desea crear como proyecto nuevo.
7. Seleccione el archivo `.spp` que se copió a la carpeta local. En nuestro ejemplo, se trata del archivo `MiProyecto.spp` situado en la carpeta `C:\M20130326`. En la aplicación se crea un proyecto nuevo llamado `MiProyecto` y este proyecto aparece en la ventana Proyecto. Los archivos del proyecto están en la carpeta `C:\M20130326`.

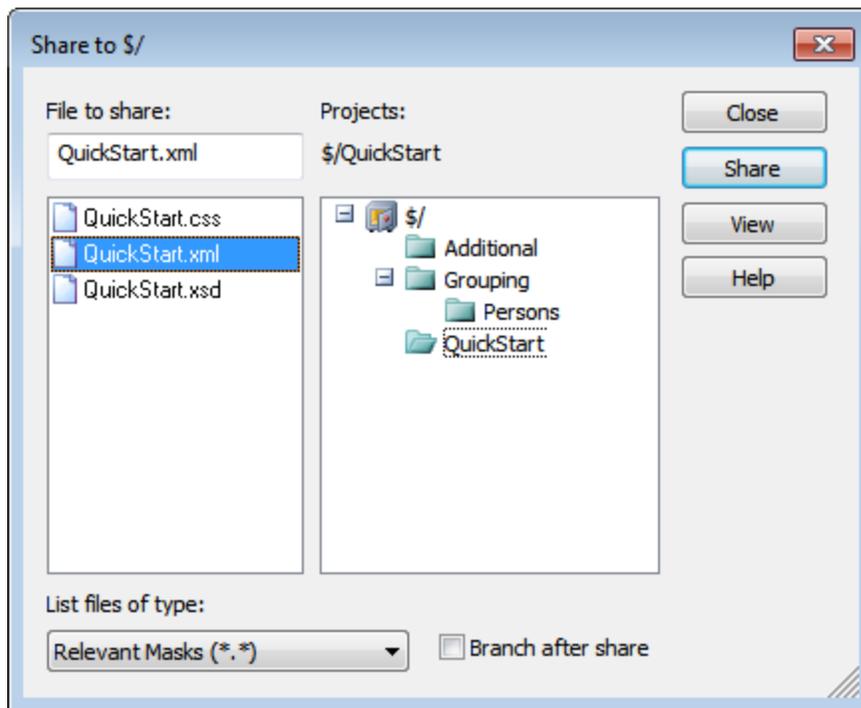
## Compartir desde el control de código fuente

El comando **Compartir desde el control de código fuente** está disponible si su sistema de control de código fuente permite compartir contenido. Por ejemplo, puede compartir un archivo para que esté disponible en varias ubicaciones locales. Si se realizan cambios en uno de estos archivos locales, los cambios se reflejan en todas las versiones "compartidas" del archivo.

En la ventana Proyecto seleccione el proyecto (*imagen siguiente*) y después haga clic en **Compartir desde el control de código fuente**.

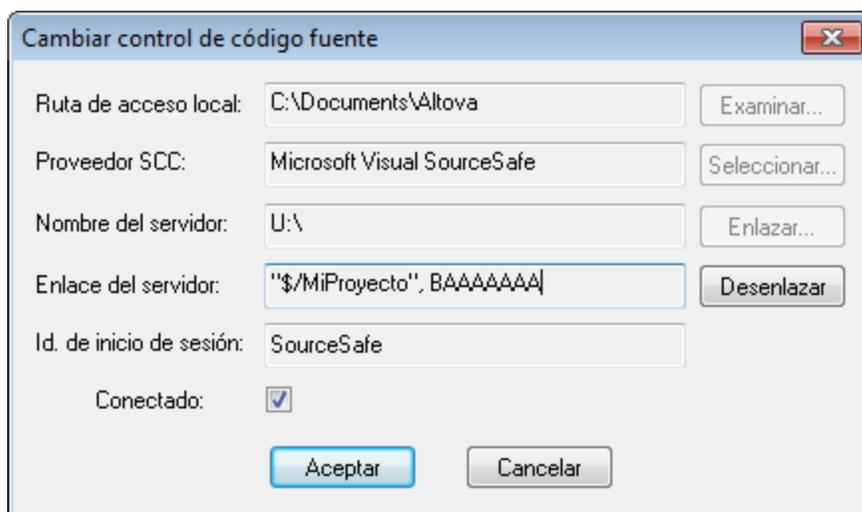


Aparece el cuadro de diálogo "Share To [carpeta]" (imagen siguiente).



Ahora puede seleccionar los archivos que desea compartir. Primero seleccione en el panel derecho la carpeta donde están los archivos. Los archivos de la carpeta elegida aparecen en el panel izquierdo. Seleccione el archivo que desea compartir (para seleccionar varios pulse la tecla **Ctrl** mientras hace clic en los archivos). Los archivos seleccionados aparecen en el cuadro situado en la parte superior izquierda del diálogo. Los archivos desaparecen del panel izquierdo. Haga clic en **Share** (*compartir*) y después en **Close** (cerrar) para copiar los archivos seleccionados a la carpeta local compartida.

El nombre de la carpeta compartida aparece en el nombre del cuadro de diálogo. En la imagen anterior, por ejemplo, se trata de la carpeta local (porque el signo  $\$$  representa la carpeta del repositorio a la que está enlazada la carpeta local). En el cuadro de diálogo "Cambiar control de código fuente" puede ver y configurar la carpeta compartida (imagen siguiente) cambiando la ruta de acceso local y el enlace de servidor.



Para más información consulte la documentación de su sistema de control de código fuente.

## 10.6.5 Cambiar control de código fuente

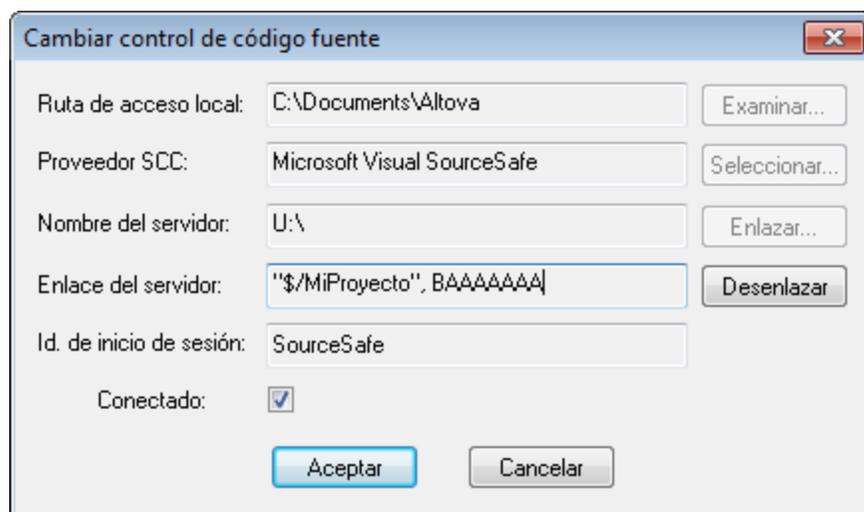
El control de código fuente se puede cambiar con dos comandos diferentes del menú **Proyecto | Control de código fuente**:

- Administrador del control de código fuente, que abre el programa de control de código fuente y permite configurar bases de datos y enlaces.
- Cambiar control de código fuente, que abre el cuadro de diálogo del mismo nombre, donde puede cambiar de sistema (ver la descripción que aparece más abajo).

El enlace actual es lo que usa el proyecto de aplicación activo para conectarse a la BD del control de código fuente. Por tanto, en enlace actual debe ser correcto. Esto significa que el archivo de proyecto de la aplicación (archivo `.spp`) debe estar en la carpeta local y la carpeta enlazada del repositorio debe ser la base de datos donde se almacenan los archivos de este proyecto. Por lo general, la carpeta enlazada y sus subcarpetas se corresponden con la carpeta de la copia de trabajo local y sus subcarpetas.

En el cuadro de diálogo "Cambiar control de código fuente" (*imagen siguiente*) puede cambiar el sistema de control (*Proveedor SCC*), la carpeta local (*Ruta de acceso local*) y el enlace al repositorio (*Nombre del servidor* y *Enlace del servidor*).

Las opciones solamente se pueden cambiar después de desenlazar el enlace actual. Para ello, pulse el botón **Desenlazar**. Ahora puede editar las demás opciones.



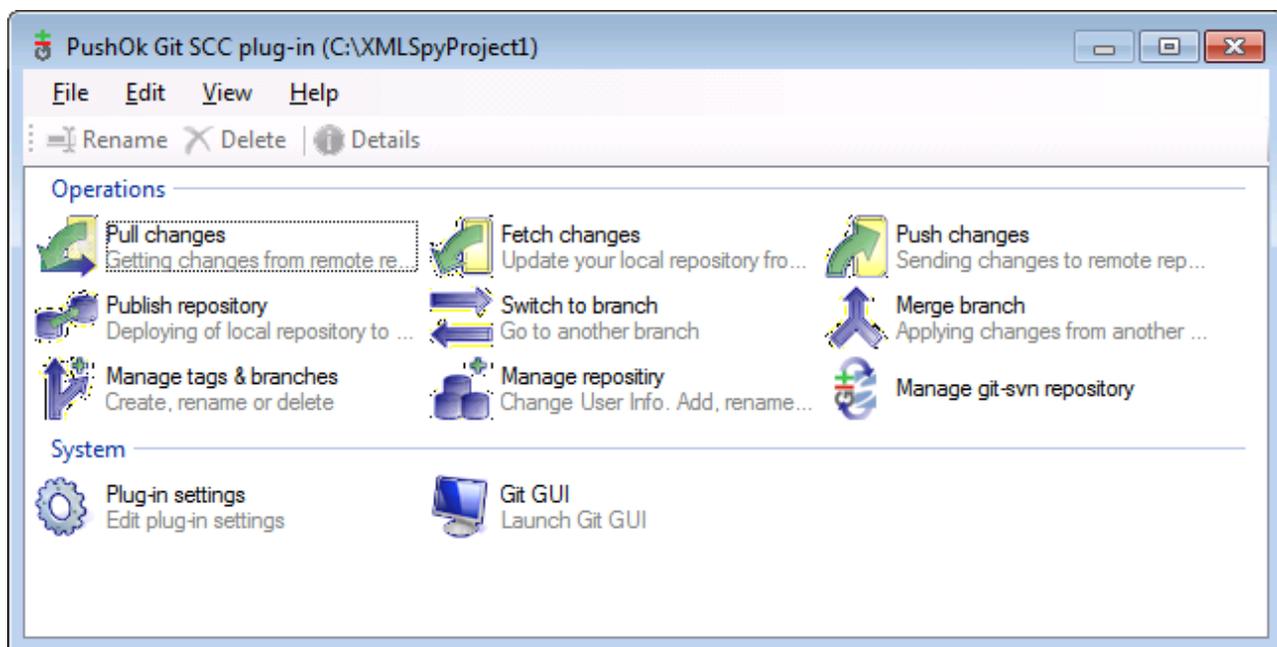
1. Con el botón **Examinar** puede buscar la carpeta local y con el botón **Seleccionar** puede seleccionar uno de los sistemas de control de código fuente instalados.
2. Después puede enlazar la carpeta local a una BD del repositorio. Para ello pulse el botón **Enlazar**. Aparece el cuadro de diálogo de conexión de sus sistema de control.
3. Si escribió un *Id. de inicio de sesión*, este identificador se pasa al sistema de control. De lo contrario, deberá escribir sus datos de inicio de sesión en el diálogo de conexión.
4. Seleccione la BD del repositorio a la que desea enlazar esta carpeta local.
5. Cuando termine haga clic en **Aceptar**.

## 10.7 Control de código fuente con Git

Authentic Desktop es compatible con el sistema de control de versiones Git por medio de un complemento externo llamado **GIT SCC plug-in** (<http://www.pushok.com/software/git.html>).

Cuando se redactó esta documentación, la versión del complemento **GIT SCC plug-in** era una versión experimental. Para usar el complemento es necesario registrarse con el autor del complemento.

El complemento GIT SCC permite trabajar con repositorios Git utilizando los comandos del menú **Proyecto | Control de código fuente** de Authentic Desktop. Recuerde que los comandos de este menú vienen de la API del complemento Microsoft Source Control, cuyo diseño es diferente al de Git. Como consecuencia, el complemento hace de intermediario entre las funciones tipo Visual Source Safe y las funciones de Git. Esto significa, por un lado, que algunos comandos como **Obtener la versión más reciente** no estarán habilitados cuando trabaje con Git. Por otro lado, hay acciones nuevas propias de Git que están disponibles en el cuadro de diálogo de administración del código fuente (**Proyecto | Control de código fuente | Administrador del control de código fuente** en Authentic Desktop).



En el menú **Proyecto | Control de código fuente** también encontrará los comandos más frecuentes de Git.

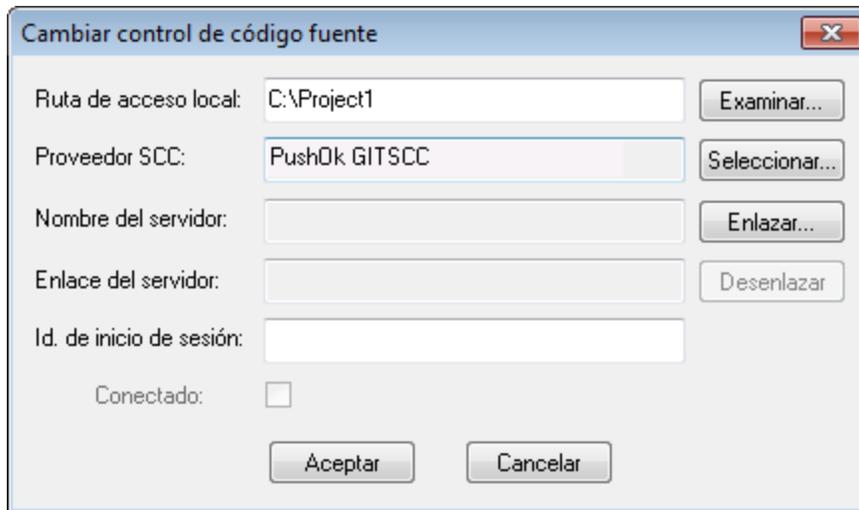
Los diferentes apartados de esta sección describen la configuración inicial del complemento y el flujo de trabajo básico:

- [Habilitar Git con el complemento GIT SCC](#) <sup>143</sup>
- [Agregar un proyecto al control de código fuente de Git](#) <sup>143</sup>
- [Clonar un proyecto desde el control de código fuente de Git](#) <sup>145</sup>

## 10.7.1 Habilitar Git con el complemento de control de código fuente

Para habilitar el control de código fuente de Git en Authentic Desktop es necesario tener instalado el complemento externo **PushOK GIT SCC plug-in**, registrarse y seleccionarlo en la lista de proveedores de control de código fuente:

1. Descargue el archivo de instalación del complemento desde el sitio web del autor (<http://www.pushok.com>), ejecútelo y siga las instrucciones que aparecen en pantalla.
2. En el menú **Proyecto** de Authentic Desktop, haga clic en **Proyecto | Control de código fuente | Cambiar de control de código fuente** y seleccione **PushOk GITSCC**. Si **Push Ok GITSCC** no aparece en la lista de proveedores, es probable que la instalación del complemento no finalizara correctamente. Consulte la documentación del autor para resolver este problema.



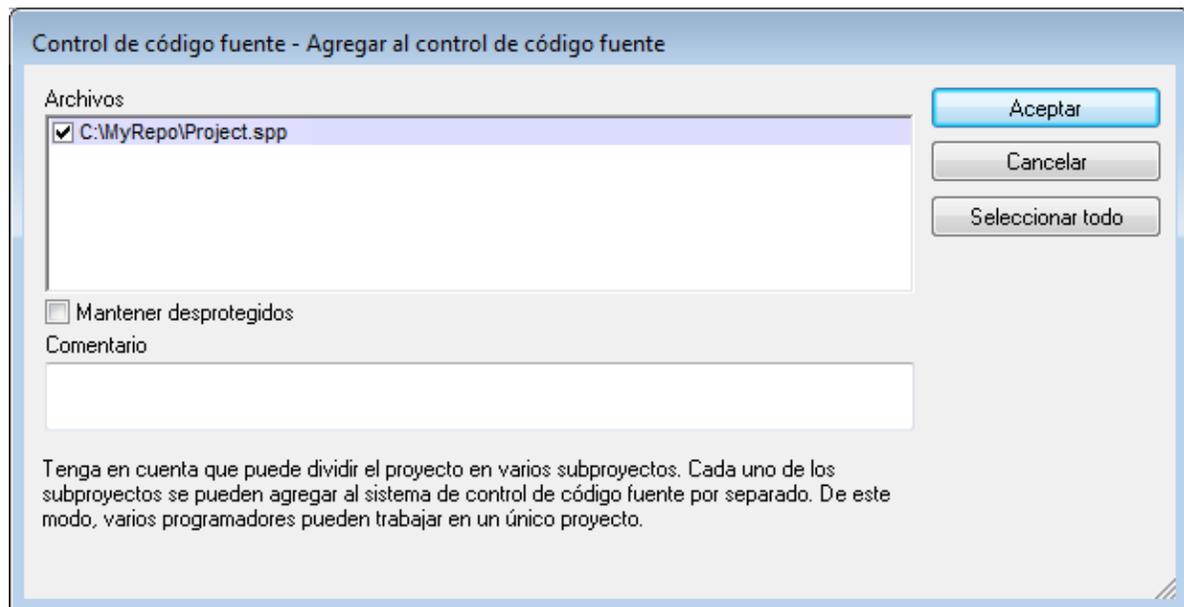
3. Para terminar debe registrar el complemento haciendo clic en **Registration**. Siga los pasos del asistente para terminar de registrar el complemento.

## 10.7.2 Agregar un proyecto al control de código fuente de Git

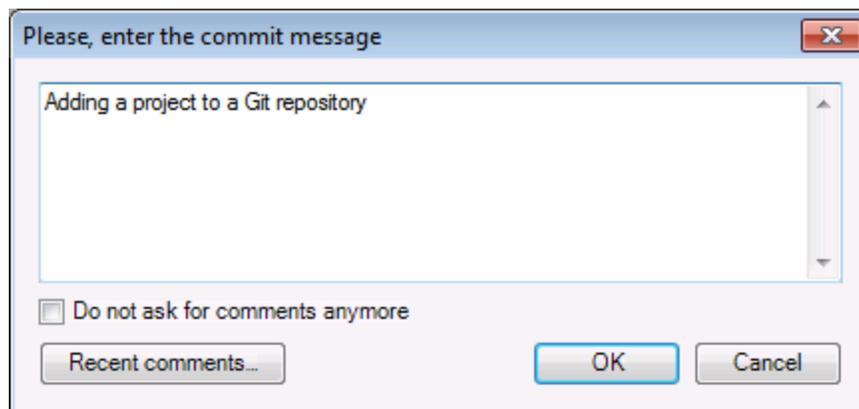
Puede guardar proyectos de Authentic Desktop como repositorios de Git. La estructura de los archivos o carpetas que añada al proyecto se corresponderán con la estructura del repositorio Git.

**Para agregar un proyecto al control de código fuente de Git:**

1. Compruebe que el proveedor de control de código fuente seleccionado es **PushOK GIT SCC Plug-in** (ver el [apartado anterior](#)<sup>143</sup>).
2. Cree un proyecto nuevo con el comando de menú **Proyecto | Crear proyecto**.
3. Guarde el proyecto en una carpeta local (p. ej. C:\MyRepo\Project.spp).
4. Ahora haga clic en **Proyecto | Control de código fuente | Agregar al control de código fuente**.



5. Haga clic en **Aceptar**.



6. Escriba el texto del mensaje de confirmación y haga clic en **OK** para agregar el proyecto al control de código fuente.

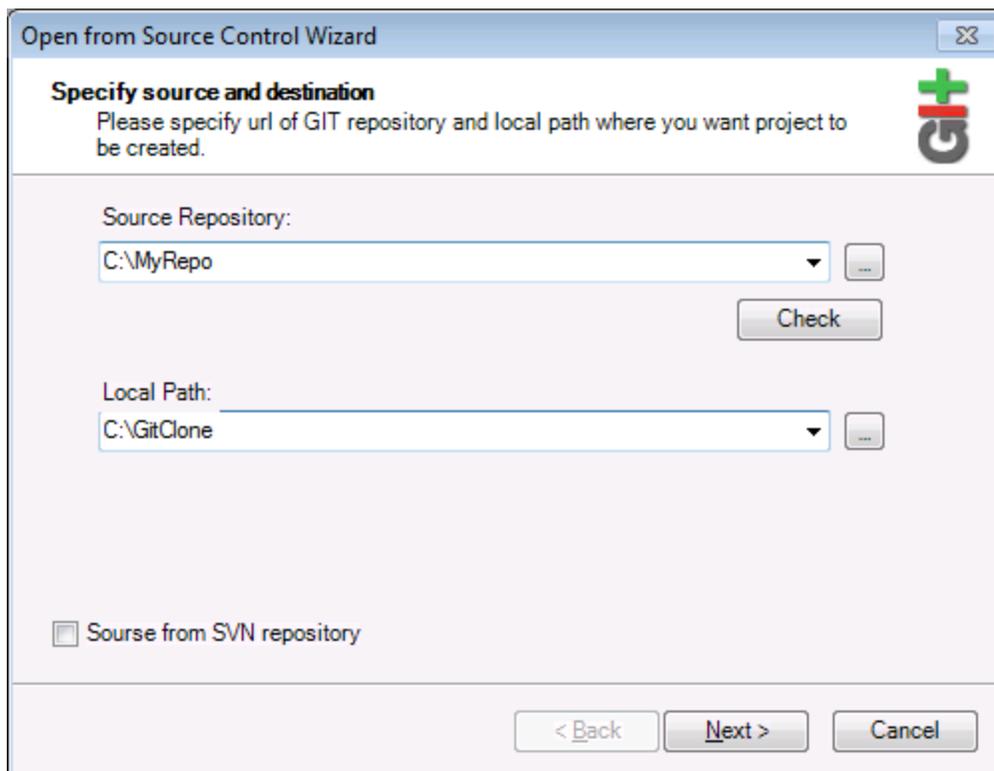
Ahora ya puede añadir archivos y carpetas al proyecto. Recuerde que todos los archivos y carpetas del proyecto deben estar bajo la carpeta raíz del proyecto. Por ejemplo, si creó el proyecto en la carpeta `C:\MyRepo`, entonces solamente podrá añadir al proyecto los archivos que estén bajo `C:\MyRepo`. Si intenta añadir archivos de proyecto que estén fuera de la carpeta raíz del proyecto, aparecerá este mensaje de advertencia:

*Sólo se pueden agregar archivos a una ubicación bajo la raíz de enlace del proyecto (C:\MyRepo).*

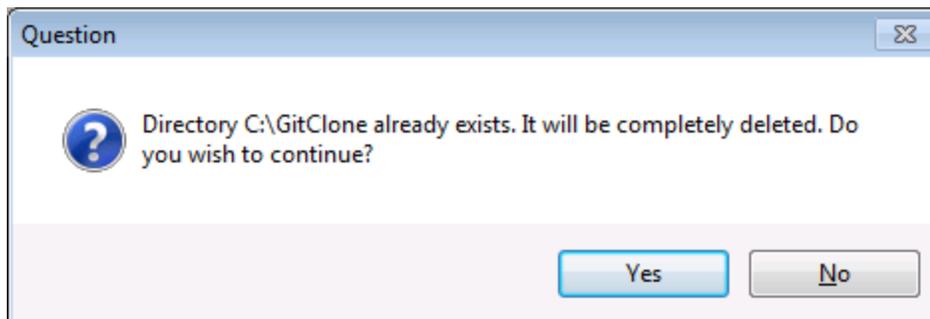
## 10.7.3 Clonar un proyecto desde el control de código fuente de Git

Los proyectos que ya estén en el control de código fuente de Git (ver el [apartado anterior](#)<sup>143</sup>) se pueden abrir desde el repositorio Git:

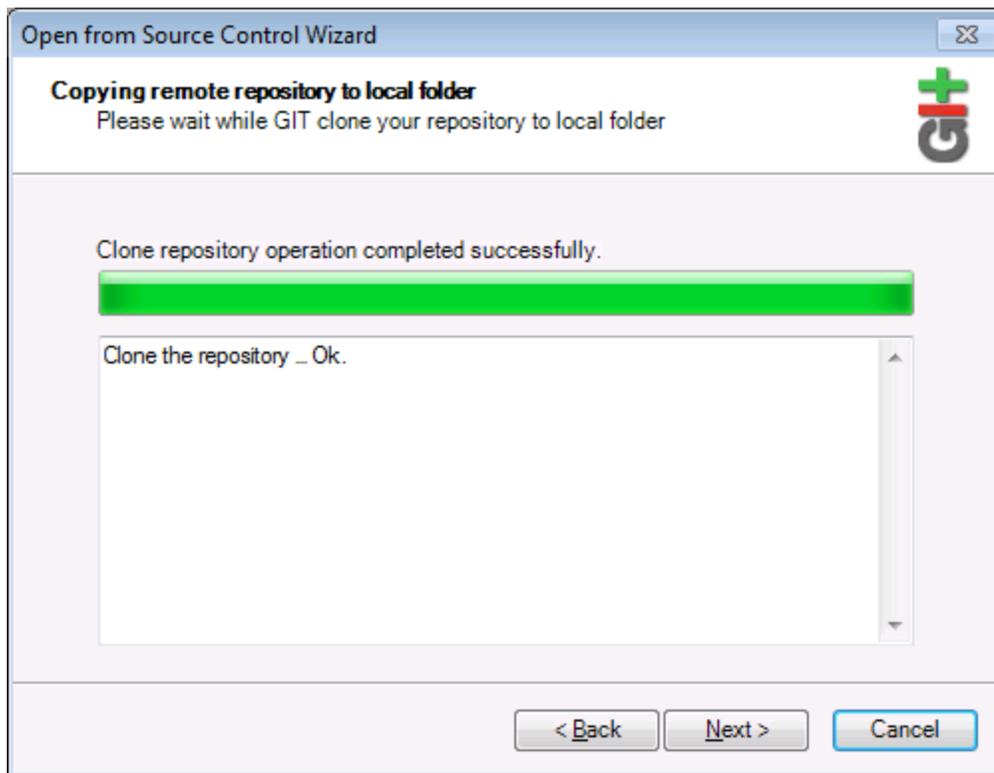
1. Compruebe que el proveedor de control de código fuente seleccionado es **PushOK GIT SCC Plug-in** (ver el apartado [habilitar Git con complemento de control de código fuente GIT SCC](#)<sup>143</sup>).
2. Haga clic en **Proyecto | Control de código fuente | Abrir desde el control de código fuente**.
3. Escriba la ruta de acceso o la URL del repositorio fuente. Haga clic en el botón **Check** para verificar la ruta de acceso o la dirección URL.



4. En el campo *Local Path* escriba la ruta de acceso de la carpeta local donde desea crear el proyecto y haga clic en **Next** para continuar. Si la carpeta local ya existe (aunque esté vacía), aparece este cuadro de diálogo preguntando si desea borrar totalmente la carpeta:



5. Haga clic en **Yes** para confirmar y después en **Next** para continuar.

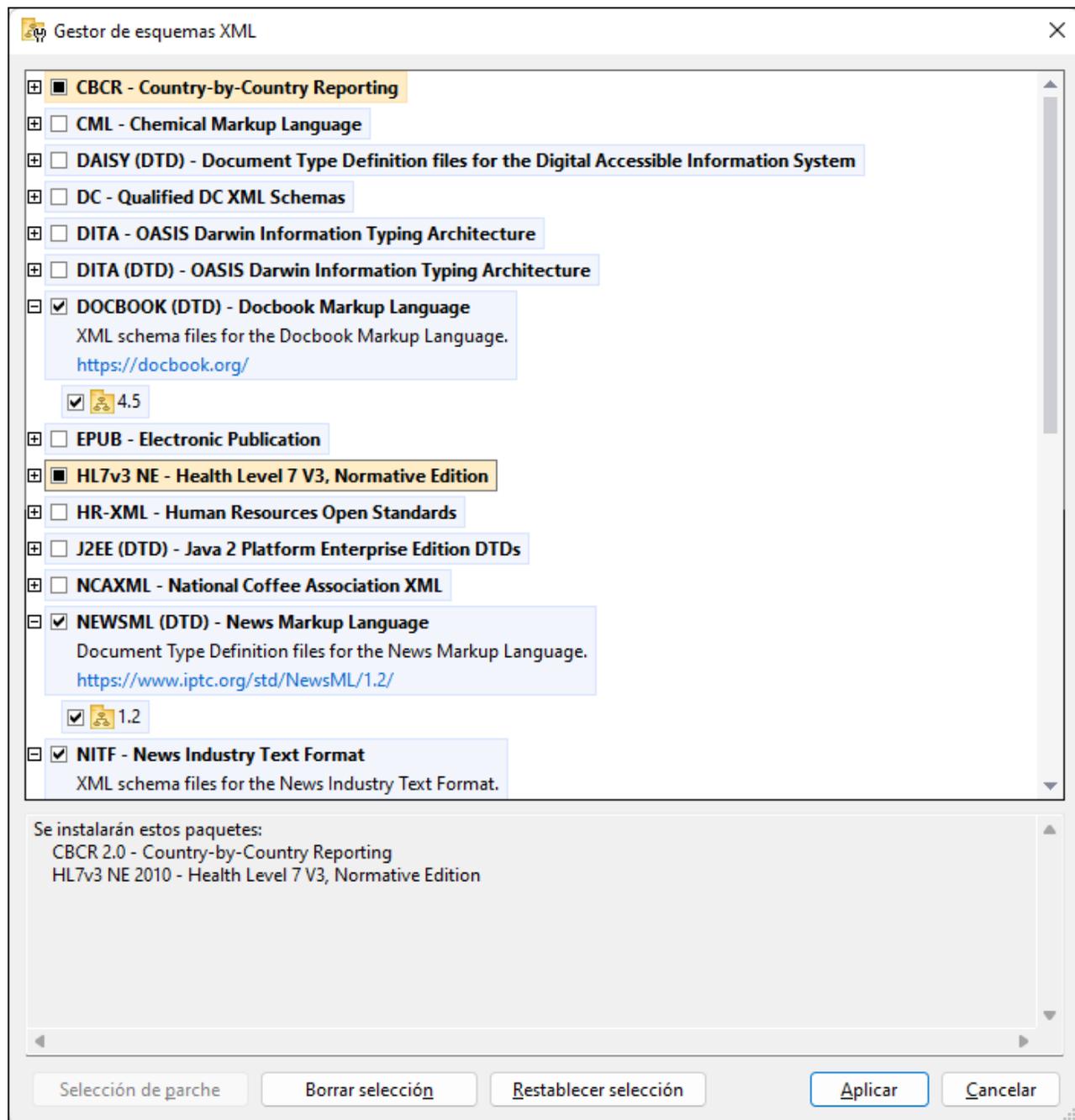


6. Siga los pasos del asistente hasta el final.
7. Al final aparece un cuadro de diálogo "Explorar" donde puede abrir el proyecto de Authentic Desktop (archivo \*.spp). Seleccione el archivo de proyecto para cargar el contenido del proyecto en Authentic Desktop.

## 11 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 [Authentic Desktop](#).

- 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*.)



*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 desinstalar 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 los 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`<sup>161</sup>, Gestor de esquemas instala los esquemas que contiene.

***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:

<i>Windows</i>	C:\ProgramData\Altova\pkgs\.cache
<i>Linux</i>	/var/opt/Altova/pkgs\.cache
<i>macOS</i>	/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 Authentic Desktop por primera vez en un mismo día natural.
- si Authentic Desktop 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](#)<sup>164</sup> 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](#)<sup>162</sup> desde la línea de comandos y después ejecute el comando [initialize](#)<sup>160</sup>.

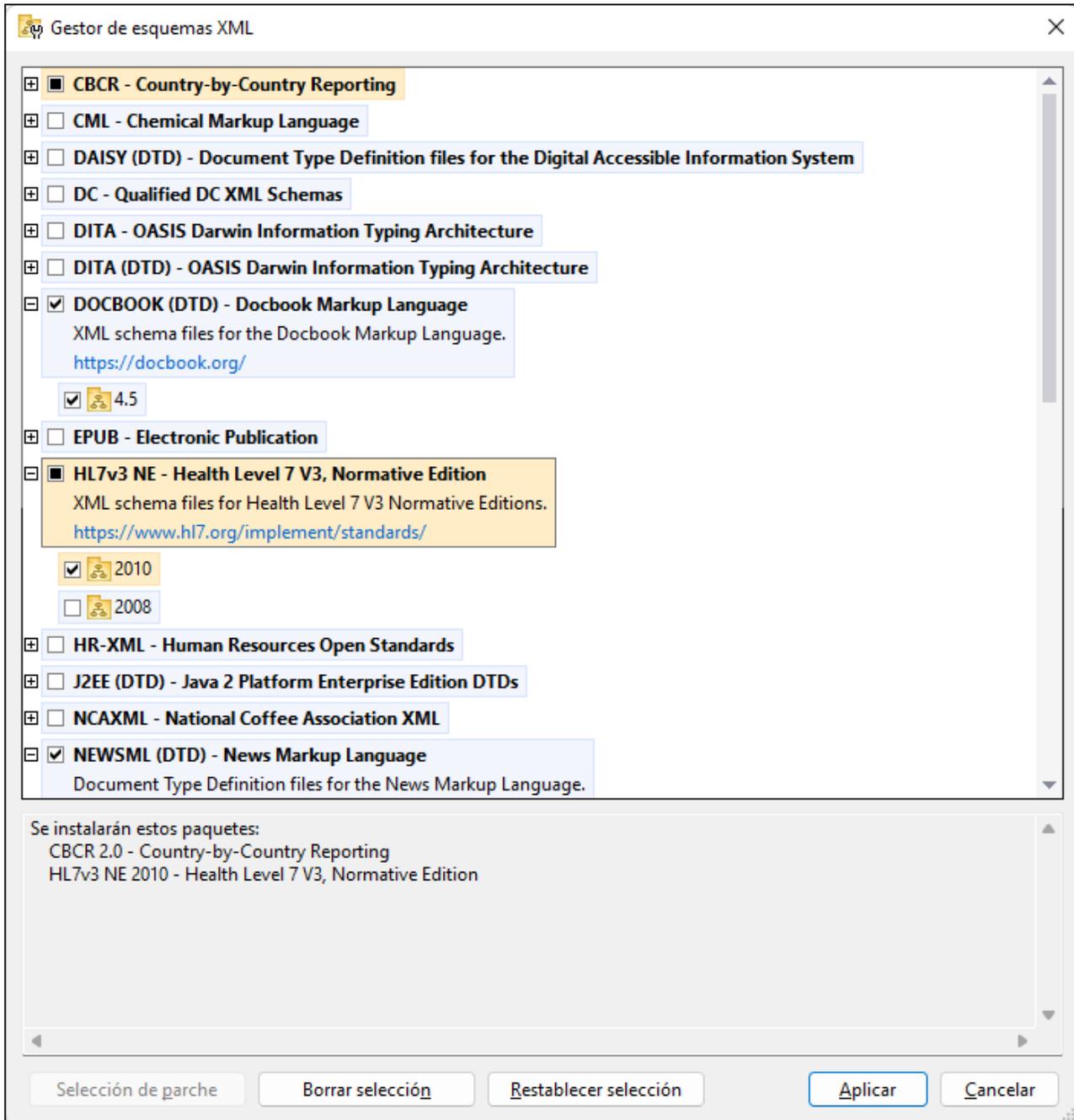
## 11.1 Ejecutar el gestor de esquemas

### Interfaz gráfica del usuario

Hay varias formas de acceder a la IGU del Gestor de esquemas:

- *Durante la instalación de Authentic Desktop*: 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 Authentic Desktop*: 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`.

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](#)<sup>159</sup>, a continuación.

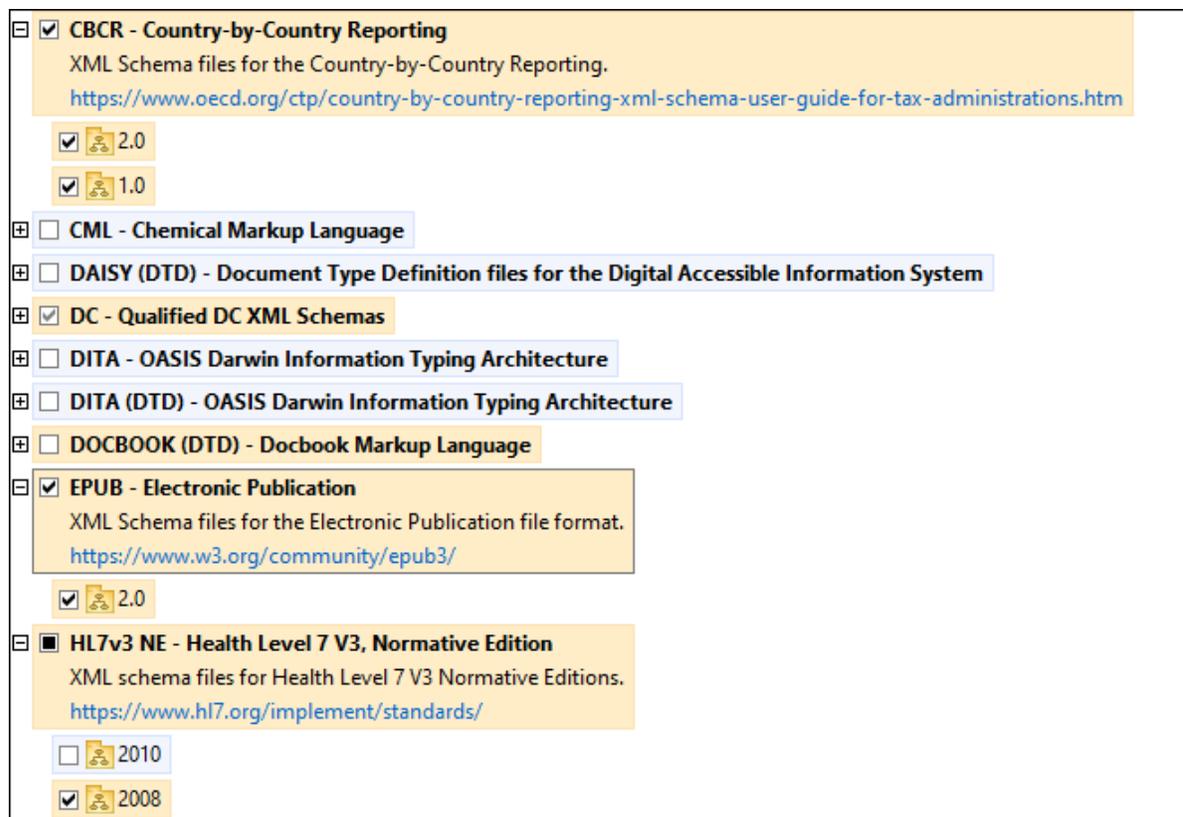
Para mostrar al ayuda de los comandos ejecute:

- *en Windows:* `xmlschemamanager.exe --help`
- *en Linux o macOS (solo para aplicaciones de servidor):* `sudo ./xmlschemamanager --help`

## 11.2 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 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:

<code>xmlschemamanager.exe list</code>	Muestra todos los esquemas instalados y disponibles; también indica qué esquemas se pueden actualizar
<code>xmlschemamanager.exe list -i</code>	Muestra solo los esquemas instalados; también indica qué esquemas se pueden actualizar
<code>xmlschemamanager.exe list -u</code>	Muestra qué esquemas se pueden actualizar

**Nota:** en Linux y macOS use `sudo ./xmlschemamanager list`

## 11.3 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](#) <sup>154</sup>.) 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 la línea de comandos:

1. Ejecute el comando `list -u` <sup>161</sup>. Aparece una lista con los esquemas para las que hay parches disponibles.
1. Ejecute el comando `upgrade` <sup>164</sup> 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 identificador con el formato `<nombre>-<versión>` que aparece junto a cada esquema que muestra el comando `list` <sup>161</sup>. Puede introducir tantos esquemas como quiera. Para más detalles consulte la descripción del comando `install` <sup>161</sup>.

**Nota:** en Linux o macOS, use el comando `sudo ./xmlschemamanager`.

***Instalar un esquema requerido***

Si ejecuta un comando en Authentic Desktop y Authentic Desktop 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**.

## 11.4 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](#)<sup>163</sup>:

```
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](#)<sup>161</sup>. Puede introducir tantos esquemas como quiera. Para más detalles consulte la descripción del comando [uninstall](#)<sup>163</sup>.

**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](#)<sup>162</sup>.

Una vez haya ejecutado este comando, asegúrese de que ejecuta también el comando [initialize](#)<sup>160</sup> para recrear el directorio caché. También puede ejecutar el comando [reset](#)<sup>162</sup> con la opción `-i`.

Recuerde que [reset -i](#)<sup>162</sup> restaura la instalación original del producto, por lo que es recomendable ejecutar el comando [update](#)<sup>164</sup> después de restaurar el gestor. Puede ejecutar el comando [reset](#)<sup>162</sup> con las opciones `-i` o `-u`.

## 11.5 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.

### 11.5.1 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
```

## 11.5.2 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](#) <sup>161</sup>.)
- 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
```

## 11.5.3 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`:

<code>--help, --h</code>	Muestra la ayuda sobre este comando en la línea de comandos.
<code>--silent, --s</code>	Muestra solamente los mensajes de error. El valor predeterminado es <code>false</code> .
<code>--verbose, --v</code>	Muestra información suplementaria durante la ejecución. El valor predeterminado es <code>false</code> .

### Ejemplo

Este comando inicializa el Gestor de esquemas:

```
xmlschemamanager initialize
```

## 11.5.4 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: `cocr-2.10`. Para ver todos los identificadores de esquemas y sus versiones ejecute el comando [list](#)<sup>161</sup>. 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](#)<sup>147</sup>.

### Opciones

Estas son las opciones del comando `install`:

<code>--help, --h</code>	Muestra la ayuda sobre este comando en la línea de comandos.
<code>--silent, --s</code>	Muestra solamente los mensajes de error. El valor predeterminado es <code>false</code> .
<code>--verbose, --v</code>	Muestra información suplementaria durante la ejecución. El valor predeterminado es <code>false</code> .

### Ejemplo

Este comando instala el esquema COCR 2.0 (Country-By-Country Reporting) y el DTD DocBook más reciente:

```
xmlschemamanager install cocr-2.0 docbook
```

## 11.5.5 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`:

<code>--help, --h</code>	Muestra la ayuda sobre este comando en la línea de comandos.
<code>--installed, --i</code>	Muestra solamente los esquemas instaladas. El valor predeterminado es <code>false</code> .
<code>--upgradeable, --u</code>	Muestra solamente los esquemas para las que hay disponible una versión más reciente (parches). El valor predeterminado es <code>false</code> .

## 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`

## 11.5.6 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](#)<sup>160</sup> 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](#)<sup>164</sup> 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`:

<code>--help, --h</code>	Muestra la ayuda sobre este comando en la línea de comandos.
<code>--init, --i</code>	Inicializa el entorno del Gestor de esquemas XML después de una restauración. El valor predeterminado es <code>false</code> .
<code>--silent, --s</code>	Muestra solamente los mensajes de error. El valor predeterminado es <code>false</code> .
<code>--update, --u</code>	Inicializa y actualiza el entorno del Gestor de esquemas XML después de una restauración. El valor predeterminado es <code>false</code> .

<code>--verbose, --v</code>	Muestra información suplementaria durante la ejecución. El valor predeterminado es <code>false</code> .
-----------------------------	---

## 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`

## 11.5.7 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`<sup>161</sup>. 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_taxonomy` descargado desde el sitio web de Altova. Para más información sobre estos archivos consulte la [Introducción al Gestor de esquemas: funcionamiento](#)<sup>147</sup>.

### Opciones

Estas son las opciones del comando `uninstall`:

<code>--help, --h</code>	Muestra la ayuda sobre este comando en la línea de comandos.
<code>--keep-references, --k</code>	Si usa esta opción, los esquemas referenciados no se desinstalan. El valor predeterminado es <code>false</code> .
<code>--silent, --s</code>	Muestra solamente los mensajes de error. El valor predeterminado es <code>false</code> .
<code>--verbose, --v</code>	Muestra información suplementaria durante la ejecución. El valor predeterminado es <code>false</code> .

## Ejemplo

Este comando desinstala los esquemas CBCR 2.0 y EPUB 2.0:

```
xmlschemamanager uninstall cbcrc-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 cbcrc-2.0
```

## 11.5.8 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](#)<sup>162</sup> e [initialize](#)<sup>160</sup>.

### Sintaxis

```
<exec> update [opciones]
```

#### Opciones

Estas son las opciones del comando `update`:

<code>--help, --h</code>	Muestra la ayuda sobre este comando en la línea de comandos.
<code>--silent, --s</code>	Muestra solamente los mensajes de error. El valor predeterminado es <code>false</code> .
<code>--verbose, --v</code>	Muestra información suplementaria durante la ejecución. El valor predeterminado es <code>false</code> .

## Ejemplo

Este comando actualiza la lista de esquemas:

```
xmlschemamanager update
```

## 11.5.9 upgrade

Este comando actualiza todos los esquemas aptos para la versión *patche* 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`:

<code>--help, --h</code>	Muestra la ayuda sobre este comando en la línea de comandos.
<code>--silent, --s</code>	Muestra solamente los mensajes de error. El valor predeterminado es <code>false</code> .
<code>--verbose, --v</code>	Muestra información suplementaria durante la ejecución. El valor predeterminado es <code>false</code> .

## 12 Authentic Desktop en Visual Studio

Authentic Desktop se puede integrar en las versiones 2012/2013/2015/2017/2019/2022 del entorno IDE Microsoft Visual Studio. Esto hace posible incorporar las avanzadas funciones de edición XML al avanzado entorno de desarrollo de Visual Studio.

En esta sección describimos:

- El [proceso general de instalación](#)<sup>167</sup> y la integración del complemento de Authentic Desktop para Visual Studio.
- Las [diferencias](#)<sup>168</sup> entre Authentic Desktop y el complemento de Authentic Desktop para Visual Studio.

## 12.1    Instalar el complemento de Authentic Desktop para Visual Studio

Siga estas instrucciones para instalar el complemento de Authentic Desktop 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 Authentic Desktop . Si tiene instalada la versión 2022 o una más avanzada, debe instalar la versión de 64 bits de Authentic Desktop.
3. Descargue y ejecute el paquete de integración de Authentic Desktop. Este paquete está disponible en la página de descargas de Authentic Desktop de [www.altova.com/es](http://www.altova.com/es).

Tras instalar el paquete de integración podrá empezar a usar Authentic Desktop en el entorno de Visual Studio.

### **Nota importante**

Debe usar el paquete de integración correspondiente a su versión de Authentic Desktop (la versión actual es 2023).

## 12.2 Diferencias entre la versión independiente y la versión para Visual Studio

Este apartado enumera las diferencias entre el complemento de Authentic Desktop para Visual Studio y la versión independiente de Authentic Desktop.

### Ayudantes de entrada (Ventanas de herramientas en Visual Studio)

Los ayudantes de entrada de Authentic Desktop se llaman *Ventanas de herramientas* en Visual Studio. Es necesario tener en cuenta los aspectos siguientes. (Para más información sobre los ayudantes de entrada de Authentic Desktop consulte la sección [Introducción](#) <sup>29</sup>.)

- Puede arrastrar las ventanas de los ayudantes de entrada y colocarlas en cualquier posición del entorno de desarrollo.
- Puede seguir personalizando la interfaz si hace clic con el botón derecho en una pestaña de un ayudante de entrada. El menú contextual que aparece incluye opciones para acoplar la ventana, ocultarla, ocultarla automáticamente o hacerla flotante.

### Comandos de Authentic Desktop como comandos de Visual Studio

Algunos comandos de Authentic Desktop son comandos de Visual Studio en la interfaz de Visual Studio. Se trata de estos comandos:

- **Deshacer/Rehacer:** estos comandos de Visual Studio afectan a todas las acciones realizadas en el entorno de desarrollo Visual Studio.
- **Proyectos:** los proyectos de Authentic Desktop se consideran proyectos de Visual Studio.
- **Personalizar las barras de herramientas/Personalizar los comandos:** las pestañas *Barras de herramientas* y *Comandos* del cuadro de diálogo Personalizar (**Herramientas | Personalizar**) incluye tanto los comandos de Visual Studio como los de Authentic Desktop.
- **Vistas:** en el menú **Ver** el comando **Authentic Desktop** incluye opciones para activar/desactivar las ventanas de los ayudantes de entrada y otras barras laterales, para cambiar de vista de edición y para activar/desactivar determinadas guías de edición.
- **Ayuda de Authentic:** este menú de Authentic Desktop es un submenú del menú **Ayuda** de Visual Studio.

**Nota:** en Visual Studio 2019 y versiones posteriores se puede acceder a las funciones de Authentic Desktop desde el menú **Extensions** de Visual Studio. En las versiones anteriores de Visual Studio puede acceder a ellas desde los menús de nivel superior de Visual Studio.

**Nota:** los comandos de la barra de herramientas no son compatibles. Si ha configurado en Authentic Desktop algún comando de la barra de herramientas que ejecute un comando o script, ese comando no estará disponible en el complemento.

### Otras notas

A continuación incluimos un consejo:

- Para editar un archivo XML con el complemento de Authentic para Visual Studio, seleccione el comando **Archivo | Abrir**. Después, en el cuadro de diálogo "Abrir archivo", utilice la opción **Abrir con** para seleccionar el complemento de Authentic.

## 13 Authentic Desktop en Eclipse

Eclipse es un entorno de desarrollo integrado de código abierto que incorpora varios tipos de aplicaciones en forma de complementos.

Gracias al complemento de Authentic Desktop para Eclipse puede usar todas las funciones de Authentic Desktop desde la plataforma Eclipse 2022-09, 2022-06, 2022-03, 2021-12. El complemento para Eclipse está disponible para todas las plataformas Windows. En esta sección explicamos [cómo instalar](#)<sup>170</sup> el complemento y cómo configurar la [perspectiva Authentic](#)<sup>172</sup>. Después de hacer esto los componentes de la interfaz de Authentic Desktop y los comandos de menú de Authentic Desktop estarán disponibles en la interfaz de Eclipse.

En esta sección explicamos:

- [Cómo instalar el complemento de Authentic Desktop para Eclipse e integrar Authentic Desktop en Eclipse](#)<sup>170</sup>
- [Perspectiva de Authentic DesktopDesktop en Eclipse](#)<sup>172</sup>
- [Otros puntos de entrada de Authentic Desktop en Eclipse](#)<sup>173</sup>

## 13.1 Instalar el complemento de Authentic Desktop para Eclipse

### Requisitos

- Eclipse 2022-09, 2022-06, 2022-03, 2021-12 (<http://www.eclipse.org>).
- Un Java JRE/JDK para la plataforma de 64 bits
- Authentic Desktop 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 Authentic Desktop 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 Eclipse una vez haya instalado el paquete de integración. El paquete de integración de Authentic Desktop 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 Authentic Desktop.

### Instalar el complemento de Authentic Desktop para Eclipse

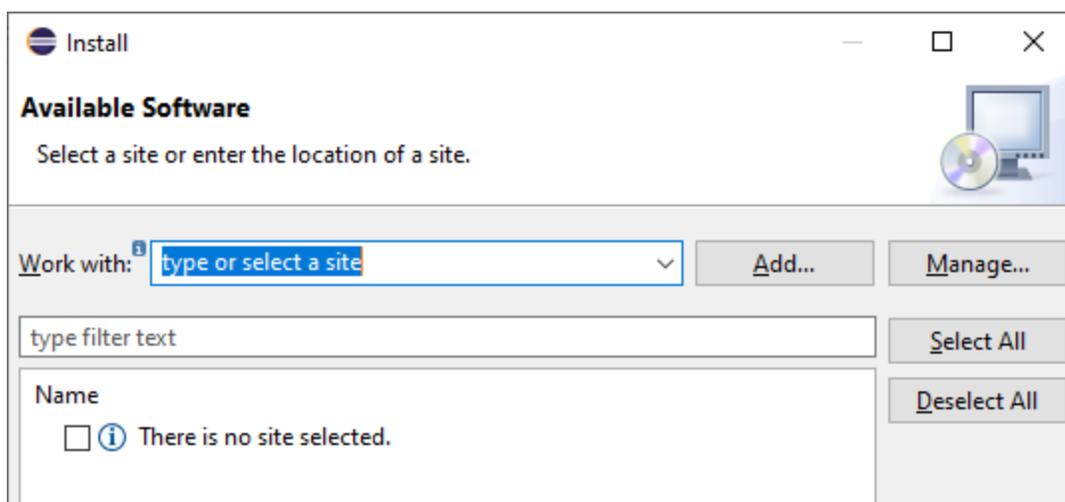
Puede integrar el complemento de Authentic Desktop en Eclipse ante la instalación del paquete de integración de Authentic Desktop. Para ello:

1. Ejecute el paquete de integración de Authentic Desktop 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 Authentic Desktop en Eclipse seleccione *Permitir integración de Authentic Desktop 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.

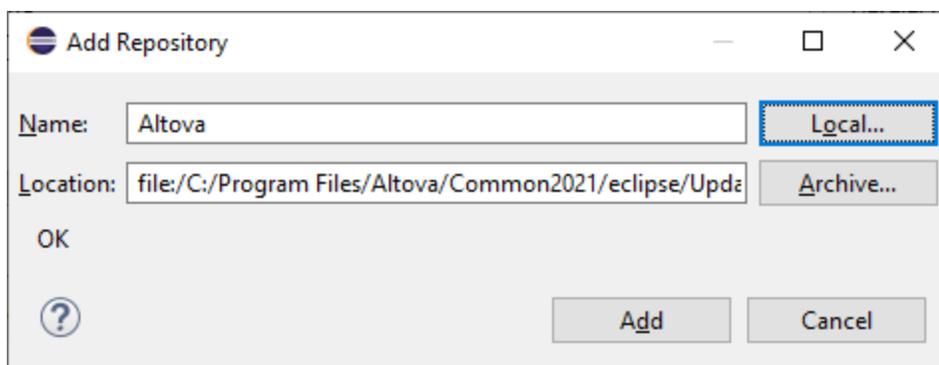
Si elige la integración automática los menús y la perspectiva de Authentic Desktop se habilitarán en Eclipse la siguiente vez que lo inicie.

### Integrar el complemento de Authentic Desktop 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**.



3. En el cuadro de diálogo "Agregar repositorio" haga clic en **Local**. Navegue hasta la carpeta `C:\Program Files\Altova\Common2023\eclipse\UpdateSite` y selecciónela. 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\Authentic Desktop2023\eclipse\UpdateSite` y un nombre como "Altova Authentic Desktop".
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`.

## 13.2 Perspectiva de Authentic Desktop en Eclipse

En Eclipse una perspectiva es una vista de la interfaz del usuario configurada con función de operador. Cuando el complemento de Authentic Desktop está integrado en Eclipse, se crea automáticamente una perspectiva Authentic Desktop predeterminada. Esta perspectiva es una interfaz gráfica del usuario que incluye los elementos de la interfaz de Authentic Desktop: sus vistas de edición, menús, ayudantes de entrada y otras barras laterales.

Al abrir un tipo de archivo asociado a Authentic Desktop (por ejemplo, `.xml`) (por ejemplo, `.mfd`), el archivo se puede editar en la perspectiva Authentic Desktop. 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:

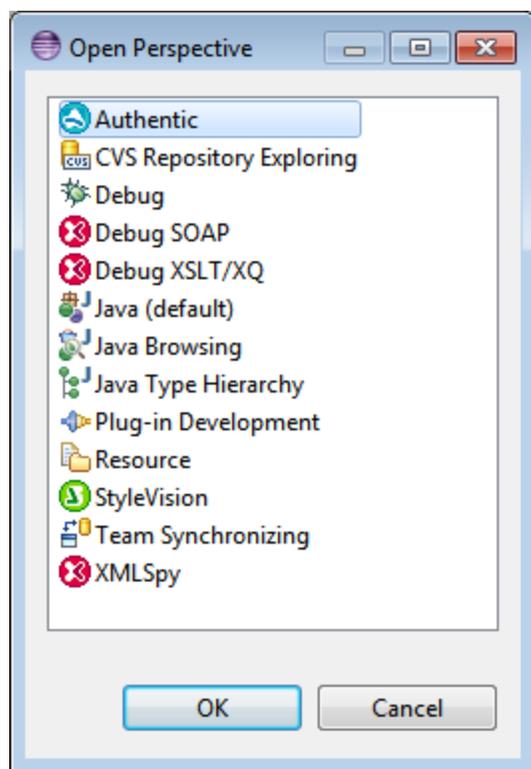
1. Permiten cambiar el entorno de trabajo para el archivo activo
2. Y permiten cambiar de un archivo a otro sin necesidad de abrir un entorno de desarrollo nuevo (el entorno asociado está disponible en la perspectiva)

Para trabajar con la perspectiva Authentic Desktop es necesario:

- Cambiar a la perspectiva Authentic Desktop.
- Configurar las preferencias de la perspectiva Authentic Desktop.
- Personalizar la perspectiva Authentic Desktop.

### Cambiar a la perspectiva Authentic Desktop

En Eclipse seleccione el comando **Window | Perspective | Open Perspective | Other**. En el diálogo que aparece (*imagen siguiente*) seleccione **Authentic Desktop** y haga clic en **OK**.



La ventana vacía o el documento activo tendrán ahora la perspectiva Authentic Desktop. Así es como se cambia de perspectiva desde el menú. Para cambiar más rápido de una perspectiva a otra se puede incluir la perspectiva deseada en el submenú **Open Perspective**, sobre la opción **Other**. Esto se hace en el cuadro de diálogo de personalización.

También puede cambiar de perspectiva al abrir un archivo o activar un archivo. La perspectiva asociada al tipo de archivo se abre automáticamente al abrir el archivo por primera vez. Antes de que cambie la perspectiva, aparece un cuadro de diálogo donde debe indicar si desea abrir automáticamente la perspectiva asociada a este tipo de archivo..

Marque la casilla *Do Not Ask Again* (No volver a preguntar) si desea asociar la perspectiva a ese tipo de archivo sin tener que confirmarlo cada vez que abra un archivo de este tipo. Después haga clic en **OK**.

### Configurar las preferencias de la perspectiva Authentic Desktop

Para abrir el cuadro de diálogo de preferencias, seleccione el comando **Window | Preferences**. En la lista de perspectivas del panel de la izquierda, seleccione Authentic Desktop 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 Authentic Desktop
- Acceso a las opciones de Authentic Desktop.

## Personalizar la perspectiva Authentic Desktop

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 Authentic Desktop) 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.

## 13.3 Otros puntos de entrada de Authentic Desktop en Eclipse

Para acceder a las funciones de Authentic Desktop puede usar estos puntos de entrada en Eclipse:

- Menú Authentic Desktop
- Barra de herramientas Authentic Desktop

### El menú de Authentic Desktop en Eclipse

El menú para Eclipse de **Authentic Desktop** contiene comandos de Authentic Desktop para acceder a las funciones correspondientes de Authentic Desktop. Estos comandos aparecen en varios menús de la versión independiente de Authentic Desktop.

### Barra de herramientas de Authentic Desktop en Eclipse

La barra de herramientas de Authentic Desktop contiene estos botones:



Estos comandos sirven para:

- abrir la Ayuda de Authentic Desktop
- acceder a los comandos de Authentic Desktop (aunque también puede acceder a ellos desde el menú **Authentic Desktop**)

**Nota:** los comandos de la barra de herramientas no son compatibles. Si ha configurado en Authentic Desktop algún comando de la barra de herramientas que ejecute un comando o script, ese comando no estará disponible en el complemento.

## 14 Comandos de menú

Esta sección ofrece una descripción completa de todos los comandos y menús de Authentic Desktop y explica cómo se utilizan. Los comandos estándar de Windows (**Abrir**, **Guardar**, **Cortar**, **Copiar**, **Pegar**, etc.) están en los menús [Archivo](#)<sup>177</sup> y [Edición](#)<sup>193</sup>.

Estos son los comandos que incluye esta sección:

- [Menú Archivo](#)<sup>177</sup>
- [Menú Edición](#)<sup>193</sup>
- [Menú Proyecto](#)<sup>196</sup>
- [Menú XML](#)<sup>230</sup>
- [Menú XSL/XQuery](#)<sup>232</sup>
- [Menú Authentic](#)<sup>239</sup>
- [Menú Vista](#)<sup>250</sup>
- [Menú Explorador](#)<sup>251</sup>
- [Menú Herramientas](#)<sup>252</sup>
- [Menú Ventanas](#)<sup>295</sup>
- [Menú Ayuda](#)<sup>297</sup>

Aquí también encontrará la descripción de los comandos que puede usar desde la [línea de comandos](#)<sup>303</sup>.

## 14.1 Menú Archivo

Los comandos del menú **Archivo** sirven para realizar operaciones con archivos y son:

- [Nuevo](#) <sup>177</sup>
- [Abrir](#) <sup>178</sup>
- [Volver a cargar](#) <sup>183</sup>
- [Codificación](#) <sup>184</sup>
- [Cerrar, Cerrar todos, Cerrar documentos inactivos](#) <sup>184</sup>
- [Guardar, Guardar como, Guardar todos](#) <sup>185</sup>
- [Enviar por correo electrónico](#) <sup>190</sup>
- [Imprimir](#) <sup>191</sup>
- [Vista previa de impresión](#) <sup>191</sup>
- [Configurar impresión](#) <sup>191</sup>
- [Archivos recientes](#) <sup>192</sup>
- [Salir](#) <sup>192</sup>

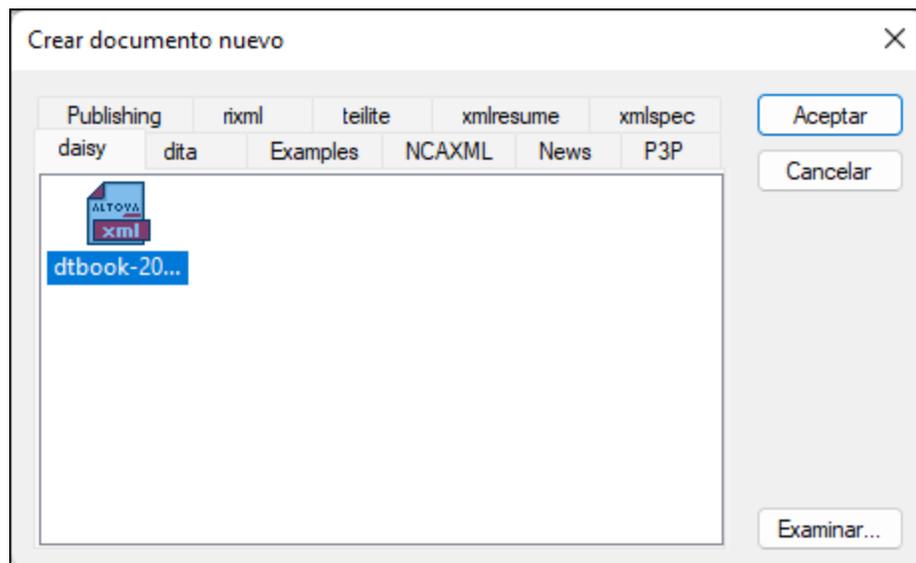
### 14.1.1 Nuevo

Icono y acceso rápido

Icono:	
Acceso rápido:	<b>Ctrl+N</b>

#### Description

Este comando sirve para abrir una plantilla de documento XML nuevo en la Authentic View. La plantilla de documento XML está basada en un archivo SPS (StyleVision Power Stylesheet) y se abre seleccionando el archivo SPS en el cuadro de diálogo "Crear documento nuevo" (*imagen siguiente*). Tras seleccionar el archivo SPS haga clic en **Aceptar**. La plantilla de documento XML definida para el archivo SPS elegido se abre en la Authentic View.



El cuadro de diálogo "Crear documento nuevo" ofrece varias plantillas de documento XML basadas en esquemas y archivos DTD frecuentes. Si lo prefiere, también puede buscar archivos SPS personalizados que tengan asignado un archivo XML. Los archivos SPS se crean con Altova StyleVision, una herramienta con la que puede diseñar plantillas de documento XML a partir de un esquema XML o un archivo DTD. Tras diseñar el archivo SPS en StyleVision puede asignarle un archivo XML como plantilla XML. Los datos de esta plantilla XML aportan los datos de partida de la plantilla de documento que se abre en la Authentic View de Authentic Desktop.

La nueva plantilla de documento XML las propiedades de presentación definidas en el archivo SPS y los datos del archivo XML que seleccionó como plantilla XML. El usuario de la Authentic View puede editar la plantilla de documento XML en una interfaz gráfica WYSIWYG y guardar los cambios en un archivo XML.

## 14.1.2 Abrir

### Icono y acceso rápido

Icono:	
Acceso rápido:	<b>Ctrl+O</b>

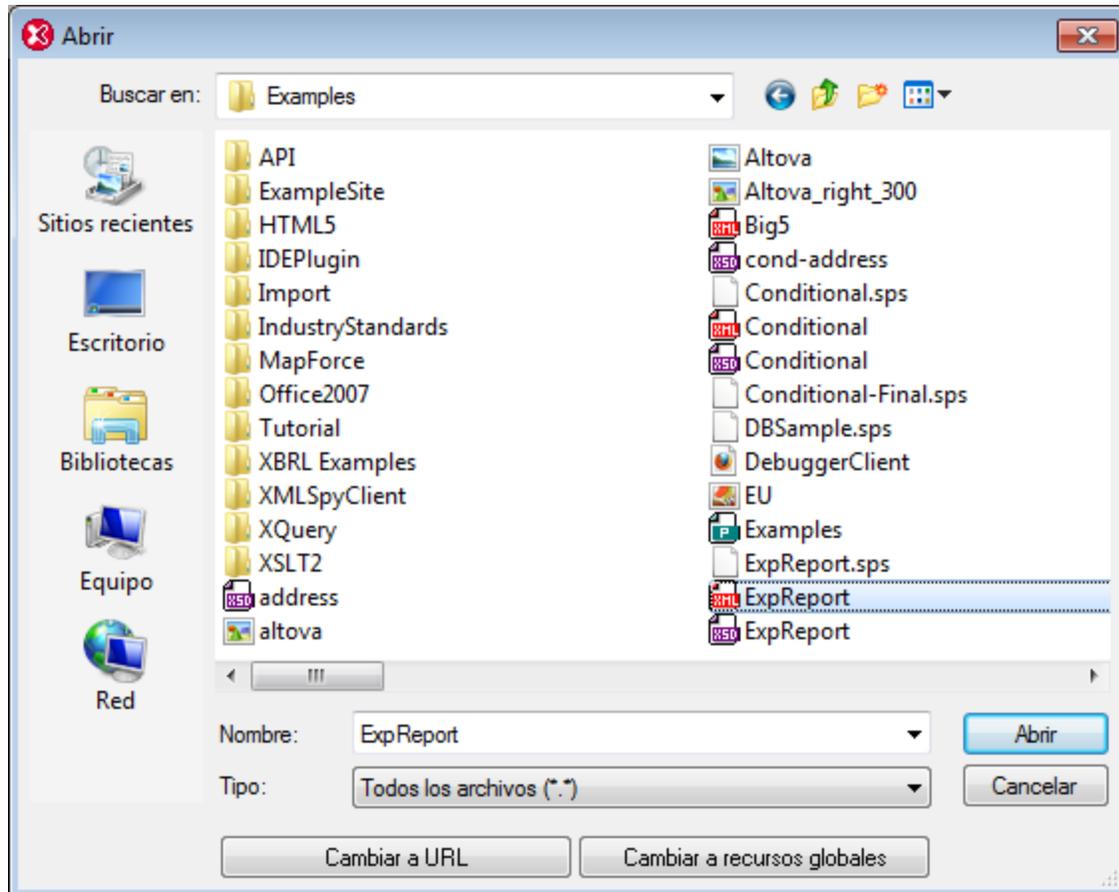
### Descripción

El comando **Abrir** abre el cuadro de diálogo "Abrir" de Windows, donde puede abrir cualquier documento XML o de texto. En el cuadro de diálogo "Abrir" puede seleccionar más de un archivo. Use el cuadro combinado *Tipo*: para filtrar la lista de archivos que aparecen en el cuadro de diálogo. (La lista de tipos de archivos disponibles se puede configurar en la sección *Tipos de archivo* del cuadro de diálogo "Opciones" ([Herramientas | Opciones](#)<sup>281</sup>)). Al abrir un archivo XML, Authentic Desktop comprueba inmediatamente si tiene un formato XML correcto. Si no tiene un formato XML correcto, entonces se emite un mensaje de error. Corrija el error y seleccione el comando de menú [XML | Comprobar formato XML \(F7\)](#)<sup>230</sup>

para volver a revisar el documento. Si seleccionó la opción de [validación automática](#)<sup>279</sup> y el archivo no es válido, entonces aparece un mensaje de error. Corrija el error y seleccione el comando [XML | Validar el documento XML \(F8\)](#)<sup>230</sup> para volver a validar el archivo.

▼ Seleccionar y guardar archivos a través de direcciones URL y recursos globales

En varios cuadros de diálogo "Abrir archivo" y "Guardar archivo" de Authentic Desktop 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*).

Abrir

Dirección URL del archivo:

Abrir como:  Modo automático  XML  DTD

Carga del archivo:  Utilizar caché o proxy  Volver a cargar

Identificación

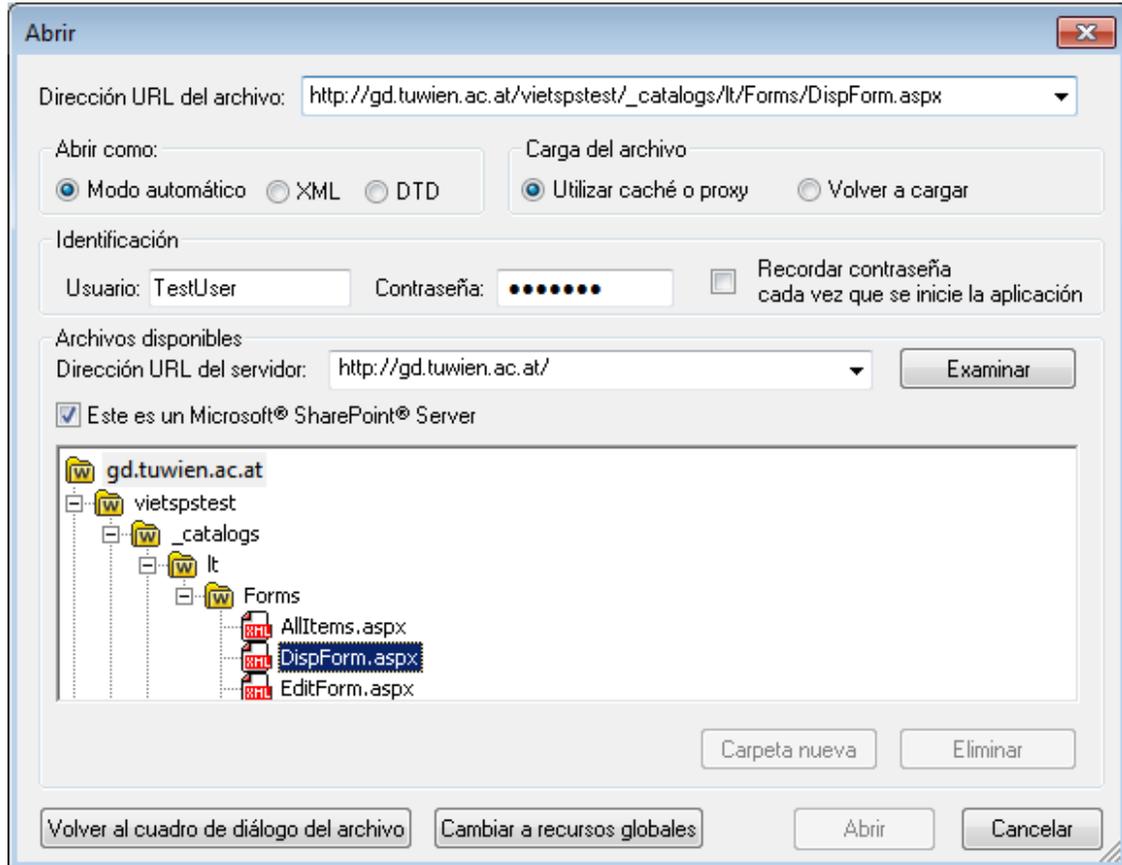
Usuario:  Contraseña:   Recordar contraseña al cerrar aplicación

Archivos disponibles

Dirección URL del servidor:

Este es un Microsoft® SharePoint® Server

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**.

- Haga clic en **Abrir** para cargar el archivo o en **Guardar** para guardarlo.

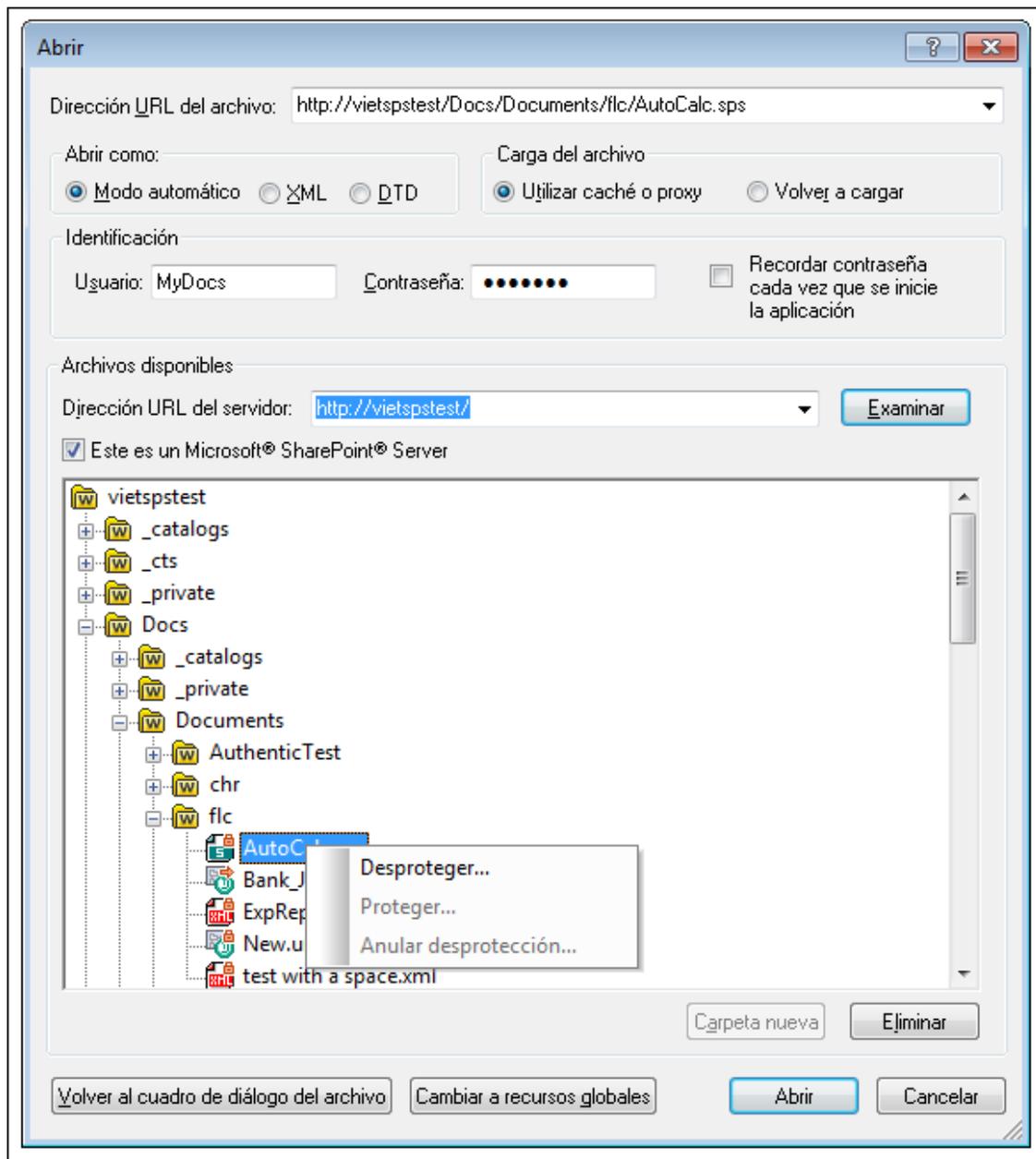
Debe tener en cuenta que:

- 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, Authentic Desktop 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*).

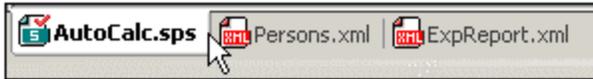
- Estos son los iconos de archivo:

	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

**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](#)<sup>117</sup>. Para obtener información sobre esta característica consulte la sección [Recursos globales](#)<sup>105</sup> de este manual.

## 14.1.3 Volver a cargar

### Icono

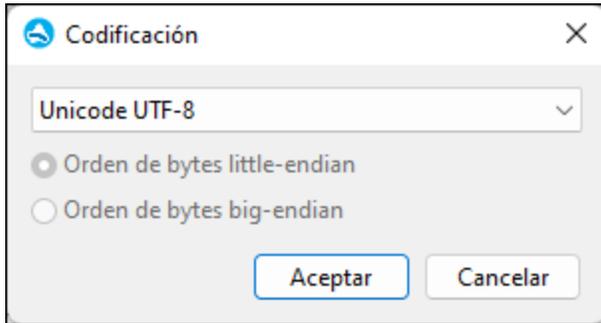


### Descripción

Vuelve a cargar los documentos abiertos que se hayan modificado fuera de Authentic Desktop. Si uno o más documentos se han modificado fuera de Authentic Desktop aparece un aviso que pregunta si quiere volver a cargar los documentos modificados. Si los vuelve a cargar se perderán todos los cambios que haya hecho y todavía no haya guardado.

## 14.1.4 Codificación

Con el comando **Codificación** puede (i) ver la codificación actual del documento activo (tanto XML como no XML) y (ii) seleccionar una codificación distinta. El documento activo se guardará con esa codificación la próxima vez que se guarde.



Si cambia la codificación de un documento XML, el atributo de codificación de la declaración XML se ajusta también. Para las codificaciones de caracteres de dos y cuatro bytes (UTF-16, UCS-2 y UCS-4) también puede especificar el orden de bytes que se debe usar para el archivo. Otra manera de cambiar la codificación de un documento XML es editar el atributo `encoding` de la declaración XML del documento.

Para configurar la codificación predeterminada de sus documentos (tanto XML como no XML) utilice la [sección](#) <sup>283</sup> [Codificación](#) <sup>283</sup> del cuadro de diálogo "Opciones".

**Nota:** al guardar un documento, Authentic Desktop revisa inmediatamente la especificación de codificación y muestra el cuadro de diálogo "Codificación", donde el usuario puede seleccionar la codificación adecuada. Si al guardar el documento se detecta que contiene caracteres que no se pueden representar en la codificación seleccionada, Authentic Desktop emite una advertencia a tal efecto.

## 14.1.5 Cerrar, Cerrar todos, Cerrar documentos inactivos

### Cerrar

El comando **Cerrar** cierra el documento activo. Si se realizaron cambios en el archivo (si hay un asterisco \* después del nombre del archivo en la barra de título), Authentic Desktop le pregunta si desea guardar el archivo antes de cerrarlo.

### Cerrar todos

El comando **Cerrar todos** cierra todos los documentos que están abiertos. Si se realizaron cambios en algún archivo (si hay un asterisco \* después del nombre del archivo en la barra de título), Authentic Desktop le pregunta si desea guardar el archivo antes de cerrarlo.

## Cerrar documentos inactivos

El comando **Cerrar documentos inactivos** cierra todos los documentos que están abiertos, excepto el documento que está activo en la ventana principal. Si se realizaron cambios en algún archivo (si hay un asterisco \* después del nombre del archivo en la barra de título), Authentic Desktop le pregunta si desea guardar el archivo antes de cerrarlo.

## 14.1.6 Guardar, Guardar como, Guardar todos

### Iconos y acceso rápido

Comando	Icono	Acceso rápido
<b>Guardar</b>		<b>Ctrl+S</b>
<b>Guardar todos</b>		

### Guardar

Este comando guarda el contenido del documento activo en el archivo desde el que se abrió. Al guardar un documento, se comprueba automáticamente [si el documento tiene un formato XML correcto](#)<sup>230</sup>. El archivo también se valida automáticamente si esta opción está seleccionada en la sección *Archivo* del cuadro de diálogo "Opciones" ([Herramientas | Opciones](#)<sup>279</sup>). Authentic Desktop también revisa la declaración XML y la especificación de [codificación](#)<sup>283</sup> y esta codificación se aplica al documento al guardarlo.

### Guardar como

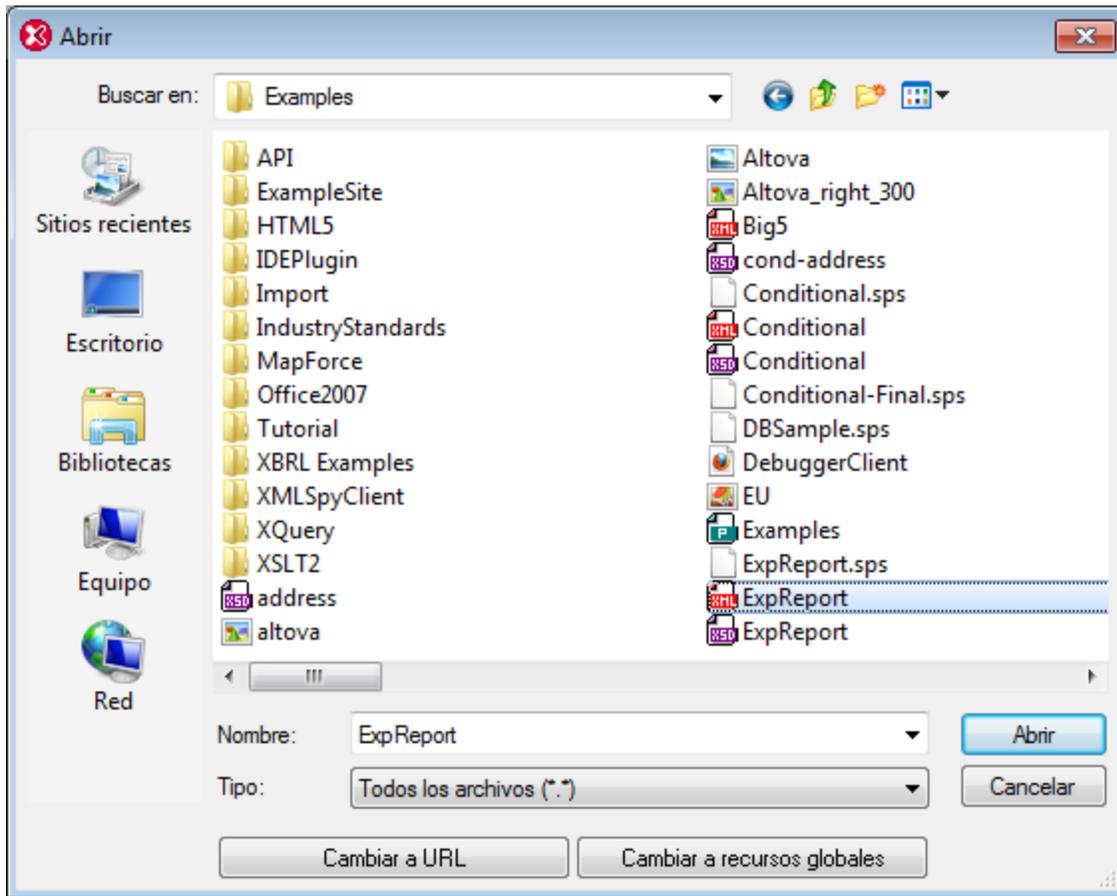
El comando **Guardar como** abre el cuadro de diálogo "Guardar como" de Windows. En este cuadro de diálogo puede introducir el nombre y la ubicación con los que desea guardar el archivo activo. Al igual que ocurre con el comando **Guardar**, la aplicación comprueba el formato XML, revisa la codificación y valida el documento.

### Guardar todos

El comando **Guardar todos** guarda todos los cambios realizados en los documentos que están abiertos. Este comando es práctico si está editando varios documentos a la vez. Si alguno de los documentos no se guardó todavía (porque se acaba de crear, por ejemplo) entonces aparece el cuadro de diálogo "Guardar como".

#### ▼ Seleccionar y guardar archivos a través de direcciones URL y recursos globales

En varios cuadros de diálogo "Abrir archivo" y "Guardar archivo" de Authentic Desktop 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*).

Abrir

Dirección URL del archivo:

Abrir como:  Modo automático  XML  DTD

Carga del archivo:  Utilizar caché o proxy  Volver a cargar

Identificación

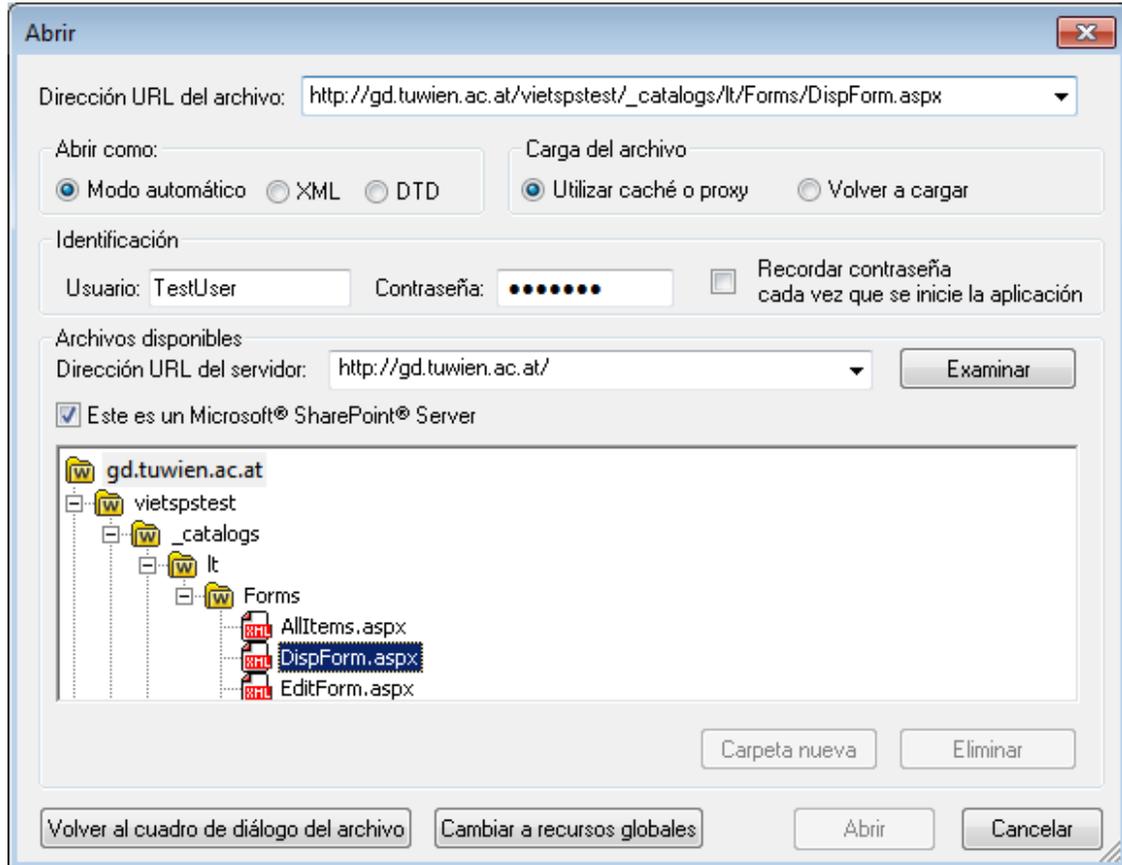
Usuario:  Contraseña:   Recordar contraseña al cerrar aplicación

Archivos disponibles

Dirección URL del servidor:

Este es un Microsoft® SharePoint® Server

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**.

- Haga clic en **Abrir** para cargar el archivo o en **Guardar** para guardarlo.

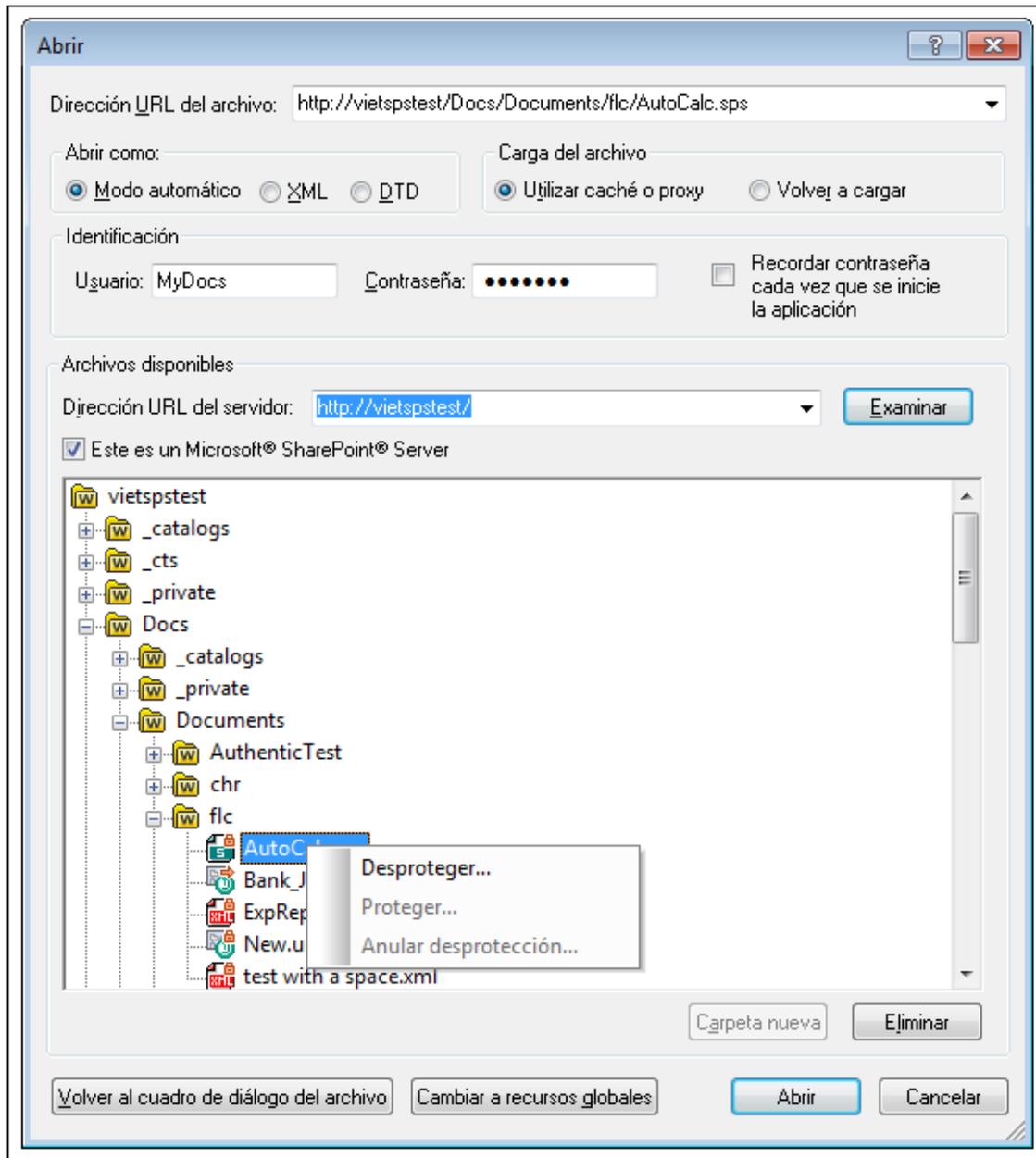
Debe tener en cuenta que:

- 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, Authentic Desktop 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

**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](#)<sup>117</sup>. Para obtener información sobre esta característica consulte la sección [Recursos globales](#)<sup>105</sup> de este manual.

## 14.1.7 Enviar por correo electrónico

### Icono

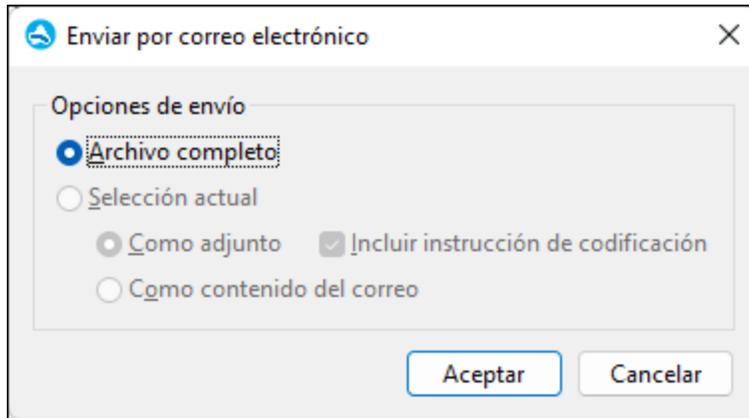


### Descripción

El comando **Enviar por correo electrónico...** sirve para enviar documentos XML o una selección de un documento XML por correo electrónico. El documento o la selección se puede enviar como documento adjunto, como contenido del correo o como enlace, dependiendo del tipo de documento que sea (*ver tabla siguiente*).

¿Qué puedo enviar?	¿Cómo?
Documento XML o PXF activo	Como adjunto
Uno o varios archivos de la ventana Proyecto	Como adjunto
Una o varias URL de la ventana Proyecto	Como adjunto o como enlace en el correo

- Cuando se invoca el comando **Enviar por correo electrónico...** para una selección del documento XML activo, aparece el cuadro de diálogo "Enviar por correo electrónico", donde puede elegir entre varias opciones (*imagen siguiente*). Si no hay texto seleccionado en el documento activo e invoca el comando **Enviar por correo electrónico...** entonces la única opción disponible en el cuadro de diálogo es la opción *Archivo completo* (las demás opciones se deshabilitan).



- Puesto que los archivos enviados desde la ventana Proyecto siempre se envían como documentos adjuntos, este cuadro de diálogo no aparece y se abre directamente el correo electrónico que incluye los archivos seleccionados en forma de adjuntos. +
- Las URL de la ventana Proyecto se pueden enviar como adjuntos o como enlaces (*imagen siguiente*). Seleccione una opción y haga clic en **Aceptar**.

## 14.1.8 Imprimir

### Icono y acceso rápido

Icono:	
Acceso rápido:	<b>Ctrl+P</b>

### Descripción

El comando **Imprimir** abre el cuadro de diálogo "Imprimir", donde puede configurar la impresión. Después puede imprimir el documento activo tal y como aparece en la vista actual.

## 14.1.9 Vista previa de impresión, Configurar impresión

### Vista previa de impresión

Al hacer clic en el comando **Vista previa de impresión** de la vista Authentic se abre una vista previa de impresión del documento activo.

El modo *Vista previa de impresión* incluye una barra de herramientas (en la parte superior izquierda) con opciones relacionadas con la impresión y con la vista previa. Los botones de navegación se encuentran en la parte inferior de la ventana de vista previa.

**Nota:** siga estos pasos para habilitar el uso de colores e imágenes de fondo en la vista previa de impresión: (i) en el menú **Herramientas** de Internet Explorer, haga clic en **Opciones de Internet** y después en la pestaña *Opciones avanzadas*. (ii) En el panel Configuración, marque la casilla *Imprimir colores e imágenes de fondo* de la sección Imprimir. (iii) Haga clic en **Aceptar** para terminar.

## Configurar impresión

El comando **Configurar impresión** abre el cuadro de diálogo "Configurar impresión" (*imagen siguiente*), donde puede seleccionar el formato del papel y la orientación de la página, entre otras opciones. Estas opciones se aplican a todos los trabajos de impresión subsiguientes.

### 14.1.10 Archivos recientes, Salir

#### Archivos recientes

Al final del menú **Archivo** aparece una lista con los nueve archivos utilizados más recientemente, empezando por el archivo que se abrió más recientemente. Puede abrir cualquiera de esos archivos haciendo clic en ellos. Par abrir un archivo de esta lista usando el teclado, pulse Alt+F para abrir el menú **Archivo** y después pulse el número que corresponda al archivo que quiere abrir.

#### Salir

Cierra Authentic Desktop. Si tiene archivos abiertos con cambios que no ha guardado, se le pedirá que guarde dichos cambios. Authentic Desktop también guarda las modificaciones en la configuración del programa y la información sobre los archivos más utilizados recientemente.

## 14.2 Menú Edición

Los comandos del menú **Edición** sirven para editar documentos en Authentic Desktop. Algunos ejemplos son los comandos **Deshacer**<sup>193</sup>, **Rehacer**<sup>193</sup>, **Cortar**<sup>193</sup>, **Copiar**<sup>193</sup>, **Pegar**<sup>193</sup>, **Eliminar**<sup>193</sup>, **Seleccionar todo**<sup>194</sup>, **Buscar**<sup>194</sup>, **Buscar siguiente**<sup>194</sup> y **Reemplazar**<sup>195</sup>.

### 14.2.1 Deshacer, Rehacer

Iconos y acceso rápido

Comando	Icono	Acceso rápido
<b>Deshacer</b>		<b>Ctrl+Z</b>
<b>Rehacer</b>		<b>Ctrl+Y</b>

#### Deshacer

El comando **Deshacer** permite eliminar todos los cambios realizados y devolver el archivo a versiones anteriores. Todos los cambios se pueden deshacer uno por uno y no hay un límite de operaciones deshacer. El historial de operaciones deshacer se mantiene tras usar el comando **Guardar**, lo cual permite devolver el archivo al estado en que estaba antes de que guardara los cambios. Es decir puede recorrer el historial de acciones con los comandos **Deshacer** y **Rehacer** (*ver siguiente*).

#### Rehacer

El comando **Rehacer** permite rehacer las acciones que deshizo con el comando **Deshacer**. Esto significa que puede ir adelante y atrás en el historial de acciones con los comandos **Deshacer** y **Rehacer**.

### 14.2.2 Cortar, Copiar, Pegar, Eliminar

Iconos y acceso rápido

Comando	Icono	Acceso rápido
<b>Cortar</b>		<b>Ctrl+X</b> o <b>Mayús+Supr</b>
<b>Copiar</b>		<b>Ctrl+C</b>
<b>Pegar</b>		<b>Ctrl+V</b>
<b>Eliminar</b>		<b>Supr</b>

#### Cortar

El comando **Cortar** copia el texto o los componentes seleccionados en el portapapeles y los elimina de su ubicación actual.

## Copiar

El comando **Copiar** copia el texto o los componentes seleccionados en el portapapeles. Puede usar este comando para duplicar datos dentro de Authentic Desktop o para mover datos a otra aplicación.

## Pegar

El comando **Pegar** inserta el contenido del portapapeles en la posición actual del cursor.

## Eliminar

El comando **Eliminar** elimina el texto o los componentes seleccionados sin colocarlos en el portapapeles.

## 14.2.3 Seleccionar todo

El comando **Seleccionar todo (Ctrl+A)** sirve para seleccionar el contenido de todo el documento.

## 14.2.4 Buscar, Buscar siguiente

### Iconos y acceso rápido

Comando	Icono	Acceso rápido
<b>Buscar</b>		<b>Ctrl+F</b>
<b>Buscar siguiente</b>		<b>F3</b>

### Buscar

El comando **Buscar** abre el cuadro de diálogo "Buscar", donde puede indicar qué cadena de texto desea buscar y configurar la búsqueda. Dependiendo de la vista que esté usando, el cuadro de diálogo "Buscar" muestra unas opciones u otras. Escriba el término de búsqueda en el campo *Buscar* o seleccione una opción del cuadro combinado, que recuerda los 10 términos de búsqueda más recientes. Después configure las opciones de búsqueda.

**Nota:** los comandos **Buscar** y **Buscar siguiente** también sirven para buscar nombres de archivo y carpeta cuando en la ventana Proyecto está seleccionado un proyecto.

### Buscar siguiente

El comando **Buscar siguiente** repite la última búsqueda realizada con el comando **Buscar** y busca la siguiente instancia del término de búsqueda.

## 14.2.5 Reemplazar

### Icono y acceso rápido

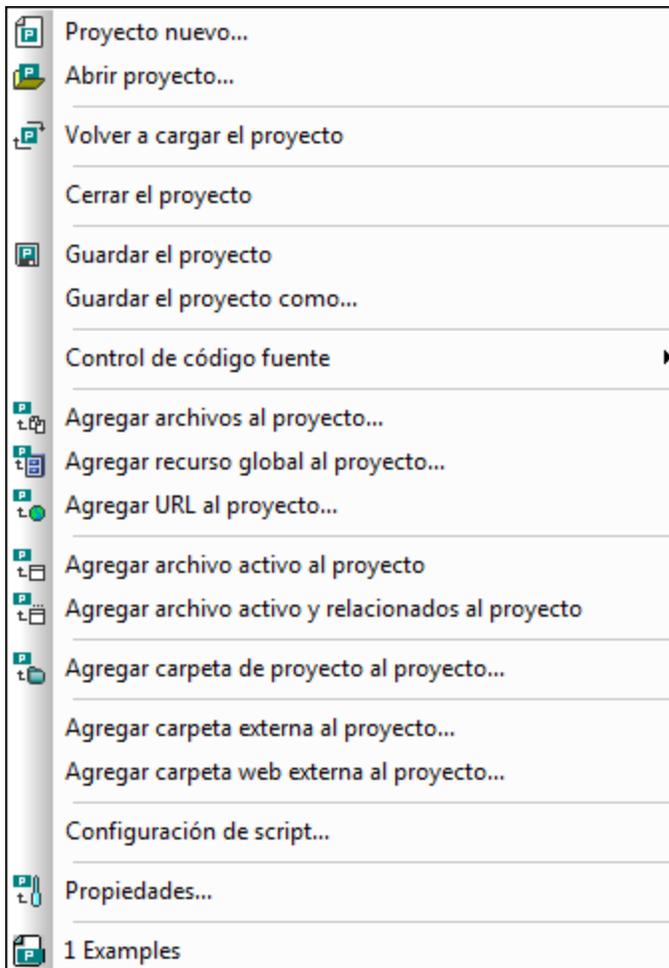
Comando	Icono	Acceso rápido
Reemplazar		Ctrl+H

### Descripción

Con el comando **Reemplazar** puede buscar y reemplazar una cadena de texto con otra. Este comando tiene las mismas características que el comando **Buscar**<sup>194</sup>. Dependiendo de la vista que esté usando, el cuadro de diálogo Reemplazar muestra unas opciones u otras. Puede reemplazar los términos encontrados uno por uno o usar el botón **Reemplazar todos** para realizar una operación global de búsqueda y reemplazo.

## 14.3 Menú Proyecto

Authentic Desktop usa una estructura en forma de árbol para gestionar y organizar varios archivos o direcciones URL en proyectos XML. Los [archivos](#)<sup>215</sup> y [direcciones URL](#)<sup>215</sup> se pueden organizar por [carpetas](#)<sup>216</sup> según su extensión de archivo o cualquier otro criterio, para que pueda estructurar su proyecto y realizar operaciones por lotes con total facilidad.



**Nota importante:** la mayoría de los comandos para trabajar con proyectos también están disponibles en el menú contextual que aparece al hacer clic con el botón derecho en cualquier elemento de la ventana Proyecto.

### Rutas de archivo relativas y absolutas

Los proyectos se guardan como archivo de proyecto y tienen la extensión `.spp`. Se trata, de hecho, de documentos XML que se pueden editar como si fueran archivos XML normales y corrientes. En el archivo de proyecto se usan rutas de archivo absolutas para archivos y carpetas que están al mismo nivel o en un nivel superior, mientras que para archivos y carpetas de la carpeta actual o subcarpetas se usan rutas de archivo relativas. Por ejemplo, si la estructura de directorios de su proyecto es como esta:

```
| -Carpeta1
```

```
|      |
|      |-Carpeta2
|      |
|      |-Carpeta3
|      |
|      |-Carpeta4
```

Y su archivo `.spp` está guardado en la `Carpeta3`, entonces las referencias a los archivos de la `Carpeta1` y `Carpeta2` serán así::

```
c:\Carpeta1\NombreDeArchivo.ext
c:\Carpeta1\Carpeta2\NombreDeArchivo.ext
```

Por su parte, las referencias a los archivos de la `Carpeta3` y `Carpeta4` serán así:

```
.\NombreDeArchivo.ext
.\Carpeta4\NombreDeArchivo.ext
```

Si desea que todos las rutas de archivo sean relativas, guarde los archivos `.spp` en el directorio raíz de su disco duro.

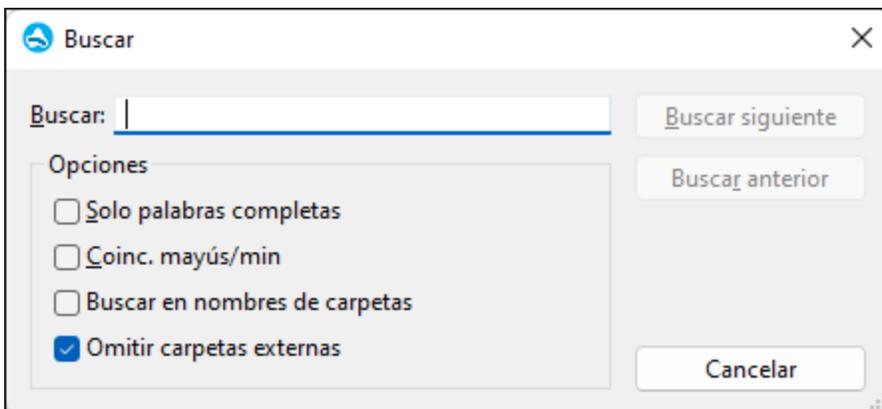
## Arrastrar y colocar

En la ventana Proyecto puede arrastrar carpetas hasta otra carpeta o hasta otra posición de la misma carpeta. Los archivos se pueden arrastrar hasta otra carpeta, pero no se pueden cambiar de posición dentro de la misma carpeta (en las carpetas los archivos se organizan alfabéticamente). Además, puede arrastrar archivos y carpetas desde el explorador de Windows hasta la ventana Proyecto de XMLSpy.

## Cómo realizar búsquedas en el proyecto

Puede buscar archivos y carpetas del proyecto por su nombre o solo parte del nombre. Si se encuentra un archivo o una carpeta que coincida con la búsqueda, estos se resaltan uno por uno en la ventana Proyecto.

Para iniciar una búsqueda, en la ventana Proyecto seleccione la carpeta de proyecto en la que desea buscar y después seleccione el comando **Edición | Buscar (Ctrl+F)**. Aparece el cuadro de diálogo Buscar (*imagen siguiente*). Escriba el texto que desea buscar y seleccione las opciones de búsqueda que correspondan (*véase más adelante*).



Este cuadro de diálogo ofrece estas opciones de búsqueda:

- Puede buscar *palabras completas* solamente. Esto limita los resultados porque el nombre de carpeta o de archivo debe incluir la palabra completa. En los nombres de archivo, la parte situada delante y detrás del punto (sin el punto) se considera una palabra.
- Puede buscar sólo los archivos o carpetas cuyo nombre coincida con el texto de búsqueda en el uso de *mayúsculas y minúsculas*.
- Puede buscar en los *nombres de carpeta* o sólo en los nombres de archivo.
- Puede incluir o excluir las [carpetas externas](#)<sup>217</sup> en la búsqueda. Las carpetas externas son carpetas reales del sistema de archivos o de la red. Por el contrario, las carpetas de proyecto se crean dentro del proyecto pero no en el sistema.

Si la búsqueda encuentra resultados, se resalta el primer resultado en la ventana Proyecto. Para navegar por el resto de resultados de la búsqueda, haga clic en **Buscar siguiente** y **Buscar anterior**.

## Actualizar los proyectos

Si se realizan cambios en una carpeta externa, el cambio no se refleja en la ventana Proyecto a no ser que se actualice el proyecto.

## Recursos globales en el menú contextual

Al hacer clic con el botón derecho en una carpeta en la ventana Proyecto, el menú contextual que aparece incluye la opción **Agregar recurso global**, que sirve para añadir [recursos globales](#)<sup>105</sup>. Al hacer clic en este comando aparece el cuadro de diálogo "Agregar recurso global". Este diálogo enumera todos los recursos globales de tipo archivo o carpeta disponibles en el archivo XML de recursos globales activo. Seleccione un recurso global y este se añade a la carpeta de proyecto seleccionada.

## Proyectos y proveedores de control de versiones

Si desea añadir un proyecto de Authentic Desktop a un repositorio de control de versiones, antes es necesario comprobar si la posición de los archivos del proyecto en la estructura del sistema de archivos permite añadir archivos desde un nivel inferior (entendiendo el directorio raíz como nivel superior de la estructura de directorios).

En otras palabras, el directorio donde está ubicado el **archivo de proyecto** básicamente representa el **directorio raíz** del proyecto en el repositorio de control de versiones. Los archivos que se añadan desde un nivel superior (al directorio raíz) se añadan al proyecto de Authentic Desktop, pero su posición en el repositorio puede ser diferente a la esperada, si es que se pueden colocar en esa posición.

Por ejemplo, en la estructura de carpetas que aparece más arriba, si un archivo de proyecto se guarda en la `Carpeta3` y se pone bajo control de versiones:

- Los archivos que se añadan en la `Carpeta1` no se pueden poner bajo control de versiones.
- Los archivos que se añadan a la `Carpeta2` se ponen bajo control de versiones en el directorio raíz del repositorio y no dentro de la carpeta de proyecto.
- Los archivos que se añadan a la `Carpeta3` y `Carpeta4` se añadan en la posición esperada y se ponen bajo control de versiones.

### 14.3.1 Proyecto nuevo



El comando **Proyecto nuevo** crea un proyecto **nuevo** en Authentic Desktop. Si al hacer clic en este comando está trabajando en otro proyecto, aparece un mensaje preguntando si desea cerrar todos los documentos que pertenecen al proyecto actual. El nombre del proyecto se le asigna al guardarlo en un archivo **.spp**.

### 14.3.2 Abrir proyecto



El comando **Abrir proyecto...** abre un proyecto de Authentic Desktop. Si al hacer clic en este comando está trabajando en otro proyecto, primero se cierra el proyecto actual.

### 14.3.3 Volver a cargar el proyecto



El comando **Volver a cargar el proyecto** vuelve a cargar el proyecto desde el disco. Si trabaja en un entorno de varios usuarios, a veces puede ser necesario volver a cargar el proyecto si otro usuario realizó cambios en el proyecto.

**Nota importante:** los archivos de proyecto (archivos **.spp**) son de hecho documentos XML que se pueden editar como cualquier otro archivo XML.

### 14.3.4 Cerrar el proyecto

El comando **Cerrar el proyecto** sirve para **cerrar** el proyecto activo. Si se realizaron cambios en el proyecto, la aplicación le pregunta si desea guardar el proyecto antes de cerrarlo. Si se realizaron cambios en el proyecto, esto se indica en la ventana Proyecto con un asterisco junto al nombre del proyecto.

## 14.3.5 Guardar el proyecto, Guardar el proyecto como



El comando **Guardar el proyecto** sirve para **guardar** el proyecto actual. También puede hacer clic en la ventana Proyecto y pulsar el icono .

El comando **Guardar el proyecto como** sirve para **guardar** el proyecto actual con un nombre nuevo.

## 14.3.6 Control de código fuente

Su aplicación de Altova es compatible con Microsoft SourceSafe y otros repositorios de archivos. El apartado [Sistemas de control de código fuente compatibles](#) <sup>124</sup> incluye una lista con todos los sistemas compatibles con su aplicación de Altova. En este apartado describimos los comandos del submenú **Proyecto | Control de código fuente**, que sirven para trabajar con el sistema de control de código fuente desde su aplicación de Altova.

### Resumen de la función de control de código fuente

Este es el mecanismo para colocar archivos de un proyecto bajo control de código fuente:

1. En Authentic Desktop se crea una carpeta de proyecto que contiene los archivos que se deben colocar bajo control de código fuente. Esta carpeta se corresponde normalmente con una carpeta local donde están guardados los archivos del proyecto. La ruta de acceso de la carpeta local se denomina ruta de acceso local.
2. En la base de datos del sistema de control de código fuente (también llamada repositorio) se crea una carpeta que contendrá los archivos que se deben poner bajo control de código fuente.
3. Los archivos del proyecto se añaden al control de código fuente con el comando [Proyecto | Control de código fuente | Agregar al control de código fuente](#) <sup>207</sup>.
4. Las acciones de control de código fuente (como proteger, desproteger y quitar archivos del control de código fuente) se pueden llevar a cabo con los comandos del submenú [Proyecto | Control de código fuente](#) <sup>200</sup>. Los comandos de este submenú se describen en los siguientes subapartados.

**Nota:** si desea cambiar de proveedor de control de código fuente, tiene dos opciones: (i) cambiar las opciones de control de código fuente ([Herramientas | Opciones | Control de código fuente](#) <sup>291</sup>) o (ii) cambiar de proveedor en el diálogo "Cambiar control de código fuente" ([Proyecto | Control de código fuente | Cambiar control de código fuente](#) <sup>214</sup>).

**Nota:** un proyecto de control de código fuente no es lo mismo que un proyecto de la aplicación. Los proyectos de control de código fuente dependen de la estructura del directorio, mientras que los proyectos de Authentic Desktop son construcciones lógicas que no dependen directamente de la estructura del directorio.

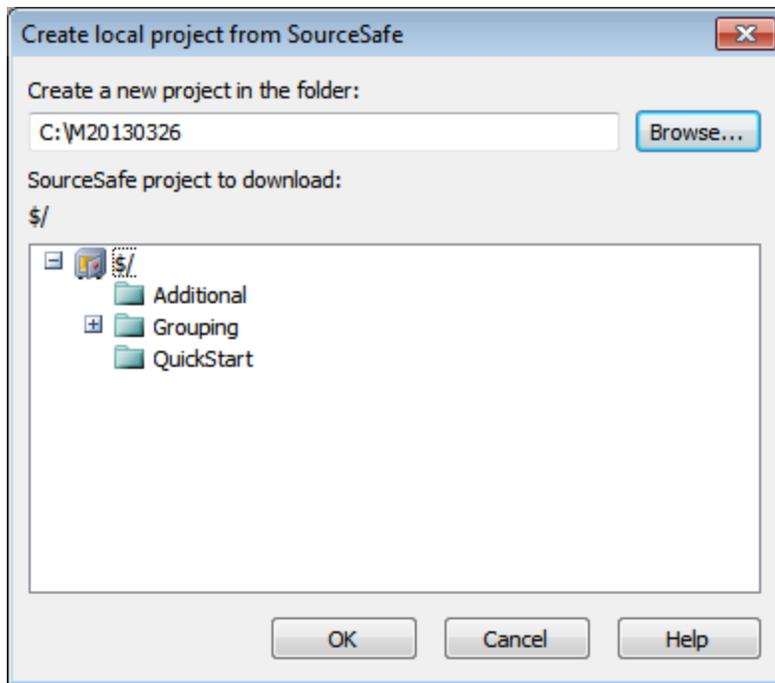
Para más información, consulte la sección [Control de código fuente](#) <sup>121</sup>.

### 14.3.6.1 Abrir desde el control de código fuente

El comando **Abrir desde el control de código fuente** crea un proyecto de aplicación nuevo a partir de un proyecto bajo control de código fuente.

Estos son los pasos para crear el proyecto:

1. Dependiendo del sistema de control de código fuente que utilice, quizás sea necesario comprobar que ningún archivo del proyecto está desprotegido antes de crear un proyecto nuevo.
2. No es necesario tener abierto un proyecto en la aplicación.
3. Seleccione el comando **Proyecto | Control de código fuente | Abrir desde el control de código fuente**.
4. El sistema de control de código fuente que tiene instalado muestra sus cuadros de diálogo de verificación y conexión. Conéctese al repositorio pertinente (es decir, a la carpeta del repositorio que se corresponde con la carpeta local).
5. En el cuadro de diálogo que aparece (*imagen siguiente*) busque la carpeta local en la que desea copiar el contenido de la carpeta del repositorio (a la que se acaba de conectar). En la imagen siguiente, por ejemplo, la carpeta asociada del repositorio se llama `MiProyecto` y se representa con el signo `$`. La carpeta local es `C:\M20130326`.



6. Haga clic en **OK**. El contenido de la carpeta asociada (`MyProject`) se copiará a la carpeta local `C:\M20130326`, y aparece un cuadro de diálogo pidiendo que seleccione el archivo de proyecto (archivo `.spp`) que desea crear como proyecto nuevo.
7. Seleccione el archivo `.spp` que se copió a la carpeta local. En nuestro ejemplo, se trata del archivo `MiProyecto.spp` situado en la carpeta `C:\M20130326`. En la aplicación se crea un proyecto nuevo llamado `MiProyecto` y este proyecto aparece en la ventana Proyecto. Los archivos del proyecto están en la carpeta `C:\M20130326`.

## Símbolos de control de código fuente

Las carpetas y los archivos aparecen con un icono diferente dependiendo de su estado:

	Protegido. Se puede desproteger.
	Desprotegido por otro usuario. No se puede desproteger.
	Desprotegido localmente. Se puede editar y proteger.

### 14.3.6.2 Habilitar control de código fuente

El comando **Habilitar control de código fuente** sirve para habilitar/deshabilitar el control de código fuente para un proyecto de la aplicación. Este comando habilita/deshabilita el control de código fuente para todo el proyecto, aunque se seleccione desde el menú contextual de un archivo o de una carpeta. Una vez habilitado el control de código fuente, el estado de los archivos se recupera y se muestra en la ventana proyecto.

	Protegido. Se puede desproteger.
	Desprotegido por otro usuario. No se puede desproteger.
	Desprotegido localmente. Se puede editar y proteger.

### 14.3.6.3 Obtener la versión más reciente

El comando **Obtener la versión más reciente** (del menú **Proyecto | Control de código fuente**) recupera y coloca en el directorio de trabajo la versión más reciente de los archivos seleccionados del control de código fuente. Los archivos se recuperan como archivos de solo lectura y no están desprotegidos. Este comando es igual que el comando **Obtener**<sup>202</sup>, la única diferencia es que no aparece el cuadro de diálogo "Obtener".

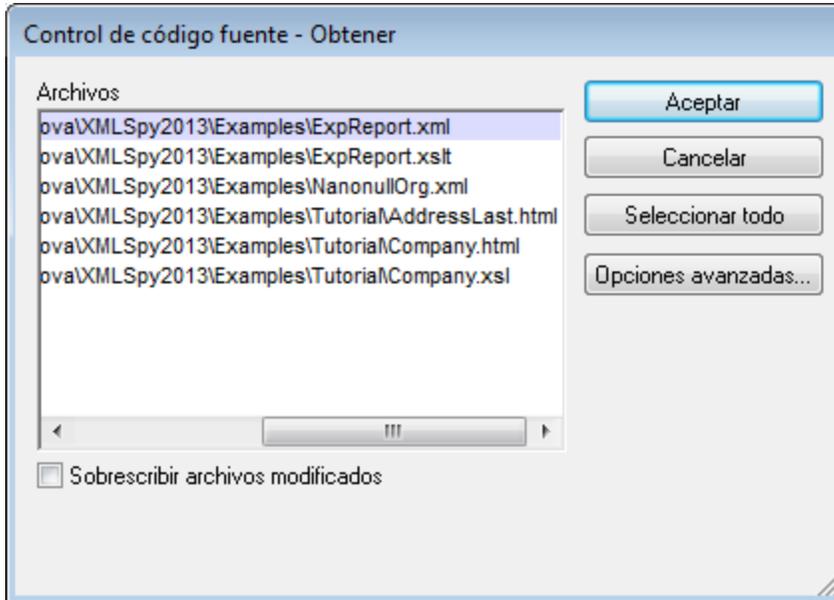
Si los archivos seleccionados están desprotegidos, la acción que se toma dependerá de lo que haga su sistema de control en situaciones así. Por lo general, el sistema de control de código fuente le preguntará si desea reemplazar el archivo desprotegido, combinarlo o dejarlo tal y como está.

**Nota:** cuando seleccione este comando para una carpeta, el comando se aplica de forma recursiva. Esto significa que se aplica a todos los archivos situados dentro de la carpeta actual.

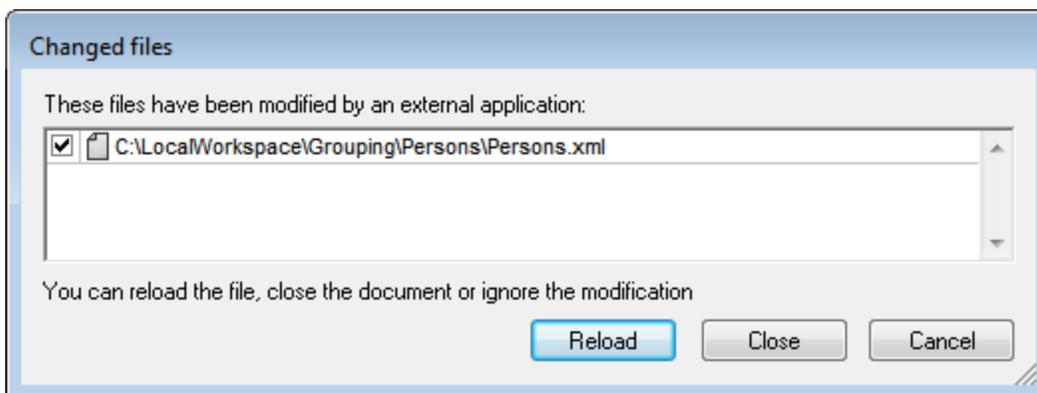
### 14.3.6.4 Obtener, Obtener carpetas

El comando **Obtener** (del menú **Proyecto | Control de código fuente**) recupera los archivos del repositorio como archivos de solo lectura. (Para poder editar un archivo, antes es necesario desprotegerlo.) El cuadro de diálogo "Obtener" enumera los archivos del objeto (proyecto o carpeta) sobre el que se ejecutó el comando **Obtener** (*imagen siguiente*). En el cuadro de diálogo puede marcar las casillas de los archivos que desea recuperar.

**Nota:** el comando **Obtener carpetas** sirve para seleccionar subcarpetas del repositorio (si su sistema de control de código fuente permite esta opción).

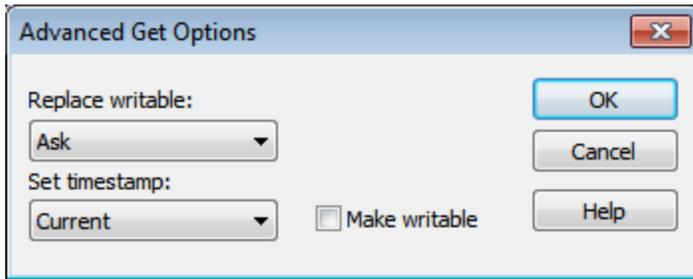


Si quiere, puede sobrescribir los archivos desprotegidos modificados marcando la opción *Sobrescribir archivos modificados* situada en la parte inferior del cuadro de diálogo. Al hacer clic en **Aceptar** los archivos se sobrescriben. Si alguno de los archivos sobrescritos está abierto, aparece un aviso preguntando si desea volver a cargar los archivos (botón **Volver a cargar**), cerrar los archivos (botón **Cerrar**) o conservar la vista actual del archivo (botón **Cancelar**).



### Opciones avanzadas

El cuadro de diálogo "Opciones avanzadas" (*imagen siguiente*) se abre con el botón **Opciones avanzadas** del cuadro de diálogo "Obtener" (*primera imagen de este apartado*).



Aquí puede seleccionar (i) si reemplaza los archivos que se pueden escribir y que están desprotegidos, (ii) la marca de tiempo y (iii) si la propiedad de solo lectura del archivo recuperado se cambia para que el archivo se pueda escribir.

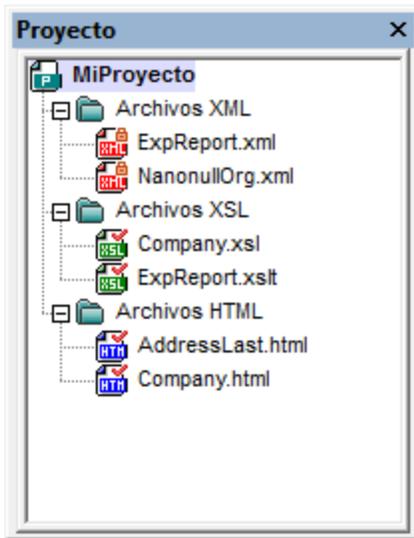
### 14.3.6.5 Desproteger, Proteger

Tras poner un archivo de proyecto bajo control de código fuente, puede desprotegerlo o protegerlo. Para ello seleccione el archivo (en la ventana Proyecto) y seleccione el comando pertinente en el menú **Proyecto | Control de código fuente**, es decir **Desproteger** o **Proteger**.

Cuando se desprotege un archivo, se pone una copia del archivo del repositorio en la carpeta local. Los archivos desprotegidos se pueden editar. Si un archivo que está bajo control de código fuente no se desprotege, no se puede editar. Después de editar el archivo, los cambios se pueden guardar en el repositorio protegiendo el archivo. Incluso si no guarda el archivo, al protegerlo se guardan los cambios en el repositorio. El icono del archivo indica si está desprotegido o protegido.

	Protegido. Se puede desproteger.
	Desprotegido por otro usuario. No se puede desproteger.
	Desprotegido localmente. Se puede editar y proteger.

Si selecciona el proyecto o una carpeta del proyecto, se seleccionan todos los archivos del objeto seleccionado. Para seleccionar varios objetos (archivos y carpetas) pulse la tecla **Ctrl** mientras hace clic en los objetos. La imagen siguiente muestra un proyecto que se desprotegió. Después se protegió el archivo `ExpReport.xml`.



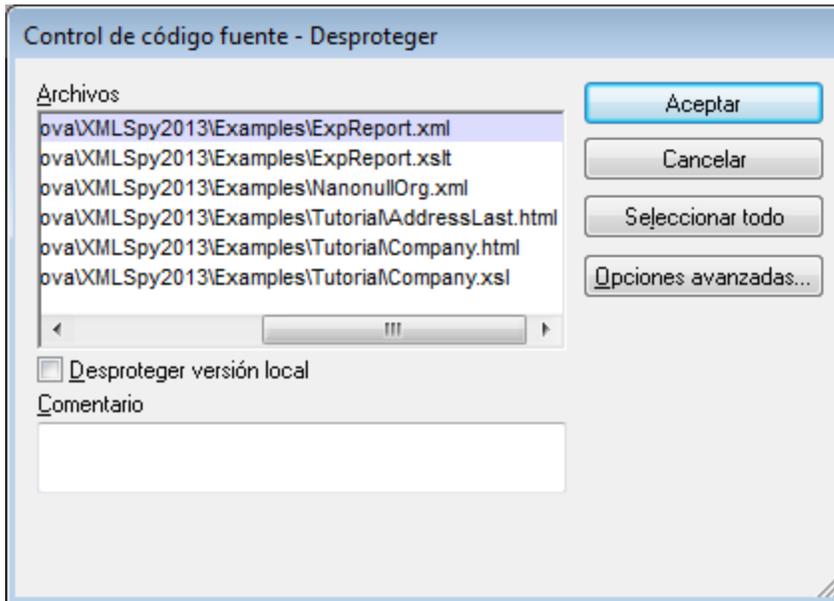
### Guardar cambios y rechazar cambios

Recuerde que al proteger un archivo puede elegir si se protegen los cambios o no. Es decir, puede guardar los cambios en el repositorio y seguir con el archivo desprotegido. Esto es muy útil si desea guardar cambios cada cierto tiempo y después seguir editando el archivo desprotegido.

Si desprotegió un archivo y realizó cambios pero quiere rechazarlos por cualquier motivo, puede volver a la versión del documento guardada en el repositorio con solo hacer clic en el comando **Proyecto | Control de código fuente | Deshacer desprotección**.

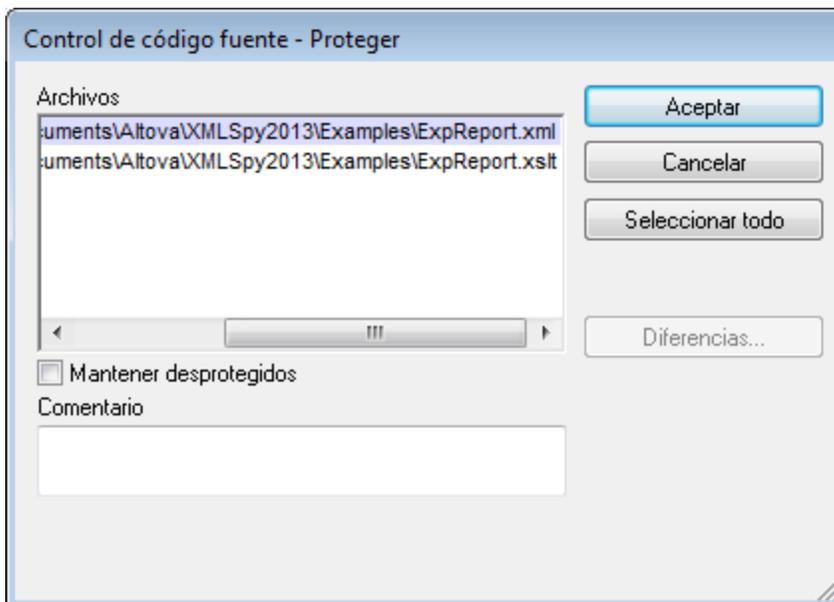
### Desproteger

El cuadro de diálogo "Desproteger" (*imagen siguiente*) sirve para: (i) seleccionar los archivos que desea desproteger y (ii) seleccionar si se desprotege la versión del repositorio o la versión local.



## Proteger

El cuadro de diálogo "Proteger" (*imagen siguiente*) sirve para: (i) seleccionar los archivos que desea proteger y (ii) mantener el archivo desprotegido.



**Nota:** en estos dos cuadros de diálogo aparecen varios archivos si el objeto seleccionado (el proyecto o las carpetas de proyecto) contienen varios archivos.

### 14.3.6.6 Anular desprotección

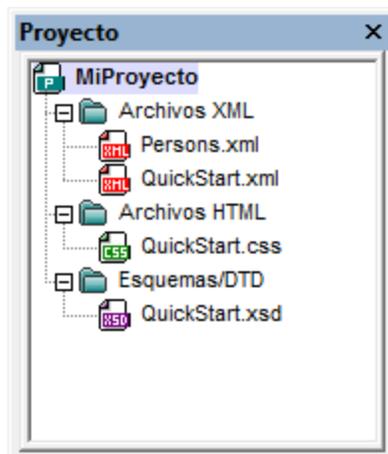
Si desprotegió un archivo y realizó cambios pero ahora prefiere rechazar esos cambios por cualquier motivo, puede volver a la versión del documento guardada en el repositorio. Para ello seleccione el comando **Proyecto | Control de código fuente | Anular desprotección**.

Las carpetas y los archivos aparecen con un icono diferente dependiendo de su estado:

	Protegido. Se puede desproteger.
	Desprotegido por otro usuario. No se puede desproteger.
	Desprotegido localmente. Se puede editar y proteger.

### 14.3.6.7 Agregar al control de código fuente

Después de añadir un proyecto al control de código fuente, puede añadir archivos o grupos de archivos al sistema de control. Seleccione el archivo en la ventana Proyecto y haga clic en el comando **Proyecto | Control de código fuente | Agregar al control de código fuente**. Para seleccionar varios archivos mantenga pulsada la tecla **Ctrl** mientras hace clic en los archivos que desea añadir. Si ejecuta el comando en una carpeta de proyecto (de color verde), entonces se agregan al control de código fuente todos los archivos y subcarpetas de la carpeta.



Cuando se agregan archivos al control de código fuente, la jerarquía de la carpeta local se copia al repositorio (no la jerarquía de la carpeta de proyecto). Es decir, si un archivo está en una subcarpeta de la carpeta local, entonces se crean en el repositorio la carpeta primaria y las demás carpetas antecesoras del archivo.

Cuando se añade el primer archivo de un proyecto al control de código fuente, se crean los enlaces correspondientes en el repositorio y se añade automáticamente el archivo de proyecto (archivo .spp). Para más información consulte el apartado [Agregar al control de código fuente](#) <sup>129</sup>.

## Símbolos de control de código fuente

Las carpetas y los archivos aparecen con un icono diferente dependiendo de su estado:

	Protegido. Se puede desproteger.
	Desprotegido por otro usuario. No se puede desproteger.
	Desprotegido localmente. Se puede editar y proteger.

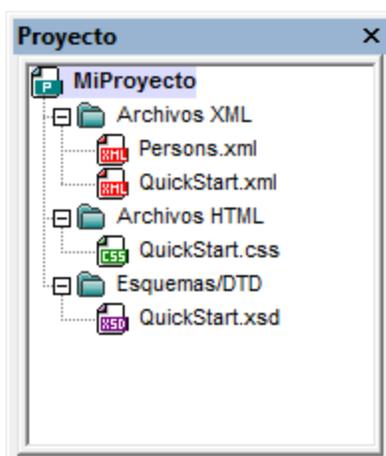
### 14.3.6.8 Quitar del control de código fuente

Para quitar un archivo del control de código fuente, selecciónelo y haga clic en el comando **Proyecto | Control de código fuente | Quitar del control de código fuente**. También puede quitar: (i) archivos de una carpeta de proyecto ejecutando el comando sobre la carpeta, (ii) varios archivos, si mantiene pulsada la tecla **Ctrl** mientras los selecciona y (iii) todo el proyecto, ejecutando el comando sobre el proyecto.

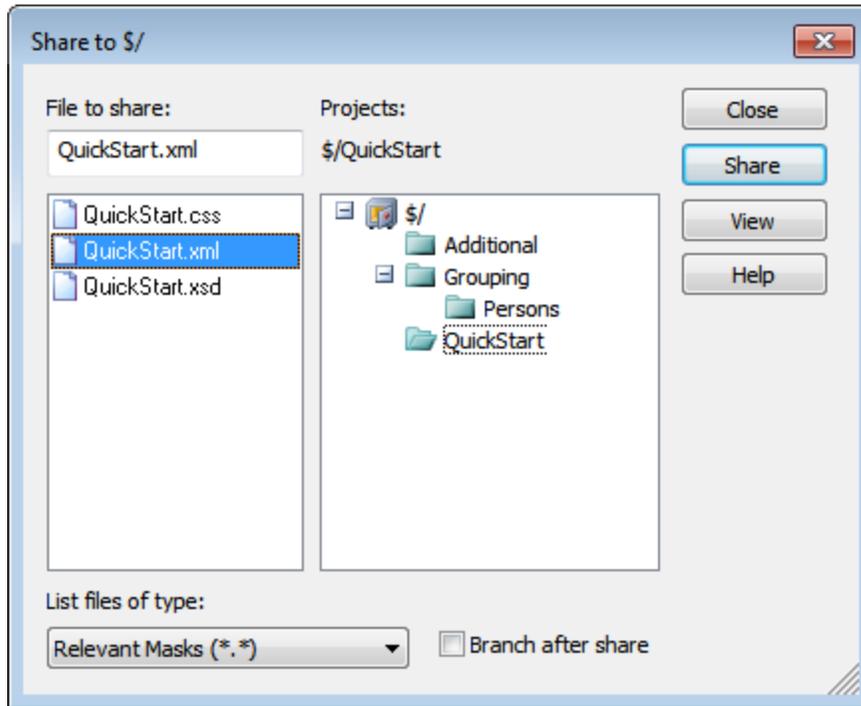
### 14.3.6.9 Compartir desde el control de código fuente

El comando **Compartir desde el control de código fuente** está disponible si su sistema de control de código fuente permite compartir contenido. Por ejemplo, puede compartir un archivo para que esté disponible en varias ubicaciones locales. Si se realizan cambios en uno de estos archivos locales, los cambios se reflejan en todas las versiones "compartidas" del archivo.

En la ventana Proyecto seleccione el proyecto (*imagen siguiente*) y después haga clic en **Compartir desde el control de código fuente**.

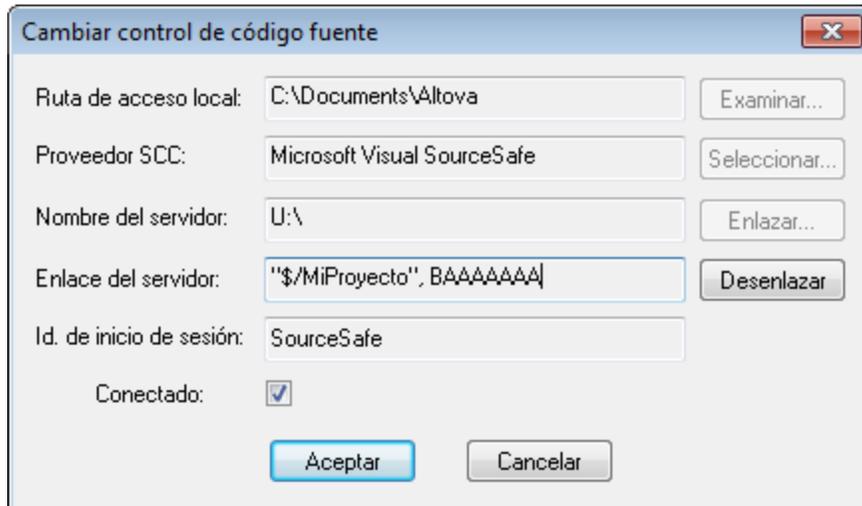


Aparece el cuadro de diálogo "Share To [carpeta]" (*imagen siguiente*).



Ahora puede seleccionar los archivos que desea compartir. Primero seleccione en el panel derecho la carpeta donde están los archivos. Los archivos de la carpeta elegida aparecen en el panel izquierdo. Seleccione el archivo que desea compartir (para seleccionar varios pulse la tecla **Ctrl** mientras hace clic en los archivos). Los archivos seleccionados aparecen en el cuadro situado en la parte superior izquierda del diálogo. Haga clic en **Share** (*compartir*) y después en **Close** (cerrar) para copiar los archivos seleccionados a la carpeta local compartida.

El nombre de la carpeta compartida aparece en el nombre del cuadro de diálogo. En la imagen anterior, por ejemplo, se trata de la carpeta local (porque el signo \$ representa la carpeta del repositorio a la que está enlazada la carpeta local). En el cuadro de diálogo "Cambiar control de código fuente" puede ver y configurar la carpeta compartida (*imagen siguiente*) cambiando la ruta de acceso local y el enlace de servidor.

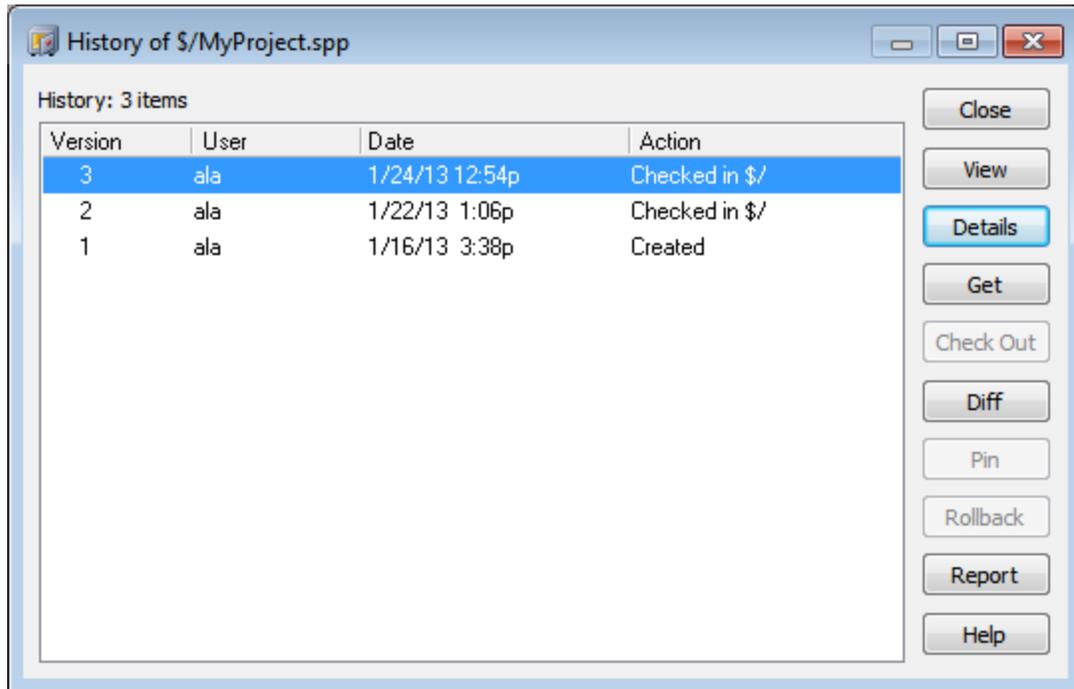


Para más información consulte la documentación de su sistema de control de código fuente.

#### 14.3.6.10 Mostrar historial

El comando **Mostrar historial** activa la función del mismo nombre del sistema de control de código fuente activo. Muestra el historia del archivo seleccionado en la ventana Proyecto. Seleccione el título del proyecto para ver el historial del archivo de proyecto (archivo .spp). Puede ver información sobre las versiones previas de un archivo y las diferencias con la versión actual y recuperar versiones previas del archivo.

La siguiente imagen muestra el diálogo "Historial" del sistema de control de versiones Visual SourceSafe y enumera todas las versiones del archivo `MyProject.spp`.



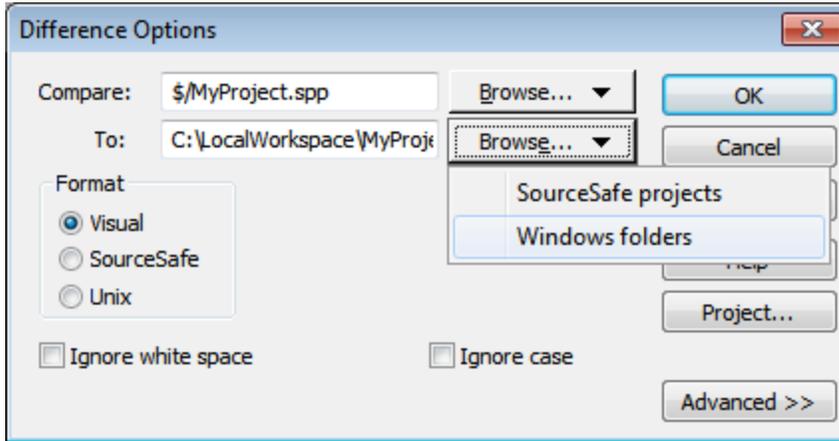
Este cuadro de diálogo permite compartir y obtener versiones determinadas del archivo en cuestión. Al hacer doble clic en un archivo de la lista se abre el cuadro de diálogo "History details" (Historial detallado). Estos son los botones de este cuadro de diálogo:

- **Close:** cierra el cuadro de diálogo.
- **View:** abre un cuadro de diálogo donde puede seleccionar el tipo de visor de archivos.
- **Details:** abre un cuadro de diálogo donde puede ver las [propiedades](#) <sup>207</sup> del archivo activo actual.
- **Get:** recupera una de las versiones previas del archivo y la coloca en el directorio de trabajo.
- **Check Out:** desprotege una versión previa del archivo.
- **Diff:** abre el cuadro de diálogo [Difference options](#) <sup>211</sup>, donde puede definir las opciones de comparación de las dos versiones del archivo. Marque dos versiones del archivo en esta ventana y después haga clic en **Diff** para ver las diferencias.
- **Pin:** ancla o desancla una versión del archivo, para que pueda definir la versión de archivo que desea usar al comparar dos archivos.
- **Rollback:** revierte el archivo a la versión seleccionada.
- **Report:** genera un informe de historial que se puede enviar a una impresora, a un archivo o al portapapeles.
- **Help:** abre la ayuda en línea del complemento de control de código fuente.

### 14.3.6.11 Mostrar diferencias

El comando **Mostrar diferencias** está disponible si selecciona un archivo de la ventana Proyecto. Para seleccionar el archivo de proyecto (.spp) haga clic en el título del proyecto en la ventana Proyecto. El comando **Mostrar diferencias** inicia la herramienta de comparación del sistema de control de código fuente. Así puede ver las diferencias directamente en la aplicación de Altova.

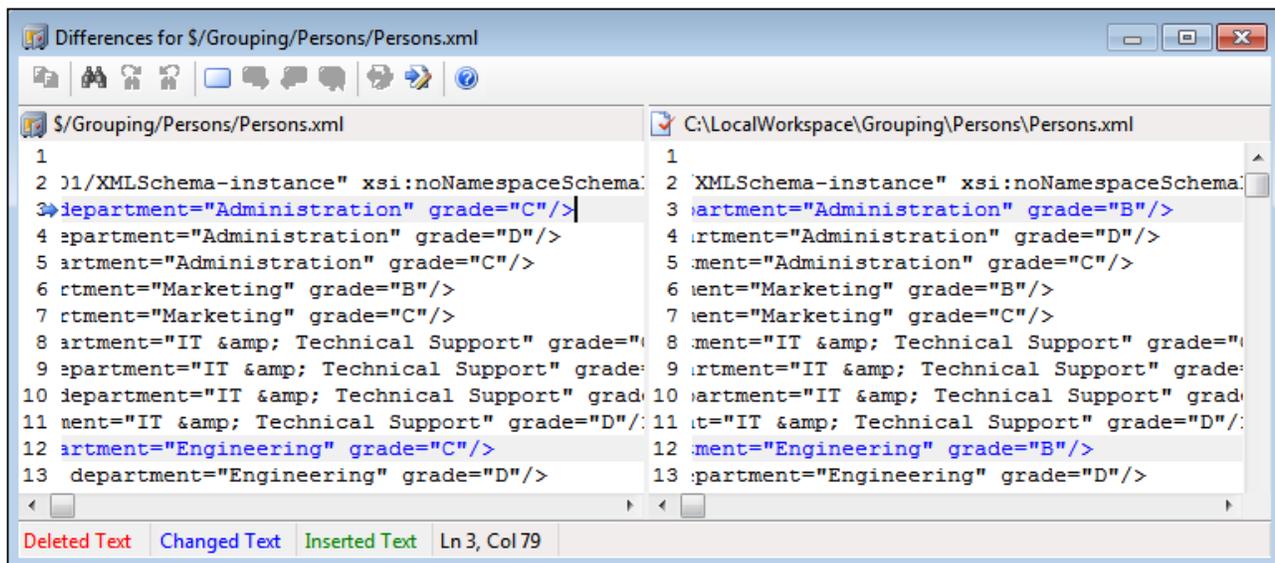
La imagen siguiente muestra la herramienta de comparación del sistema de control Visual SourceSafe.



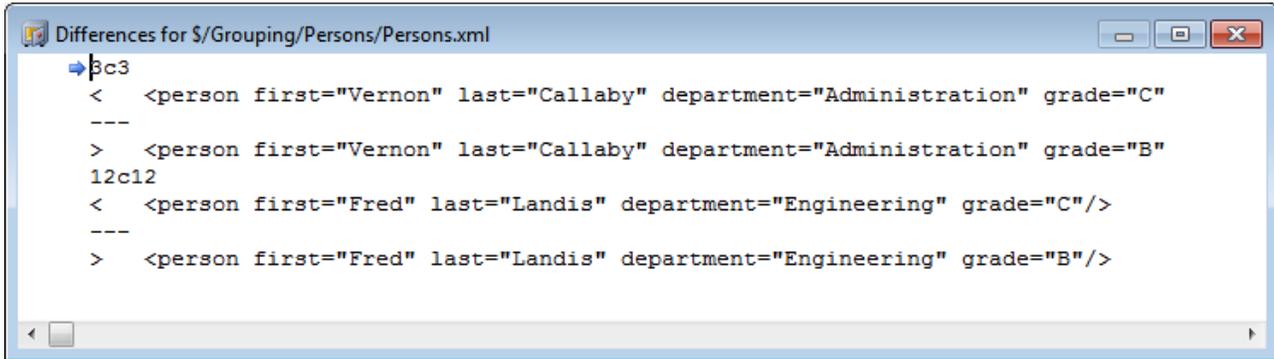
En los campos de texto *Compare* y *To* aparecen por defecto la versión del repositorio y la versión local respectivamente. Para buscar otros archivos:

1. Seleccione la opción *SourceSafe projects* en la lista desplegable del cuadro combinado **Examinar** (para buscar archivos del repositorio) o la opción *Windows folders* (para buscar carpetas locales).
2. Busque los archivos y selecciónelos.

Seleccione las opciones que desea usar y haga clic en **OK** para iniciar la comparación. Los resultados aparecen en otra ventana. Las imágenes que aparecen a continuación muestran los resultados de una misma comparación en dos formatos diferentes.



La primera imagen muestra los resultados de la comparación de Visual SourceSafe en formato visual (ver la opción *Format* > *Visual* del cuadro de diálogo de la primera imagen). La imagen que aparece a continuación muestra el resultado en formato Unix. En ambos casos, se encontraron dos diferencias.

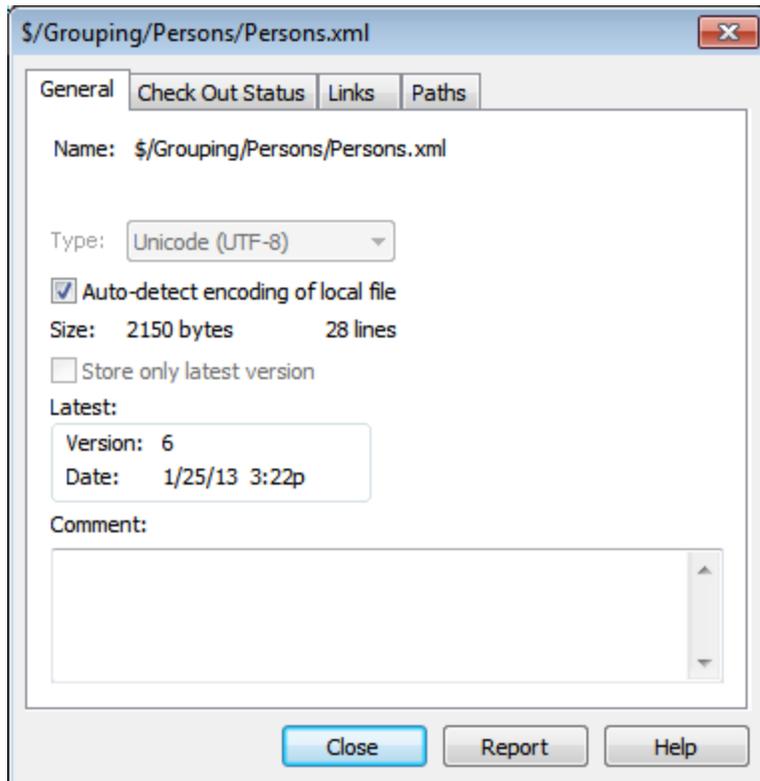


```
Differences for $/Grouping/Persons/Persons.xml
↳ Bc3
< <person first="Vernon" last="Callaby" department="Administration" grade="C"
---
> <person first="Vernon" last="Callaby" department="Administration" grade="B"
12c12
< <person first="Fred" last="Landis" department="Engineering" grade="C"/>
---
> <person first="Fred" last="Landis" department="Engineering" grade="B"/>
```

Para más información consulte la documentación de su sistema de control de código fuente.

### 14.3.6.12 Mostrar propiedades

El comando **Mostrar propiedades** muestra las propiedades del archivo que está seleccionado (*imagen siguiente*). Las propiedades que aparecen dependen del sistema de control de código fuente utilizado. La imagen siguiente corresponde al sistema Visual SourceSafe.



Recuerde que este comando se habilita solamente si selecciona un solo archivo.

Para más información consulte la documentación de su sistema de control de código fuente.

### 14.3.6.13 Actualizar estado

El comando **Actualizar estado** actualiza el estado de todos los archivos del proyecto independientemente de cuál sea su estado actual.

### 14.3.6.14 Administrador del control de código fuente

El comando **Administrador del control de código fuente** inicia el software de control de código fuente en su interfaz nativa.

### 14.3.6.15 Cambiar control de código fuente

El enlace actual es lo que usa el proyecto de aplicación activo para conectarse a la BD del control de código fuente. Por tanto, en enlace actual debe ser correcto. Esto significa que el archivo de proyecto de la aplicación (archivo .spp) debe estar en la carpeta local y la carpeta enlazada del repositorio debe ser la base de datos donde se almacenan los archivos de este proyecto. Por lo general, la carpeta enlazada y sus subcarpetas se corresponden con la carpeta de la copia de trabajo local y sus subcarpetas.

En el cuadro de diálogo "Cambiar control de código fuente" (*imagen siguiente*) puede cambiar el sistema de control (*Proveedor SCC*), la carpeta local (*Ruta de acceso local*) y el enlace al repositorio (*Nombre del servidor* y *Enlace del servidor*).

Las opciones solamente se pueden cambiar después de desenlazar el enlace actual. Para ello, pulse el botón **Desenlazar**. Ahora puede editar las demás opciones.



1. Con el botón **Examinar** puede buscar la carpeta local y con el botón **Seleccionar** puede seleccionar uno de los sistemas de control de código fuente instalados.

2. Después puede enlazar la carpeta local a una BD del repositorio. Para ello pulse el botón **Enlazar**. Aparece el cuadro de diálogo de conexión de sus sistema de control.
3. Si escribió un *Id. de inicio de sesión*, este identificador se pasa al sistema de control. De lo contrario, deberá escribir sus datos de inicio de sesión en el diálogo de conexión.
4. Seleccione la BD del repositorio a la que desea enlazar esta carpeta local.
5. Cuando termine haga clic en **Aceptar**.

### 14.3.7 Agregar archivos al proyecto



El comando **Proyecto | Agregar archivos al proyecto** agrega archivos al proyecto actual. Use este comando para agregar archivos a cualquier carpeta del proyecto. Puede seleccionar un solo archivo o un grupo de archivos en el cuadro de diálogo Abrir. Si agrega archivos al proyecto (en lugar de agregarlos a una carpeta del proyecto), estos se distribuyen en las carpetas del proyecto dependiendo de las extensiones de tipo de archivo definidas en el cuadro de diálogo [Propiedades del proyecto](#)<sup>226</sup>.

### 14.3.8 Agregar recurso global al proyecto

El comando **Proyecto | Agregar recurso global al proyecto** abre el cuadro de diálogo "Agregar recurso global", donde puede seleccionar un recurso global de tipo archivo o carpeta y agregarlo al proyecto. Si selecciona un recurso global de tipo archivo, el archivo se añade a la carpeta correspondiente según las extensiones de tipo de archivo definidas en el cuadro de diálogo [Propiedades del proyecto](#)<sup>226</sup>. Si selecciona un recurso global de tipo carpeta, la carpeta se abre en un cuadro de diálogo "Abrir" y debe seleccionar un archivo. El archivo seleccionado en este cuadro de diálogo se añade a la carpeta de proyecto correspondiente según las extensiones de tipo de archivo definidas en el cuadro de diálogo [Propiedades del proyecto](#)<sup>226</sup>. Para más información, consulte la sección del *Manual del usuario* dedicada a los recursos globales.

### 14.3.9 Agregar URL al proyecto



El comando **Proyecto | Agregar URL al proyecto** agrega una URL al proyecto actual. Las direcciones URL de un proyecto incluyen su objeto de destino en el proyecto. Si ejecuta una operación por lotes en una URL o en una carpeta que contiene una URL, Authentic Desktop recupera el documento de la URL y ejecuta en él la operación solicitada.

### 14.3.10 Agregar archivo activo al proyecto



El comando **Proyecto | Agregar archivo activo al proyecto** añade el archivo activo al proyecto actual. Si acaba de abrir un archivo desde el disco duro o desde una URL, puede agregar el archivo al proyecto con este comando.

### 14.3.11 Agregar archivo activo y relacionados al proyecto



El comando **Proyecto | Agregar archivo activo y relacionados al proyecto** añade al proyecto el archivo XML que está activo y todos los archivos relacionados con él. Si está trabajando con un documento XML que está basado en una DTD o en un esquema, este comando no sólo añade al proyecto el documento XML activo, sino también todos los archivos relacionados con él (por ejemplo, la DTD y todas las entidades externas analizadas a las que hace referencia la DTD).

**Nota importante:** no se consideran archivos relacionados los archivos a los que se hace referencia mediante instrucciones de procesamiento (por ejemplo, archivos XSLT).

### 14.3.12 Agregar carpeta de proyecto al proyecto



El comando **Proyecto | Agregar carpeta de proyecto al proyecto** añade una carpeta nueva al proyecto actual. Use este comando para agregar una carpeta nueva al proyecto actual o una subcarpeta a una carpeta de proyecto. Este comando también está disponible en el menú contextual que aparece al hacer clic con el botón derecho en una carpeta en la ventana Proyecto.

**Nota:** las carpetas de proyecto se pueden arrastrar y colocar en otra carpeta de proyecto o en otra posición del proyecto. Además, puede arrastrar carpetas desde el explorador de Windows hasta cualquier carpeta de la ventana Proyecto.

**Nota:** las carpetas de proyecto son verdes, mientras que las [carpetas externas](#)<sup>217</sup> son amarillas.

### 14.3.13 Agregar carpeta externa al proyecto

El comando **Proyecto | Agregar carpeta externa al proyecto** añade una carpeta externa nueva al proyecto actual. Use este comando para agregar una carpeta de red o local al proyecto actual. Este comando también está disponible en el menú contextual que aparece al hacer clic con el botón derecho en una carpeta en la ventana Proyecto.

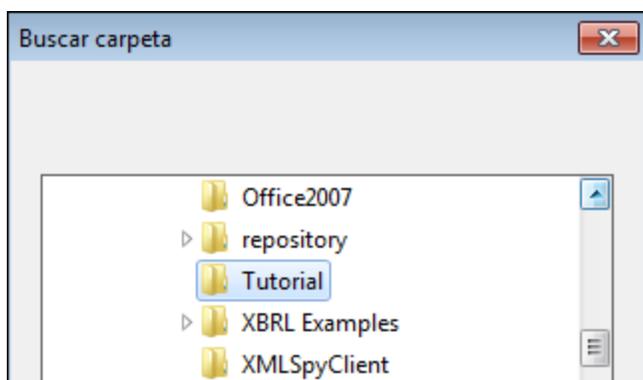
**Nota:** las carpetas externas son amarillas, mientras que las [carpetas de proyecto](#)<sup>216</sup> son verdes.

**Nota:** los archivos de las carpetas externas no se pueden poner bajo control de código fuente.

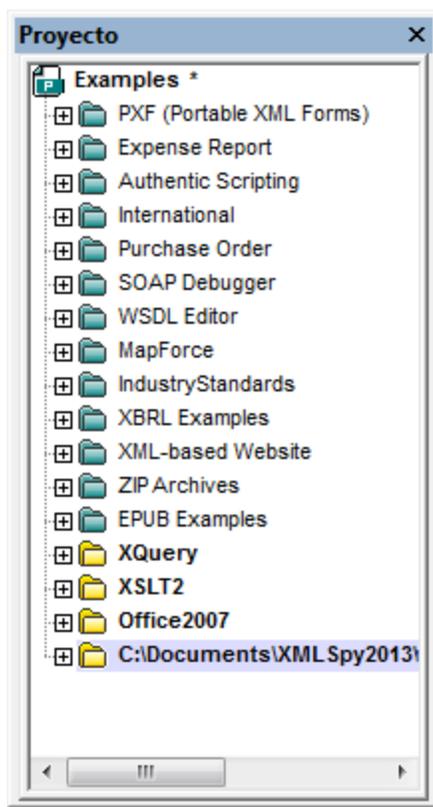
#### Agregar carpetas externas al proyecto

Siga estos pasos para agregar una carpeta externa al proyecto:

1. Seleccione la opción de menú **Proyecto | Agregar carpeta externa al proyecto**
2. En el cuadro de diálogo "Buscar carpeta" seleccione la carpeta que desea agregar y haga clic en **Aceptar**. La carpeta seleccionada aparece en la ventana Proyecto.



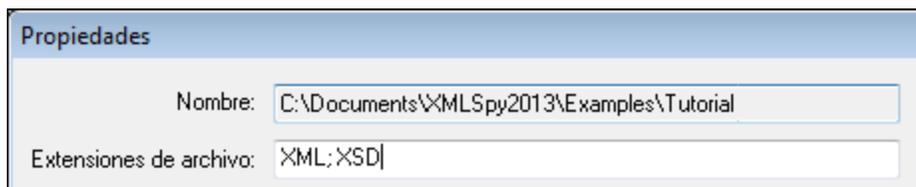
3. Haga clic en el icono + para ver el contenido de la carpeta.



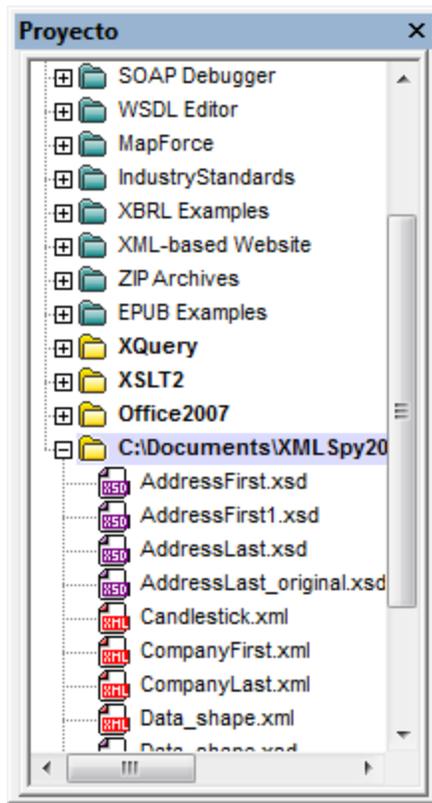
### Filtrar el contenido de las carpetas

Siga estos pasos para filtrar el contenido de la carpeta:

1. Haga clic con el botón derecho en la carpeta local y seleccione la opción de menú **Propiedades**. Aparece el cuadro de diálogo Propiedades.



2. En el campo *Extensiones de archivo*: escriba las extensiones de archivo de los tipos de archivo que desea ver. Para definir varios tipos de archivo, separe las extensiones con un **punto y coma** (*XML y XSD en el ejemplo de la imagen anterior*).
3. Para confirmar haga clic en **Aceptar**.



Ahora la ventana Proyecto muestra solamente los archivos XML y XSD de la carpeta externa seleccionada.

## Validar carpetas externas

Siga estos pasos para validar y comprobar si los documentos de sus carpetas externas tienen un formato XML correcto:

1. Seleccione qué tipo de archivos desea ver y validar en la carpeta externa (siguiendo los pasos descritos más arriba).
2. Seleccione la carpeta y haga clic en el comando de menú **XML | Comprobar formato XML** o **Validar el documento XML (F7 y F8** respectivamente).
3. Corrija el error y ejecute de nuevo el proceso de validación para asegurarse de que se solucionó el problema.

## Actualizar las carpetas de proyecto externas

Es posible que añada o elimine archivos en la carpeta local o de red. Cuando lo haga, será necesario actualizar la vista de la carpeta externa. Para ello, haga clic con el botón derecho en la carpeta externa y seleccione la opción **Actualizar** en el menú contextual.

### Eliminar carpetas externas y sus archivos

Para eliminar la carpeta externa de la ventana Proyecto, selecciónela y pulse la tecla **Suprimir**. También puede hacer clic con el botón derecho en la carpeta externa y seleccionar el comando **Eliminar**. Recuerde que la carpeta sólo se elimina en la ventana Proyecto, pero no se elimina ni en el disco duro ni en la red.

Para eliminar un archivo de una carpeta externa es necesario eliminar el archivo en el disco duro o en la red. Para que el cambio se refleje en el proyecto es necesario actualizar el contenido de la carpeta (haga clic en la carpeta con el botón derecho y seleccione **Actualizar**).

**Nota:** las carpetas externas se pueden arrastrar y colocar en otra carpeta de proyecto o en otra posición del proyecto. Además, puede arrastrar carpetas externas desde el explorador de Windows hasta cualquier carpeta de la ventana Proyecto.

### 14.3.14 Agregar carpeta web externa al proyecto

Este comando añade una carpeta web externa al proyecto actual. Este comando también está disponible en el menú contextual que aparece al hacer clic con el botón derecho en una carpeta de la ventana Proyecto. Recuerde que los archivos de carpetas externas no se pueden poner bajo control de código fuente.

#### Agregar carpetas web externas al proyecto

Siga estos pasos para agregar una carpeta web externa al proyecto:

1. Seleccione la opción de menú **Proyecto | Agregar carpeta web externa al proyecto**. Aparece el cuadro de diálogo "Agregar carpeta web al proyecto" (*imagen siguiente*).

Agregar carpeta web al proyecto

Dirección URL del archivo:

Identificación

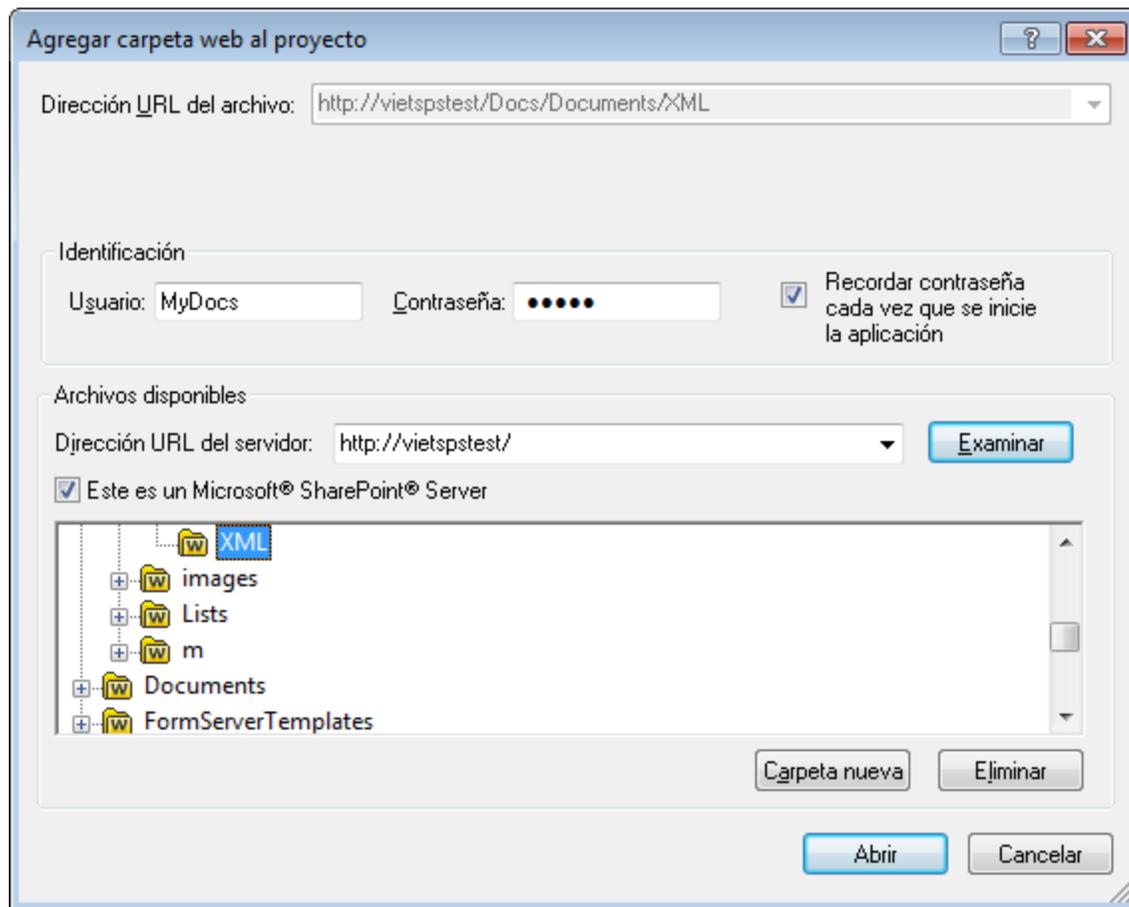
Usuario:  Contraseña:   Recordar contraseña cada vez que se inicie la aplicación

Archivos disponibles

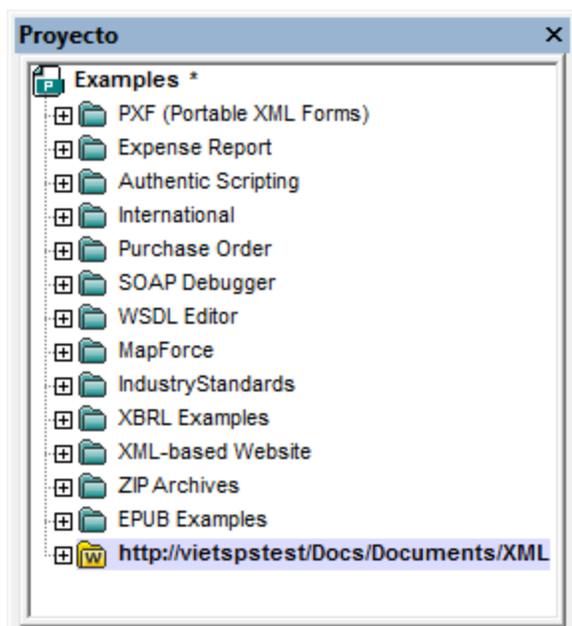
Dirección URL del servidor:

Este es un Microsoft® SharePoint® Server

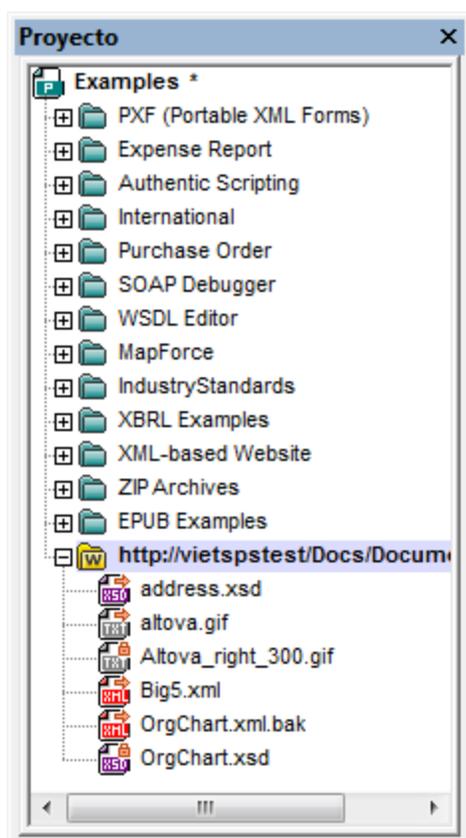
2. Haga clic en el campo *Dirección URL del servidor* y escriba la URL del servidor. Si se trata de un servidor Microsoft® SharePoint® Server, marque la casilla. Consulte el subapartado titulado **Carpetas de los servidores Microsoft® SharePoint® Server** de esta página para más información sobre cómo trabajar con archivos de este tipo de servidor.
3. Si el servidor está protegido con contraseña, escriba su Id. de usuario y contraseña en los campos correspondientes.
4. Haga clic en **Examinar** para conectarse al servidor y ver las carpetas que están disponibles.



5. Haga clic en la carpeta que desea agregar al proyecto. El botón **Abrir** no se habilita hasta que no seleccione una carpeta. La URL de la carpeta aparece ahora en el campo *Dirección URL del archivo*.
6. Haga clic en **Abrir** para agregar la carpeta al proyecto.



7. Haga clic en el icono + para ver el contenido de la carpeta.



## Filtrar el contenido de las carpetas

Para filtrar el contenido de una carpeta, haga clic con el botón derecho en la carpeta y seleccione **Propiedades** en el menú contextual. Aparece el cuadro de diálogo Propiedades. Haga clic en el campo *Extensiones de archivo* y escriba los tipos de archivo que desea ver en la carpeta (por ejemplo, archivos XML y XSD). Separe cada tipo de archivo con un punto y coma (por ejemplo, `xml; xsd; sps`). A partir de ahora la ventana Proyecto muestra solamente los tipos de archivo que tienen las extensiones indicadas.

## Validar carpetas web externas

Para validar los archivos de una carpeta y comprobar si tienen un formato XML correcto, seleccione la carpeta y después haga clic en el icono **Comprobar formato XML** o **Validar el documento XML (F7 y F8)** respectivamente). La operación se ejecuta en todos los archivos que están visibles en la carpeta. Si un archivo no es válido o tiene un formato XML incorrecto, entonces el archivo se abre en la ventana principal para que pueda editarlo. Corrija el error y reinicie el proceso para volver a revisar el resto de la carpeta. Recuerde que también puede seleccionar varios archivos de la carpeta si mantiene pulsada la tecla **Ctrl** mientras hace clic en los archivos uno por uno. En este caso, al pulsar **F7** o **F8**, se revisan únicamente los archivos seleccionados.

## Actualizar el contenido de las carpetas de proyecto

Es posible que se añadan o eliminen archivos en la carpeta web. Cuando esto ocurra, será necesario actualizar la vista de la carpeta externa. Para ello, haga clic con el botón derecho en la carpeta externa y seleccione la opción **Actualizar** en el menú contextual.

## Eliminar carpetas y archivos

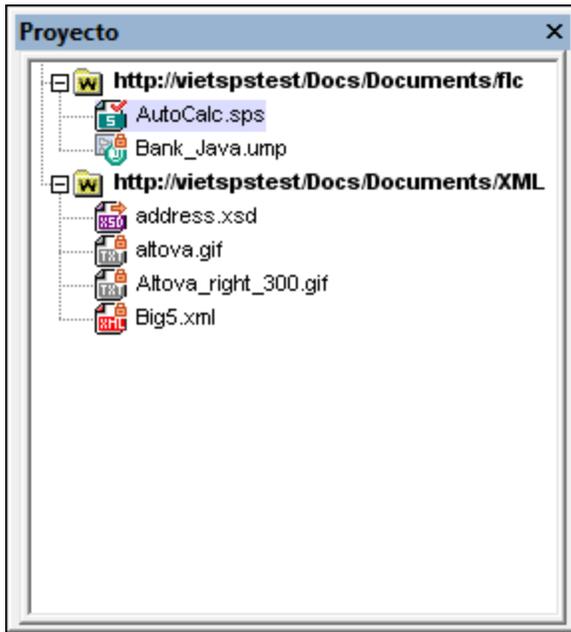
Puesto que lo que se añadió al proyecto es la carpeta web, solamente se puede eliminar del proyecto la carpeta web y no los archivos que están dentro de la carpeta. Para eliminar una carpeta web del proyecto, tiene dos opciones: (i) puede hacer clic con el botón derecho en la carpeta y seleccionar **Eliminar** o (ii) puede seleccionar la carpeta y pulsar la tecla **Suprimir**. Recuerde que la carpeta se elimina solamente en la ventana Proyecto, no en el servidor web.

**Nota:** si en la ventana Proyecto hace clic con el botón derecho en un archivo de la carpeta web y pulsa la tecla **Suprimir**, el archivo no desaparece de la ventana Proyecto. Para que desaparezca es necesario eliminar el archivo en el servidor y después actualizar el contenido de la carpeta externa.

## Carpetas de los servidores Microsoft® SharePoint® Server

Cuando añada al proyecto una carpeta de un servidor Microsoft® SharePoint® Server, los archivos de la carpeta se pueden proteger y desproteger con los comandos del menú contextual de la lista de archivos de la ventana Proyecto (*imagen siguiente*). Para usar estos comandos haga clic con el botón derecho en el archivo con el que desea trabajar y seleccione el comando pertinente (**Desproteger**, **Proteger** o **Deshacer desprotección**).

El Id. de usuario y la contraseña se pueden guardar en las [propiedades de las carpetas del proyecto](#) <sup>226</sup> para que no sea necesario pasar por el proceso de verificación cada vez que acceda al servidor.



En la ventana Proyecto (*imagen anterior*), los iconos de archivo tienen símbolos que indican el estado de los archivos en el servidor:

	Protegido. Se puede desproteger.
	Desprotegido por otro usuario. No se puede desproteger.
	Desprotegido localmente. Se puede editar y proteger.

Recuerde que:

- Después de desproteger un archivo, puede editarlo en su aplicación de Altova y guardarlo con el comando **Archivo | Guardar (Ctrl+S)**.
- Puede proteger el archivo editado con el menú contextual de la ventana Proyecto (*imagen anterior*) o con el menú contextual que aparece al hacer clic con el botón derecho en la pestaña de archivo de la ventana principal de la aplicación (*imagen siguiente*).

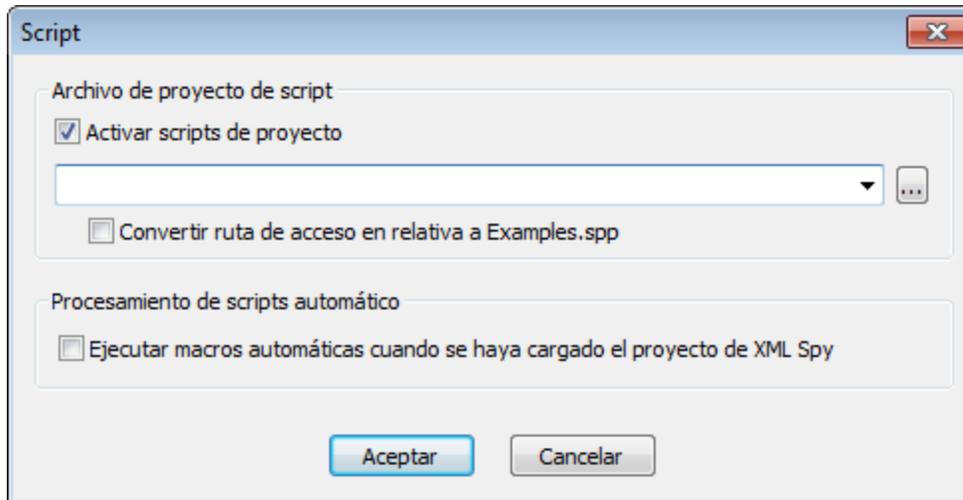


- Cuando otro usuario desprotege un archivo, este archivo no se puede desproteger.
- Cuando desproteja un archivo localmente, puede deshacer la desprotección con el comando **Deshacer desprotección** del menú contextual. Como resultado se devuelve el archivo al servidor sin ningún cambio.
- Si desprotege un archivo en una aplicación de Altova, no lo puede desproteger en otra aplicación de Altova porque se entiende que está desprotegido. A partir de ese momento los comandos disponibles en las aplicaciones de Altova compatibles con Microsoft® SharePoint® Server serían **Proteger** y **Deshacer desprotección**.

### 14.3.15 Configuración de script

Siga estos pasos para asignar un script de proyecto a un proyecto de Authentic Desktop:

1. En la IGU de Authentic Desktop, abra el proyecto de aplicación correspondiente.
2. Seleccione el comando de menú **Proyecto | Configuración de script**. Se abre el cuadro de diálogo Script (*imagen siguiente*).



3. Marque la casilla Activar scripts de proyecto y seleccione el proyecto correspondiente (archivo `.asprj`). Si quiere ejecutar macros automáticas cuando se haya cargado el proyecto de Authentic Desktop, marque la casilla de la parte inferior del cuadro de diálogo.
4. Haga clic en **Aceptar** para finalizar.

**Nota:** para que el proyecto de script deje de estar asignado a un proyecto de Authentic Desktop desactive la casilla Activar scripts de proyecto.

### 14.3.16 Propiedades



El comando **Proyecto | Propiedades** abre el cuadro de diálogo "Propiedades" del proyecto activo (*imagen siguiente*). Si hace clic con el botón derecho en una carpeta de la ventana Proyecto (en lugar de en la carpeta del proyecto), entonces se abre el cuadro de diálogo "Propiedades" de esa carpeta. A continuación describimos las opciones de este cuadro de diálogo.

**Nota:** si el archivo de proyecto está bajo control de código fuente, entonces aparece un mensaje preguntando si desea desproteger el archivo de proyecto (archivo `.spp`). Haga clic en **Aceptar** si desea editar las propiedades y guardarlas.

Propiedades

Nombre: Facturas-EU

Extensiones de archivo:

Validación

Validar con:

Transformación XSL de archivos XML

Utilizar este XSL: C:\Facturas\informes.xslt

Transformación XSL:FO de archivos XML

Utilizar este XSL: C:\Facturas\informesFO.xslt

Transformación XQuery/XQuery Update de archivos XML

Utilizar este XQuery:

XML de entrada para la transformación XSL/XQuery/XQuery Update

Utilizar este XML:

Archivos de salida para la transformación XSL/XQuery/XQuery Update

Guardar en la carpeta: C:\Facturas\Informes

Extensión de archivo: .html

Ejecución XULE

Utilizar este XBRL: C:\Facturas\XBRL\FacturasEU.xbrl

Vista Authentic

Utilizar configuración:

Archivos conforme a JSON

Tratar como: Detectar automáticamente

## Opciones del cuadro de diálogo

### Extensiones de archivo

Este campo se habilita cuando se trata del cuadro de diálogo "Propiedades" de una carpeta y no del proyecto. Cuando se añade un archivo al proyecto, se añadirá a la carpeta donde se definió su extensión de archivo. Por ejemplo, imagine que añade el archivo `MiInforme.xml` al proyecto. Si definió la extensión de archivo `.xml` en la carpeta `Facturas-EU`, entonces el archivo `MiInforme.xml` se añadirá a esa carpeta. Si desea añadir archivos XML en varias carpetas distintas, deberá hacerlo a mano carpeta por carpeta.

### Usuario y contraseña para carpetas externas

Si se trata de una carpeta externa (incluidas carpetas web externas), en este cuadro de diálogo también puede guardar el id. de usuario y la contraseña necesarios para acceder al servidor.

### Validación

Aquí puede indicar qué archivo DTD, esquema XML o esquema JSON debe utilizarse para [validar](#)<sup>231</sup> los archivos que está en la carpeta (o en el proyecto entero).

### Transformación XSL de archivos XML

Esta es la hoja de estilos XSLT que se debe utilizar para ejecutar la [transformación XSL](#)<sup>232</sup> de los archivos XML de la carpeta (o del proyecto).

### Transformación XSL-FO de archivos XML

Esta es la hoja de estilos XSLT que se debe utilizar para transformar en XSL-FO los archivos XML de la carpeta (o del proyecto).

### Transformación XQuery/XQuery Update de archivos XML

El archivo XQuery o XQuery Update que se debe utilizar para ejecutar XQuery o XQuery Update en archivos XML de la carpeta.

### XML de entrada para la transformación XSL/XQuery/XQuery Update

El archivo XML que se debe utilizar como entrada para las transformaciones XSLT o para ejecutar XQuery/XQuery Update, con los archivos de la carpeta correspondientes en cada caso.

### Archivos de salida para la transformación XSL/XQuery/XQuery Update

El directorio de destino de las transformaciones y, opcionalmente, la extensión de archivo del documento resultante.

### Ejecución XULE

El archivo de instancia XBRL que se procesa con el documento XULE que esté activo en la ventana de la aplicación XMLSpy.

### Vista Authentic

En este campo puede especificar el archivo SPS que se debe usar para la representación de los archivos XML de la carpeta en la Authentic View. Recuerde que los archivos XML deben ser válidos según el esquema utilizado para el archivo SPS.

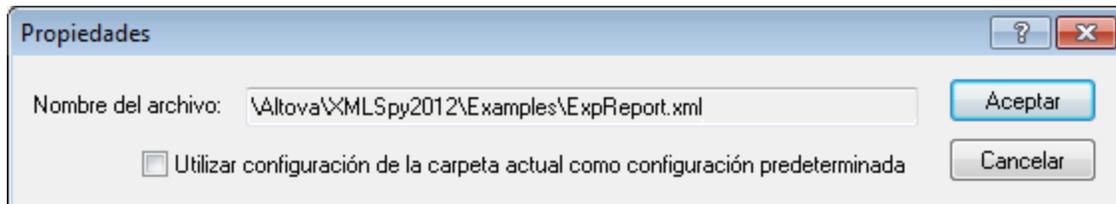
### Archivos conforme a JSON

En esta propiedad se indica si una carpeta de proyecto contiene archivos de esquema JSON o archivos de instancia JSON, lo que puede ser útil para identificar archivos de esquema JSON si estos no están claramente identificados como tales con la palabra clave `$schema` y los archivos hacen referencia unos a otros. Puede elegir entre *Instancia JSON*, *Esquema JSON* o *Detectar automáticamente*. La opción predeterminada es *Detectar automáticamente* y hace que XMLSpy compruebe la estructura y el contenido de los archivos JSON para determinar de qué tipo son.

## Notas sobre las propiedades de proyecto

Reglas de prioridad:

- Cuando se llevan a cabo validaciones o transformaciones XSLT/XQuery desde el menú contextual de una carpeta de proyecto, entonces los archivos de validación y transformación especificados en este cuadro de diálogo tienen prioridad sobre los archivos asignados al archivo XML. Además, las opciones definidas para las diferentes carpetas del proyecto tienen prioridad sobre las opciones definidas para carpetas de niveles superiores.
- Si un archivo está presente en varias carpetas del proyecto y se le asignaron archivos de validación o transformación diferentes en estas carpetas, entonces puede especificar qué asignación se utiliza cuando el archivo se procesa fuera del proyecto. Esto se hace de la siguiente manera: busque el archivo de la carpeta del proyecto cuya asignación desea utilizar. Haga clic con el botón derecho en el archivo en la carpeta correspondiente y seleccione **Propiedades** en el menú. Aparece un cuadro de diálogo (*imagen siguiente*). Aquí debe marcar la casilla *Utilizar configuración de la carpeta actual como configuración predeterminada* (la carpeta actual es la carpeta del proyecto donde está el archivo). Si esta casilla está deshabilitada, significa que la configuración de la carpeta actual ya está definida como configuración predeterminada. Recuerde que, si el archivo tiene una asignación local (es decir, la asignación está dentro del archivo propiamente dicho), entonces se utiliza esta asignación local y se ignora la configuración de la carpeta predeterminada.



### 14.3.17 Proyectos usados recientemente

Este comando muestra el nombre de archivo y la ruta de acceso de los últimos nueve proyectos utilizados, para que pueda abrirlos rápidamente.

Recuerde que Authentic Desktop abre automáticamente el [último proyecto utilizado](#)<sup>279</sup> nada más iniciarse la aplicación (**Herramientas | Opciones | Archivo, Proyecto | Abrir el último proyecto al iniciarse el programa**).

## 14.4 Menú XML

Los comandos del menú **XML** sirven para trabajar con documentos XML. Entre las tareas XML más comunes se encuentran la [comprobación de formato XML](#)<sup>230</sup> y la [validación](#)<sup>231</sup> de documentos XML. Los comandos para estas operaciones se encuentran en este menú.

### 14.4.1 Comprobar formato XML

**F7**

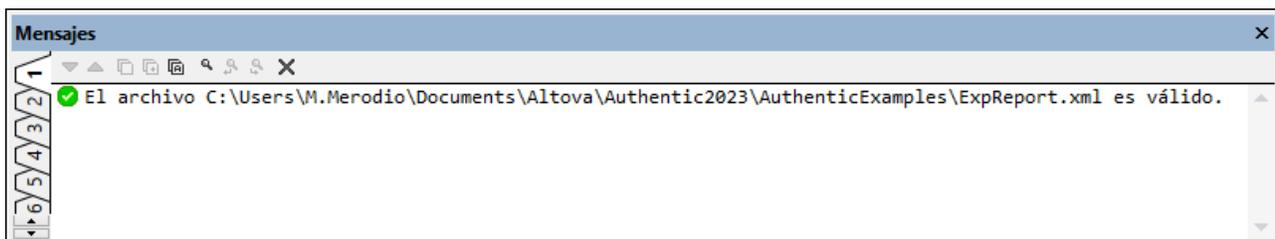
El comando **XML | Comprobar formato XML (F7)** revisa el documento activo y comprueba si el formato es correcto según las definiciones de la especificación XML 1.0. Es obligatorio que el documento XML tenga un formato XML correcto. Authentic Desktop hace una comprobación de formato XML cada vez que se abre o guarda un documento o cuando se cambia de la vista Texto a otra vista. Si se detecta un error durante la comprobación de formato XML, en la ventana Mensajes aparece un mensaje de error explicando el problema y el documento XML aparece en la vista Texto. Puede arreglar el problema en la vista Texto y volver después a la vista Authentic.

### 14.4.2 Validar el documento XML

**F8**

El comando **XML | Validar el documento XML (F8)** sirve para validar documentos XML con esquemas XML, DTD y otros esquemas. Si lo desea, puede configurar la aplicación para que valide automáticamente los documentos al abrirlos o guardarlos (**Herramientas | Opciones | Archivo**). También puede activar esta opción con el comando [Validar al editar](#)<sup>231</sup> para ir validando conforme edita los datos.

Si el documento es válido, aparece este mensaje en la ventana Mensajes:



**Nota:** si el documento no es válido, entonces aparece un mensaje describiendo el error. Para ir al nodo del documento XML que contiene el error, haga clic en el enlace que incluye el mensaje. UNA vez haya subsanado el error es recomendable que vuelva a ejecutar el comando **Validar (F8)** para asegurarse de que el error está corregido.

#### Validación desde la ventana Proyecto

El comando **Validar el documento XML** también se puede aplicar a un archivo, una carpeta o un grupo de archivos del proyecto activo. En la ventana Proyecto seleccione el archivo, la carpeta o el grupo de archivos que desea validar. Después haga clic en **XML | Validar el documento XML** o pulse **F8**. Los archivos que no

sean válidos se abren en la ventana principal y aparece un mensaje de tipo *El archivo no es válido* en la ventana Mensajes.

### Automatizar la validación de documentos XML con RaptorXML 2023

RaptorXML es una aplicación independiente de Altova para validar XML, transformar XSLT y ejecutar XQuery. Se puede usar desde la línea de comandos, mediante la interfaz de COM, en programas Java y en aplicaciones .NET.

Por tanto, puede automatizar sus tareas de validación con ayuda de RaptorXML. Por ejemplo, puede crear un archivo de proceso por lotes que llame a RaptorXML para ejecutar la validación en un conjunto de documentos y que envíe el resultado de la validación a un archivo de texto. Para más información, consulte la [documentación de RaptorXML](#) en nuestro sitio web.

## 14.4.3 Validar al editar

El comando **Validar al editar** activa y desactiva el modo del mismo nombre, que valida el documento conforme usted teclea. También puede activar o desactivar este modo con el botón **Validar al editar** de la barra de herramientas o con la opción *Al editarlos* de la [sección Archivo del cuadro de diálogo "Opciones"](#) <sup>279</sup>.

## 14.5 Menú XSL/XQuery

El lenguaje de transformación XSL permite especificar cómo se debe convertir un documento XML en otros documentos XML o archivos de texto. Un tipo de documento XML que se genera con un documento XSLT es el documento FO, que más adelante se puede procesar para generar archivos PDF. Authentic Desktop contiene procesadores XSLT integrados (para XSLT 1.0, 2.0 y 3.0) y puede conectarse a un procesador FO del sistema del usuario para transformar archivos XML y generar archivos en diferentes formatos de salida. La ubicación del procesador FO debe indicarse en la sección XSL del cuadro de diálogo "Opciones" ([Herramientas | Opciones](#)<sup>286</sup>) para que Authentic Desktop pueda utilizarlo.

Estos son los comandos que puede encontrar en el menú XSL/XQuery:

- [Transformación XSL](#)<sup>232</sup>
- [Transformación XSL-FO](#)<sup>233</sup>
- [Parámetros de XSL / Variables de XQuery](#)<sup>234</sup>

### 14.5.1 Transformación XSL



F10

El comando **XSL/XQuery | Transformación XSL** transforma un documento XML usando una hoja de estilos XSLT. La transformación se puede realizar usando el motor XSLT integrado de Altova (el motor XSLT 1.0 para las hojas de estilos XSLT 1.0, el motor XSLT 2.0 para las hojas de estilos XSLT 2.0 y el motor XSLT 3.0 para hojas de estilos XSLT 3.0), el módulo MSXML de Microsoft o un procesador XSLT externo. El procesador que se usa junto con este comando se especifica en la [sección](#)<sup>286</sup> [XSL](#)<sup>286</sup> del cuadro de diálogo "Opciones" ([Herramientas | Opciones](#)).

Si el documento XML contiene una referencia a una hoja de estilos XSLT, esta hoja de estilos se usa para la transformación. (Para asignar una hoja de estilos XSLT al documento XML utilice el comando [Asignar XSL](#)<sup>233</sup>). Si no se asignó ninguna hoja de estilos al archivo XML, aparece un mensaje solicitando la hoja de estilos XSLT que se debe usar. También puede seleccionar un archivo mediante un recurso global o una URL (haga clic en el botón [Examinar](#)<sup>178</sup>) o un archivo de las ventanas Authentic Desktop que están abiertas (haga clic en el botón **Ventana**).

#### Automatizar transformaciones XSLT con RaptorXML 2023

RaptorXML es una aplicación independiente de Altova para validar XML, transformar XSLT y ejecutar XQuery. Se puede usar desde la línea de comandos, por una interfaz COM, en programas Java y en aplicaciones .NET. Por tanto, puede automatizar las tareas de transformación XSLT con RaptorXML. Por ejemplo, puede crear un archivo de procesamiento por lotes que llame a RaptorXML para transformar un grupo de documentos. Para más información, consulte la [documentación de RaptorXML](#).

#### Transformaciones con archivos ZIP como destino

Para desviar los documentos de salida de las transformaciones a un archivo ZIP (incluidos los documentos OOXML, como `.docx`, por ejemplo) es necesario especificar el protocolo ZIP en la ruta de acceso del archivo de salida. Por ejemplo:

```
nombreArchivo.zip|zip/nombreArchivo.xxx
```

```
nombreArchivo.docx|zip/nombreArchivo.xxx
```

**Nota:** puede que sea necesario crear la estructura de directorios antes de ejecutar la transformación. Si está generando archivos para un archivo Open Office XML, es necesario comprimir los ficheros del archivo para poder crear el archivo OOXML de nivel superior (por ejemplo, .docx).

## 14.5.2 Transformación XSL-FO



**Ctrl+F10**

FO es un formato XML que describe documentos paginados. Los procesadores FO, como el procesador FOP de Apache XML Project, toman un archivo FO de entrada y generan un archivo PDF de salida. Esto significa que el proceso para generar documentos PDF a partir de documentos XML tiene dos fases:

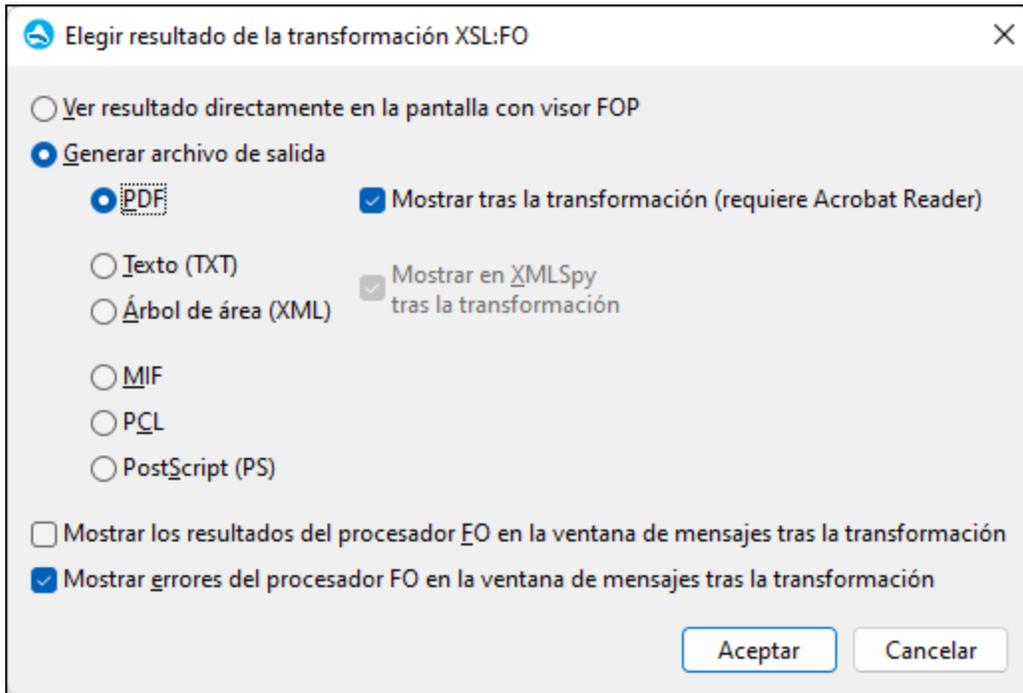
1. Primero el documento XML se transforma en un documento FO usando una hoja de estilos XSLT.
2. Después el procesador FO procesa el documento FO y genera un archivo PDF.

El comando **XSL/XQuery | Transformación XSL-FO** transforma un documento XML o un documento FO en PDF.

- Si ejecuta el comando **Transformación XSL-FO** en un documento XML de origen, entonces se ejecutan los dos pasos descritos anteriormente en secuencia, uno después del otro. Si en el documento XML no hay una referencia a la hoja de estilos XSLT necesaria para transformarlo en documento FO, la aplicación solicita que le asigne una hoja de estilos. Recuerde que también puede seleccionar un archivo mediante un recurso global o una URL (botón **Examinar**<sup>176</sup>) o seleccionar uno de los archivos que están abiertos en las ventanas de Authentic Desktop (botón **Ventana**). La transformación del documento XML en XSL-FO la realiza el procesador XSLT que se indica en la [sección](#)<sup>286</sup> **XSL**<sup>286</sup> del cuadro de diálogo "Opciones" (**Herramientas | Opciones**). El procesador XSLT predeterminado es el procesador XSLT integrado de Authentic Desktop. El documento FO resultante lo procesa el procesador FO que se indica en la [sección](#)<sup>286</sup> **XSL**<sup>286</sup> en el cuadro de diálogo "Opciones" (**Herramientas | Opciones**).
- Si el comando **Transformación XSL-FO** se ejecuta en un documento FO, el documento se procesa con el procesador FO indicado en la [sección](#)<sup>286</sup> **XSL**<sup>286</sup> del cuadro de diálogo "Opciones" (**Herramientas | Opciones**).

### Resultado de la transformación XSL:FO

Al hacer clic en el comando **Transformación XSL-FO** aparece el cuadro de diálogo "Elegir resultado de la transformación XSL:FO" (*imagen siguiente*). Recuerde que si el documento activo es un documento XML que no tiene asignado ninguna hoja de estilos XSLT, XMLSpy primero le pide un archivo XSLT.



El resultado generado por el procesador FO se puede ver directamente en pantalla usando un visor FOP o puede generar un archivo de salida en uno de estos formatos: PDF, texto, árbol de área, MIF, PCL o PostScript. Si lo desea, los mensajes del procesador FO y los mensajes de error del procesador pueden aparecer en la ventana Mensajes. Para activar estas opciones marque las casillas correspondientes situadas al final del cuadro de diálogo.

#### Nota:

- A no ser que desactive la casilla correspondiente en el programa de instalación de Authentic Desktop, el procesador FOP de [Apache XML Project](#) se instalará en la carpeta `c:\ProgramData\Altova\SharedBetweenVersions`. Y si el procesador está instalado, su ruta de acceso aparecerá automáticamente en la [sección 286 XSL 286](#) del cuadro de diálogo "Opciones" (**Herramientas | Opciones**). Si quiere puede cambiar esta ruta de acceso para usar otro procesador FO distinto.
- El comando **Transformación XSL-FO** no solamente se puede usar con el archivo activo en la ventana principal. También puede usarlo con cualquier archivo o carpeta del proyecto activo. Para ello, seleccione los archivos o carpetas en la ventana Proyecto, haga clic con el botón derecho y seleccione el comando **Transformación XSL-FO**. Para la transformación se usa la hoja de estilos XSLT asignada a la carpeta de proyecto seleccionada.

### 14.5.3 Parámetros de XSL / Variables de XQuery

El comando **XSL/XQuery | Parámetros de XSL / Variables de XQuery** abre el cuadro de diálogo "Parámetros de entrada de XSL y variables externas de XQuery" (*imagen siguiente*). Aquí puede escribir el nombre de los parámetros que desea pasar a la hoja de estilos XSLT y las variables externas de XQuery que

desea pasar al documento XQuery, así como sus correspondientes valores. En Authentic Desktop estos parámetros se usan de esta manera:

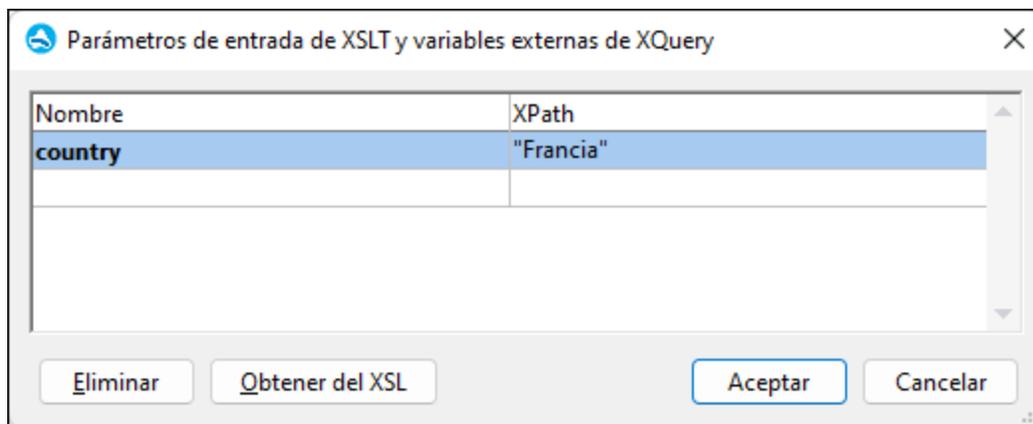
- Cuando se usa el comando **XSL/XQuery | Transformación XSL** para transformar un documento XML, los valores del parámetros guardados en este cuadro de diálogo se pasan al archivo XSLT seleccionado y utilizado para la transformación.
- Cuando se usa el comando **XSL/XQuery | Ejecutar XQuery** para procesar un documento XQuery, los valores de las variables externas de XQuery guardadas en este cuadro de diálogo se pasan al documento XQuery para la ejecución.

**Nota:** los parámetros y las variables que introduzca en este cuadro de diálogo se pasan solamente al motor XSLT integrado de Altova. Por tanto, si usa MSXML o un motor externo, estos parámetros no se pasan al motor de procesamiento.

**Nota:** no es erróneo definir una variable externa de XQuery (o un parámetro de XSLT) en el diálogo "Parámetros de entrada de XSL y variables externas de XQuery" y después no utilizarla en el documento XQuery. Tampoco es erróneo definir un parámetro de XSLT (o una variable externa de XQuery) en este cuadro de diálogo y después no usarlo en una transformación XSLT.

## Cómo usar los parámetros de XSLT

El valor del parámetro puede ser una expresión XPath sin comillas o una cadena de texto delimitada por comillas. Si el documento activo es un archivo XSLT, entonces está habilitado el botón **Obtener desde XSL**. Al hacer clic en este botón se insertan en el cuadro de diálogo los parámetros declarados en el archivo XSLT y sus valores predeterminados. Es decir, este botón le permite incluir rápidamente los parámetros declarados y después cambiar sus valores predeterminados si lo necesita.



Una vez introducidos en este cuadro de diálogo, los pares de parámetros y valores se usan para todas las transformaciones subsiguientes hasta que se eliminen del cuadro de diálogo o se reinicie la aplicación. Los parámetros introducidos en este cuadro de diálogo se definen para toda la aplicación y se pasan al documento XSLT correspondiente en cada transformación que se realice a partir de ese momento a través del entorno IDE. En otras palabras:

- los parámetros no están asociados a ningún tipo de documento en particular
- los parámetros introducidos en el cuadro de diálogo "Parámetros de entrada de XSL y variables externas de XQuery" se borran cuando se cierra Authentic Desktop.

**Nota:** el botón **Obtener desde XSL** se habilita en la vista Authentic solamente si el documento activo es un documento XSLT e inserta parámetros declarados en el documento XSLT activo en el cuadro de diálogo junto con los valores predeterminados de esos parámetros.

## Ejemplo de uso de parámetros de XSLT

En este ejemplo tenemos un documento XML que contiene los nombres de países y sus respectivas capitales:

```
<document>
  <countries>
    <country name="USA" capital="Washington DC"/>
    <country name="UK" capital="London"/>
    <country name="France" capital="Paris"/>
    <country name="Russia" capital="Moscow"/>
    <country name="China" capital="Beijing"/>
  </countries>
</document>
```

El siguiente documento XSLT generará un documento XML que muestre uno de esos países con su capital. Se selecciona un país introduciendo su nombre como valor del parámetro llamado **country** (resaltado en amarillo en el ejemplo siguiente).

```
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:param name="country" select="'USA'"/>
  <xsl:template match="countries">
    <xsl:for-each select="country[@name=$country]">
      <country>
        <name><xsl:value-of select="$country"/></name>
        <capital><xsl:value-of select="@capital"/></capital>
      </country>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

Al ejecutar este documento XSLT en el documento XML mencionado más arriba, el resultado será el siguiente:

```
<country><name>USA</name><capital>Washington DC</capital></country>
```

Si se crea un parámetro llamado **country** en el cuadro de diálogo Parámetros de entrada XSLT/Variables externas XQuery y se le da un valor (*imagen anterior*), este valor pasará al parámetro **country** en la hoja de estilos para la transformación. De esta manera puede pasar distintos valores a diferentes parámetros en tiempo de ejecución.

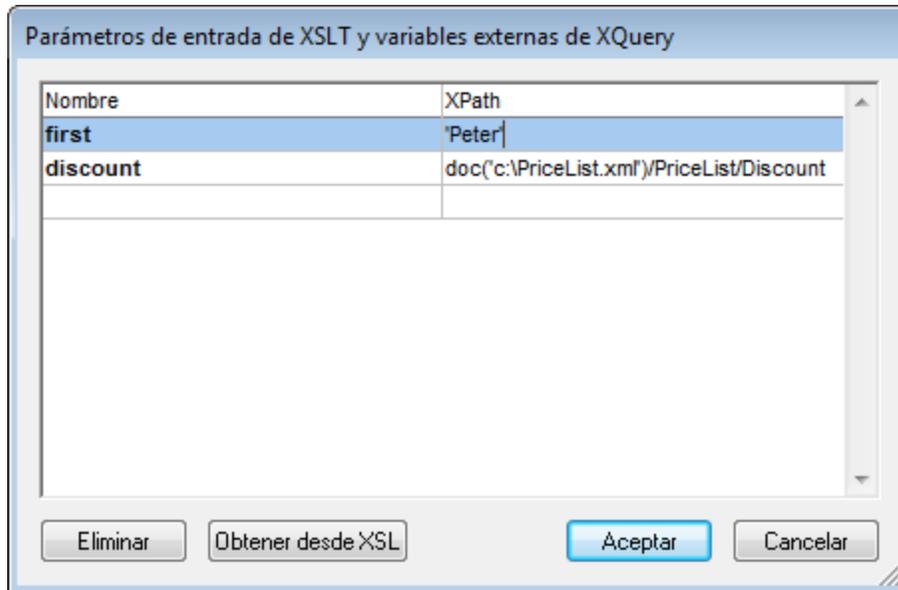
### Nota:

- si usa el comando **XSL/XQuery | Transformación XSL-FO**, entonces los parámetros introducidos en el cuadro de diálogo "Parámetros de entrada de XSLT y variables externas de XQuery" **no se pasan** a la hoja de estilos. Para que estos parámetros se usen en el archivo PDF de salida, antes es necesario transformar el documento XML en FO con el comando **XSL/XQuery | Transformación XSL** y después transformar el FO en PDF con el comando **XSL/XQuery | Transformación XSL-FO**.

- si usa un procesador XSLT que no sea uno de los motores XSLT integrados de Altova, los parámetros que introduzca en el diálogo "Parámetros de entrada de XSLT y variables externas de XQuery" no se pasan al procesador externo.

## Cómo usar las variables externas de XQuery

El valor de las variables externas de XQuery puede ser una expresión XPath sin comillas o una cadena de texto delimitada por comillas. El tipo de datos de la variable externa se especifica en la declaración de variable en el documento XQuery.



**Nota:** una vez introducidas en este cuadro de diálogo, las variables externas de XQuery se utilizan para todas las ejecuciones subsiguientes hasta que se borren del cuadro de diálogo o se reinicie la aplicación. Las variables introducidas en este cuadro de diálogo se definen para toda la aplicación y se pasan al documento XQuery correspondiente en cada ejecución que se realice a partir de ese momento a través del entorno IDE. En otras palabras:

- las variables no están asociadas a ningún tipo de documento en particular
- las variables introducidas en el cuadro de diálogo "Parámetros de entrada de XSL y variables externas de XQuery" se borran cuando se cierra Authentic Desktop.

## Ejemplo de uso de variables externas de XQuery

En este ejemplo, en el documento XQuery se declaró una variable `$first` que se utiliza en la cláusula `return` de la expresión FLWOR:

```
xquery version "1.0";
declare variable $first as xs:string external;
let $last := "Jones"
return concat($first, " ", $last )
```

Este XQuery devuelve `Peter Jones`, si el valor de la variable externa (introducida en el cuadro de diálogo "Parámetros de entrada de XSL y variables externas de XQuery") es `Peter`. Es necesario tener en cuenta que:

- La palabra clave `external` de la declaración de variable del documento XQuery indica que esta variable es una variable externa.
- Definir el tipo estático de la variable es opcional. Si no se indica un tipo de datos para la variable en la declaración de variable, entonces al valor de la variable se le asigna el tipo `xs:untypedAtomic`.
- Si se declara una variable externa en el documento XQuery, pero no se pasa al documento XQuery ninguna variable externa que tenga ese mismo nombre, entonces se emite un mensaje de error.
- Si se declara y se introduce una variable externa en el cuadro de diálogo "Parámetros de entrada de XSL y variables externas de XQuery", entonces se considera que está en el ámbito del documento XQuery que se está ejecutando. Si se declara una variable nueva con el mismo nombre dentro del documento XQuery, la variable nueva reemplaza temporalmente a la variable externa. Por ejemplo, el XQuery que aparece a continuación devuelve `Paul Jones` aunque la variable externa `$first` que está en el ámbito del documento tiene el valor `Peter`.

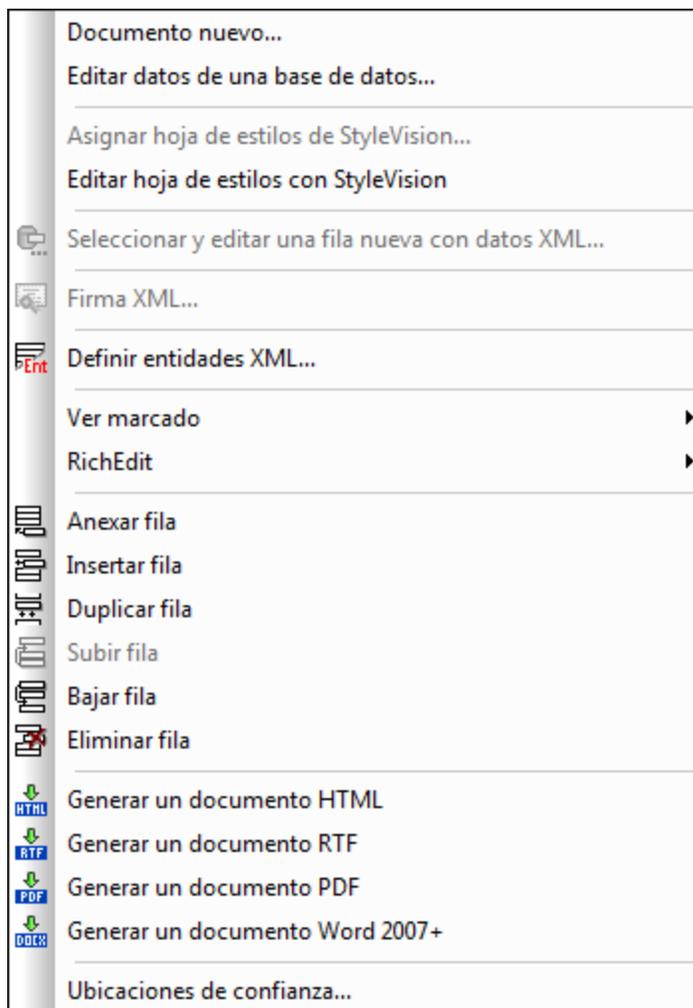
```
xquery version "1.0";
declare variable $first as xs:string external;
let $first := "Paul"
let $last := "Jones"
return concat($first, " ", $last )
```

## 14.6 Menú Authentic

En la Authentic View puede editar documentos XML basados en **archivos SPS creados con Altova StyleVision**. Los archivos SPS son hojas de estilos que permiten mostrar el archivo XML de forma gráfica en la Authentic View. Además de contener información sobre la presentación, el archivo SPS también permite escribir datos en el archivo XML. Estos datos se procesan de forma dinámica usando las funciones de las hojas de estilos XSLT y su resultado se reproduce instantáneamente en la Authentic View.

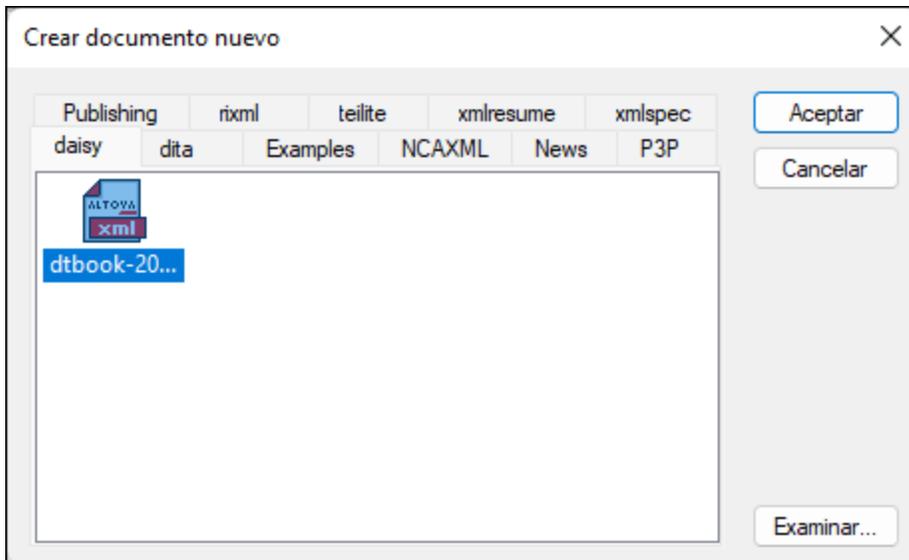
También puede usar archivos SPS para mostrar una vista XML editable de una base de datos. El archivo SPS contiene información para conectarse a la base de datos, mostrar los datos de la base de datos en la Authentic View y volver a escribir datos en la base de datos.

El menú **Authentic** incluye todos los comandos necesarios para editar documentos XML en la Authentic View. Para aprender a usar la Authentic View, consulte [este tutorial](#)<sup>35</sup>.



## 14.6.1 Documento nuevo

Este comando sirve para abrir una plantilla de documento XML nuevo en la Authentic View. La plantilla de documento XML está basada en un archivo SPS (StyleVision Power Stylesheet) y se abre seleccionando el archivo SPS en el cuadro de diálogo "Crear documento nuevo" (*imagen siguiente*). Tras seleccionar el archivo SPS haga clic en **Aceptar**. La plantilla de documento XML definida para el archivo SPS elegido se abre en la Authentic View.



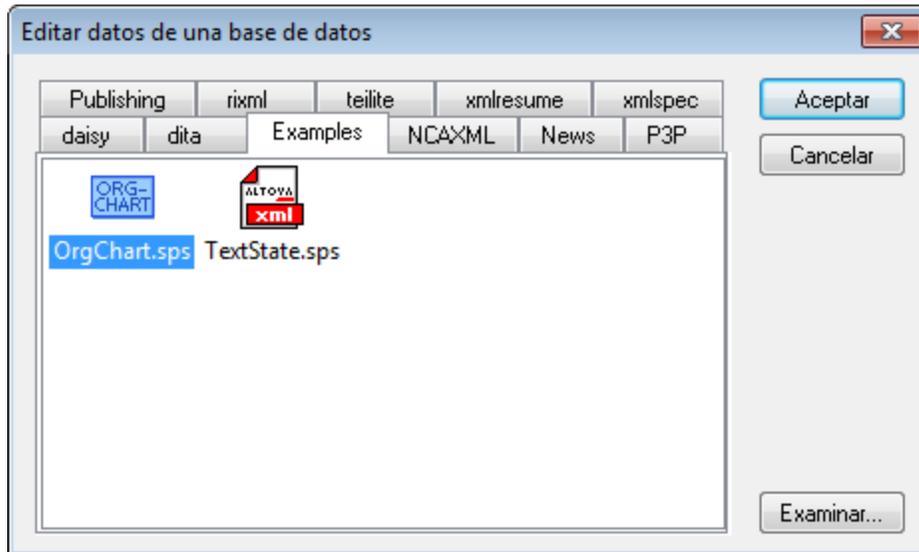
El cuadro de diálogo "Crear documento nuevo" ofrece varias plantillas de documento XML basadas en esquemas y archivos DTD frecuentes. Si lo prefiere, también puede buscar archivos SPS personalizados que tengan asignado un archivo XML. Los archivos SPS se crean con Altova StyleVision, una herramienta con la que puede diseñar plantillas de documento XML a partir de un esquema XML o un archivo DTD. Tras diseñar el archivo SPS en StyleVision puede asignarle un archivo XML como plantilla XML. Los datos de esta plantilla XML aportan los datos de partida de la plantilla de documento que se abre en la Authentic View de Authentic Desktop.

La nueva plantilla de documento XML las propiedades de presentación definidas en el archivo SPS y los datos del archivo XML que seleccionó como plantilla XML. El usuario de la Authentic View puede editar la plantilla de documento XML en una interfaz gráfica WYSIWYG y guardar los cambios en un archivo XML.

## 14.6.2 Editar datos de una base de datos

El comando **Authentic | Editar datos de una base de datos** sirve para abrir una vista editable de una base de datos (BD) en la Authentic View. El archivo SPS contiene la información necesaria para conectarse a la BD y sobre cómo presentar la BD y aceptar los cambios realizados en ella en la Authentic View. El archivo SPS basado en la BD se abre con el comando **Editado datos de una base de datos...** Al abrirlo se establece una conexión a la BD y sus datos se representan (a través de la lente de XML) en la Authentic View.

Al hacer clic en el comando **Editado datos de una base de datos...** aparece un cuadro de diálogo del mismo nombre.



Busque el archivo SPS que desea usar, selecciónelo y haga clic en **Aceptar**. A continuación se establece la conexión con la BD y en la Authentic View se abre una vista editable de la BD. El diseño de la vista de la BD que aparece en la Authentic View se define dentro del archivo SPS.

**Nota:** si con el comando **Editar datos de una base de datos** intenta abrir un archivo SPS que no está basado en una BD o si intenta abrir un archivo SPS basado en una BD que se creó con una versión de StyleVision anterior a StyleVision 2005, entonces recibirá un error.

**Nota:** los archivos SPS se crean con Altova StyleVision.

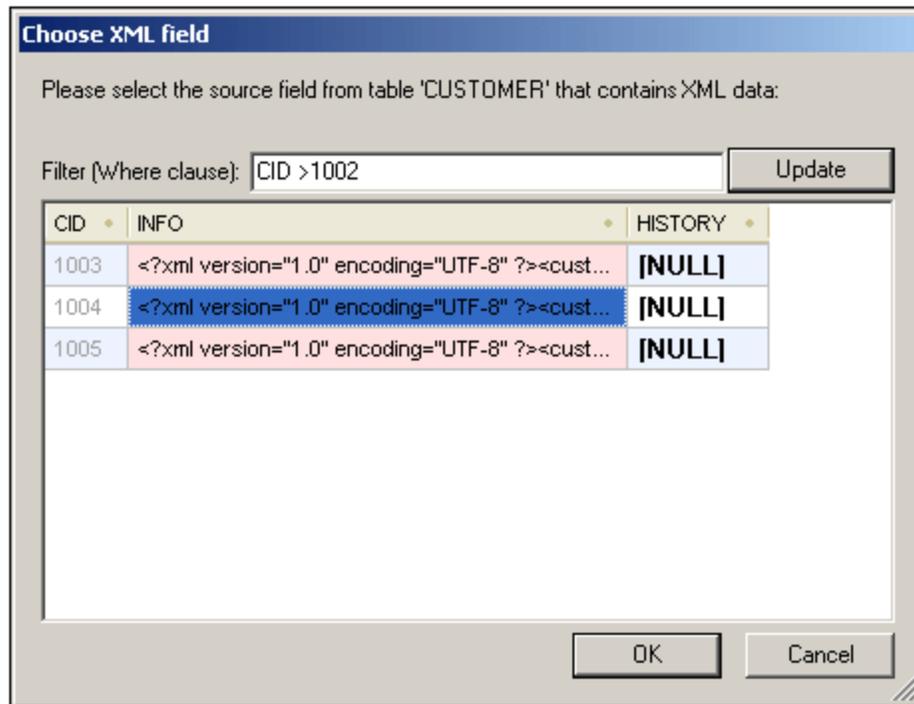
### 14.6.3 Editar una hoja de estilos de StyleVision

El comando **Authentic | Editar una hoja de estilos de StyleVision** solo está disponible en la Vista Authentic. Es decir, solo si se ha asignado al documento una hoja de estilos de StyleVision. Este comando inicia la aplicación StyleVision para que pueda editar el archivo SPS seleccionado.

### 14.6.4 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 Authentic View, 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 Authentic View 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.



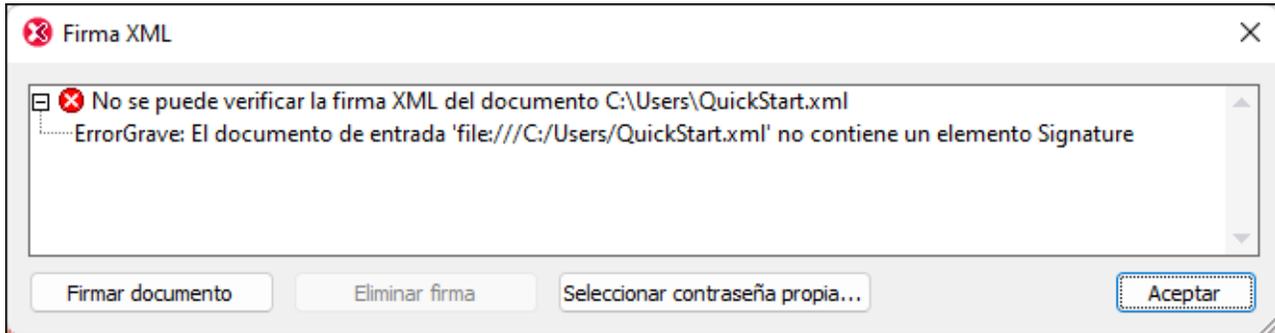
En este cuadro de diálogo puede introducir un filtro para filtrar la tabla. Este filtro debe ser una cláusula `WHERE` SQL (solamente la condición, sin la palabra clave `WHERE`. Por ejemplo `CID>1002`). Haga clic en **Actualizar** para actualizar el cuadro de diálogo. En la imagen anterior puede ver la 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 Authentic View.

### 14.6.5 Firma XML

El comando **Firma XML** está disponible en la vista Authentic cuando el archivo SPS asociado tiene firmas XML. El comando **Firma XML** también está disponible en forma de icono en la barra de herramientas de Authentic.

## Verificación de la firma y contraseña/certificado propios

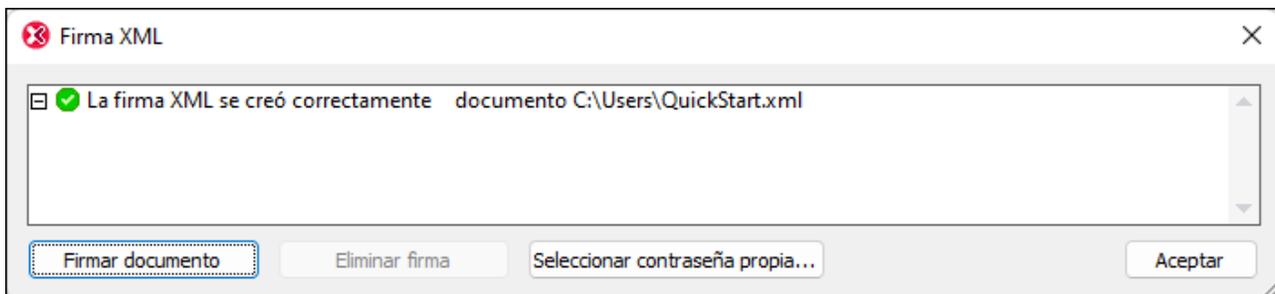
Al hacer clic en el comando **Firma XML** se inicia el proceso de verificación de la firma. Si el documento no tiene una firma, aparece el cuadro de diálogo "Firma XML" con un mensaje a tal efecto (*ver imagen*) y el cuadro de diálogo incluye un botón con el que el usuario de la Authentic View puede firmar el documento.



Si en este cuadro de diálogo aparece el botón **Seleccionar certificado propio** o **Seleccionar contraseña propia**, significa que se definió la posibilidad de seleccionar un certificado o una contraseña propios en la Authentic View. (Cuando configura la firma, el diseñador del archivo SPS decide si se puede elegir un certificado o contraseña). La selección del usuario de la Authentic View se almacena en memoria y solamente es válida para la sesión actual. Si tras seleccionar un certificado o contraseña se cierra el documento o la aplicación, el documento vuelve a su configuración original definida en el SPS.

## Verificación e información de autenticación

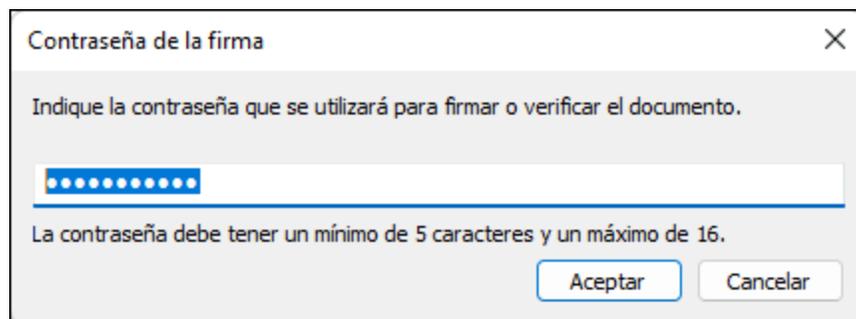
Si el proceso de verificación se ejecuta en un documento firmado, hay dos posibilidades. Primero, que esté disponible la información de autenticación (en la firma o en el SPS). En este caso se ejecuta el proceso de verificación y el resultado aparece en el cuadro de diálogo "Firma XML" (*imagen siguiente*).



Cuando hablamos de información de autenticación nos referimos a la información de clave del certificado o a la contraseña. El diseñador del archivo SPS especifica si la información de clave del certificado se guarda en la firma cuando se firma el documento XML o, si se trata de una firma con contraseña, si la contraseña se guarda en el SPS. En ambos casos, la información de autenticación está disponible. Por consiguiente, el proceso de verificación se ejecuta directamente, sin solicitar más información al usuario de la Authentic View.

La segunda posibilidad es que la información de autenticación no esté disponible en la firma (falta la información de clave del certificado) ni en el archivo SPS (falta la contraseña). En este caso, se le pide al

usuario de la Authentic View que ofrezca más información: una contraseña (como en la imagen siguiente) o la ubicación del certificado.



## 14.6.6 Definir entidades XML

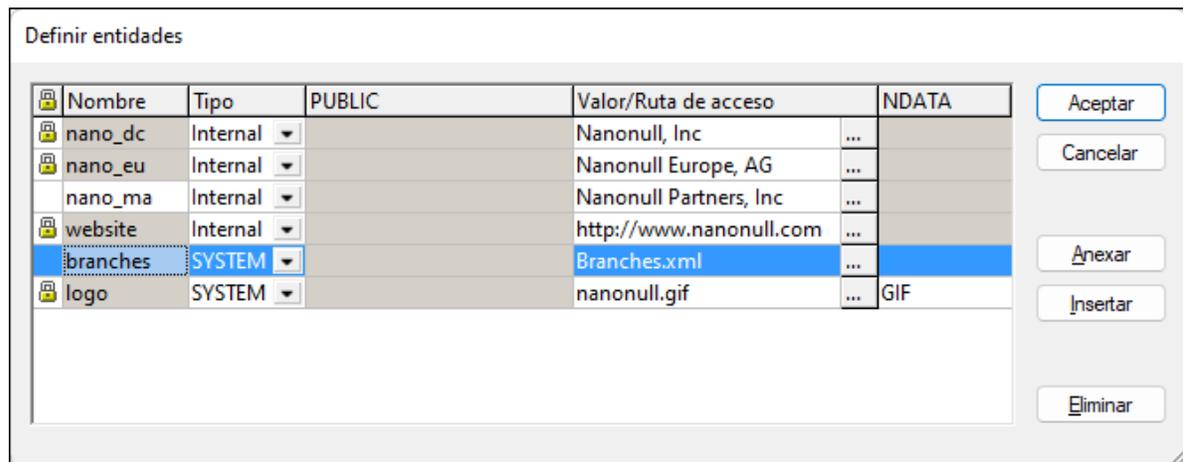
Tanto si su documento está basado en una DTD como si se basa en un esquema XML, puede definir las entidades que se deben usar en la Authentic View. Una vez definidas, estas entidades aparecen en el ayudante de entrada Entidades y en el submenú **Insertar entidad** del menú contextual. Al hacer doble clic en una entidad del ayudante de entrada Entidades, esta entidad se inserta en el punto de inserción del cursor.

Las entidades son útiles cuando hay una cadena de texto, un fragmento XML o un recurso externo que se va a utilizar en varios puntos del documento. La entidad (básicamente un nombre corto que representa los datos) se define en el cuadro de diálogo "Definir entidades". Después de definirla, puede usar la entidad en varios puntos del documento. Así puede ahorrar tiempo y mantener el documento más fácilmente.

En un documento puede usar dos tipos de entidades: **entidades analizadas**, es decir, datos XML (una cadena de texto o un fragmento de un documento XML) o **entidades sin analizar**, es decir, datos no XML como un archivo binario (por lo general, objetos gráficos, sonoros o multimedia). Cada entidad tiene un nombre y un valor. En el caso de las entidades analizadas, la entidad es un marcador de posición para los datos XML. El valor de la entidad es los datos XML propiamente dichos o un URI que apunta a un archivo XML que contiene los datos XML. En el caso de las entidades sin analizar, el valor de la entidad es un URI que apunta al archivo de datos no XML.

Siga estos pasos para definir una entidad:

1. Haga clic en **Authentic | Definir entidades XML...** y aparece el cuadro de diálogo "Definir entidades".



2. Escriba el nombre de su entidad en el campo **Nombre**. Se trata del nombre que aparecerá en el ayudante de entrada Entidades.
3. En la lista desplegable del campo **Tipo** seleccione el tipo de entidad. Hay tres tipos. Una entidad interna (*Internal*) es una cuyo texto se almacena en el documento XML propiamente dicho. Las opciones **PUBLIC** y **SYSTEM** indican que el recurso está fuera del XML y que se localizará usando un identificador público o de sistema. Un identificador de sistema es un URI que ofrece la ubicación del recurso. Un identificador público es un identificador que no depende de la ruta de acceso y permite a algunos procesadores identificar el recurso. Si se indica tanto un identificador público como de sistema, el identificador público se resuelve como identificador de sistema y se usa el identificador de sistema.
4. Si selecciona el tipo **PUBLIC**, escriba el identificador público de su recurso en el campo **PUBLIC**. Si selecciona el tipo *Internal* o **SYSTEM**, se deshabilita el campo **PUBLIC**.
5. En el campo **Valor/Ruta de acceso** tiene estas opciones:
  - Si el tipo de entidad es **Internal**, escriba la cadena de texto de la entidad. La cadena de texto no puede estar delimitada por comillas. Si escribe comillas, se entenderá que forman parte de la cadena de texto.
  - Si el tipo de entidad es **SYSTEM**, escriba el URI del recurso o seleccione un recurso de su red local con el botón **Examinar**. Si el recurso contiene datos analizados, este debe ser un archivo XML (es decir, debe tener la extensión `.xml`). El recurso también puede ser un archivo binario, como un archivo GIF.
  - Si el tipo de entidad es **PUBLIC**, debe introducir también un identificador de sistema en este campo.
6. El campo **NDATA** le dice al procesador que la entidad no se debe analizar sino que se debe enviar al procesador adecuado. Por tanto, rellene el campo **NDATA** solamente si la entidad es una entidad sin analizar.

## Características del cuadro de diálogo

En este cuadro de diálogo puede anexar, insertar y eliminar entidades haciendo clic en los botones correspondientes. También puede ordenar las entidades por orden alfabético en base al valor de cualquiera de las columnas. Al hacer clic una vez en el encabezado de la columna, la columna se ordena alfabéticamente en orden ascendente. Si hace clic dos veces en el encabezado, la columna se ordena por orden descendente. También puede ajustar el tamaño del cuadro de diálogo y el ancho de las columnas.

Una vez utilizadas en el documento XML, las entidades se bloquean y no se pueden editar en el cuadro de diálogo "Definir entidades". Las entidades bloqueadas se señalizan con un símbolo en forma de candado junto a la primera columna del diálogo. Así se garantiza la validez del documento (el documento no es válido si se hace referencia a una entidad que no se definió).

También recibirá una advertencia si define la misma entidad dos veces.

## Limitaciones de las entidades

- Una entidad que está dentro de otra entidad no se resuelve, ni en el cuadro de diálogo, ni en la Authentic View ni en el resultado XSLT. Además, el carácter & de dichas entidades se presenta con caracteres de escape, es decir, &amp; ; .
- Las entidades externas no se resuelven en la Authentic View, excepto cuando la entidad es un archivo de imagen y se introduce como valor de un atributo que está definido en el esquema como atributo de tipo ENTITY o ENTITIES. Este tipo de entidades se resuelven cuando el documento se procesa con un XSLT generado a partir del archivo SPS.

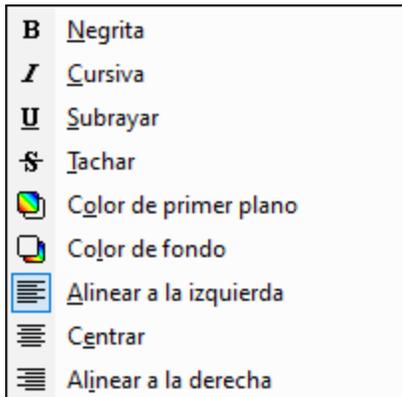
### 14.6.7 Ver marcado

El comando **Ver marcado** tiene un submenú con opciones para controlar el marcado del documento XML en la vista Authentic. A continuación describimos estas opciones.

	El comando <b>Ocultar marcado</b> oculta los símbolos de marcado del documento en la Authentic View.
	El comando <b>Mostrar marcado en tamaño pequeño</b> muestra símbolos pequeños de marcado en la Authentic View.
	El comando <b>Mostrar marcado en tamaño grande</b> muestra símbolos grandes de marcado en la Authentic View.
	El comando <b>Mostrar marcado mixto</b> muestra símbolos de marcado de tamaño mixto en la Authentic View. 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 Authentic View utiliza este tipo personalizado de marcado.

### 14.6.8 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 Authentic View y solamente cuando se coloca el cursor dentro de un elemento que se creó como componente RichEdit en el diseño SPS.



Las propiedades de estilo del menú **RichEdit** se aplican al texto seleccionado al hacer clic en el uno de los comandos de marcado RichEdit. Además de la fuente y el tamaño de fuente especificado en la barra de herramientas Authentic, el usuario de la Authentic View puede especificar el peso de la fuente, su estilo, decoración, color, color de fondo y la alineación del texto seleccionado.

### 14.6.9 Anexar/insertar/duplicar/eliminar fila

	El comando <b>Anexar fila</b> anexa una fila a la tabla seleccionada en la Authentic View.
	El comando <b>Insertar fila</b> inserta una fila en la tabla seleccionada en la Authentic View.
	El comando <b>Duplicar fila</b> crea una copia de la fila seleccionada en la tabla de la Authentic View.
	El comando <b>Eliminar fila</b> elimina la fila seleccionada en la tabla de la Authentic View.

### 14.6.10 Mover fila, Eliminar fila

Los comandos siguientes permiten mover las filas de las tablas de la vista Authentic:

- En una tabla de la Authentic View el comando Subir fila sube la fila seleccionada una fila más arriba.
- En una tabla de la Authentic View el comando Bajar fila baja la fila seleccionada una fila más abajo.

### 14.6.11 Generar un documento HTML, RTF, PDF o Word 2007+

Estos cuatro comandos sirven para generar documentos de salida a partir del documento XML de un archivo PXF en la Authentic View:

- **Generar un documento HTML**
- **Generar un documento RTF**

- **Generar un documento PDF**
- **Generar un documento Word 2007+**
- **Generar un documento de texto**

Estos cinco comandos también están disponibles en la barra de herramientas "Formulario XML Portátil" (*imagen siguiente*).

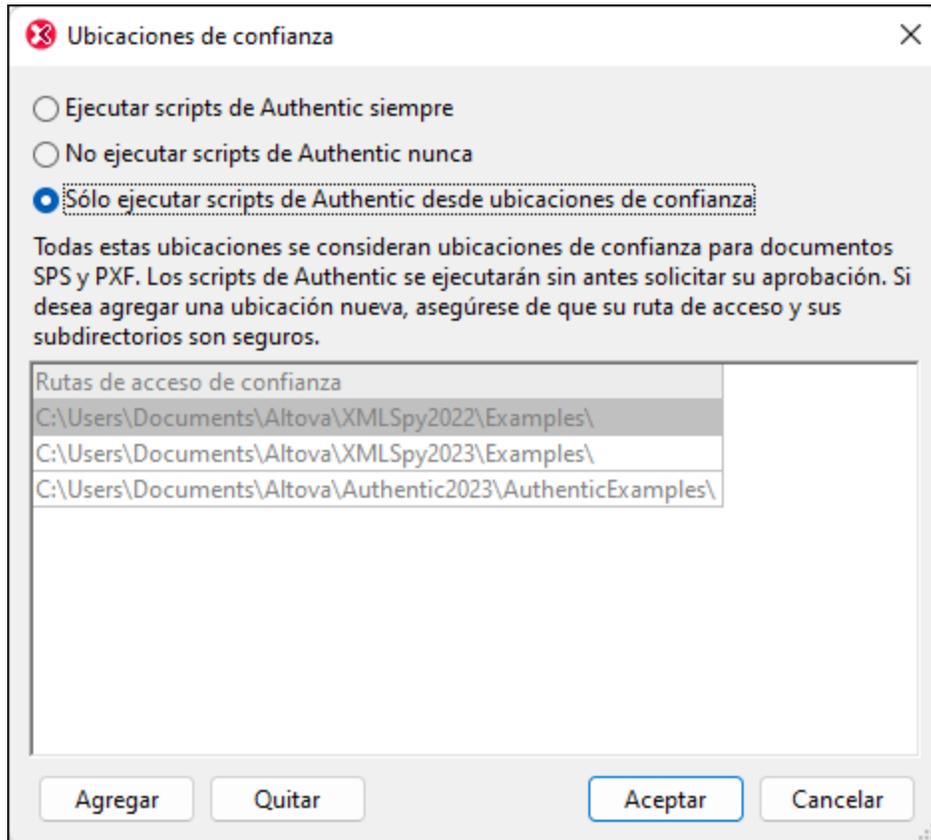


Al hacer clic en estos comandos o botones se genera un documento HTML, RTF, PDF o DocX, dependiendo del comando seleccionado.

Estos comandos se habilitan solamente si el archivo abierto en la Authentic View es un archivo PXF. Además, si el archivo PXF contiene una hoja de estilos para un formato de salida concreto, solamente se habilita el comando correspondiente. Por ejemplo, si el archivo PXF contiene hojas de estilos XSLT para HTML y RTF, entonces solamente se habilitan los comandos para generar documentos HTML y RTF, mientras que los comandos para PDF y DocX se deshabilitan.

## 14.6.12 Ubicaciones de confianza

El comando **Ubicaciones de confianza** abre el cuadro de diálogo del mismo nombre (*imagen siguiente*). En este cuadro de diálogo puede configurar opciones de seguridad para los scripts del archivo SPS. Cuando se abre en la vista Authentic un archivo XML basado en un SPS que contiene scripts, los scripts se ejecutan o no dependiendo de las opciones elegidas en este cuadro de diálogo.



Hay tres posibilidades:

- Los scripts de Authentic se ejecutan siempre
- Los scripts de Authentic no se ejecutan nunca
- Se ejecutan solamente los scripts de Authentic que están en ubicaciones de confianza. La lista de ubicaciones (carpetas) de confianza aparece en el panel inferior del cuadro de diálogo. Use el botón **Agregar** para buscar una carpeta y añadirla a la lista. Para quitar una carpeta de la lista, selecciónela y después haga clic en **Quitar**. Para confirmar haga clic en **Aceptar**.

## 14.7 Menú Vista

El menú **Vista** controla la visualización de la [ventana principal](#)<sup>16</sup> activa y permite definir la presentación del documento en pantalla. En esta sección explicamos los comandos del menú **Vista**.

### 14.7.1 Vista Authentic

Este comando **cambia** la vista actual del documento a la [Authentic View](#)<sup>51</sup>.

En la Authentic View puede editar documentos XML **basados en plantillas StyleVision Power Stylesheet creadas con Altova StyleVision**. Estas plantillas muestran los documentos XML en formato gráfico, lo cual facilita la edición de documentos XML.

### 14.7.2 Vista Explorador

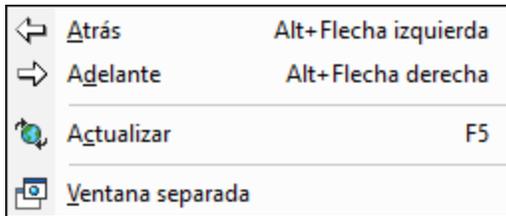


Este comando **cambia** la vista actual del documento a la [vista Explorador](#)<sup>103</sup>. El explorador web preparado para XML de esta vista representa los documentos XML sirviéndose de la información de las hojas de estilos CSS/XSL disponibles.

Si selecciona la opción *Validar archivos automáticamente al guardarlos* en la [sección](#)<sup>279</sup> [Archivo](#)<sup>279</sup> [del cuadro de diálogo "Opciones"](#)<sup>279</sup> (**Herramientas | Opciones**), el documento se valida automáticamente antes de cambiar a la vista Explorador. Para más información, consulte la sección [Vista Explorador](#)<sup>103</sup>.

## 14.8 Menú Explorador

Los comandos del menú **Explorador** están disponibles solamente en la [vista Explorador](#)<sup>103</sup>.



### Atrás, Adelante

El comando **Atrás** (**Alt + flecha izda**) está disponible en las vistas Explorador y Esquema. También puede usar la tecla **Retroceso**. Este comando es muy práctico si hace clic en un enlace del documento XML y quiere volver al documento.

El comando **Adelante** (**Alt + flecha der**) permite avanzar por (i) páginas vistas previamente en la vista Explorador y (ii) vistas previas de componentes en la vista Esquema.

### Actualizar

El comando **Actualizar** (**F5**) está disponible en la vista Explorador y sirve para actualizar la vista, es decir, se vuelve a cargar el documento y su documentos relacionados (las hojas de estilos CSS/XSL, los documentos DTD, etc.).

### Fuentes

Al hacer clic en el comando **Fuentes** aparece un submenú donde puede seleccionar el tamaño de fuente predeterminado en el que se representa el texto del documento XML. Este comando está disponible en la vista Explorador.

### Ventana separada

El comando **Ventana separada** está disponible en la vista Explorador y abre la vista Explorador en otra ventana diferente, separada de las demás vistas. Así puede ver la vista Explorador en paralelo a una vista de edición, por ejemplo.

Si tras realizar cambios en la vista de edición desea actualizar la ventana separada de la vista Explorador, pulse **F5** en la vista de edición. Si desea volver a acoplar la ventana separada en la interfaz, haga clic en la ventana de la vista Explorador y después en el comando **Ventana separada**.

## 14.9 Menú Herramientas

Con el menú **Herramientas** puede:

- Revisar la [ortografía](#)<sup>252</sup> de sus documentos XML
- Acceder al [entorno de scripting](#)<sup>306</sup> de Authentic Desktop para crear, gestionar y almacenar sus propios formularios, macros y controladores de eventos.
- [Ver](#)<sup>258</sup> las macros asignadas actualmente.
- Acceder a comandos personalizados que usan aplicaciones externas. Estos comandos se pueden crear en [la pestaña](#)<sup>264</sup> [Herramientas](#)<sup>264</sup> [del cuadro de diálogo Personalizar](#)<sup>264</sup>.
- [Definir recursos globales](#)<sup>258</sup>
- [Personalizar](#)<sup>260</sup> su versión de Authentic Desktop: definiendo sus propias barras de herramientas, teclas de acceso rápido, macros y menús.
- Definir [opciones de configuración](#)<sup>278</sup> globales de Authentic Desktop

### 14.9.1 Ortografía

El corrector ortográfico de Authentic Desktop (y sus diccionarios integrados) está disponible en la Authentic View.

**Nota:** los diccionarios integrados que vienen con el software de Altova no son un reflejo de las preferencias lingüísticas de Altova. Los diccionarios integrados del corrector ortográfico se seleccionaron porque su licencia permite su distribución con software comercial ([MPL](#), [LGPL](#) o [BSD](#), por ejemplo). Existen muchos otros diccionarios de código abierto, pero se distribuyen con licencias más restrictivas, como la licencia [GPL](#). Estos diccionarios pueden descargarse e instalarse por separado desde la página <http://www.altova.com/es/dictionaries>. Elija los diccionarios que quiere usar en base a las licencias de estos y a sus propias necesidades.

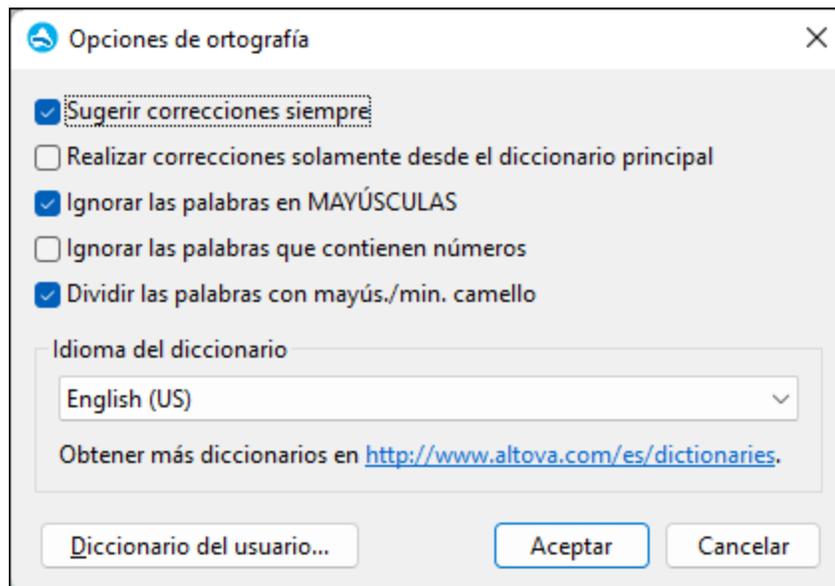
Este apartado describe el uso del corrector ortográfico y tiene tres partes:

- [Seleccionar el idioma del corrector ortográfico](#)<sup>252</sup>
- [Ejecutar la revisión ortográfica](#)<sup>253</sup>

#### Seleccionar el idioma del corrector ortográfico

Siga estos pasos para seleccionar el idioma del corrector ortográfico:

1. Haga clic en el comando de menú **Herramientas | Opciones de ortografía**.
2. En el cuadro de diálogo "Opciones de ortografía" que aparece seleccione uno de los diccionarios instalados de la lista desplegable del cuadro combinado *Idioma del diccionario*.

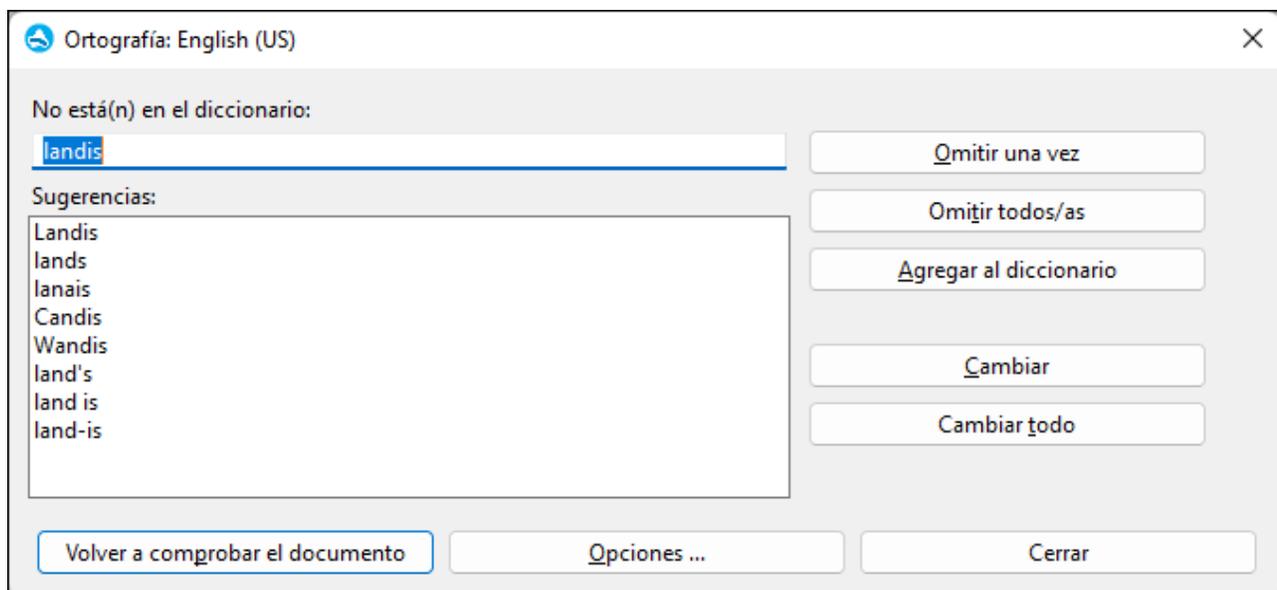


3. Para terminar haga clic en **Aceptar**.

El idioma del diccionario seleccionado será utilizado durante la revisión ortográfica. Si el idioma que desea utilizar no está instalado, puede descargar más diccionarios desde el sitio web de Altova. Para más información consulte el apartado [Opciones de ortografía \(añadir diccionarios para el corrector ortográfico\)](#) <sup>256</sup>.

### Ejecutar la revisión ortográfica

El comando **Herramientas | Ortografía (Mayús+F7)** empieza a revisar el documento XML activo inmediatamente. Si el corrector encuentra una palabra desconocida, aparece el cuadro de diálogo *Ortografía: No está en el diccionario* (imagen siguiente). De lo contrario, el corrector recorre el documento hasta el final.



A continuación describimos el cuadro de diálogo *Ortografía: No está en el diccionario* y las opciones disponibles en él.

#### No está en el diccionario

Este cuadro de texto contiene la palabra desconocida (es decir, una palabra que no está en el diccionario seleccionado ni en el diccionario del usuario). En caso de encontrar una palabra desconocida hay varias opciones:

- Puede editar la palabra en el cuadro de texto manualmente o seleccionar una de las opciones del panel *Sugerencias*. Después haga clic en el botón **Modificar** para reemplazar la palabra en el documento XML con la palabra nueva. (Si hace doble clic en una sugerencia, la palabra sugerida se inserta directamente en el documento XML.) La palabra que aparece en el cuadro *No está en el diccionario* también aparece resaltada en el documento XML para que pueda editarla directamente en el documento si lo prefiere. Si hace clic en el botón **Modificar todos** se reemplazan todas las instancias de la palabra en el documento con la palabra nueva.
- Puede decidir no realizar ningún cambio y omitir la advertencia del corrector (para la instancia actual de la palabra o para todas las instancias de la palabra).
- Puede añadir la palabra al diccionario del usuario para que a partir de ahora la palabra sea correcta.

#### Sugerencias

Este cuadro sugiere palabras que se parecen a la palabra desconocida (las palabras sugeridas provienen del diccionario integrado y de los diccionarios del usuario). Si hace doble clic en una palabra de la lista, la palabra elegida se inserta automáticamente en el documento y el proceso de revisión ortográfica continúa.

#### Omitir una vez

Este botón permite seguir revisando el documento e ignorar la primera aparición de la palabra desconocida. Si la misma palabra aparece otra vez en el documento, volverá a aparecer en el recuadro *No está en el diccionario*.

#### Omitir todos

Este botón permite seguir revisando el documento e ignorar todas las apariciones de la palabra desconocida en todo el documento.

#### Agregar al diccionario

Este botón añade la palabra desconocida al diccionario del usuario. Para tener acceso al diccionario del usuario (para poder editarlo) utilice el cuadro de diálogo [Opciones de ortografía](#)<sup>255</sup>.

#### Modificar

Este botón reemplaza la palabra desconocida en el documento XML con la palabra (nueva) del cuadro *No está en el diccionario*.

#### Modificar todos

Este botón reemplaza todas las instancias de la palabra desconocida en el documento XML con la palabra (nueva) del cuadro *No está en el diccionario*.

#### Volver a revisar el documento

Este botón vuelve a revisar el documento desde el principio.

#### Opciones

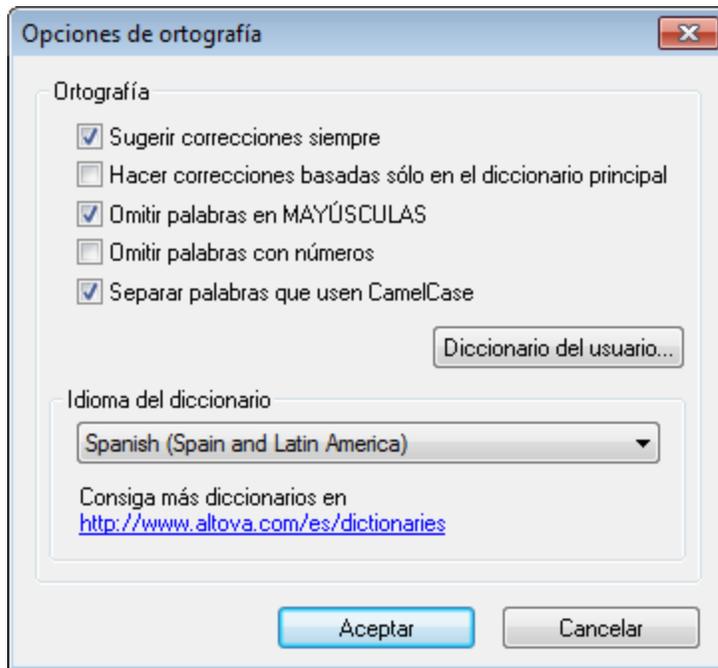
Si hace clic en Opciones se abre el cuadro de diálogo [Opciones de ortografía](#)<sup>255</sup>.

#### Cerrar

Este botón cierra el cuadro de diálogo Ortografía.

## 14.9.2 Opciones de ortografía

El cuadro de diálogo **Opciones de ortografía** sirve para configurar el corrector ortográfico de forma general.



### *Sugerir correcciones siempre*

Si marca esta casilla, el cuadro *Sugerencias* siempre mostrará opciones del diccionario integrado elegido y del diccionario del usuario. Si desactiva esta opción, el corrector no ofrecerá ninguna sugerencia.

### *Hacer correcciones basadas sólo en el diccionario principal*

Si marca esta casilla, solamente se usan sugerencias del diccionario integrado elegido (diccionario principal). El diccionario del usuario no se utilizará para ofrecer sugerencias. Si marca esta casilla, se deshabilita el botón **Diccionario del usuario**. Es decir, mientras esté activa esta opción el diccionario del usuario no se podrá editar.

### *Omitir palabras en MAYÚSCULAS*

Si marca esta casilla, el corrector ortográfico pasa por alto las palabras que están en mayúsculas.

### *Omitir palabras con números*

Si marca esta casilla, el corrector ortográfico pasa por alto las palabras que contienen números.

### *Separar palabras que usen CamelCase*

Las palabras que usan CamelCase están formadas por palabras que empiezan por mayúsculas, unidas unas a otras sin espacios. Por ejemplo, la palabra *CamelCase* está formada por las palabras *Camel* y *Case*, ambas en mayúsculas y unidas sin espacios. Este tipo de palabras no suele aparecer en los diccionarios y, por tanto, el corrector ortográfico las marcará como erróneas. Marque la casilla *Separar palabras que usen CamelCase* para que el corrector revise por separado cada una de las palabras que forman la palabra CamelCase. Esta

opción está marcada por defecto.

### *Idioma del diccionario*

Use este cuadro combinado para seleccionar el idioma del diccionario integrado del corrector ortográfico. La opción predeterminada es **US English** (inglés EE UU). Para descargar gratis más diccionarios en otros idiomas, visite el [sitio web de Altova](#).

## 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:

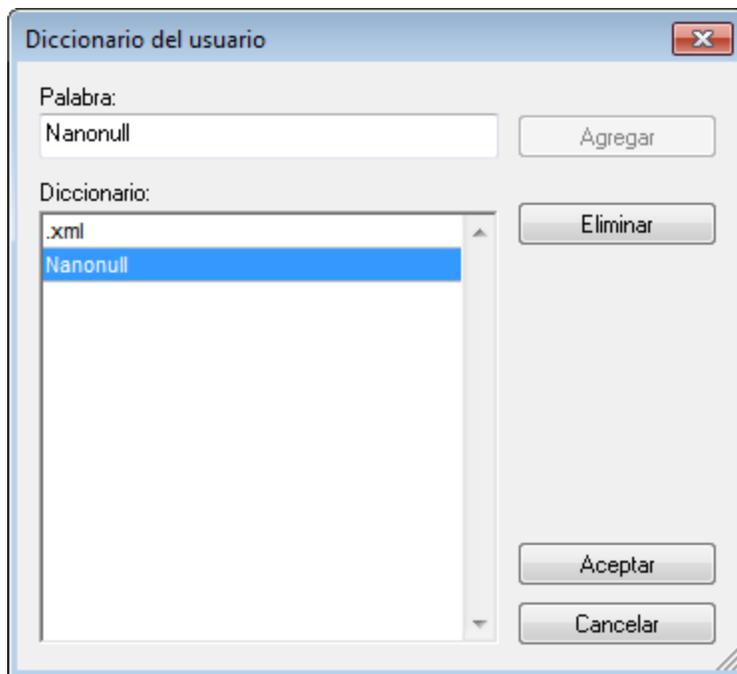
- Puede añadir diccionarios Hunspell a una subcarpeta nueva de la carpeta `Lexicons`. Los diccionarios Hunspell se pueden descargar desde <https://wiki.services.openoffice.org/wiki/Dictionaries> o desde <https://extensions.services.openoffice.org/en/dictionaries>, por ejemplo. (Recuerde que OpenOffice 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](#) <sup>252</sup>. 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\\Documentos\Altova\SpellChecker\Lexicons\user.dic

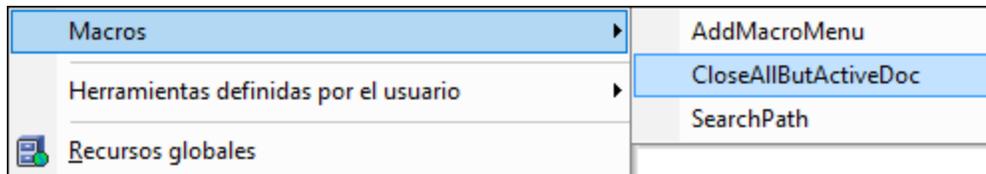
### 14.9.3 Editor de script

El comando **Editor de script** abre la ventana del editor de script. Para más información sobre cómo trabajar con el editor de script, consulte la sección [Editor de scripts](#)<sup>306</sup> de la presente documentación.

**Nota:** es necesario tener instalado .NET Framework versión 2.0 (o superior) para poder ejecutar el editor de scripts.

### 14.9.4 Macros

1. Al pasar el ratón por el comando Macros aparece un submenú con todas las macros definidas en el proyecto de script que está activo en Authentic Desktop (imagen siguiente).
2. El proyecto de script activo se especifica en [la sección Scripting del cuadro de diálogo "Opciones"](#)<sup>290</sup>.



3. Al hacer clic en una macro del submenú (imagen anterior) la macro se ejecuta.

### 14.9.5 Herramientas definidas por el usuario

Al pasar el puntero del ratón por el comando **Herramientas definidas por el usuario** aparece un submenú con comandos hechos a medida que usan aplicaciones externas. Para crear estos comandos, use [la pestaña](#)<sup>264</sup> [Herramientas](#)<sup>264</sup> [del cuadro de diálogo Personalizar](#)<sup>264</sup>. Al hacer clic en uno de estos comandos personalizados, se ejecuta la acción asociada al comando.

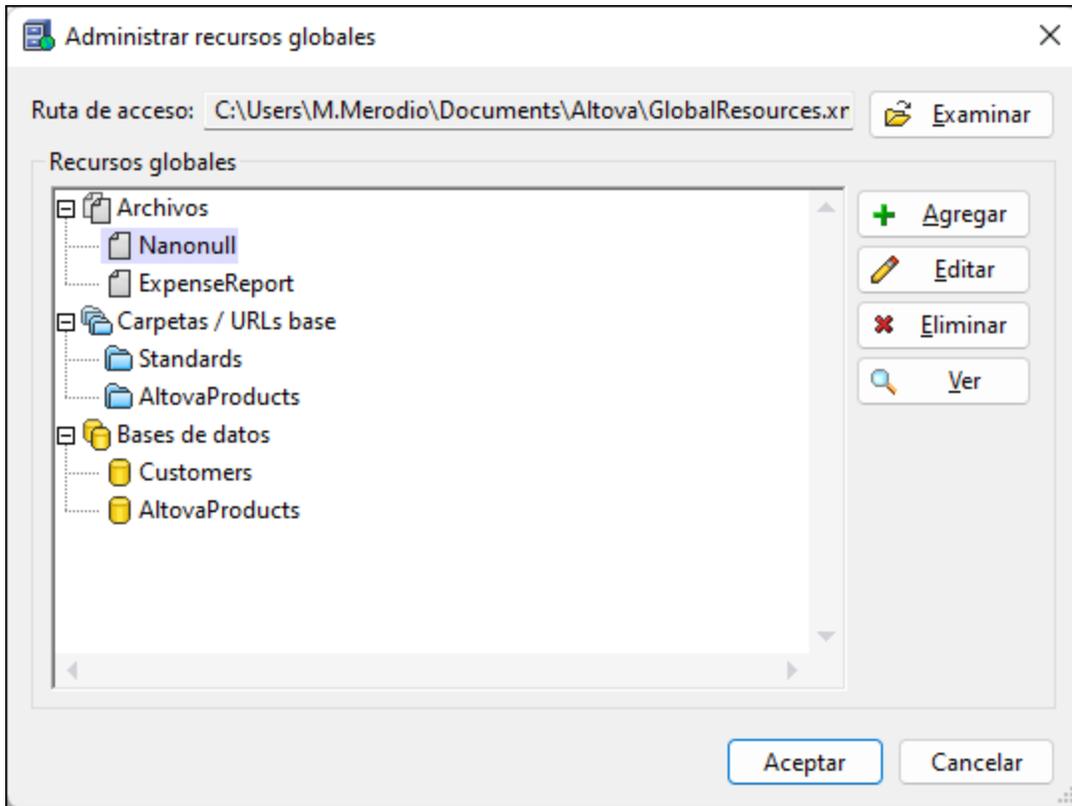
El comando **Herramientas definidas por el usuario | Personalizar** abre [la pestaña](#)<sup>264</sup> [Herramientas](#)<sup>264</sup> [del cuadro de diálogo Personalizar](#)<sup>264</sup> (donde puede crear comandos hechos a medida para el submenú del comando **Herramientas definidas por el usuario**.)

### 14.9.6 Recursos globales

El comando **Recursos globales** abre el cuadro de diálogo "Administrar recursos globales" (*imagen siguiente*), donde puede:

- Seleccionar qué archivo XML de recursos globales se utiliza para los recursos globales.
- Añadir recursos globales (o alias) de tipo archivo, carpeta o base de datos.

- Especificar varias configuraciones para cada recurso global (alias). Cada configuración se asigna a un recurso concreto.

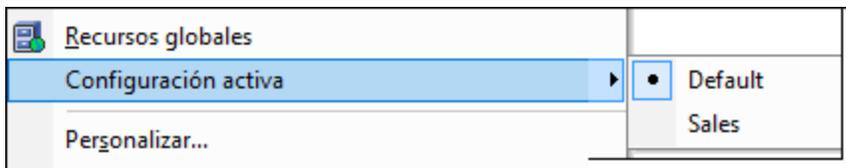


Para más información consulte la sección [Definir recursos globales](#) <sup>105</sup>.

**Nota:** también puede abrir el cuadro de diálogo "Administrar recursos globales" desde la [barra de herramientas Recursos globales](#) <sup>262</sup> (**Herramientas | Personalizar | Barras de herramientas | Recursos globales**).

## 14.9.7 Configuración activa

Al pasar el puntero del ratón sobre el comando **Configuración activa** aparece un submenú con todas las configuraciones definidas en el [archivo XML de recursos globales](#) <sup>258</sup> 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](#) <sup>262</sup> (Herramientas | Personalizar | Barras de herramientas | Recursos globales).

## 14.9.8 Personalizar

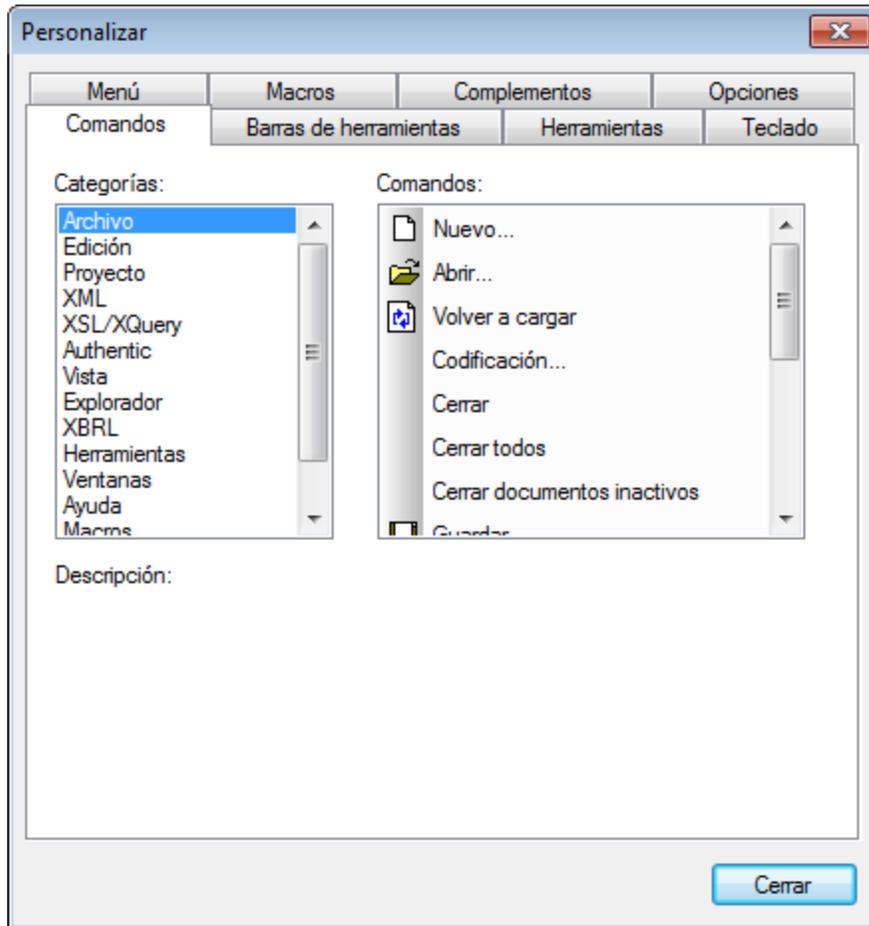
El comando **Personalizar** sirve para adaptar los menús y barras de herramientas de la aplicación a sus requisitos. Al hacer clic en este comando aparece el cuadro de diálogo "Personalizar", que está compuesto por estas pestañas:

- [Comandos](#) <sup>260</sup>: desde esta pestaña puede arrastrar comandos de la aplicación y de macros hasta barras de menú, menús y barras de herramientas.
- [Barras de herramientas](#) <sup>262</sup>: aquí puede activar, desactivar y restaurar cada una de las barras de herramientas.
- [Herramientas](#) <sup>252</sup>: aquí puede añadir comandos que abren programas externos desde dentro de la interfaz.
- [Teclado](#) <sup>265</sup>: aquí puede crear teclas de acceso directo para cada comando de la aplicación y de macros.
- [Menú](#) <sup>268</sup>: aquí puede seleccionar y activar las barras de menú y los menús contextuales que desea personalizar. Esta pestaña funciona conjuntamente con la pestaña *Comandos*.
- [Macros](#) <sup>270</sup>: aquí puede asignar comandos nuevos a las macros.
- [Complementos](#) <sup>272</sup>: aquí puede activar complementos e integrarlos a la interfaz.
- [Opciones](#) <sup>278</sup>: aquí puede configurar la presentación de las barras de herramientas.

En este apartado también describimos el [menú contextual](#) <sup>275</sup> que aparece si hace clic con el botón derecho en una barra de menú, en un menú o en una barra de herramientas de la aplicación mientras está abierto el cuadro de diálogo "Personalizar".

### 14.9.8.1 Comandos

En la pestaña **Comandos** puede personalizar sus menús y barras de herramientas y añadir comandos a los menús y a las barras de herramientas, dependiendo de lo que necesite. No obstante, tenga en cuenta que no puede crear comandos ni menús nuevos para la aplicación.



Para añadir un comando a una barra de herramientas o menú:

1. Seleccione el comando **Herramientas | Personalizar**. Se abre el cuadro de diálogo "Personalizar".
2. Seleccione la pestaña **Comandos**. En el cuadro de lista *Categorías* seleccione la opción **Todos los comandos**. Todos los comandos disponibles aparecen en el cuadro de lista *Comandos*.
3. Haga clic en un comando del cuadro de lista *Comandos* y arrástrelo a un menú o barra de herramientas ya existente. Al pasar el puntero por encima de una posición donde se puede colocar el comando aparece el icono **I**.
4. Cuando encuentre la posición donde desea colocar el comando, suelte el botón del ratón.

Tenga en cuenta que:

- Mientras arrastra el comando, aparece un pequeño botón al final del puntero del ratón. Esto indica que el comando está siendo arrastrado.
- Si el comando no se puede colocar en la posición actual del cursor, debajo del puntero aparece una **X**.
- Si el cursor está en una posición donde se puede colocar el comando (en una barra de herramientas o en un menú), la **X** desaparece y el icono **I** indica que la posición es válida.
- Los comandos se pueden colocar en menús o barras de herramientas. Si [creó una barra de herramientas nueva](#) <sup>262</sup>, puede usar este mecanismo de personalización para rellenar la barra de herramientas con comandos.

- Si pasa el cursor por un menú que está cerrado, el menú se abre y puede insertar el comando en cualquier parte del menú.

## Agregar comandos a menús contextuales

También puede añadir comandos a menús contextuales arrastrando comandos del cuadro de lista *Comandos* hasta el menú contextual:

1. Haga clic en la pestaña [Menú](#)<sup>268</sup> del cuadro de diálogo "Personalizar".
2. En el cuadro combinado del panel *Menús contextuales* seleccione un menú contextual. El menú contextual seleccionado aparece en pantalla.
3. Vuelva a la pestaña *Comandos* del cuadro de diálogo "Personalizar".
4. Seleccione un comando en el cuadro de lista *Comandos* y arrástrelo hasta la posición deseada del menú contextual.

## Eliminar un comando o menú

Por último, puede eliminar un comando de un menú, menú contextual (ver párrafo anterior) o barra de herramientas o eliminar un menú entero:

1. Abra el cuadro de diálogo "Personalizar" (**Herramientas | Personalizar**).
2. Seleccione cualquier pestaña del cuadro de diálogo "Personalizar". Haga clic con el botón derecho en un menú o comando de menú y seleccione **Eliminar** en el menú contextual que aparece. Si lo prefiere, también puede arrastrar el menú o comando de menú hasta que aparezca el icono **X** debajo del puntero del ratón y suelte el menú o comando de menú. Como resultado se elimina el menú o comando de menú.

Para volver a instalar los comandos de menú eliminados, utilice los mecanismos descritos en este apartado. Para restablecer un menú eliminado, seleccione **Herramientas | Personalizar | Menú** y pulse el botón **Restaurar** del panel *Menús del marco de la aplicación*. Otra opción es seleccionar **Herramientas | Personalizar | Barras de herramientas**, hacer clic en la barra de herramientas pertinente y pulsar el botón **Restaurar**.

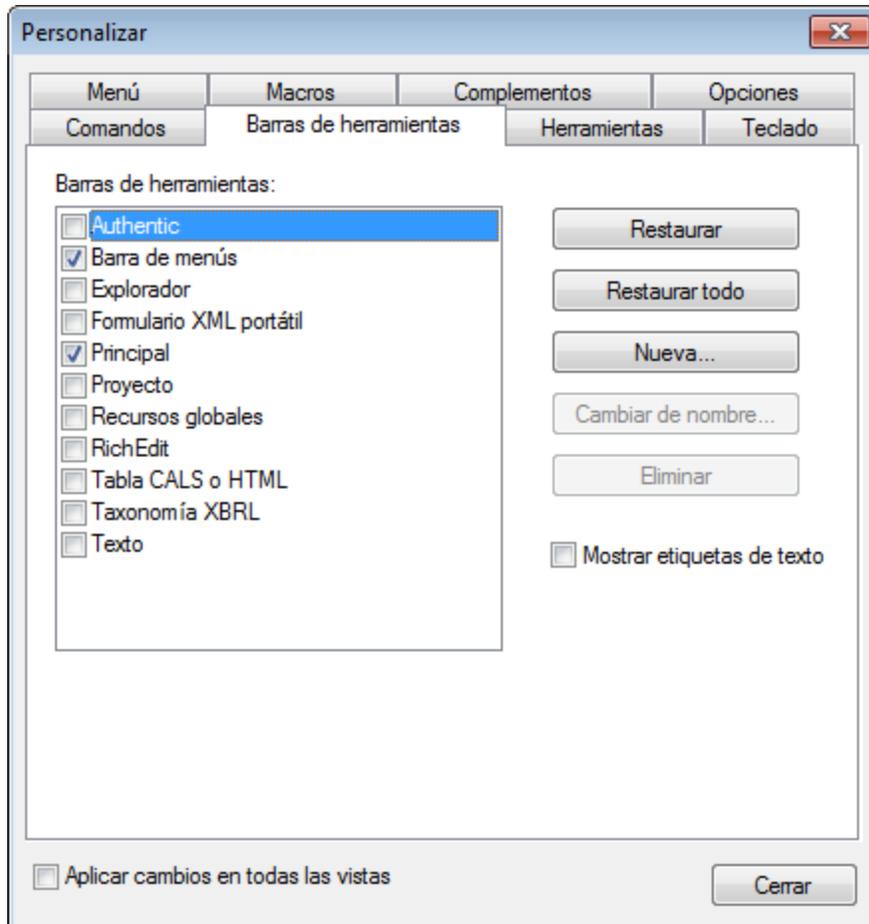
### 14.9.8.2 Barras de herramientas

En la pestaña **Barras de herramientas** puede: (i) activar o desactivar barras de herramientas (es decir, decidir qué barras de herramientas aparecen en la interfaz), (ii) definir qué iconos aparecen en cada barra de herramientas y (iii) crear barras de herramientas personalizadas.

Las barras de herramientas incluyen iconos para los comandos de menú más utilizados. Además, al pasar el puntero sobre un icono, se ofrece información rápida sobre el icono en un mensaje emergente y en la barra de estado de la aplicación. Las barras de herramientas se pueden colocar en cualquier posición de la pantalla, donde aparece como ventana flotante.

**Nota:** para añadir un comando a una barra de herramientas, arrastre un comando del cuadro de lista *Comandos* de la pestaña [Comandos](#)<sup>260</sup> hasta la barra de herramientas. Para eliminar un comando de una barra de herramientas, abra el cuadro de diálogo Personalizar, seleccione cualquier pestaña y arrastre el comando fuera de la barra de herramientas (para más información consulte el apartado [Comandos](#)<sup>260</sup>).

**Nota:** la configuración de una barra de herramientas definida en una vista determinada afecta solamente a dicha vista. Para que la configuración afecte a todas las vistas, marque la casilla situada al final de la pestaña *Barras de herramientas* del cuadro de diálogo Personalizar.



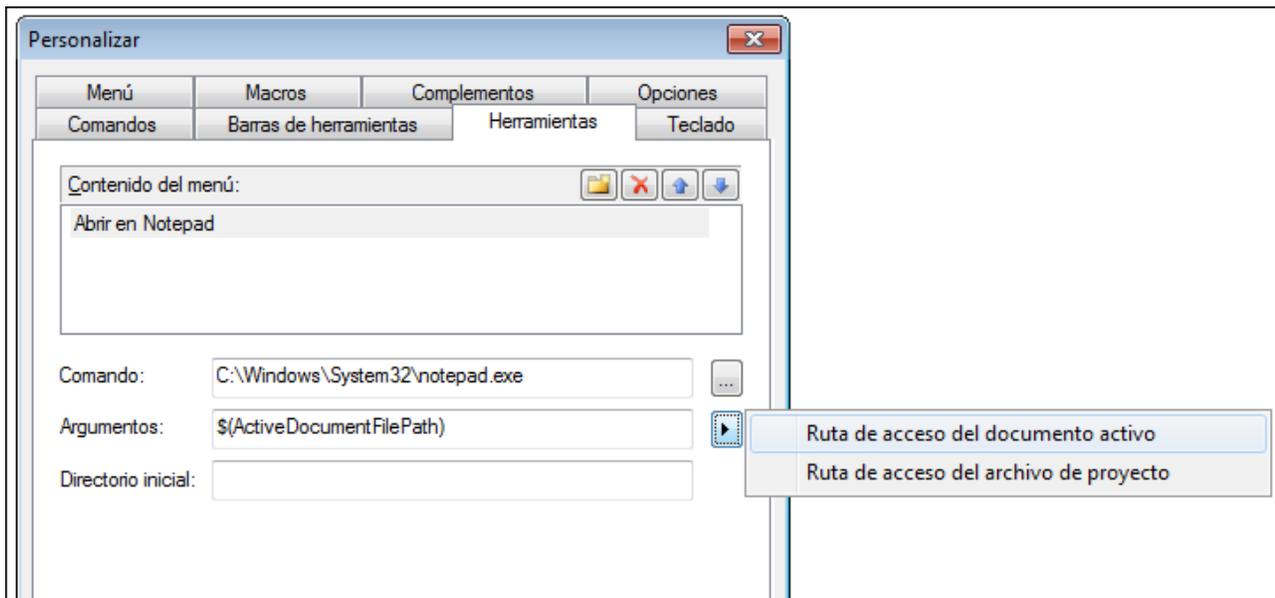
En esta pestaña puede llevar a cabo estas operaciones:

- *Para activar/desactivar una barra de herramientas:* marque su casilla en el cuadro de lista *Barras de herramientas*.
- *Para aplicar los cambios a todas las vistas:* marque la casilla situada al final de la pestaña. De lo contrario, los cambios realizados afectan solamente a la vista activa. Recuerde que los cambios realizados **después** de marcar la casilla *Aplicar cambios en todas las vistas* afectarán a todas las vistas.
- *Para añadir una barra de herramientas nueva:* pulse el botón **Nueva...** y escriba el nombre de la barra de herramientas nuevas en el cuadro de diálogo "Nombre de la barra de herramientas" que aparece. Arrastre comandos desde la pestaña [Comandos](#)<sup>260</sup> hasta la barra de herramientas nueva.
- *Para cambiar el nombre de una barra de herramientas nueva:* seleccione la barra de herramientas en el panel *Barra de herramientas* y pulse el botón **Cambiar de nombre**. Edite el nombre en el cuadro de diálogo "Nombre de la barra de herramientas" que aparece.

- *Para restaurar la barra de menús:* seleccione *Barra de menús* en el panel *Barras de herramientas* y pulse el botón **Restaurar**. La barra de menús vuelve a su estado original de instalación.
- *Para restaurar todas las barras de herramientas y comandos de menú:* pulse el botón **Restaurar todo**. Todas las barras de herramientas y menús vuelven a su estado original de instalación.
- *Para eliminar una barra de herramientas:* seleccione la barra de herramientas en el panel *Barras de herramientas* y pulse el botón **Eliminar**.
- *Para mostrar las etiquetas de texto de una barra de herramientas:* seleccione la barra de herramientas y marque la casilla *Mostrar etiquetas de texto*. Recuerde que debe activar las etiquetas de texto de cada barra de herramientas por separado.

### 14.9.8.3 Herramientas

En la pestaña **Herramientas** puede crear comandos para poder usar aplicaciones externas desde Authentic Desktop directamente. Estos comandos se añaden al menú **Herramientas | Herramientas definidas por el usuario**. Por ejemplo, el archivo activo de la ventana principal de Authentic Desktop se puede abrir en una aplicación externa como Notepad haciendo clic en un comando del menú **Herramientas | Herramientas definidas por el usuario** creado previamente.



Para crear un comando y usarlo con una aplicación externa siga estos pasos:

1. En el panel *Contenido del menú* (imagen anterior), haga clic en el icono **Nuevo** de la barra de título del panel y aparece una línea nueva. Escriba el nombre del comando de menú que desea crear. En el ejemplo de la imagen anterior introducimos el comando de menú **Abrir en Notepad**. La idea es usar este comando para abrir el documento activo en la aplicación externa Notepad. Puede añadir más comandos a la lista de comandos haciendo clic en el icono **Nuevo**. Para subir o bajar un comando de la lista, use los iconos **Subir** y **Bajar** respectivamente. Para eliminar un comando, selecciónelo y haga clic en el icono **Eliminar**.

2. Para asignar una aplicación externa al comando, seleccione el comando en el panel *Contenido del menú*. En el campo *Comando* escriba la ruta de acceso del archivo ejecutable de la aplicación externa o búsquelo. En el ejemplo de la imagen anterior introdujimos la ruta de acceso de la aplicación Notepad en el campo *Comando*.
3. Las acciones que se pueden llevar a cabo con la aplicación externa aparecen al hacer clic en el botón desplegable del campo *Argumentos (imagen anterior)*. Estas acciones se describen más abajo. Seleccione una acción y en el campo *Argumentos* aparece una cadena de código para la acción elegida.
4. Si desea especificar un directorio de trabajo actual, introdúzcalo en el campo *Directorio inicial*.
5. Haga clic en **Cerrar** para finalizar.

Los comandos creados en esta pestaña aparecen en el menú **Herramientas | Herramientas definidas por el usuario** y en el menú contextual de los archivos y carpetas de la ventana Proyecto (en el submenú **Herramientas definidas por el usuario**).

Al hacer clic en uno de los comandos de **Herramientas | Herramientas definidas por el usuario**, se ejecuta la acción asignada al comando. El comando del ejemplo de la imagen hace esto: abre en Notepad el documento que está activo en la ventana principal de Authentic Desktop. El comando de la aplicación externa también está disponible en el menú contextual de los archivos de la ventana Proyecto (clic con el botón derecho en un archivo de la ventana Proyecto). En la ventana Proyecto también puede abrir varios archivos (si la aplicación externa lo permite) seleccionando varios archivos y haciendo clic en el comando del menú contextual.

## Argumentos

El campo *Argumentos* especifica la acción que debe ejecutar el comando de la aplicación externa. Hay dos argumentos posibles:

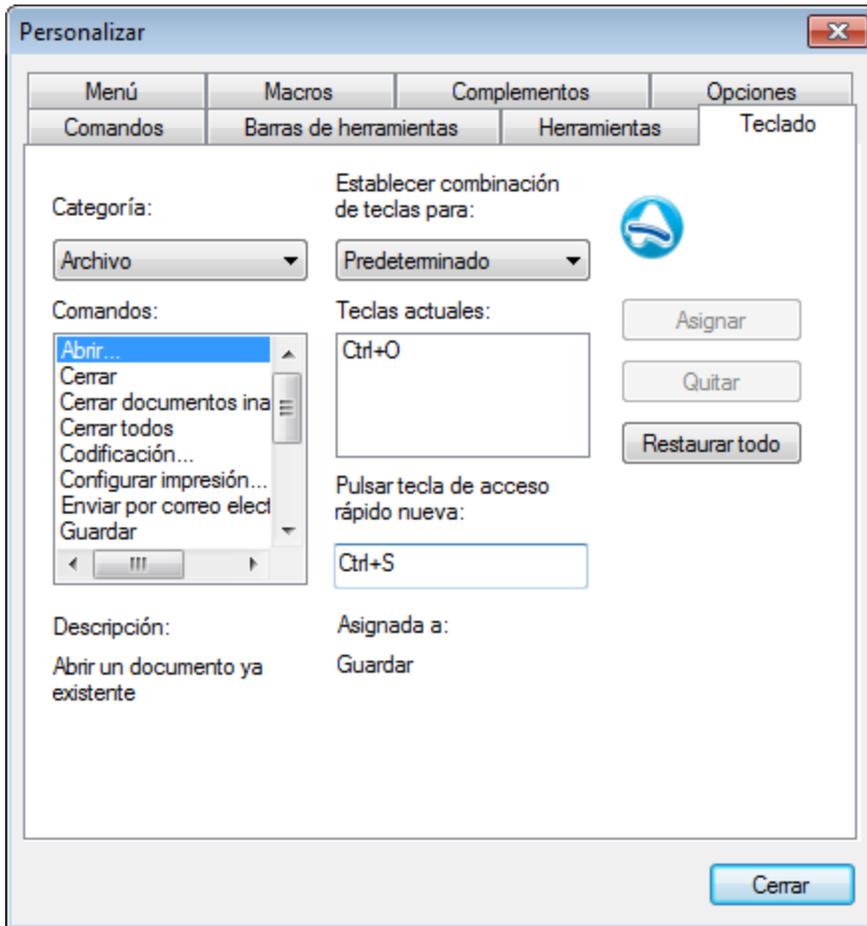
- *Ruta de acceso del documento activo*: el comando de **Herramientas | Herramientas definidas por el usuario** abre en la aplicación externa el documento que está activo en Authentic Desktop. El comando del menú contextual de un archivo de la ventana Proyecto abre el archivo seleccionado en la aplicación externa
- *Ruta de acceso del archivo de proyecto*: abre en la aplicación externa el archivo de proyecto de Authentic Desktop (el archivo `.spp`)

## Directorio inicial

El campo *Directorio inicial* es opcional y es la ruta de acceso que se usa como directorio actual.

## 14.9.8.4 Teclado

En la pestaña **Teclado** puede crear teclas de acceso rápido nuevas o cambiar las teclas de acceso rápido ya existentes para cualquier comando de la aplicación.



Siga estos pasos para asignar una tecla de acceso rápido nueva a un comando o cambiar una tecla de acceso rápido ya existente:

1. En el cuadro combinado *Categoría* seleccione la opción *Todos los comandos*. Recuerde que si [seleccionó una macro como comando asociado](#)<sup>270</sup>, la macro también estará disponible en el cuadro combinado *Categoría* y podrá crear una tecla de acceso rápido para ella.
2. En el cuadro de lista *Comandos* seleccione el comando al que desea asignar una tecla de acceso rápido nueva o el comando cuya tecla de acceso rápido desea cambiar.
3. Haga clic dentro del cuadro *Pulsar tecla de acceso rápido nueva* y pulse la tecla de acceso rápido que desea asignar al comando. La tecla de acceso rápido aparece en el cuadro *Pulsar tecla de acceso rápido nueva*. Si la tecla de acceso rápido no se asignó todavía a ningún comando, se habilita el botón **Asignar**. Si la tecla ya se asignó a un comando, el comando aparece debajo del cuadro y el botón **Asignar** está deshabilitado. (Para borrar el contenido del cuadro *Pulsar tecla de acceso rápido nueva* pulse **Ctrl**, **Alt** o **Mayús**).
4. Haga clic en el botón **Asignar**. La tecla de acceso rápido aparece ahora en el cuadro de lista *Teclas actuales*. Puede asignar varias teclas de acceso rápido al mismo comando si lo desea.
5. Para confirmar los cambios pulse el botón **Cerrar**.

## Eliminar una tecla de acceso rápido.

No se puede asignar la misma tecla de acceso rápido a comandos diferentes. Si desea eliminar una tecla de acceso rápido, selecciónela en el cuadro *Teclas actuales* y después pulse el botón **Quitar**.

## Establecer combinación de teclas para

Actualmente solo se pueden establecer combinaciones de teclas predeterminadas.

## Teclas de acceso rápido predeterminadas

La tabla que aparece a continuación enumera las teclas de acceso rápido asignadas a los comandos más utilizados en el momento de la instalación. En la aplicación puede ver una lista de los comandos, su descripción y sus teclas de acceso rápido en el cuadro de diálogo "Mapa de teclado" ([Ayuda | Mapa de teclado](#)<sup>298</sup>).

### Teclas de función (también para validación y transformación)

<b>F1</b>	Menú Ayuda
<b>F1 + Alt</b>	Abrir último archivo
<b>F3</b>	Buscar siguiente
<b>F4 + CTRL</b>	Cerrar ventana activa
<b>F4 + Alt</b>	Cerrar Authentic Desktop
<b>F5</b>	Actualizar
<b>F6 + CTRL</b>	Recorrer ventanas abiertas
<b>F7</b>	Comprobar formato XML
<b>F8</b>	Validar
<b>F10</b>	Transformación XSL
<b>F10 + CTRL</b>	Transformación XSL:FO

### Comandos de archivo y de aplicación

<b>Alt + F1</b>	Abrir último archivo
<b>CTRL + O</b>	Abrir archivo
<b>CTRL + N</b>	Archivo nuevo
<b>CTRL + P</b>	Imprimir archivo
<b>CTRL + S</b>	Guardar archivo
<b>CTRL + F4</b>	Cerrar ventana activa
<b>CTRL + F6</b>	Recorrer ventanas abiertas
<b>CTRL + TAB</b>	Cambiar de documento abierto
<b>Alt + F4</b>	Cerrar Authentic Desktop

### Otras teclas

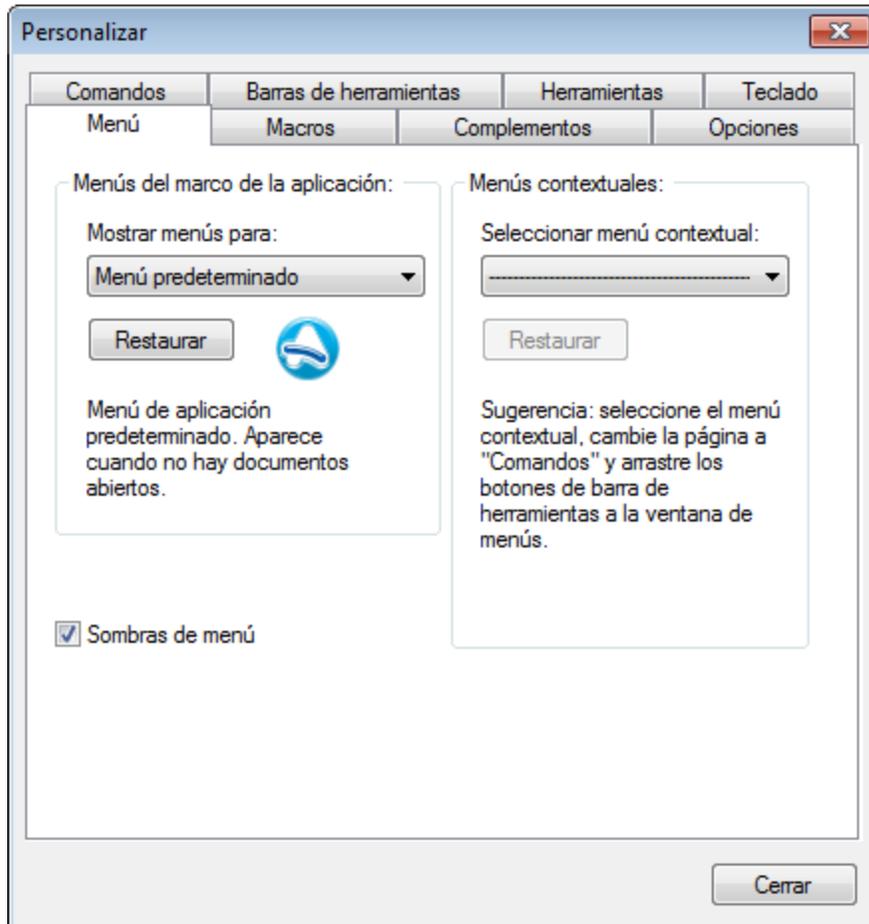
<b>Teclas de flecha arriba/abajo</b>	Mover el cursor o la barra de selección
<b>Esc</b>	Abandonar cambios o cerrar cuadro de diálogo
<b>Entrar</b>	Confirmar selección
<b>Supr</b>	Eliminar carácter o elemento seleccionado
<b>Mayús + Supr</b>	Cortar

☐ *Comandos de edición*

<b>CTRL + A</b>	Seleccionar todo
<b>CTRL + F</b>	Buscar
<b>CTRL + G</b>	Ir a la línea o carácter
<b>CTRL + H</b>	Reemplazar
<b>CTRL + V</b>	Pegar
<b>CTRL + X</b>	Cortar
<b>CTRL + Y</b>	Rehacer
<b>CTRL + Z</b>	Deshacer

### 14.9.8.5 Menú

En la pestaña **Menú** puede personalizar las dos barras de menú principales (la barra de menú predeterminada y la barra de menú de la aplicación) así como los menús contextuales de la aplicación.



## Personalizar la barra de menú predeterminada y la barra de menú de la aplicación

La barra de menú predeterminada es la barra de menú que aparece cuando no hay ningún documento abierto en la ventana principal. La barra de menú de la aplicación es la barra que aparece cuando hay un documento abierto en la ventana principal. Cada una de estas barras de menú se puede personalizar y los cambios realizados en una de las barras de menú no afecta a la otra.

Para personalizar una barra de menú, selecciónela en el cuadro combinado *Mostrar menús para:* de la pestaña *Menú* (imagen anterior). Después cambie a la [pestaña Comandos](#) del cuadro de diálogo *Personalizar* y arrastre comandos desde el cuadro de lista *Comandos* hasta la barra de menú.

## Eliminar comandos de menús y restaurar las barras de menú

Para **eliminar** un menú entero o un comando de un menú:

1. En el panel *Menús del marco de la aplicación* seleccione *Menú predeterminado* (que muestra los menús disponibles aunque no haya ningún documento abierto) o *XMLSpy* (que muestra los menús disponibles si hay al menos un documento abierto).
2. Con el cuadro de diálogo "Personalizar" abierto, seleccione (i) el menú que quiere eliminar de la barra de herramientas de la aplicación o (ii) el comando que quiere eliminar de uno de estos menús.

3. Ahora puede (i) arrastrar el menú para eliminarlo de la barra de herramientas o arrastrar el comando de menú para eliminarlo del menú, o bien (ii) hacer clic con el botón derecho en el menú o en el comando de menú y seleccionar **Eliminar**.

Para restaurar estas dos barras de menú (la barra de menú predeterminada y la de la aplicación) a su estado original de instalación seleccione el menú en el cuadro combinado *Mostrar menús para:* y haga clic en el botón **Restaurar** situado bajo el cuadro combinado.

### Personalizar los menús contextuales de la aplicación

Los menús contextuales son los menús que aparecen cuando se hace clic con el botón derecho en determinados objetos de la interfaz de la aplicación. Siga estos pasos para personalizar un menú contextual:

1. Seleccione el menú contextual en el cuadro combinado *Seleccionar menú contextual*. Aparece el menú contextual.
2. Pase a la [pestaña Comandos del cuadro de diálogo Personalizar](#).
3. Arrastre un comando del cuadro de lista *Comandos* al menú contextual.
4. Si desea eliminar un comando del menú contextual, haga clic en él con el botón derecho y seleccione **Eliminar**. También puede seleccionar el comando y arrastrarlo fuera del menú contextual.

Para restaurar un menú contextual a su estado original de instalación seleccione el menú en el cuadro combinado *Seleccionar menú contextual* y después pulse el botón **Restaurar**, situado bajo el cuadro combinado.

### Sombras de menú

Marque la casilla *Sombras de menú* para dar sombra a todos los menús.

## 14.9.8.6 Macros

En la pestaña **Macros** puede crear comandos de aplicación para macros creadas con el editor de script de Authentic Desktop. Más tarde puede incluir estos comandos de aplicación (que ejecutan las macros asociadas a ellos) en menús y barras de herramientas, bien desde la pestaña **Macros** directamente o bien con los mecanismos disponibles en la [pestaña Comandos del cuadro de diálogo Personalizar](#). Al ser comandos de aplicación, también pueden tener asignadas teclas de acceso rápido (desde la [pestaña Teclado del cuadro de diálogo Personalizar](#).)

### Cómo funcionan las macros en Authentic Desktop

En Authentic Desktop las macros funcionan así:

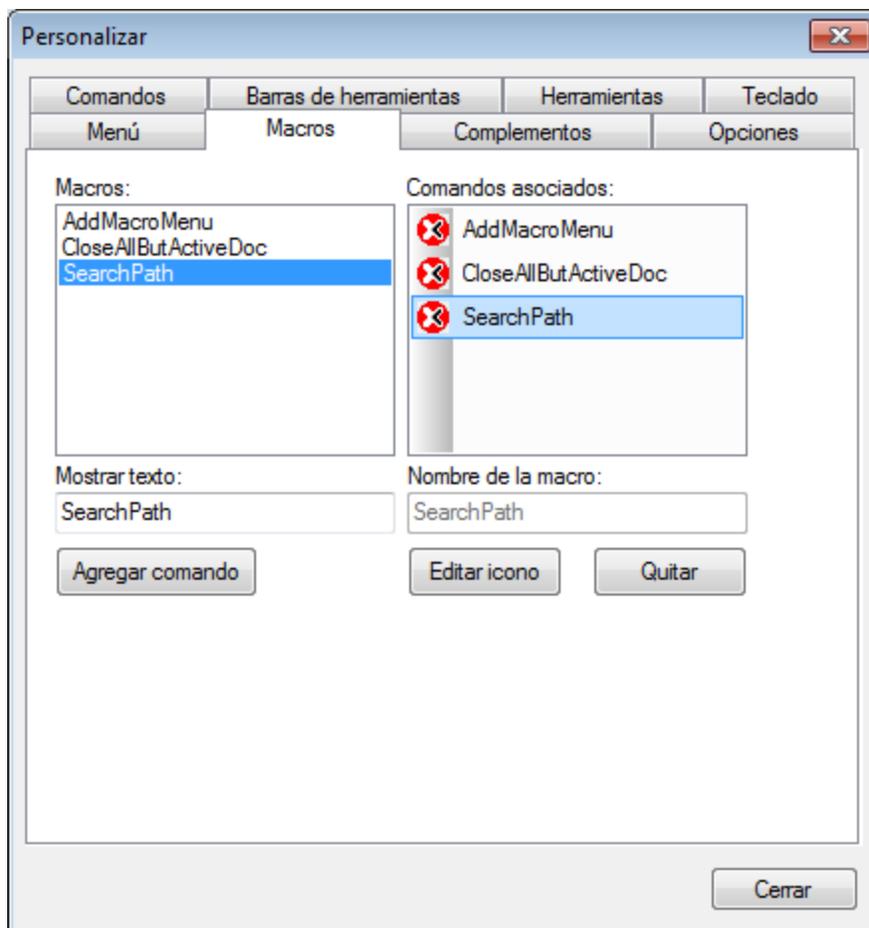
- Los proyectos de script de Altova (archivos `.asprj`) se crean en el [editor de script](#) de Authentic Desktop. Estos proyectos de script contienen las macros que se utilizan en Authentic Desktop.
- En Authentic Desktop puede haber dos proyectos de script activos al mismo tiempo: (i) un proyecto de script de aplicación, que se especifica en la [sección Scripting del cuadro de diálogo "Opciones"](#) y (ii) el proyecto de script del [proyecto de Authentic Desktop](#), que se especifica en el cuadro de diálogo [Script \(Proyecto | Configuración de script\)](#).
- Las macros de estos dos proyectos de script están disponibles en la aplicación en el menú **Herramientas | Macros** (desde donde puede ejecutarlas) y en la pestaña *Macros* del cuadro de diálogo Personalizar (*imagen siguiente*), donde se pueden configurar como comandos de aplicación.

Después de configurar una macro como comando de aplicación, el comando se puede colocar en un menú o barra de herramientas.

### Cómo crear un comando de aplicación para una macro

En el [editor de script](#) <sup>306</sup> ([Herramientas | Editor de script](#) <sup>306</sup>) cree la macro que desea utilizar y guárdela en un proyecto de script. Después defina esta proyecto de script como proyecto de script de aplicación (en la [sección](#) <sup>290</sup> [Scripting](#) <sup>290</sup> [del cuadro de diálogo "Opciones"](#) <sup>290</sup>) o como proyecto de script del proyecto activo (en el cuadro de diálogo [Script](#) <sup>226</sup> del proyecto ([Proyecto | Configuración de script](#) <sup>226</sup>)). A continuación aparecen las macros del proyecto de script en el panel *Macros* de la pestaña *Macros* (*imagen siguiente*).

Para crear un comando de aplicación para una macro, seleccione la macro en el panel *Macros*, escriba el texto del comando en el cuadro *Mostrar texto* y pulse el botón **Agregar comando** (*imagen siguiente*). A continuación se añade a la lista *Comandos asociados* un comando para la macro seleccionada.



Para editar el icono de un comando asociado, seleccione el comando y pulse el botón **Editar icono**. Para eliminarlo, seleccione el comando asociado y pulse **Quitar**.

### Cómo poner un comando asociado a una macro en un menú o barra de herramientas

Hay dos maneras de poner un comando asociado a una macro en un menú o en una barra de herramientas:

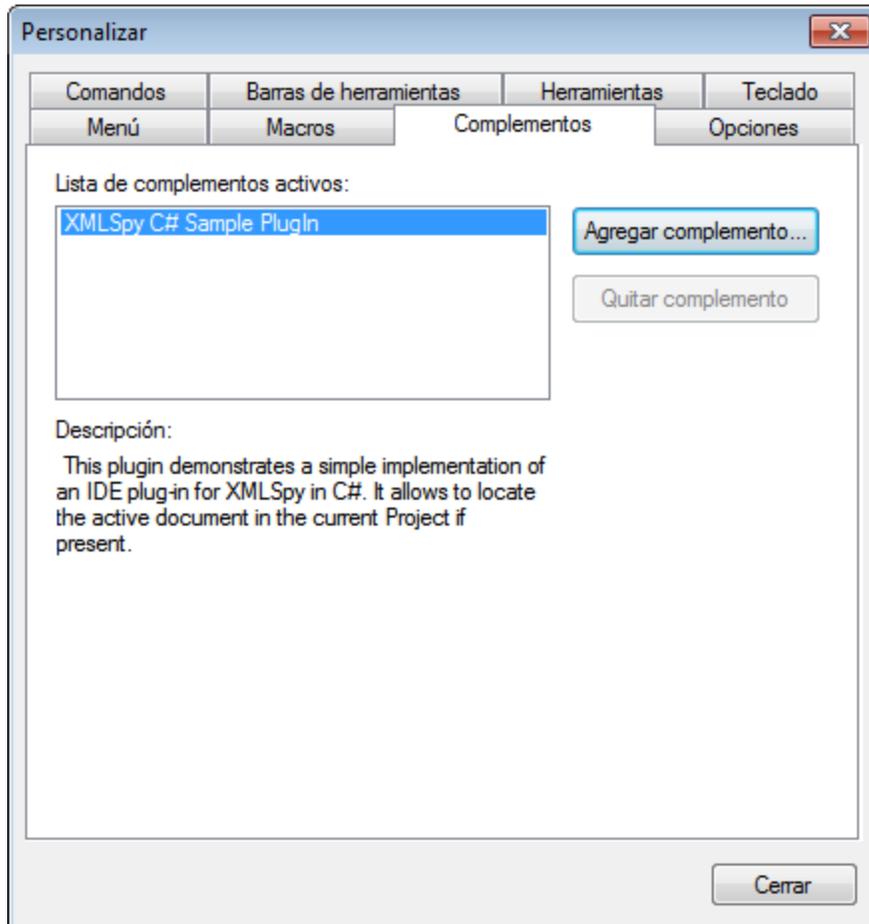
- Arrastrando el comando desde el cuadro de lista *Comandos asociados* hasta la posición deseada en el menú o barra de herramientas de destino.
- Usando los mecanismos disponibles en la [pestaña Comandos del cuadro de diálogo Personalizar](#).

En ambos casos se crea el comando en la posición deseada del menú o de la barra de herramientas y al hacer clic en el comando se ejecuta la macro.

**Nota:** si definió una macro como comando asociado, también puede asignarle una [tecla de acceso rápido](#). En la [pestaña Teclado del cuadro de diálogo Personalizar](#), seleccione la opción *Macros* en el cuadro combinado *Categoría*. Después seleccione la macro y defina su tecla de acceso rápido. Para poder asignarle la tecla antes debe definir la macro como comando asociado.

### 14.9.8.7 Complementos

Desde la pestaña **Complementos** puede integrar complementos y colocar comandos (si se han programado) en un menú o en una barra de herramientas de la aplicación. Pulse el botón **Agregar complemento** de la pestaña *Complementos* (*imagen siguiente*) y busque el archivo DLL del complemento (ver más abajo el párrafo *Cómo crear complementos*). Haga clic en **Aceptar** para agregar el complemento. Si quiere, puede agregar varios complementos.



En el cuadro de diálogo aparece una descripción de los complementos añadidos y se habilita el botón **Quitar complemento**. Si el código del complemento crea barras de herramientas y menús, estos aparecen inmediatamente en la interfaz de la aplicación. Si desea quitar un complemento, selecciónelo y pulse el botón **Quitar complemento**.

### Cómo crear complementos

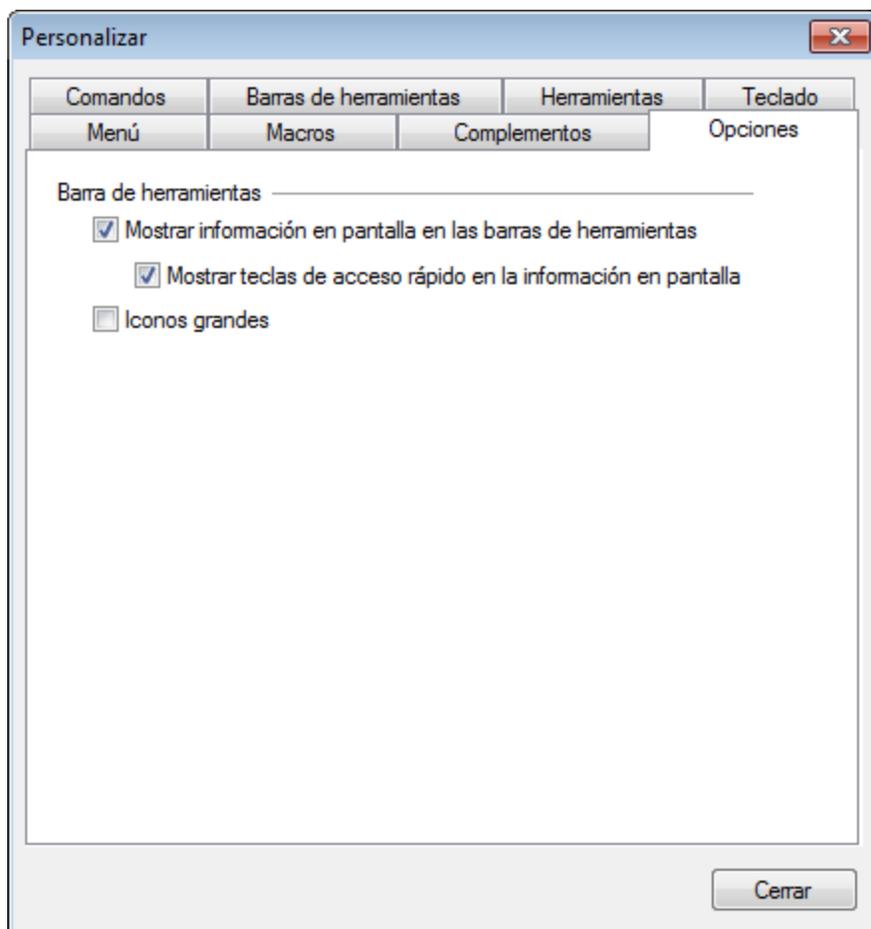
En la carpeta `Examples\IDEPlugin` de la carpeta [\(Mis\) Documentos](#)<sup>12</sup> de la aplicación encontrará código fuente para crear ejemplos de complementos en Authentic Desktop. Siga estas instrucciones para crear un complemento a partir de este código fuente:

1. Abra la solución que desea crear como complemento en Visual Studio.
2. Cree el complemento con el comando del menú **Compilación**.
3. El archivo DLL del complemento se creará en la carpeta `Bin` o `Debug`. Este archivo DLL es el que se debe añadir como complemento (ver instrucciones más arriba).

Para más información consulte la sección [Complementos para entornos IDE](#)<sup>334</sup>.

### 14.9.8.8 Opciones

En la pestaña **Opciones** puede definir la configuración general del entorno.

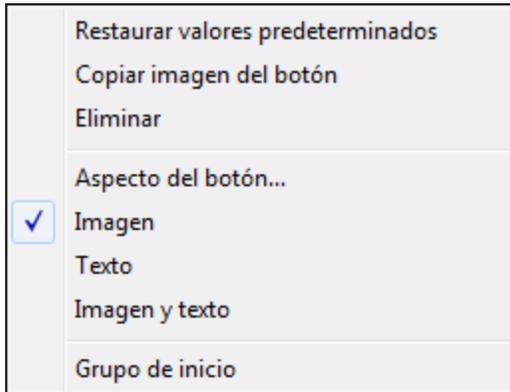


Marque las casillas para activar estas opciones:

- *Mostrar información en pantalla en las barras de herramientas*: al pasar el puntero del ratón sobre los iconos de las barras de herramientas aparece un mensaje emergente con una breve descripción de la función del icono, así como su tecla de acceso rápido (si le asignó una y si marcó la casilla *Mostrar teclas de acceso rápido*).
- *Mostrar teclas de acceso rápido en la información en pantalla*: la información en pantalla de los iconos de las barras de herramientas incluye la tecla de acceso rápido del comando.
- *Iconos grandes*: alterna el tamaño de los iconos de las barras de herramientas entre tamaño estándar y tamaño grande.

### 14.9.8.9 Menú contextual Personalizar

El menú contextual **Personalizar** (*imagen siguiente*) es el menú que aparece al hacer clic con el botón derecho en un menú, un comando de menú o en un icono de barra de herramientas de la aplicación mientras está abierto el cuadro de diálogo Personalizar.

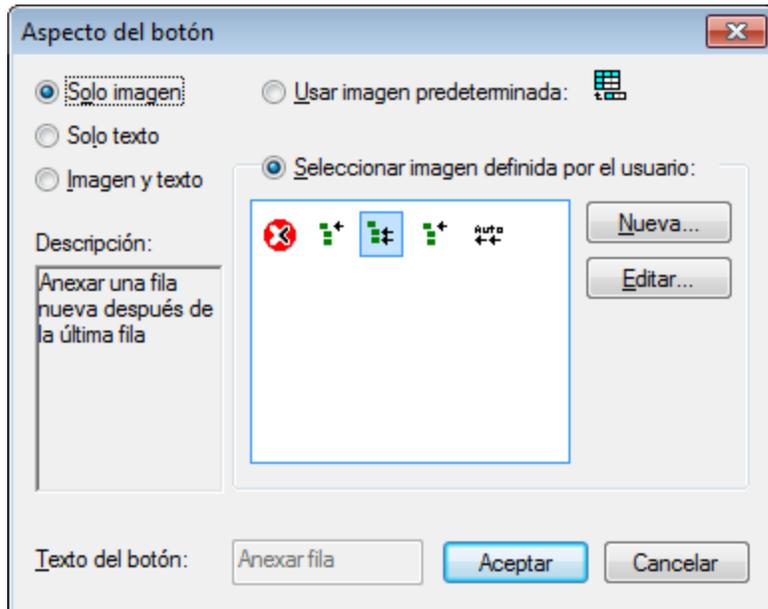


Estas son las opciones del menú contextual **Personalizar**:

- *Restaurar valores predeterminados* (por ahora no tiene función alguna).
- *Copiar imagen del botón*: copia la imagen del icono en el portapapeles.
- *Eliminar*: elimina el menú, comando o icono seleccionado. Para más información sobre cómo restaurar los componentes eliminados lea el último párrafo de este apartado.
- *Aspecto del botón*: abre el cuadro de diálogo "Aspecto del botón" (*imagen siguiente*), donde puede configurar las propiedades que definen el aspecto del icono seleccionado. Consulte el resto de este apartado para más información.
- *Imagen*, *Texto*, *Imagen y texto*: estas tres opciones son mutuamente excluyentes y determinan si el icono seleccionado debe ser solamente icono, solamente texto o ambos. Esto también se puede definir en el cuadro de diálogo "Aspecto del botón".
- *Grupo de inicio*: inserta un divisor de grupo vertical a la izquierda del icono seleccionado. Como resultado el icono seleccionado se convierte en el primer icono del grupo.

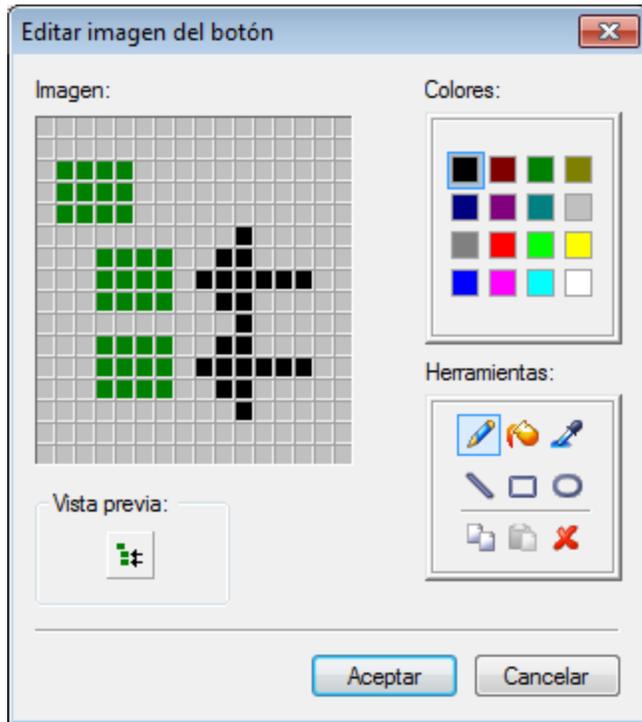
#### El cuadro de diálogo "Aspecto del botón"

Si hace clic con el botón derecho en un icono de una barra de herramientas (mientras está abierto el cuadro de diálogo Personalizar) y después selecciona **Aspecto del botón**, aparece el cuadro de diálogo "Aspecto del botón" (*imagen siguiente*). En este cuadro de diálogo puede editar la imagen del icono y su texto. Actualmente sólo puede editar iconos de macros y complementos en este cuadro de diálogo.



Estas son las funciones de edición disponibles en este cuadro de diálogo para el icono seleccionado:

- *Solo imagen*, *Solo texto* o *Imagen y texto*: seleccione uno de estos tres botones de opción para definir el formato del icono de la barra de herramientas.
- *Edición de la imagen*: si selecciona la opción *Solo imagen* o *Imagen y texto*, se habilitan las opciones de edición de la imagen. Pulse el botón **Nueva** para crear una imagen nueva, que se añadirá al panel de imágenes definidas por el usuario. Después seleccione una imagen y pulse **Editar**.



- *Selección de la imagen:* seleccione una imagen del panel de imágenes definidas por el usuario y después pulse **Aceptar** para usar la imagen seleccionada como icono de la barra de herramientas.
- *Edición y selección del texto:* si selecciona la opción *Solo texto* o *Imagen y texto*, se habilita el cuadro de texto *Texto del botón*. Escriba el texto y haga clic en **Aceptar** para usar este texto con el icono de la barra de herramientas.

**Nota:** el cuadro de diálogo "Aspecto del botón" también se puede usar para editar el texto de los comandos de menú. Haga clic con el botón derecho en el comando de menú (cuando está abierto el cuadro de diálogo Personalizar), seleccione **Aspecto del botón** y edite el texto del comando de menú en el cuadro de texto *Texto del botón*.

## Restaurar menús, comandos e iconos de barras de herramientas eliminados

Si elimina un menú, comando o icono de barra de herramientas con el comando **Eliminar** del menú contextual Personalizar, puede restaurarlos siguiendo estos pasos:

- *Menús:* seleccione [Herramientas | Personalizar | Menú](#) <sup>268</sup> y haga clic en el botón **Restaurar** del panel *Menús del marco de la aplicación*. Otra opción es seleccionar [Herramientas | Personalizar | Barras de herramientas](#) <sup>262</sup>, marcar la casilla *Barra de menús* y pulsar el botón **Restaurar**.
- *Comandos de menú:* seleccione [Herramientas | Personalizar | Comandos](#) <sup>260</sup> y arrastre el comando desde el cuadro de lista *Comandos* hasta el menú.
- *Iconos de barra de herramientas:* seleccione [Herramientas | Personalizar | Comandos](#) <sup>260</sup> y arrastre el comando desde el cuadro de lista *Comandos* hasta la barra de herramientas.

## 14.9.9 Restaurar barras de herramientas y ventanas

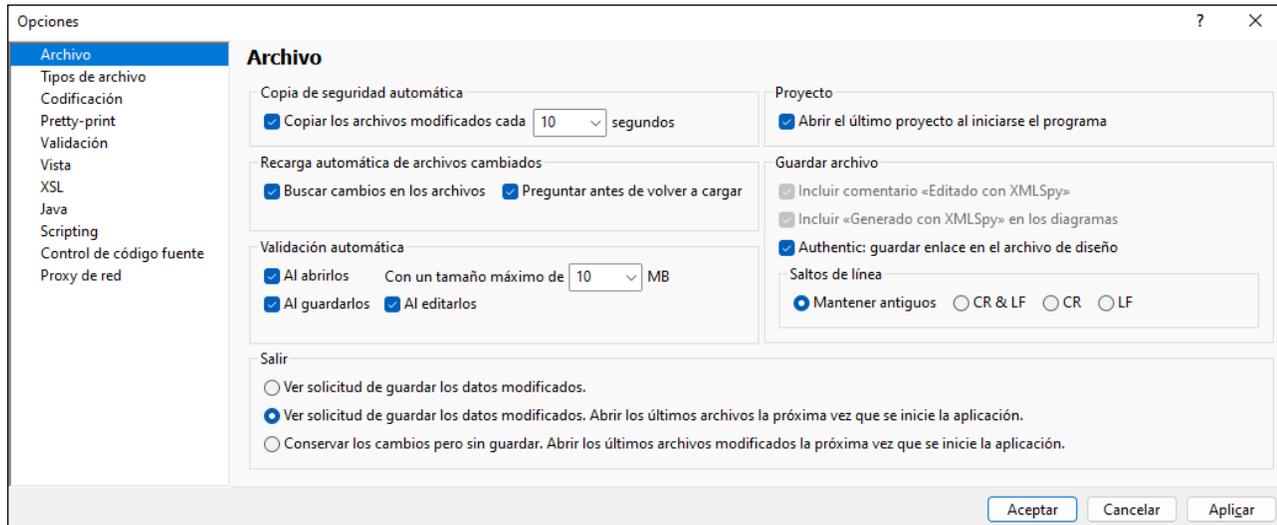
El comando **Restaurar barras de herramientas y ventanas** cierra Authentic Desktop y lo reinicia con su configuración predeterminada. Antes de cerrarse, Authentic Desktop le pregunta si desea cerrar o no la aplicación (*imagen siguiente*).



Este comando es muy práctico si movió ventanas o barras de herramientas de sitio, si las ocultó o si ajustó su tamaño y desea poner todas estas barras de herramientas y ventanas como estaban en un principio.

## 14.9.10 Opciones

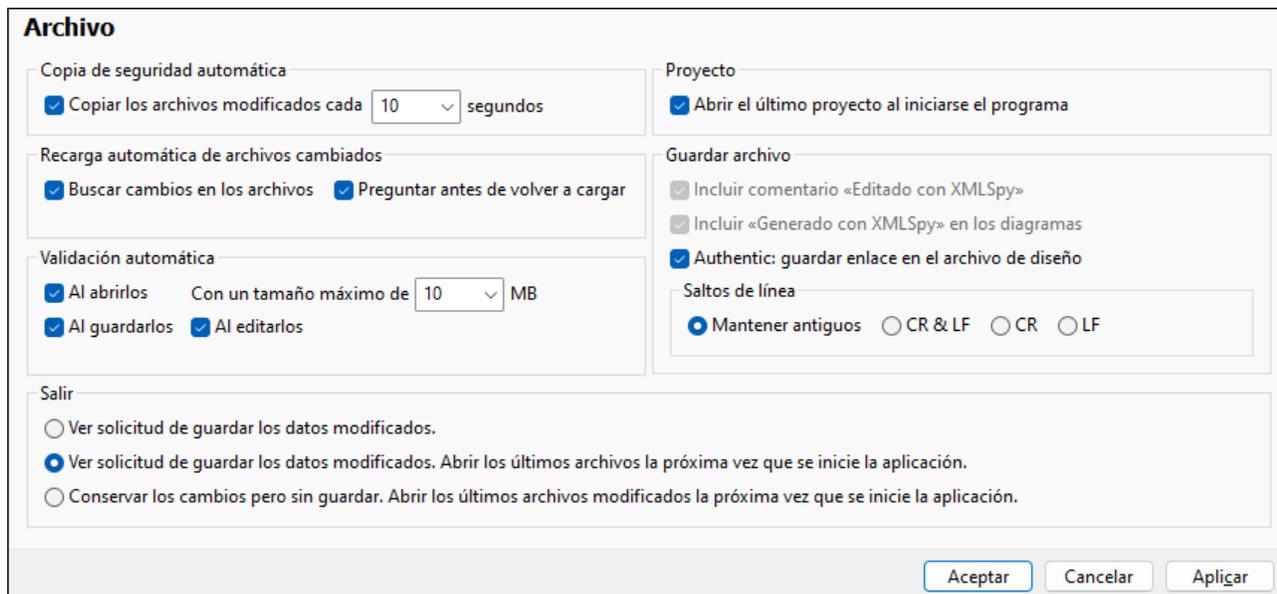
El comando **Herramientas | Opciones** sirve para definir opciones de configuración globales de la aplicación. Estas opciones están organizadas en secciones (*ver panel izquierdo de la imagen siguiente*). Por ejemplo, la sección **Archivo** <sup>279</sup> (*imagen siguiente*) contiene opciones que especifican cómo se deben abrir y guardar archivos en Authentic Desktop. Para definir opciones de configuración de una determinada sección, haga clic en dicha sección en el panel izquierdo del cuadro de diálogo y especifique los valores correspondientes. Al hacer clic en el botón **Aceptar** se guardan los cambios en el registro y se cierra el cuadro de diálogo. El botón **Aplicar** efectúa los cambios en los documentos que están abiertos.



En los apartados siguientes se describen todas las secciones del cuadro de diálogo "Opciones".

### 14.9.10.1 Archivo

La sección **Archivo** define cómo se abren y guardan los documentos en Authentic Desktop. La sección [Codificación](#) <sup>283</sup> incluye opciones de configuración relacionadas.



#### Copia de seguridad automática

Si esta opción está activada, la aplicación guarda automáticamente una copia de seguridad de los archivos que edite. Puede seleccionar en el cuadro combinado con qué frecuencia se guarda la copia de seguridad de

entre 5 y 60 segundos o introducir un valor de hasta 300 segundos. Para más información consulte la sección [Copia de seguridad automática](#)<sup>69</sup>.

## Recarga automática de archivos cambiados

Si está trabajando en un entorno de varios usuarios o si trabaja con archivos que se generan de forma dinámica en un servidor, puede vigilar si se realizan cambios en los archivos que están abiertos en la interfaz. Cada vez que Authentic Desktop detecte un cambio en un documento de los que están abiertos, la aplicación le pregunta si desea volver a cargar el archivo en el que se realizaron los cambios.

## Validación automática

Si usa archivos DTD o esquemas para definir la estructura de sus documentos XML, puede comprobar automáticamente la validez del documento:

- Cada vez que lo abra, si el archivo no supera el límite de MB que haya establecido
- Cada vez que lo guarde
- Mientras edita el archivo.

Si el documento no es válido, aparece un mensaje de error. Si es válido no aparece ningún mensaje y la operación de lleva a cabo sin ninguna notificación.

## Proyecto

Cuando inicie Authentic Desktop, puede abrir automáticamente el último proyecto que utilizó.

## Guardar archivo

Cuando guarda un documento XML, Authentic Desktop incluye un breve comentario (`<!-- Edited with Authentic Desktop http://www.altova.com -->`) casi al principio del archivo. Esta opción solamente se puede deshabilitar si el usuario compró una licencia del producto y afecta a los archivos que se editan o guardan en las vistas Cuadrícula y Esquema.

Cuando guarda un diagrama del modelo de contenido (**Diseño de esquemas | Generar documentación**), Authentic Desktop incluye el logotipo de Authentic Desktop en la documentación. Esta opción solamente se puede deshabilitar si el usuario compró una licencia del producto.

Si el archivo XML tiene un archivo SPS asociado y marca la casilla *Authentic: guardar enlace al archivo de diseño*, el enlace al archivo SPS se guardará en el archivo XML.

## Saltos de línea

Cuando abra un archivo, la codificación de caracteres para los saltos de línea del archivo se mantiene si selecciona la opción *Mantener antiguos*. Si lo prefiere, puede codificar los saltos de línea con una de estas tres codificaciones: **CR&LF** (para PC), **CR** (para MacOS) o **LF** (para Unix).

## Modo de salida

Aquí puede definir qué ocurre con los archivos abiertos al cerrar Authentic Desktop. Hay varias opciones:

- *Ver solicitud de guardar los datos modificados:* si un archivo abierto contiene cambios sin guardar, el programa le preguntará si quiere guardarlos. Según la respuesta que dé, el archivo se guarda o no y después la aplicación se cierra.
- *Ver solicitud de guardar los datos modificados. Abrir los últimos archivos la próxima vez que se inicie la aplicación:* aparece el cuadro de diálogo "Guardar" para los archivos abiertos que contienen cambios sin guardar. Puede elegir si quiere o no guardar esos cambios. Cuando vuelva a abrir la aplicación, los archivos que estaban abiertos al cerrarla se volverán a abrir. (Si no se guardan los cambios realizados antes de salir de la aplicación, estos se perderán.)
- *Conservar los cambios pero sin guardar. Abrir los últimos archivos modificados la próxima vez que se inicie la aplicación:* la aplicación se cierra directamente sin guardar ningún cambio. Al volver a abrirla, todos los archivos que estaban abiertos se vuelven abrir y contienen los cambios no guardados. Básicamente es como si retomara el trabajo donde lo dejó.

Al salir de la aplicación por primera vez, verá el cuadro de diálogo del modo de salida, donde podrá escoger el comportamiento que mejor le convenga. En adelante puede cambiar estas opciones en la sección Archivo del cuadro de diálogo "Opciones".

## Guardar y salir

Cuando termine de modificar las opciones de configuración, haga clic en **Aceptar** para terminar.

### 14.9.10.2 Tipos de archivo

En la sección **Tipos de archivo** puede adaptar a sus requisitos el comportamiento de Authentic Desktop dependiendo del tipo de archivo. Elija un tipo de archivo en el panel Tipos de archivo y después personalice las funciones para ese tipo de archivo concreto tal y como se explica a continuación. Observe que hay dos entradas especiales en la lista Tipos de archivo:

- *<predeterminado>* se puede usar para indicar qué tratamiento quiere dar a los archivos que tengan una extensión que no esté incluida en la lista de tipos de archivos.
- *<ninguno>* se puede usar para indicar qué tratamiento quiere dar a los archivos que no tengan ninguna extensión.

## Configuración del explorador de Windows

Puede definir la descripción del tipo de archivo y el tipo de contenido MIME utilizado por Windows Explorer. También puede elegir si Authentic Desktop es el editor predeterminado para los archivos de este tipo.

## Conformidad

Authentic Desktop ofrece funciones de edición inteligente y otras funciones para distintos tipos de archivo. Authentic Desktop determina las características propias para cada tipo de archivo dependiendo de la configuración de conformidad elegida en este grupo de opciones. La configuración predeterminada de XMLSpy abarca un gran número de tipos de archivo. Por eso no recomendamos cambiar esta configuración predeterminada a no ser que sea para añadir tipos de archivo nuevos o si necesita asignar otra clase de conformidad a un tipo de archivo determinado.

### Vista predeterminada

En este panel puede definir la vista predeterminada de cada tipo de archivo. Si un tipo de archivo conforme con determinada especificación solamente se puede ver en una vista, entonces dicha vista estará seleccionada por defecto y la selección de vista se deshabilita.

### Vista Cuadrícula

Aquí puede elegir si la vista Cuadrícula genera tablas automáticamente o no.

### Vista Texto

Aquí puede elegir si se usa color de sintaxis o no.

### Deshabilitar validación automática

Si marca esta casilla, se deshabilita la validación automática del tipo de archivo seleccionado. Por lo general, la validación automática se produce al abrir o guardar el archivo o cuando se cambia de vista.

### Agregar extensión de archivo nueva

Añade un tipo de archivo nuevo a la lista *Tipos de archivo*. Después debe definir la configuración de este tipo de archivo nuevo usando las demás opciones de esta pestaña.

### Eliminar extensión de archivo seleccionada

Elimina el tipo de archivo seleccionado y todas sus opciones de configuración.

### Guardar y salir

Cuando la configuración esté lista, haga clic en **Aceptar** para terminar.

### 14.9.10.3 Codificación

En la sección **Codificación** puede configurar las opciones de codificación de archivos.

#### Codificación

Codificación predeterminada para archivos XML nuevos

Unicode UTF-8

Orden de bytes little-endian

Orden de bytes big-endian

Marca BOM

Siempre crear marca BOM (excepto para UTF-8)

Al guardar, conservar la marca BOM detectada

Abrir archivos XML con codificación desconocida como

Unicode UTF-8

Abrir archivos no XML en

Codepage 1252 (Western)

#### Codificación predeterminada para archivos XML nuevos

En esta lista desplegable puede elegir la codificación predeterminada para los archivos XML nuevos. Cada documento nuevo se crea con una declaración XML que contiene el valor de codificación elegido aquí. Si selecciona una codificación de dos o cuatro bytes como codificación predeterminada (es decir, UTF-16, UCS-2 o UCS-4) también puede elegir entre el orden de bytes little-endian o big-endian.

La codificación de los archivos XML ya existentes se conserva y solamente se puede cambiar con el comando [Archivo | Codificación](#)<sup>184</sup>.

#### Abrir archivos XML con codificación desconocida como

Si no se puede determinar la codificación de un archivo XML o si el documento XML no especifica ninguna codificación, el archivo se abre con la codificación que elija en este cuadro combinado.

#### Abrir archivos no XML en

Los archivos no XML nuevos y los ya existentes se abren con la codificación que elija en este cuadro combinado. Puede cambiar la codificación del documento con el comando [Archivo | Codificación](#)<sup>184</sup>.

#### Marca BOM (orden de bytes)

Los documentos que tienen una codificación de caracteres de dos o cuatro bytes se pueden guardar:

- con un orden de bytes little-endian y una marca BOM little-endian (opción *Siempre crear marca BOM (excepto para UTF-8)*) o
- con el orden de bytes detectado y la marca BOM detectada (opción *Al guardar, conservar la marca BOM detectada*).

## Guardar y salir

Cuando termine de modificar las opciones de configuración, haga clic en **Aceptar** para terminar.

### 14.9.10.4 Pretty Printing

La sección **Pretty Print** permite indicar cómo aparece el texto en el documento XML representado en la Vista Authentic. Aunque Authentic Desktop no muestra el documento XML en forma de texto, las opciones que configure en esta sección se aplicarán al texto sin formato del archivo XML que está editando en la Vista Authentic.

Las opciones son:

- Puede configurar si quiere usar el tabulador o espacios.
- Puede decidir cómo se escriben los elementos vacíos. Un elemento que se cierra automáticamente combina las etiquetas de apertura y cierre en una sola: `<element/>` or `<element />`.

## Guardar y salir

Cuando la configuración esté lista, haga clic en **Aceptar** para terminar.

### 14.9.10.5 Validación

En la opción **Validación** puede configurar las opciones de validación de documentos XML y JSON.

#### Validación

XML

Almacenar archivos DTD o de esquema en memoria

Versión del esquema

Versión 1.1 si `<xs:schema vc:minVersion="1.1" ... >`  
De lo contrario, versión 1.0

Versión 1.1 siempre

Versión 1.0 siempre

Límites del mensaje

Errores:

Inconsistencias:

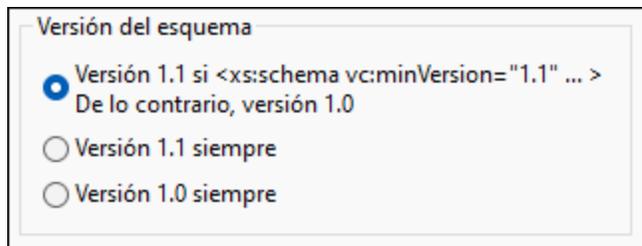
Advertencias:

## XML

Authentic Desktop puede almacenar en su memoria caché archivos DTD y XML Schema para no tener que cargarlos continuamente (por ejemplo, si el esquema no es local sino que se accede a él con un URL). Sin embargo, recuerde que si usa las versiones de la memoria caché de los esquemas y hace cambios en un esquema, estos no se reflejan de inmediato en la validación, sino que necesita volver a cargar el archivo XML o reiniciar Authentic Desktop.

### Versión del esquema

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



En definitiva pueden darse todas las situaciones que aparecen en la siguiente tabla. Con *Versión de XML Schema* nos referimos a la opción seleccionada en el panel de opciones Versión de XML Schema. La segunda columna (atributo `vc:minVersion`) corresponde al valor del atributo `xs:schema/@vc:minVersion` del esquema XML.

Versión de XML Schema	Atributo <code>vc:minVersion</code>	Modo XSD
<i>Versión 1.0 siempre</i>	Ausente (o presente con cualquier valor)	1.0
<i>Versión 1.1 siempre</i>	Ausente (o presente con cualquier valor)	1.1
<i>Valor de @vc:minVersion</i>	El atributo tiene el valor 1.1	1.1
<i>Valor de @vc:minVersion</i>	Falta el atributo o está presente y tiene cualquier valor menos 1.1	1.0

## Límite de los mensajes

Estas opciones permiten establecer distintos límites para los errores, las inconsistencias XBRL y las advertencias que aparecen. El número predeterminado para cada una de estas categorías es 100. Puede cambiarlo al número que quiera.

## Guardar y salir

Cuando la configuración esté lista, haga clic en **Aceptar** para terminar.

## 14.9.10.6 Vista

En la sección **Vista** puede personalizar la presentación de los documentos XML en Authentic Desktop.

### Mostrar logotipo

Si desea que la aplicación se inicie más rápido, puede desactivar la pantalla de presentación que aparece al iniciarse el programa. Además, si compró una licencia del producto, tiene la opción de no incluir el logotipo del programa, el aviso de copyright ni la información de registro en los documentos que imprima desde Authentic Desktop.

### Título de la pantalla

El título de la pantalla de cada ventana de documento puede llevar el nombre del archivo solamente o el nombre completo de la ruta de acceso del archivo.

### Motor del navegador

El motor del navegador que se utiliza en la vista Authentic y la vista Explorador es actualmente Internet Explorer (IE), por lo que IE es el motor del navegador por defecto para estas dos vistas. Como alternativa, puede utilizar Microsoft Edge Web View 2 como motor para la vista Explorador. Si Edge no está instalado en su equipo, vaya a la [página de descarga de WebView2](#), desde donde puede instalar el Evergreen Bootstrapper. Esto le permitirá utilizar Microsoft Edge WebView2 como motor para la vista Explorador.

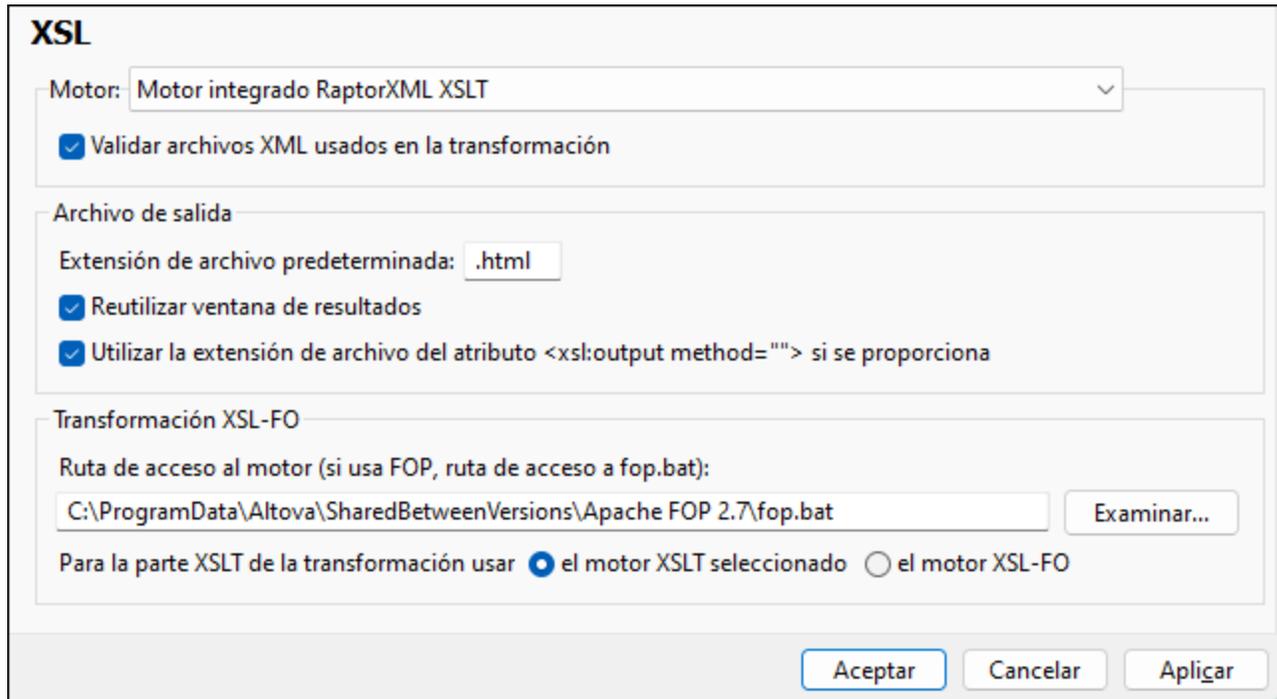
Consulte el tema [Vista Explorador](#)<sup>103</sup> para obtener más información.

### Guardar y salir

Cuando termine de modificar las opciones de configuración, haga clic en **Aceptar** para terminar.

## 14.9.10.7 XSL

En la sección **XSL** (*imagen siguiente*) puede definir las opciones de configuración para las [transformaciones XSL](#)<sup>287</sup> y las [transformaciones XSL-FO](#)<sup>289</sup> realizadas dentro de la aplicación.



## Configuración del motor

Puede configurar un procesador XSLT para ejecutar transformaciones XSLT cuando se invoca el comando [Transformación XSLT](#)<sup>232</sup>.

Hay varios motores XSLT de entre los que elegir:

- El motor XSLT integrado de RaptorXML
- El analizador MSXML de Microsoft (MSXML)
- Un procesador XSLT externo

**Nota:** para depurar XSLT en Authentic Desktop se usa siempre el motor XSLT integrado de RaptorXML, incluso aunque aquí se haya seleccionado otro motor XSLT para las transformaciones.

### Motor XSLT de Altova RaptorXML

Authentic Desktop incluye los motores XSLT 1.0, XSLT 2.0 y XSLT 3.0 de Altova Raptor XML para transformaciones XSLT. Dependiendo del valor del atributo `version` del elemento `xsl:stylesheet` o `xsl:transform`, en la transformación se utiliza un motor u otro. Ocurre lo mismo al depurar XSLT con el depurador XSLT/XQuery de XMLSpy.

Para validar los archivos XML de las transformaciones, seleccione la opción *Validar* (*imagen anterior*).

### Analizador (MSXML) de Microsoft

El analizador MSXML 3.0, 4.0 o 6.0 es una utilidad que viene preinstalada en su equipo. Si sabe qué versión utiliza su equipo, seleccione la opción correspondiente. Si no lo sabe, puede pedir a Authentic Desktop que

elija la versión automáticamente (*Elegir la versión automáticamente* es la opción predeterminada). En ese caso, Authentic Desktop intenta seleccionar la versión más reciente.

### Procesador XSLT externo

Puede elegir el procesador XSLT externo que usted prefiera introduciendo la ruta de acceso a su archivo ejecutable.

**XSL**

Motor: Programa externo de transformación XSL ▼

Escriba la línea de comando para ejecutar un programa externo de transformación XSL de la siguiente forma

Program.exe %1 %2 %3

donde %1 es el nombre del archivo XML de entrada, %2 es el nombre del archivo de salida y %3 (opcional) es el nombre del archivo de la hoja de estilos XSL. También puede añadir cualquier otro parámetro que requiera el programa externo.

c:\MyEngine.exe o %2 %1 %3 date=2023 Examinar...

Mostrar los resultados del programa externo en la ventana de mensajes tras la transformación

Mostrar errores del programa externo en la ventana de mensajes tras la transformación

Importante: para depurar XSLT se usa siempre el motor integrado.

Para ello basta con indicar la cadena de línea de comandos que el procesador XSLT externo utiliza para ejecutar transformaciones. Estas son las variables que puede usar para crear la cadena de línea de comandos:

- %1 = documento XML que se debe procesar
- %2 = archivo de salida que se debe generar
- %3 = hoja de estilos XSLT que se debe usar (si el documento XML no contiene una referencia a la hoja de estilos)

Por ejemplo, imagine que tiene un procesador que utiliza este comando para ejecutar transformaciones XSLT:

```
mimotorxslt.exe -o salida.xml entrada.xml hojaestilos.xslt nombre-parámetro=valor-parámetro
```

Entonces en Authentic Desktop puede usar las variables descritas más arriba para generar este comando. Seleccione el botón de opción *Programa externo de transformación XSL* y escriba esta línea en el cuadro de texto:

```
c:\mimotorxslt\mimotorxslt.exe -o %2 %1 %3 nombre-parámetro=valor-parámetro
```

Por último, si desea ver los resultados y los errores del programa externo en la ventana Mensajes de Authentic Desktop marque las casillas correspondientes.

Para ver los mensajes de salida y de error del programa externo en la ventana Mensajes de Authentic Desktop marque la casilla correspondiente.

**Nota:** los parámetros definidos en el cuadro de diálogo Parámetros de entrada XSLT (al que se accede desde el menú XSL) solo se pasan a los motores XSLT internos de Altova, pero no a otros motores XSLT aunque se hayan seleccionado como procesador XSLT predeterminado.

## Opciones del archivo de salida

Para configurar el archivo de salida tiene varias opciones:

- *Extensión de archivo predeterminada:* usa una extensión de archivo predeterminada para los archivos de salida; esta extensión se puede sobrescribir con la extensión indicada en el elemento XSLT `xsl:output` (último punto).
- Si marca la casilla *Reutilizar ventana de resultados*, las transformaciones que realice a partir de ahora mostrarán el documento de resultados en la misma ventana. Si el archivo XML pertenece a un proyecto y la casilla *Reutilizar ventana de resultados* está desactivada, la opción solamente surte efecto si también se desactiva la casilla *Guardar en la carpeta* de las [propiedades del proyecto](#)<sup>226</sup> (imagen siguiente).



- *Usar la extensión de archivo del elemento `xsl:output`:* puede decidir si la extensión de archivo indicada en el elemento `xsl:output` de la hoja de estilos XSLT sobrescribe la extensión predeterminada indicada en la primera opción.

## Transformaciones XSL-FO

Los documentos FO se procesan usando un procesador FO y la ruta de acceso del ejecutable del procesador FO debe indicarse en la casilla del motor de transformación XSL-FO. La transformación se lleva a cabo con el comando de menú [XSL/XQuery | Transformación XSL-FO](#)<sup>283</sup>. Si el archivo de entrada (el documento que estaba activo cuando se ejecutó el comando) es un documento XSL-FO, se invoca al procesador FO para la transformación. Si el documento de entrada es un documento XML, entonces es necesario realizar primero una transformación XSLT para convertir el documento XML en un documento XSL-FO. Esta transformación XSLT se puede llevar a cabo bien con el motor XSLT que indicó como motor predeterminado ([véase más arriba](#)<sup>287</sup>) o bien con el motor XSLT que está integrado en el procesador FO que indicó como procesador FO predeterminado para la aplicación. Seleccione cuál de estos dos motores desea usar para esta transformación.

**Nota:** a no ser que desactive la casilla correspondiente en el programa de instalación de Authentic Desktop, el procesador FOP de [Apache XML Project](#) se instalará en la carpeta `c:\ProgramData\Altova\SharedBetweenVersions`. Y si el procesador está instalado, su ruta de acceso aparecerá automáticamente en el campo *Motor XSL-FO*. Si quiere puede cambiar esta ruta de acceso para usar otro procesador FO distinto. Sin embargo, recuerde que los demás productos de Altova que usan procesadores FO utilizarán la misma ruta de acceso (StyleVision y Authentic Desktop).

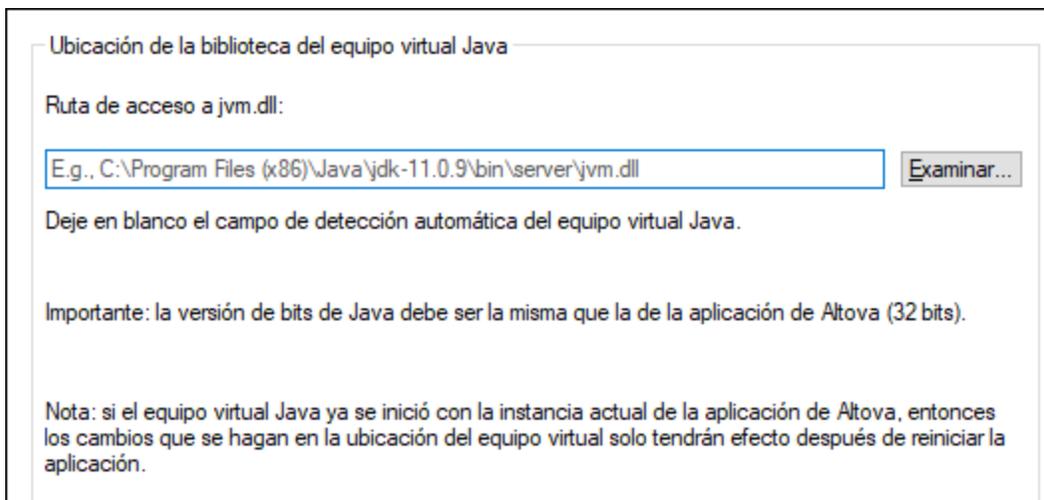
## Guardar y salir

Cuando termine de modificar las opciones de configuración, haga clic en **Aceptar** para terminar.

### 14.9.10.8 Java

En la pestaña *Java* puede introducir la ruta de acceso a un equipo virtual java en su sistema de archivos. Tenga en cuenta que no siempre es necesario agregar una ruta de acceso personal a un equipo virtual. Por defecto, Authentic Desktop 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 Authentic Desktop haya detectado automáticamente.

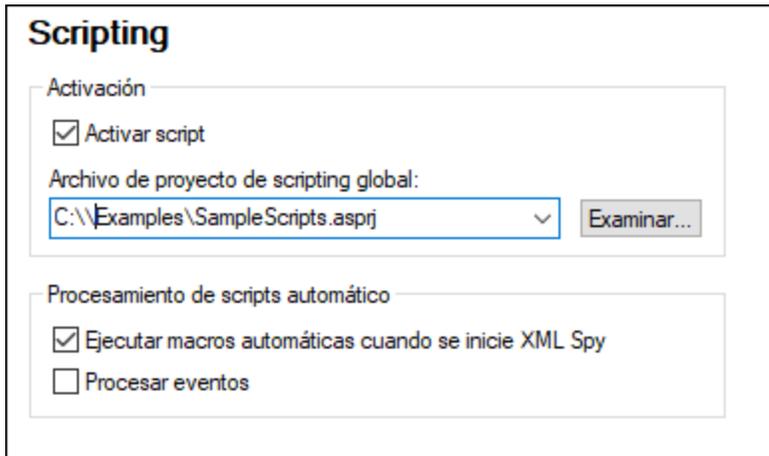


Observe lo siguiente:

- 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 Authentic Desktop (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 Authentic Desktop para que surta efecto la nueva configuración.

### 14.9.10.9 Scripting

En la sección **Script** (*imagen siguiente*) puede habilitar el [entorno de scripting](#) <sup>252</sup> al iniciarse la aplicación. Para ello marque la casilla *Activar script*. Después puede seleccionar el archivo de proyecto de script global (*imagen siguiente*).



Para definir el proyecto de script global de Authentic Desktop, marque la casilla *Activar script* y busque el archivo de proyecto de script de Altova (.asprj) que desea utilizar. También puede elegir si (i) las macros automáticas del proyecto se ejecutan automáticamente cuando se inicia Authentic Desktop y (ii) si los controladores de eventos del proyecto de script de la aplicación se ejecutan automáticamente o no.

### Guardar y salir

Cuando termine de modificar las opciones de configuración, haga clic en **Aceptar** para terminar. Las macros del proyecto de script global aparecerán en el submenú del comando **Macros**.

### 14.9.10.10 Control de código fuente

En la sección **Control de código fuente** (imagen siguiente) puede especificar el proveedor de control de código fuente, configurarlo y definir el id. de inicio de sesión para el proveedor.

Complemento actual de control de código fuente:

Jalindi Igloo Opciones avanzadas...

Id. de inicio de sesión (Jalindi Igloo):

Administrador

Realizar actualizaciones de estado en segundo plano cada 500 ms

Mostrar mensajes de salida del complemento

Obtener todo al abrir un proyecto

Proteger todo al cerrar un proyecto

No mostrar el cuadro de diálogo Desprotección al desproteger elementos

No mostrar el cuadro de diálogo Protección al proteger elementos

Mantener elementos desprotegidos cuando se protejan o añadan elementos

Crear y usar archivos de instantánea automáticamente (para fusión a tres bandas)

Si se ocultaron los diálogos al seleccionar "No volver a mostrar", haga clic en "Restaurar" para poder volver a verlos.

Restaurar

## Complemento de control de código fuente

Puede elegir un complemento de control de código fuente de los sistemas de control de código fuente que tiene instalados en el equipo. Para ello seleccione una opción de la lista desplegable del cuadro combinado. Después especifique el id. de inicio de sesión. El botón **Opciones avanzadas** abre un cuadro de diálogo del sistema de control de código fuente elegido, donde puede definir opciones de configuración propias de ese control de código fuente concreto.

## Preferencias de usuario

En esta pestaña también puede definir las preferencias de usuario:

- Puede realizar actualizaciones de estado en segundo plano pasado un intervalo de tiempo definido por el usuario. También puede desactivar por completo estas actualizaciones. Las bases de datos de control de código fuente de gran tamaño pueden consumir muchos recursos de red y de CPU. Pero si deshabilita las actualizaciones de estado en segundo plano o utiliza un intervalo de tiempo mayor entre actualización y actualización el sistema puede ser más rápido.
- Cuando se abre y se cierra un proyecto, los archivos se pueden desproteger y proteger (respectivamente) de forma automática.
- Puede evitar que aparezcan los cuadros de diálogo Desprotección y Protección.
- El botón **Restaurar** se habilita si marcó/activó la opción *No volver a mostrar* en alguno de los cuadros de diálogo. Al pulsar el botón **Restaurar** se rehabilita el botón/la opción *No volver a mostrar*.

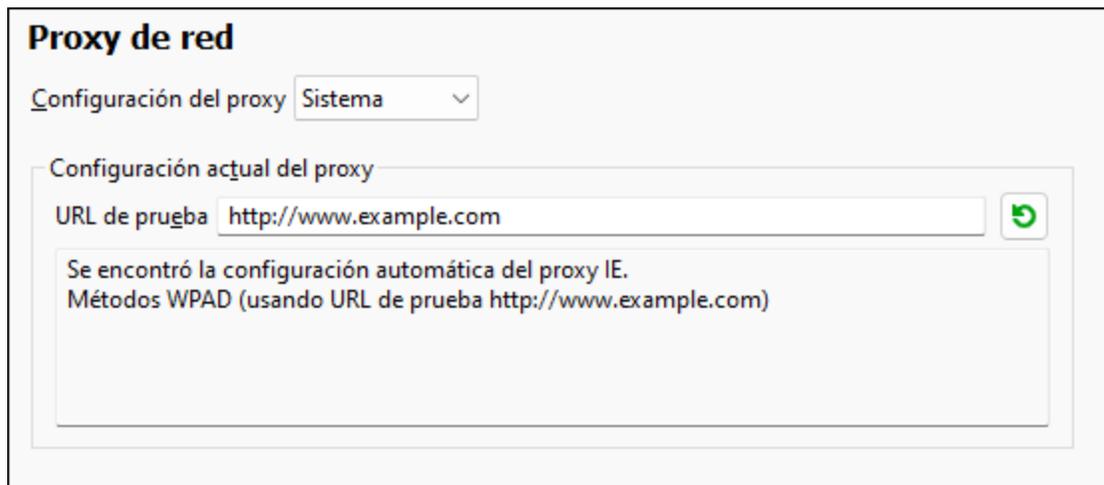
## Guardar y salir

Cuando termine de modificar las opciones de configuración, haga clic en **Aceptar** para terminar.

### 14.9.10.11 Proxy de red

El cuadro de diálogo **Proxy de red** (*imagen siguiente*) permite personalizar la configuración del proxy de red. El sistema viene con una configuración predeterminada para el proxy, por lo que este funcionará sin necesidad de configurarlo, pero si quiere usar un proxy de red alternativo puede usar estas opciones para cambiar la configuración como quiera.

**Nota:** la configuración del proxy de red es común a todas las aplicaciones de Altova MissionKit. En consecuencia, si cambia esta configuración en cualquiera de esas aplicaciones, el cambio afectará automáticamente a todas las demás.



#### Usar la configuración del proxy del sistema

Usa los parámetros de Internet Explorer (IE), que se pueden configurar desde las opciones del proxy de red. También consulta los parámetros configurados con `netsh.exe winhttp`.

#### Configuración automática del proxy

Existen las siguientes opciones:

- *Configuración de detección automática:* consulta un script WPAD (`http://wpad.LOCALDOMAIN/wpad.dat`) vía DHCP o DNS y lo usa para configurar el proxy.
- *URL del script:* indica una HTTP URL a un script (`.pac`) de configuración automática del proxy cuyos parámetros se aplican para configurar el proxy.
- *Volver a cargar:* reinicia y vuelve a cargar la configuración automática actual del proxy. Esta acción requiere Windows 8 o superior y puede llegar a tardar 30 segundos en tener efecto.

#### Configuración manual del proxy

Puede indicar manualmente el nombre completo de host y el puerto para los proxys de los respectivos protocolos. Es posible que haya un esquema compatible incluido en el nombre de host (por ejemplo: `http://hostname`). Si el proxy es compatible no es necesario que el esquema sea el mismo que el protocolo correspondiente.

### Proxy de red

Configuración del proxy Manual

Proxy HHTTP  Puerto 0

Usar este servidor de proxy para todos los protocolos

Proxy SSSL  Puerto 0

Ningún proxy para

No usar el servidor proxy para direcciones locales

---

Configuración actual del proxy

URL de prueba  ↻

(usando URL de prueba http://www.example.com)  
No se está usando ningún proxy.

Aceptar
Cancelar
Aplicar

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 <localhost> 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 (.).

## 14.10 Menú Ventanas

El menú **Ventanas** contiene comandos estándar que se usan en la mayoría de las aplicaciones Windows y con los que podrá organizar las ventanas de documento de la sesión activa de Authentic Desktop. Las ventanas que están abiertas se pueden organizar en cascada o en mosaico. Desde este menú también puede activar o desactivar (mostrar/ocultar) las ventanas de los ayudantes de entrada y las diferentes ventanas de documento.

### En cascada, En mosaico horizontal, En mosaico vertical

El comando **En cascada** reorganiza todas las ventanas de documento que están abiertas en forma de cascada (es decir, las ventanas se apilan una encima de otra).

Los comandos **En mosaico horizontal** y **En mosaico vertical** reorganiza todas las ventanas de documento que están abiertas y no minimizadas en forma de mosaico, es decir, se pueden ver todas las ventanas a la vez y se distribuyen de forma vertical u horizontal.

### Ventanas Proyecto, Información, Ayudantes de entrada y Resultados

Estos comandos sirven para activar/desactivar (mostrar/ocultar) las ventanas [Proyecto](#)<sup>27</sup>, [Información](#)<sup>29</sup>, [Ayudantes de entrada](#)<sup>19</sup> y [Resultados](#)<sup>20</sup>.

Estas ventanas son acoplables. Si hace clic en la barra de título de una de ellas y la arrastra, la ventana se desprende de su posición actual y se convierte en una ventana flotante. Haga clic con el botón derecho en la barra de título de la ventana flotante para acoplarla otra vez o para ocultarla.

### Proyecto y ayudantes de entrada

Este comando activa/desactiva (muestra/oculta) a la vez la [ventana Proyecto](#)<sup>17</sup> y las [ventanas de los ayudantes de entrada](#)<sup>19</sup>. De esta forma no tiene que activar/desactivar estas ventanas individualmente.

### Activar o desactivar todas

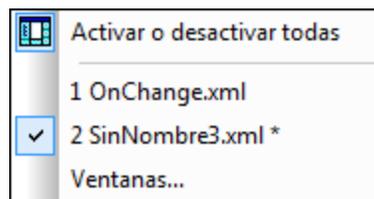
Este comando sirve para activar o desactivar todas las ventanas acoplables:

- la ventana [Proyecto](#)<sup>27</sup>
- la ventana [Información](#)<sup>29</sup>
- las ventanas de los [ayudantes de entrada](#)<sup>19</sup>
- las ventanas de [Resultados](#)<sup>25</sup>

Este comando es muy práctico si quiere ocultar rápidamente las ventanas que no sean de documento y conseguir el mayor espacio posible en la interfaz para ver mejor el documento en el que está trabajando.

### Lista de ventanas abiertas

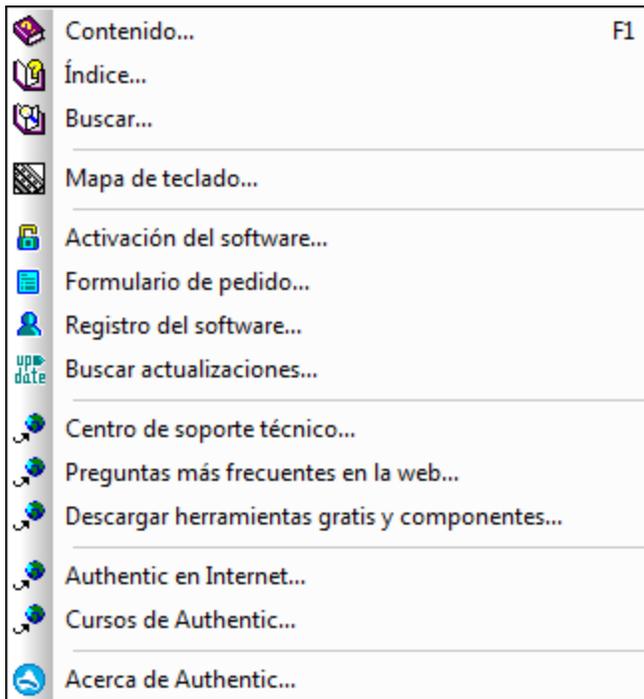
Esta lista muestra todas las ventanas que están abiertas en cada momento y permite cambiar de una ventana a otra rápidamente.



También puede usar las teclas de acceso rápido **Ctrl+F6** para recorrer todas las ventanas que están abiertas.

## 14.11 Menú Ayuda

Los comandos del menú **Ayuda** le ayudarán a obtener ayuda o información sobre Authentic Desktop y entre ellos se encuentran enlaces a páginas del sitio web de Altova.



El menú **Ayuda** también sirve para abrir el cuadro de diálogo [Activación del software](#) <sup>298</sup>, donde puede introducir su código clave de licencia cuando haya comprado el producto.

### 14.11.1 Contenido, Índice, Buscar...

#### ☐ Contenido

Abre la ayuda en pantalla de Authentic Desktop 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 Authentic Desktop 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.

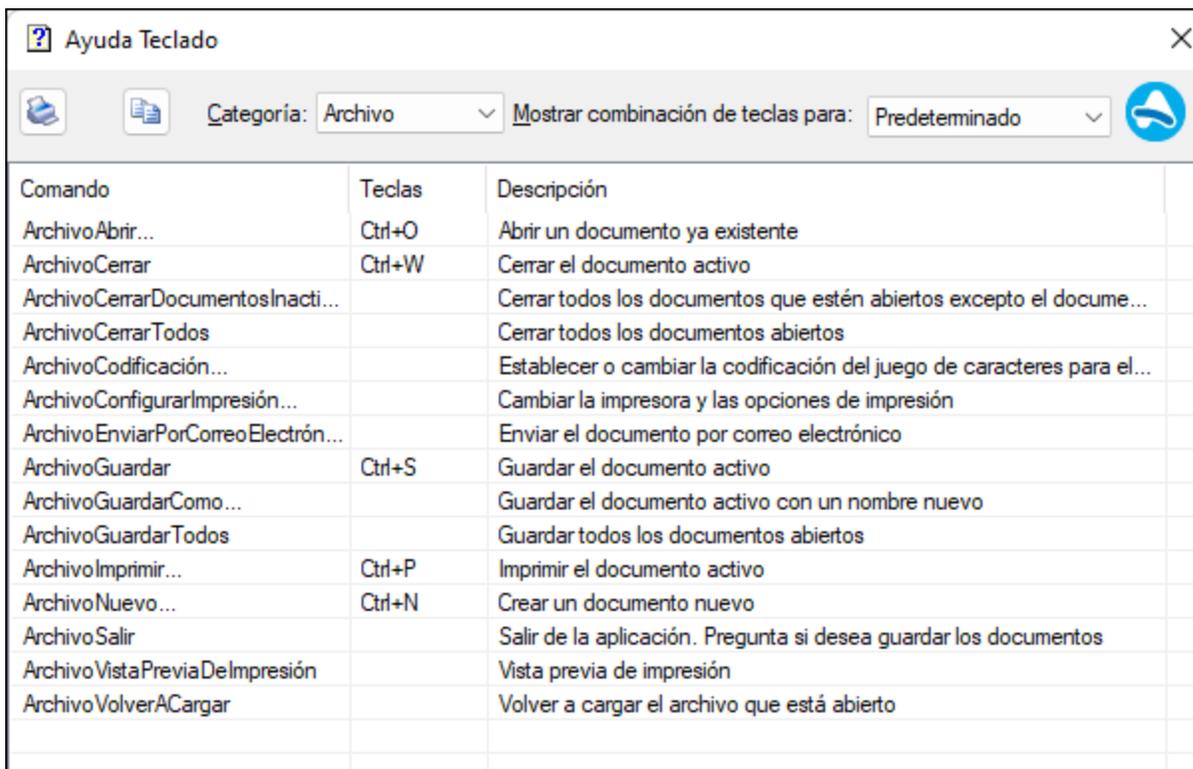
#### ☐ Buscar

Abre la ayuda en pantalla de Authentic Desktop 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.

## 14.11.2 Mapa de teclado

El comando **Ayuda | Mapa de teclado...** abre un cuadro de diálogo informativo que enumera todos los comandos de todos los menús de Authentic Desktop. Los comandos de menú van acompañados de una breve descripción y de sus teclas de acceso rápido.



Comando	Teclas	Descripción
ArchivoAbrir...	Ctrl+O	Abrir un documento ya existente
ArchivoCerrar	Ctrl+W	Cerrar el documento activo
ArchivoCerrarDocumentosInacti...		Cerrar todos los documentos que estén abiertos excepto el docume...
ArchivoCerrarTodos		Cerrar todos los documentos abiertos
ArchivoCodificación...		Establecer o cambiar la codificación del juego de caracteres para el...
ArchivoConfigurarImpresión...		Cambiar la impresora y las opciones de impresión
ArchivoEnviarPorCorreoElectrón...		Enviar el documento por correo electrónico
ArchivoGuardar	Ctrl+S	Guardar el documento activo
ArchivoGuardarComo...		Guardar el documento activo con un nombre nuevo
ArchivoGuardarTodos		Guardar todos los documentos abiertos
ArchivoImprimir...	Ctrl+P	Imprimir el documento activo
ArchivoNuevo...	Ctrl+N	Crear un documento nuevo
ArchivoSalir		Salir de la aplicación. Pregunta si desea guardar los documentos
ArchivoVistaPreviaDeImpresión		Vista previa de impresión
ArchivoVolverACargar		Volver a cargar el archivo que está abierto

Para ver los comandos de un menú determinado, seleccione el nombre del menú en el cuadro combinado *Categoría*. Si quiere imprimir el comando, haga clic en el icono en forma de impresora.

## 14.11.3 Activación, formulario de pedido, registro, actualizaciones

### ☒ Activación del software

Tras descargar el producto de software de Altova puede registrarlo o activarlo con una clave de evaluación gratuita o con una clave de licencia permanente.

- **Licencia gratuita:** cuando inicie el software por primera vez aparecerá el cuadro de diálogo "Activación del software". Este cuadro de diálogo incluye un botón para solicitar una licencia de evaluación gratuita. Introduzca su nombre, el nombre de su compañía y su dirección de correo

electrónico y haga clic en **Enviar solicitud**. Altova le enviará un archivo de licencia al correo electrónico proporcionado. Guarde el archivo en su equipo. Al hacer clic en **Enviar solicitud** aparece un campo en la parte inferior del cuadro de diálogo, que es donde debe introducir la ruta de acceso al archivo de licencia. Puede escribir la ruta de acceso o navegar hasta el archivo de licencia. Por último, haga clic en **Aceptar**. (En el cuadro de diálogo de activación del software también puede hacer clic en **Cargar licencia nueva** para acceder al cuadro de diálogo en el que se introduce la ruta de acceso.) El software permanecerá desbloqueado durante 30 días.

- **Clave de licencia permanente:** el cuadro de diálogo "Activación del software" también incluye un botón para comprar una clave de licencia permanente. Este botón conduce a la tienda en línea de Altova, donde podrá adquirir una clave de licencia permanente para el producto. Recibirá por correo electrónico un archivo que contiene los datos de la licencia. Existen tres tipos de licencias permanentes: de tipo instalado, de usuario concurrente y de usuario designado. Las *licencias de tipo instalado* son cada una para un único equipo. En el caso de las *licencias de usuario concurrentes*, si adquiere una de estas licencias para  $n$  usuarios podrá instalar la licencia en un máximo de  $10n$  equipos pero solo  $n$  usuarios podrán usar el software al mismo tiempo. Las *licencias de usuario designado* autorizan a un usuario específico a usar el software en un máximo de 5 equipos distintos. Para activar su software haga clic en **Cargar una licencia nueva** e introduzca la ruta de acceso al archivo de licencia en el cuadro de diálogo "Activación del software" que aparece. Por último, haga clic en **Aceptar**.

**Nota:** en el caso de licencias para varios usuarios, se le pedirá a cada usuario que introduzca su nombre.

*Claves por correo electrónico y las distintas formas de activar las licencias de los productos de Altova*

El correo electrónico que recibirá de Altova contiene, en un adjunto, el archivo de la licencia, que tiene la extensión `.altova_licenses`.

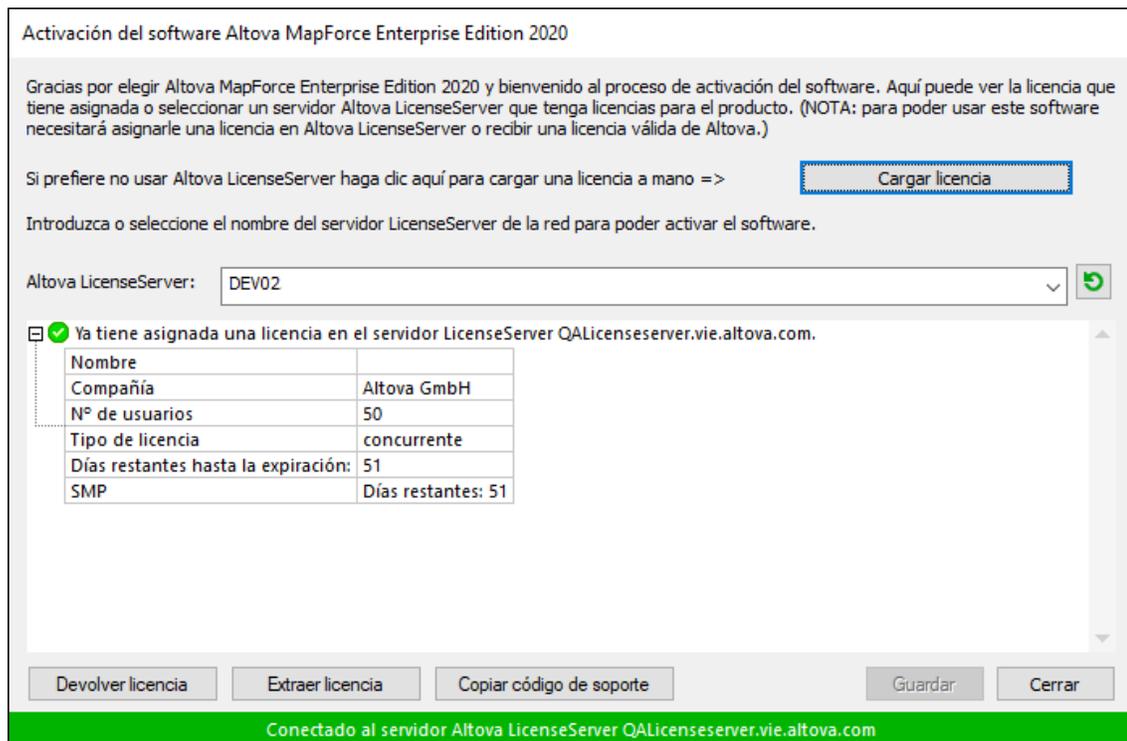
Para activar su producto de Altova, puede optar por una de las siguientes opciones:

- Guardar el archivo de licencia (`.altova_licenses`) en su equipo, hacer doble clic en el archivo de licencia, introducir los detalles necesarios en el cuadro de diálogo que aparece y finalmente hacer clic en **Aplicar claves**.
- Guardar el archivo de licencia (`.altova_licenses`) en su equipo. En su producto de Altova seleccione el comando de menú **Ayuda | Activación del software** y después **Cargar una licencia nueva**. Navegue hasta el archivo de licencia o introduzca la ruta de acceso al archivo de licencia en el cuadro de diálogo "Activación del software" y haga clic en **Aceptar**.
- Guardar el archivo de licencia (`.altova_licenses`) en su equipo y cargarlo desde esa ubicación a su Altova LicenseServer. Puede: (i) adquirir la licencia de su producto Altova con el cuadro de diálogo de activación de software del producto (*véase más abajo*) o (ii) asignar la licencia al producto de Altova LicenseServer. Para obtener más información sobre la gestión de licencias con el LicenseServer, lea el resto de esta sección.

El cuadro de diálogo "Activación del software" (*imagen siguiente*) se abre con el comando **Ayuda | Activación del software**.

Hay dos maneras de activar el software:

- registrando la licencia en el cuadro de diálogo "Activación del software". Para ello haga clic en **Cargar una licencia nueva** y navegue hasta el archivo de la licencia. Haga clic en **Aceptar** para confirmar la ruta de acceso al archivo de licencia y para confirmar los datos que haya introducido (su nombre, en el caso de licencias para más de un usuario); a continuación, haga clic en **Guardar** para finalizar el proceso.
- asignando una licencia a través de un servidor Altova LicenseServer de la red: para adquirir una licencia a través de un servidor Altova LicenseServer de la red haga clic en el botón **Usar Altova LicenseServer**, situado al final del cuadro de diálogo. Seleccione el equipo en el que está instalado el LicenseServer que quiere usar. Tenga en cuenta que la autodetección de los License Servers funciona con emisiones enviadas por LAN. Este tipo de emisiones se limitan a una subred, por lo que Altova License Server debe estar en la misma subred que el equipo del cliente para que funcione la autodetección. Si esta no funciona, introduzca el nombre del servidor. Para ello es necesario que el servidor LicenseServer tenga una licencia para su producto en el repositorio de licencias. Si así es, el cuadro de diálogo "Activación del software" emite un mensaje a tal efecto (*imagen siguiente*). Haga clic en el botón **Guardar** para adquirir la licencia.



Una vez se ha adquirido una licencia para un equipo específico (es decir, "instalada") del servidor LicenseServer, no se puede devolver al mismo hasta 7 días después. Transcurridos estos 7 días podrá devolver la licencia de ese equipo (con el botón **Devolver licencia**) para que pueda ser adquirida por otro cliente. No obstante, el administrador de LicenseServer puede anular asignaciones de licencias desde la interfaz web del servidor LicenseServer en cualquier momento. Observe que únicamente se pueden devolver las licencias instaladas en equipos específicos, no las licencias concurrentes.

#### Extracción de licencias

Puede extraer una licencia del repertorio durante un período máximo de 30 días de modo que la licencia se almacene en el equipo donde se ejecuta el producto. Esto le permitirá trabajar sin

conexión a Internet, lo cual puede ser útil si desea trabajar en un entorno que no dispone de acceso a su servidor Altova LicenseServer (p. ej. cuando el producto servidor de Altova está instalado en un equipo portátil y el usuario se encuentra de viaje). Mientras la licencia esté extraída, LicenseServer indicará que la licencia está en uso y no podrá ser utilizada por ningún otro equipo. La licencia vuelve automáticamente a su estado insertado cuando finaliza el período de extracción de la licencia. La licencia extraída también se puede insertar en el servidor en cualquier momento con el botón **Insertar** del cuadro de diálogo "Activación del software".

Siga estas instrucciones para extraer una licencia:

1. En el cuadro de diálogo "Activación del software" haga clic en el botón **Extraer licencia** (*imagen anterior*).
2. Aparece el cuadro de diálogo "Extracción de licencias". Seleccione el periodo de extracción y haga clic en **Extraer**.
3. La licencia se extrae y ocurren dos cosas: (i) el cuadro de diálogo "Activación del software" muestra información sobre la extracción de la licencia, incluida la fecha y la hora en la que expira el plazo de extracción y (ii) en lugar del botón **Extraer licencia**, aparece el botón **Insertar licencia**. Para insertar la licencia basta con hacer clic en este botón. Como la licencia vuelve automáticamente a su estado de inserción cuando finaliza el plazo de extracción, compruebe que el plazo seleccionado coincide con el período de tiempo que tiene pensado trabajar sin conexión a Internet.

Para devolver una licencia esta debe ser de la misma versión principal que el producto de Altova con el que se extrajo. Por tanto, es recomendable devolver la licencia antes de actualizar el producto de Altova correspondiente a la siguiente versión principal.

**Nota:** para poder extraer licencias esta característica debe estar habilitada en el servidor LicenseServer. Si esta característica no está habilitada, recibirá un mensaje de error a tal efecto cuando trate de extraer una licencia. Cuando esto ocurra, póngase en contacto con el administrador de su servidor LicenseServer.

#### Copiar código de soporte

Haga clic en **Copiar código de soporte** para copiar los detalles de la licencia en el portapapeles. Esta es la información que deberá introducir al ponerse en contacto con el equipo de soporte técnico a través del [formulario de soporte técnico](#).

Altova LicenseServer es una práctica herramienta para administrar en tiempo real todas las licencias de Altova de la red y ofrece información detallada sobre cada licencia, asignaciones a clientes y uso de las licencias. La ventaja de usar este producto está en sus características administrativas. Altova LicenseServer puede descargarse gratis del [sitio web de Altova](#). Para más información consulte la [documentación de Altova LicenseServer](#).

#### ☒ Formulario de pedido

Hay dos maneras de comprar licencias para los productos de Altova: con el botón **Comprar una licencia permanente** del cuadro de diálogo "Activación del software" (*ver apartado anterior*) o con el comando **Ayuda | Formulario de pedido**, que le lleva directamente a la tienda en línea de Altova, donde puede adquirir claves de licencias para los productos de Altova.

#### ☒ Registro del software

Este comando abre la página de registro de productos de Altova en una pestaña del explorador. Si registra el software, recibirá información sobre actualizaciones y versiones nuevas del producto.

#### ⊕ Buscar actualizaciones

Comprueba si existe una versión más reciente del producto en el servidor de Altova y emite un mensaje a tal efecto.

## 14.11.4 Otros comandos

#### ☐ Centro de soporte técnico

Es un enlace al centro de soporte técnico del [sitio web de Altova](#). El centro de soporte técnico incluye preguntas frecuentes, foros de debate y un formulario para ponerse en contacto con el equipo de soporte técnico de Altova.

#### ☐ 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.

#### ☐ Authentic Desktop en Internet

Es un enlace al [sitio web de Altova](#), donde encontrará más información sobre Authentic Desktop, otros productos de Altova y tecnologías relacionadas.

#### ☐ Acerca de Authentic Desktop

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.

## 14.12 Línea de comandos

Algunas acciones de Authentic Desktop se pueden llevar acabo desde la línea de comandos. Se trata de estos comandos:

### Abrir un archivo

```
authentic.exe file.xml
```

Abre el archivo, `file.xml`, in Authentic Desktop

### Abrir varios archivos

```
authentic.exe file1.xml file2.xml
```

Abre los archivos, `file1.xml` and `file2.xml`, in Authentic Desktop

### Asignar un archivo SPS a un archivo XML para editarlo en la vista Authentic

```
authentic.exe myxml.xml /sps mysps.sps
```

Abre el archivo XML, `myxml.xml`, en la vista Authentic con `mysps.sps` como archivo SPS asociado. La marca `/sps` especifica que el archivo SPS que sigue a la marca debe usarse con el archivo XML que precede a la marca `/sps` (para poder editar el archivo XML en la vista Authentic).

### Abrir una plantilla XML nueva a través de un archivo SPS

```
authentic.exe mysps.sps
```

Abre un archivo XML nuevo en la vista Authentic. La presentación del archivo se basa en el SPS y el archivo XML nuevo tendrá una estructura inicial basada en el esquema del SPS. Debe asignar un nombre al archivo XML nuevo cuando lo guarde.

## 15 Referencia del programador

Authentic Desktop es un servidor de automatización: expone objetos programables a otras aplicaciones, que se denominan clientes de automatización. Como resultado los clientes de automatización pueden tener acceso directo a los objetos y a las funciones que pone a su disposición el servidor de automatización. Por ejemplo, los clientes de automatización de Authentic Desktop pueden usar la función de validación XML de Authentic Desktop. Por consiguiente, los desarrolladores pueden mejorar sus aplicaciones con las funciones predefinidas de Authentic Desktop.

Los objetos programables de Authentic Desktop se ponen a disposición de los clientes de automatización por medio de la API de Authentic Desktop, que es una API de COM. En la presente documentación describimos el modelo de objetos de la API de la aplicación y todos los objetos disponibles (consulte la sección [API de la aplicación](#)<sup>351</sup>).

A la API de la aplicación se puede acceder desde estos entornos:

- [El editor de scripts](#)<sup>306</sup>
- [Los complementos para entornos IDE](#)<sup>334</sup>
- [Programas externos](#)<sup>351</sup>
- [La integración con ActiveX](#)<sup>618</sup>

A continuación describimos estos entornos brevemente.

### Editor de scripts: personalizar y modificar las funciones de Authentic Desktop

Puede personalizar su instalación de Authentic Desktop modificando sus funciones o agregando funciones nuevas. También puede crear formularios y modificar la interfaz del usuario añadiendo comandos de menú e iconos nuevos. Para ello basta con escribir scripts que interactúen con objetos de la API de la aplicación. A fin de ayudarle a realizar estas tareas de forma eficaz, Authentic Desktop viene con un editor de scripts integrado. Para más información sobre las funciones del editor de scripts e instrucciones de uso, consulte la sección [Editor de scripts](#)<sup>306</sup> de esta referencia. Los lenguajes de programación compatibles con el editor de scripts son **JScript** y **VBScript**.

### Complementos para entornos IDE: crear complementos para Authentic Desktop

En Authentic Desktop puede crear sus propios complementos e integrarlos en Authentic Desktop. Para ello puede usar la interfaz especial de Authentic Desktop para complementos. Para más información sobre cómo crear complementos, consulte la sección [Complementos para entornos IDE](#)<sup>334</sup>.

Un objeto de la aplicación se pasa a la mayoría de los métodos que debe implementar el complemento para entornos IDE y la aplicación llama al objeto. Los lenguajes que se suelen utilizar para implementar un complemento para entornos IDE son **C#** y **C++**. Para más información, consulte la sección [Complementos para entornos IDE](#)<sup>334</sup>.

### Programas externos

También puede manipular Authentic Desktop con scripts externos. Por ejemplo, puede escribir un script para abrir Authentic Desktop a una hora concreta, abrir un archivo XML en Authentic Desktop, validarlo e imprimirlo. Los scripts externos utilizan la API de la aplicación para llevar a cabo estas tareas. Para más información, consulte la sección [API de la aplicación](#)<sup>351</sup>.

Para usar la API de la aplicación desde fuera de XMLSpy es necesario iniciar una instancia de XMLSpy primero. La manera de iniciar la instancia de Authentic Desktop depende del lenguaje de programación utilizado. Para más información consulte el apartado [Lenguajes de programación](#)<sup>353</sup>.

Básicamente Authentic Desktop se inicia por medio de su registro COM. Después se devuelve el objeto `Application` asociado con la instancia de Authentic Desktop. Dependiendo de la configuración COM, puede devolverse un objeto asociado con una instancia de Authentic Desktop que ya se esté ejecutando. Puede utilizarse cualquier lenguaje de programación que permita crear e invocar objetos COM. A continuación enumeramos los más comunes:

- Los archivos de script de [JScript](#)<sup>373</sup> y [VBScript](#)<sup>359</sup> tienen una sintaxis sencilla y están diseñados para acceder a objetos COM. Pueden ejecutarse directamente desde la línea de comandos de DOS o al hacer doble clic en ellos desde el explorador de Windows. Son ideales para tareas de automatización sencillas.
- [C#](#)<sup>362</sup> es un lenguaje de programación con un gran número de funciones. El acceso a objetos COM se puede encapsular automáticamente con C#.
- C++ permite un control directo sobre el acceso a COM pero requiere cantidades de código relativamente grandes en comparación con otros lenguajes.
- [Java](#)<sup>373</sup>: los productos de Altova vienen con clases nativas de Java, que encapsulan la API de la aplicación y ofrecen un entorno similar a Java.
- Otros lenguajes de programación que pueden ser útiles son Visual Basic for Applications, Perl y Python.

## Integración con ActiveX

Otra manera de acceder a la API de la aplicación es por medio del control ActiveX de Authentic Desktop. Esta característica sólo está disponible si tiene instalado el [paquete de integración de Authentic Desktop](#)<sup>618</sup>. Todos los controles ActiveX tienen una propiedad que devuelve un objeto COM equivalente para su función subyacente. El control del administrador ofrece un objeto `Application`, el control de documentos ofrece un objeto `Document` y el objeto marcador de posición, si contiene un árbol de proyecto, devuelve el objeto `Project`. Los métodos compatibles con estos objetos se describen en detalle en el [apartado Interfaces de la sección API de la aplicación](#)<sup>381</sup>. Debe tener cuidado y no utilizar métodos que no tienen sentido en el contexto de la integración para el control ActiveX. Para más información consulte la sección [Integración con ActiveX](#)<sup>618</sup>.

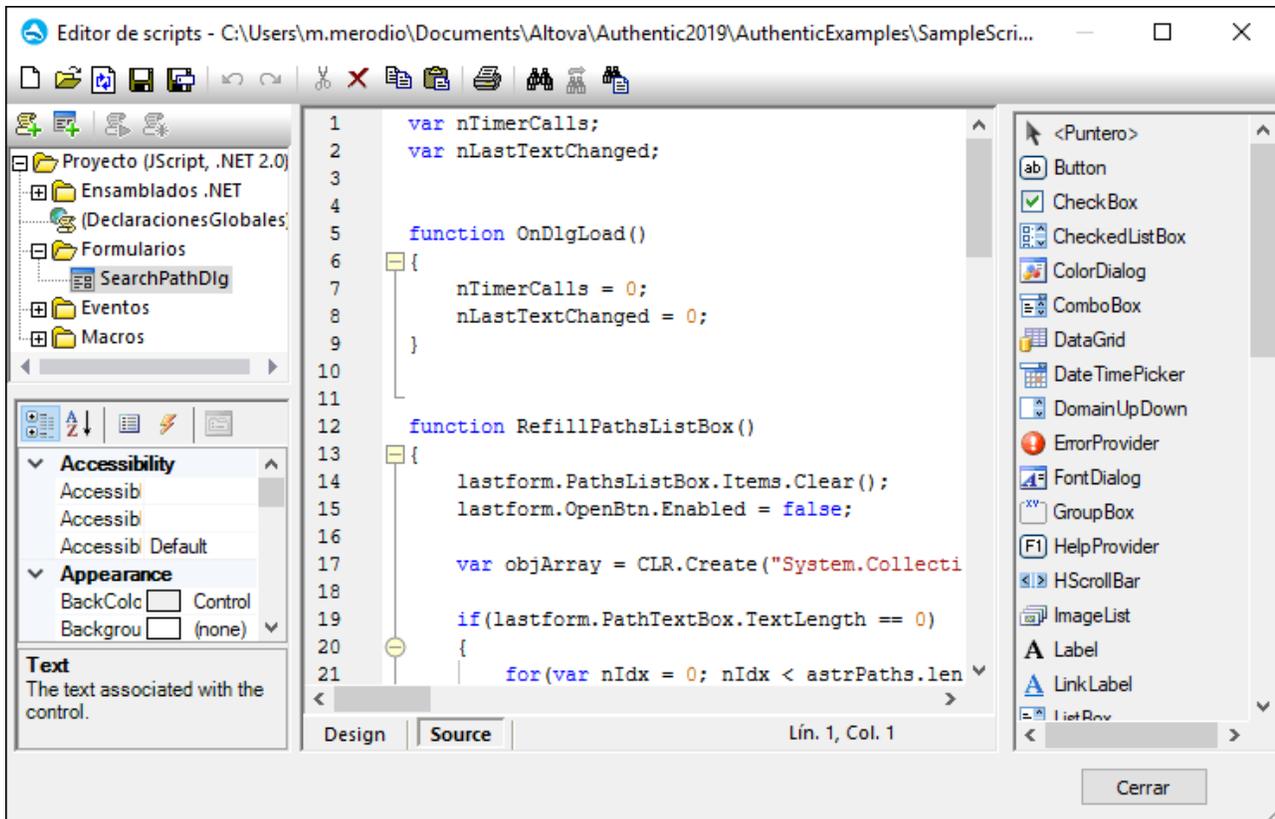
## Acerca de la referencia del programador

La *Referencia del programador* de la documentación de Authentic Desktop se divide en estas secciones:

- [Editor de scripts](#)<sup>306</sup>: una referencia del usuario para el entorno de scripting incluido en Authentic Desktop
- [Complementos para entornos IDE](#)<sup>334</sup>: explica cómo crear complementos para Authentic Desktop
- [API de la aplicación](#)<sup>351</sup>: una referencia de la API de la aplicación
- [Integración para ActiveX](#)<sup>618</sup>: una guía y referencia para aprender a integrar la IGU y las funciones de Authentic Desktop usando el control ActiveX

## 15.1 Editor de scripts

El Editor de scripts un entorno de desarrollo integrado en Authentic Desktop desde donde puede personalizar las funciones de Authentic Desktop con ayuda de los scripts JScript o VBScript. Por ejemplo, puede añadir un elemento nuevo de menú que ejecute una tarea personalizada en el proyecto o puede hacer que Authentic Desktop desencadene algún comportamiento cada vez que se abra o cierre un documento. Para ello debe crear proyectos de scripting, que son archivos con la extensión `.asprj` (Altova Scripting Project).



Editor de scripts

Los proyectos de scripting suelen incluir una o varias macros, que son programas que ejecutan diversas tareas personalizadas cuando se invocan. Puede ejecutar macros bien de forma explícita desde un elemento de menú (o un botón de la barra de herramientas, si está configurado para ello) o configurar una para que se ejecute automáticamente siempre que se inicie Authentic Desktop. El entorno de scripting también se puede integrar con la API COM de Authentic Desktop. Por ejemplo, los scripts que cree con VBScript o JScript pueden gestionar eventos de aplicación o documentos como iniciar o cerrar Authentic Desktop, abrir o cerrar un proyecto, etc. Los proyectos de scripting pueden incluir formularios de Windows Forms, que puede diseñar de forma visual de forma parecida a como cuando trabaja con Visual Studio. También existen varios comandos integrados de los que se puede ayudar para instanciar y usar clases .NET de código VBScript o JScript.

Una vez haya completado el proyecto de scripting puede habilitarlo de forma global en todo Authentic Desktop o solamente para proyectos específicos.

El Editor de scripts necesita que instale .NET Framework 2.0 o más avanzado antes de instalar Authentic Desktop.

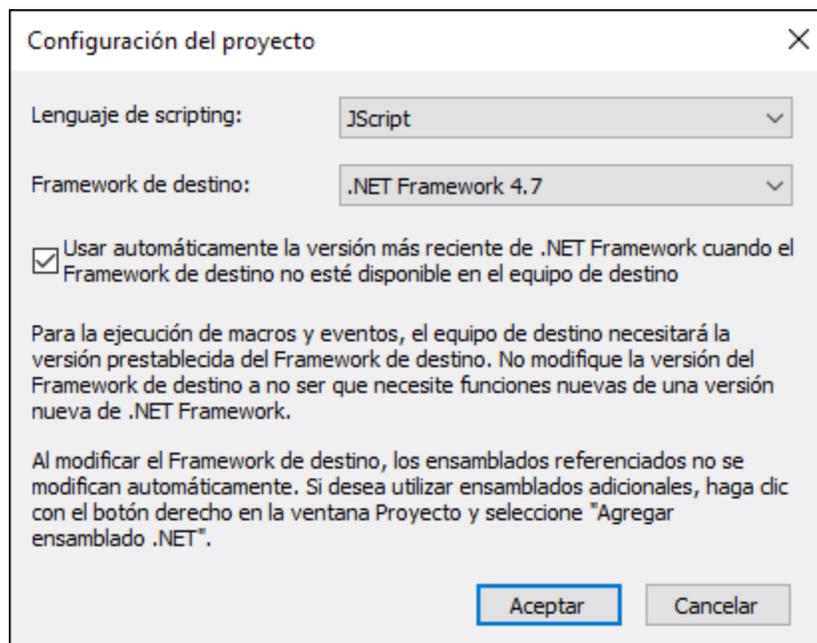
## 15.1.1 Crear un proyecto de scripting

Todos los scripts, así como la información de scripting que se crean en el Editor de scripts se guardan en los proyectos de scripting de Altova (archivos .asprj). Un proyecto de scripting puede contener macros, controladores de eventos de aplicaciones y formularios (que también pueden tener sus propios controladores de eventos). También puede añadir variables globales y funciones a un script "Declaraciones globales", que hace que esas variables y funciones sean accesibles en todo el proyecto.

Para empezar un proyecto nuevo ejecute el comando **Herramientas | Editor de scripts**.

Los lenguajes que se pueden usar en los proyectos de scripting son JScript y VBScript (no debe confundirse con Visual Basic, que no es compatible). Estos motores de scripting están disponibles por defecto en Windows y no precisan de requisitos especiales para ejecutarse. Para seleccionar el lenguaje de scripting que quiere usar:

1. Haga clic con el botón derecho en el elemento Proyecto, en el panel superior izquierdo y seleccione **Configuración del proyecto** en el menú contextual.
2. Seleccione un lenguaje (JScript o VBScript) y haga clic en **Aceptar**.



Desde el cuadro de diálogo "Configuración el proyecto", en la imagen anterior, también puede cambiar la versión de .NET Framework de destino. Esto suele ser necesario si su proyecto de scripting requiere alguna función que solamente exista en las versiones más recientes de .NET Framework. Tenga en cuenta que cualquier cliente que use su proyecto de scripting tendrá que tener instalada la misma versión de .NET Framework que usted (o una más reciente, siempre que sea compatible).

Por defecto, un proyecto de scripting hace referencia a varios ensamblados .NET, como `System`, `System.Data`, `System.Windows.Forms` entre otros. Si lo necesita puede importar más ensamblados .NET, incluidos 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](#)<sup>324</sup> desde el código.

Puede crear varios proyectos de scripting y guardar uno en disco para después volver a cargarlo más tarde en el Editor de scripts. Para ello use los botones estándar de Windows de la barra de herramientas: **Nuevo**, **Abrir**, **Guardar**, **Guardar como**. Una vez haya probado el proyecto y esté listo para implementarse puede cargarlo en Authentic Desktop y ejecutar cualquiera de sus macros o controladores de eventos. Para más información consulte el apartado [Habilitar scripts y macros](#)<sup>331</sup>.

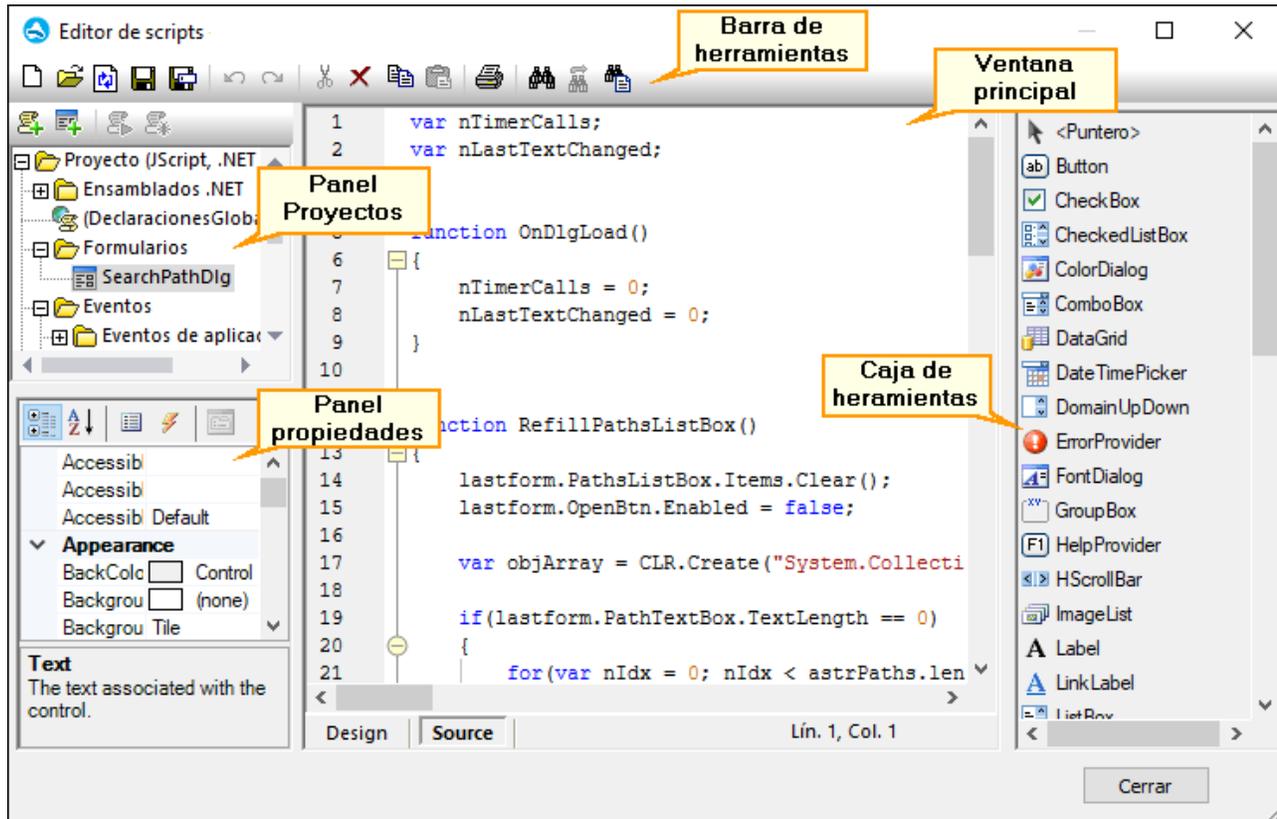
También puede encontrar un proyecto de scripting de ejemplo en: **C:**  
**\Usuarios\<user>\Documentos\Altova\Authentic2023\AuthenticExamples\SampleScripts.asprj.**

Las secciones siguientes se centran en las partes de un proyecto de scripting: declaraciones globales, macros, formularios y eventos.

### 15.1.1.1 Información general

El Editor de scripts contiene estas partes:

- Barra de herramientas
- Panel Proyectos
- Panel Propiedades
- Ventana principal
- Cuadro de herramientas



## Barra de herramientas

La barra de herramientas incluye comandos estándar de Windows para la gestión de archivos (**Nuevo**, **Abrir**, **Guardar**, **Guardar como**) y comandos de edición (**Copiar**, **Cortar**, **Eliminar**, **Pegar**). Al editar código fuente también se activan los comandos **Buscar** y **Reemplazar**, así como el comando **Imprimir**.

## Panel Proyectos

En el panel Proyectos puede ver y gestionar la estructura del proyecto. Un proyecto de scripting consiste en varios componentes que pueden funcionar juntos y que no tienen que crearse en un orden determinado:

- *Un script "Declaraciones globales"*. Como el nombre indica, este script contiene información que se usa en todo el proyecto. En este script puede declarar cualquier variable o función que necesite tener disponible para todos los formularios, scripts de gestión de eventos y macros.
- *Formularios*. Los formularios suelen ser necesarios para recopilar información de los usuarios o para suministrar cuadros de diálogo informativos. Un formulario se invoca llamándolo sea con una función (en el script *Declaraciones globales*) o directamente con una macro.
- *Eventos*. La carpeta "Eventos" muestra eventos de la aplicación Authentic Desktop 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 Authentic Desktop. 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**.

**Ejecutar macro**

Ejecuta la macro seleccionada.

**Depurar macro**

Ejecuta la macro seleccionada en modo depuración.

## Panel Propiedades

El panel Propiedades es muy parecido al de Visual Studio y contiene:

- Propiedades del formulario, si se selecciona uno
- Propiedades del objeto, si se selecciona uno
- Eventos del formulario, si se selecciona uno
- Eventos del objeto, si se selecciona uno

Para alternar entre las propiedades y los eventos del componente seleccionado haga clic en los botones Propiedades  o Eventos .

Los iconos **Por categorías**  y **Alfabético**  muestran las propiedades o los eventos organizados por categorías o por orden alfabético ascendente.

Cuando se selecciona una propiedad o un evento aparece una breve descripción de ese elemento en la parte inferior del panel Propiedades.

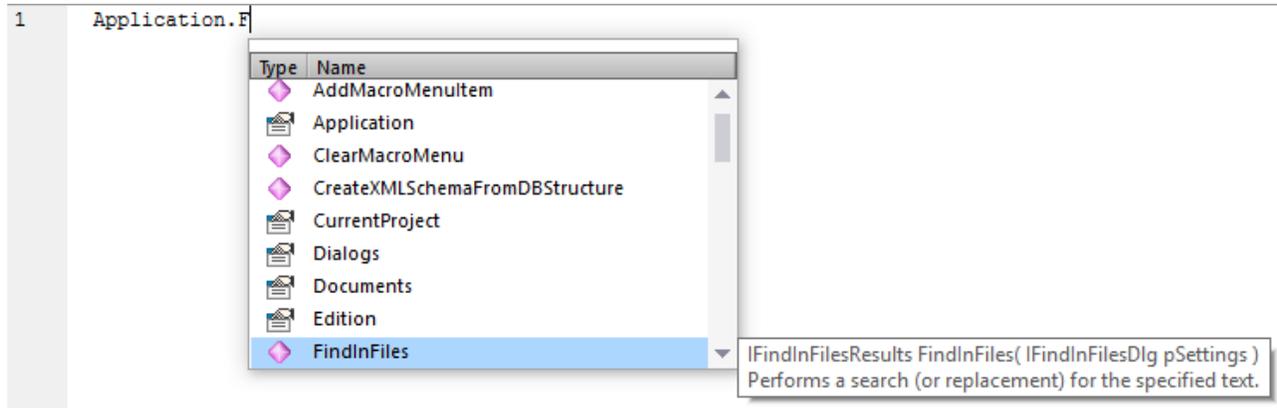
## Ventana principal

La ventana principal es el área de trabajo donde puede introducir código fuente o modificar el diseño del formulario. Al editar formularios puede trabajar en dos pestañas: *Design* y *Source*. La pestaña *Design* muestra el diseño del formulario, mientras que la pestaña *Source* contiene el código fuente, como métodos de control de eventos.

El editor de código fuente cuenta con ayudas para la edición de código, como color de sintaxis, plegamiento de código, resaltado de los corchetes de inicio y final, zoom, sugerencias de finalización automática y marcadores.

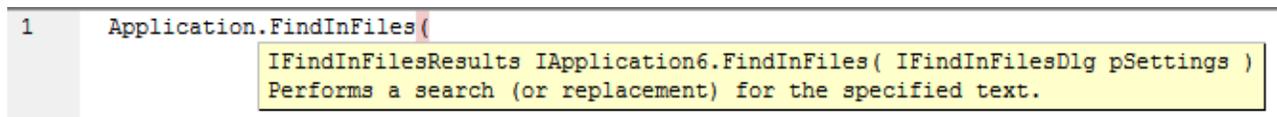
### Sugerencias de finalización automática

JScript y VBScript son dos lenguajes sin tipo, por lo que la finalización automática se limita a los nombres de la API COM y a los [comandos](#) <sup>321</sup> integrados de Authentic Desktop. El método completo o la firma de la propiedad aparecen junto al ayudante de entrada de finalización automática.



Si los nombres empiezan por `objDocument`, `objProject`, `objXMLData`, or `objAuthenticRange`, entonces se mostrarán los miembros de la interfaz correspondiente.

Para ver la firma (y la documentación, si la hay) de un método o una propiedad conocidos, coloque el cursor del ratón sobre el elemento en cuestión, por ejemplo:



El ayudante de entrada de finalización automática suele aparecer automáticamente al editar, pero también puede activarlo pulsando **Ctrl+Barra espaciadora**.

### Marcadores

- Para guardar o eliminar un marcador, haga clic dentro de una línea y después pulse **Ctrl+F2**
- Para ir al marcador siguiente pulse **F2**
- Para ir al marcador anterior pulse **Mayús+F2**
- Para eliminar todos los marcadores pulse **Ctrl+Mayús+F2**

### Alejarse y acercarse con el zoom

- Para alejarse y acercarse con el zoom mantenga pulsada la tecla Ctrl y pulse las teclas "+" o "-", o gire la rueda del ratón.

### Configurar la vista Texto

Para activar estas opciones haga clic con el botón derecho en el editor y seleccione **Configuración de la vista Texto** en el menú contextual.

### Fuentes

Para cambiar la fuente haga clic con el botón derecho en el editor y seleccione **Fuentes de la vista Texto...** en el menú contextual.

## Caja de herramientas

La Caja de herramientas contiene todos los objetos que hay disponibles para diseñar formularios, como son los botones, las cajas de texto, los cuadros combinados, etc.

### 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](#)<sup>314</sup>.

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.

### 15.1.1.2 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.

### 15.1.1.3 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](#)<sup>313</sup>).

```
// 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](#)<sup>321</sup>.

Puede agregar varias macros al mismo proyecto y puede convertir cualquier macro en "automática". Al hacerlo, la macro se ejecuta automáticamente cuando se inicia Authentic Desktop. Para definir una macro como automática haga clic con el botón derecho en ella y seleccione **Establecer como macro automática** en el menú contextual.

Solo se puede ejecutar una macro a la vez. Una vez se ha ejecutado la macro (o el evento) el script se cierra y las variables globales pierden sus valores.

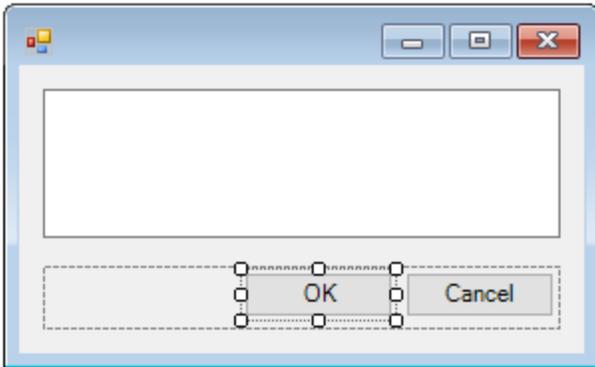
Para ejecutar una macro directamente en el Editor de scripts haga clic en Ejecutar macro . Para depurar una macro con el depurador de Visual Studio haga clic en Depurar macro . Para más información sobre cómo habilitar y ejecutar macros en Authentic Desktop, consulte el apartado [Habilitar scripts y macros](#)<sup>331</sup>.

#### 15.1.1.4 Formularios

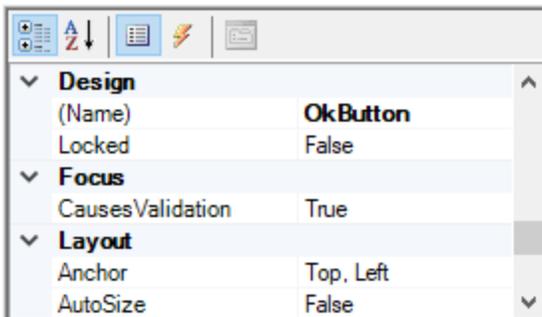
Los formularios son especialmente prácticos si necesita recolectar datos de entrada de usuarios o mostrar datos a usuarios. Para llevar a cabo esas acciones un formulario puede contener varios controles como botones, casillas de verificación, cuadros combinados, etc.

Para añadir un formulario haga clic con el botón derecho en el panel Proyecto y seleccione **Agregar formulario** en el menú contextual. Para añadir un control a un formulario arrástrelo desde la Caja de herramientas que está a la derecha del Editor de scripts y suéltela en el formulario.

Puede cambiar la posición y el tamaño de los controles directamente en el formulario usando las manijas que aparecen al hacer clic en un control, por ejemplo:



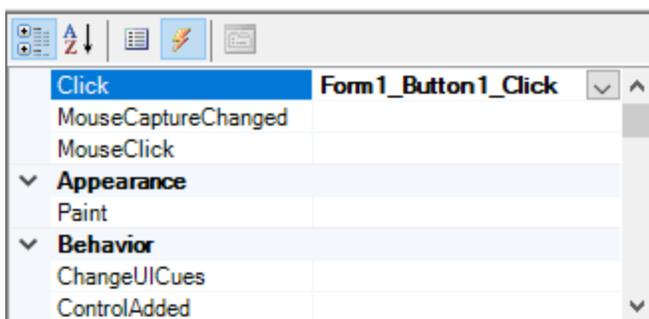
Todos los controles de los formularios tienen propiedades que puede ajustar en el panel Propiedades. Para ello, seleccione primero el control en el formulario y después edite las propiedades en cuestión en el panel Propiedades.



## 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 Authentic Desktop 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

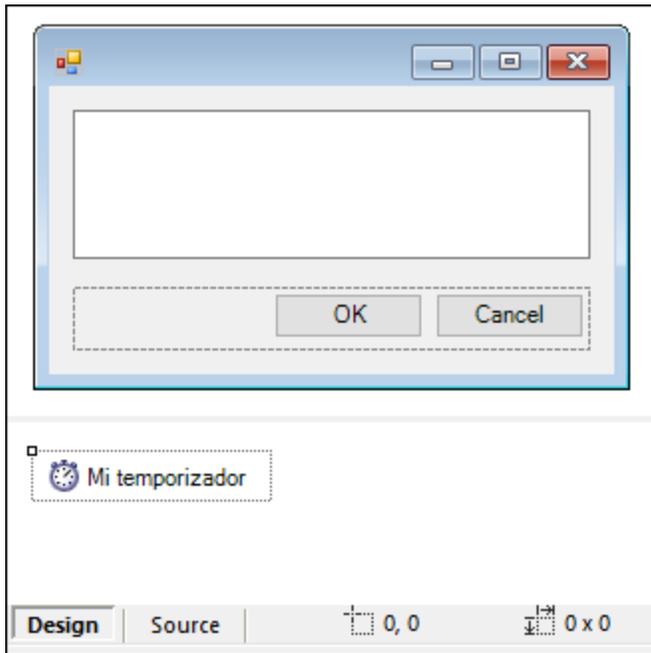
Puede acceder a cualquier componente de un formulario desde el código usando sintaxis de acceso a los campos. Por ejemplo, imagine que tiene un formulario con este diseño:

```
// 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:



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`:

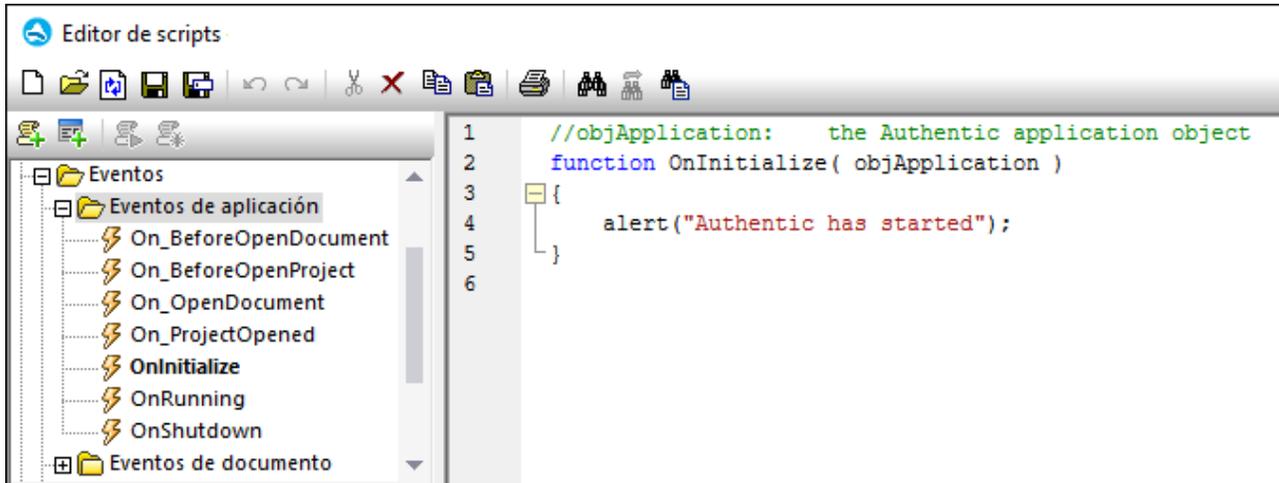
```
var ocx = lastform.AxMediaPlayer1.OCX; // obtener objeto COM subyacente
ocx.enableContextMenu = true;
ocx.URL = "mms://apasf.apa.at/fm4_live_worldwide";
```

### 15.1.1.5 Eventos

Un proyecto de scripting puede incluir scripts que manejen eventos de Authentic Desktop como abrir, cerrar o guardar un documento, iniciar o cerrar Authentic Desktop, añadir un elemento a un diagrama, etc. Estos eventos los suministra la API COM de Authentic Desktop y los puede encontrar en la carpeta "Eventos" del proyecto de scripting. Tenga en cuenta que estos eventos son eventos específicos de Authentic Desktop, en contraposición a los eventos de formulario. Los eventos están organizados en carpetas:

- Eventos de aplicación
- Eventos de documento
- Eventos de la vista Authentic
- Eventos de la vista Cuadrícula
- Eventos de la vista Texto

Para crear un script de controlador de eventos haga clic con el botón derecho en un evento y seleccione **Abrir** en el menú contextual (o haga doble clic en el evento). El script del controlador de eventos aparece en la ventana principal, donde puede editarlo. Por ejemplo, el controlador de eventos de la imagen siguiente muestra una alerta cada vez que se inicia Authentic Desktop:



Tenga en cuenta que:

- El comando `alert` se puede aplicar a JScript. El equivalente en VBScript es `MsgBox`. Véase también [alert](#)<sup>322</sup>.
- No debe cambiar el nombre de la función del controlador de eventos; de lo contrario no se llamará al script del controlador de eventos.
- Para que se procesen eventos debe marcar la casilla *Procesar eventos* al habilitar el proyecto de scripting en Authentic Desktop. Para más información consulte el apartado [Habilitar scripts y macros](#)<sup>331</sup>.

También puede definir variables locales y funciones de ayuda dentro del script del controlador, por ejemplo:

```
var local;

function OnInitialize( objApplication )
{
    local = "OnInitialize";
    Helper();
}

function Helper()
{
    alert("I'm a helper function for " + local);
}
```

### 15.1.1.6 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 Authentic Desktop.

## Parámetros out

Los parámetros out de los métodos de NET Framework requieren variables especiales en JScript. Por ejemplo:

```
var dictionary =
CLR.Create("System.Collections.Generic.Dictionary<System.String, System.String>");
dictionary.Add("1", "A");
dictionary.Add("2", "B");

// use el método de JScript para acceder a los parámetros out
var strOut = new Array(1);
if ( dictionary.TryGetValue("1", strOut) ) // TryGetValue establece el parámetro out
    alert( strOut[0] ); // use out parameter
```

## Números enteros como argumentos

Los métodos .NET que requieren números enteros como argumentos no se deben llamar directamente con los objetos numéricos de JScript, que son valores de puntos flotantes. Por ejemplo, en lugar de:

```
var objCustomColor = CLR.Static("System.Drawing.Color").FromArgb(128,128,128);
```

use:

```
var objCustomColor =
CLR.Static("System.Drawing.Color").FromArgb(Math.floor(128),Math.floor(128),Math.floor(128));
```

## 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 Authentic Desktop (no necesita conocer su valor numérico) puede usar:

```
objExportXMIFileDialog.XMIType = eXMI21ForUML23;
```

### 15.1.1.7 Ejemplo: proyecto de scripting

Authentic Desktop viene con un proyecto de scripting de ejemplo que puede encontrar aquí: **C:\Users\<user>\Documents\Altova\Authentic2023\AuthenticExamples\SampleScripts.asprj**. Este proyecto consiste en unas pocas macros y un formulario de Windows.

**Para cargar el proyecto de scripting en el Editor de scripts:**

1. En el menú **Herramientas** haga clic en **Editor de scripts**.
2. Haga clic en **Abrir** y navegue hasta el archivo **SampleScripts.asprj** desde la ruta indicada más arriba.

El proyecto contiene vaciar macros en el directorio "Macros".

Macro	Descripción
<b>AddMacroMenu</b>	<p>Esta macro añade un elemento de menú nuevo a Authentic Desktop invocando el método <code>Application.AddMacroMenuItem</code> de la API COM. El primer argumento del método <code>Application.AddMacroMenuItem</code> es el nombre de la macro que se quiere añadir (en este ejemplo, "CloseAllButActiveDoc") y el segundo argumento es el texto que aparece con ese elemento del menú.</p> <p>Siempre que se ejecute esta macro se añade un comando de menú nuevo llamado "CloseAllButActiveDoc" al menú <b>Herramientas</b>. Para eliminar los elementos de menú de la macro creados anteriormente puede reiniciar Authentic Desktop o crear una macro que llame al método de la API <code>Application.ClearMacroMenu</code>.</p>

<b>CloseAllButActiveDocument</b>	Cuando se ejecuta, esta macro itera por todos los documentos abiertos en Authentic Desktop y los cierra todos, excepto el que esté activo.
<b>SearchPath</b>	<p>Esta macro muestra un formulario que permite a los usuarios buscar archivos dentro del proyecto actual. El formulario está en el directorio "Formularios", donde puede ver el diseño y los controladores de eventos asociados.</p> <p>El método <code>GetAllPathsFromProject()</code> devuelve como matriz todas las rutas de acceso a archivos que pertenecen al proyecto abierto en ese momento. La definición de este método está en el script <b>GlobalDeclarations</b> del proyecto. El método <code>InsertStringInArrayUnique</code> controla que solamente se añadan rutas de acceso únicas a la matriz. A continuación, el formulario se inicializa con <a href="#">CreateForm</a><sup>327</sup>. Por último, la matriz se convierte en un tipo .NET con el método <a href="#">CreateForm</a><sup>327</sup>, y el formulario se rellena con la colección <code>ArrayList</code> resultante.</p> <p>El botón Abrir del formulario tiene un controlador que llama al método <code>Application.Documents.OpenFile</code> de la API para abrir el archivo seleccionado.</p>

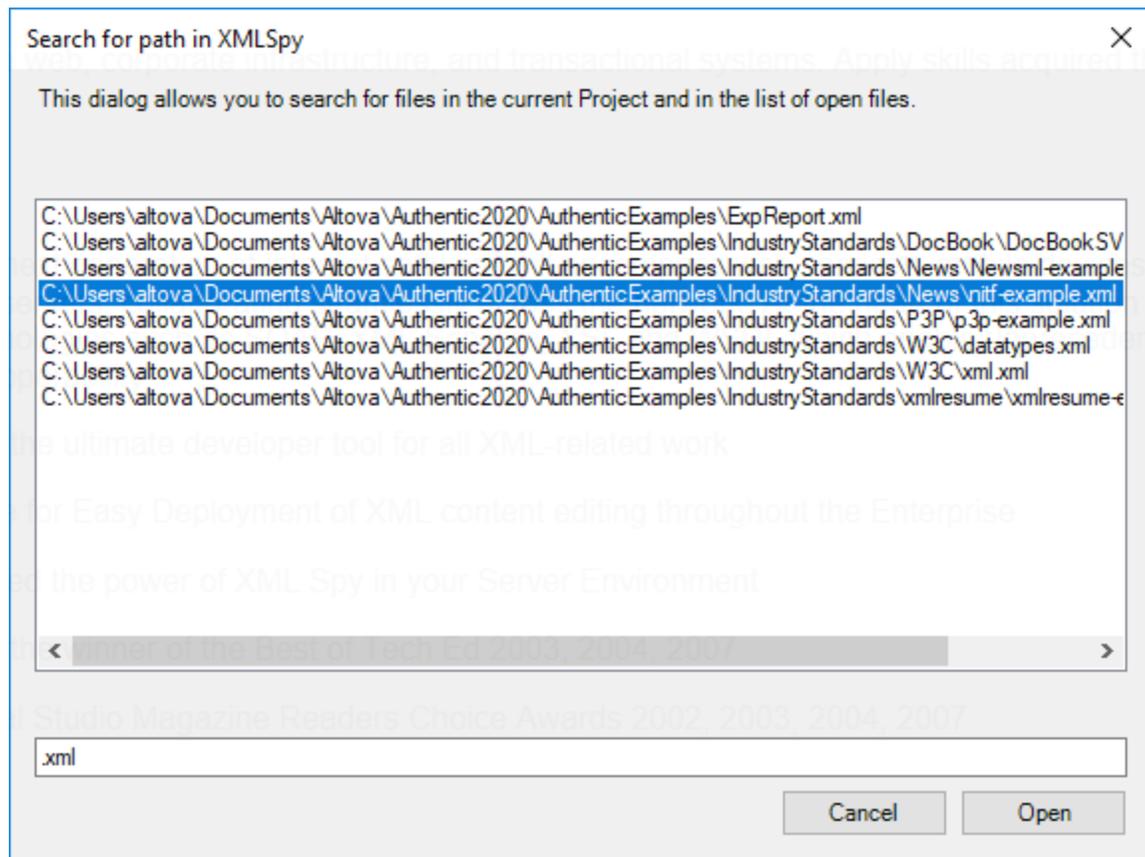
#### Para habilitar el proyecto de scripting como proyecto global de sc de Authentic Desktop:

1. En el menú **Herramientas** haga clic en **Opciones**.
2. Haga clic en la pestaña *Scripting*.
3. En "Archivo de proyecto de scripting global" haga clic en **Examinar** y seleccione el archivo **SampleScripts.asprj** en la ruta indicada más arriba.
4. Este proyecto de scripting no tiene macros o controladores de eventos automáticos, por lo que no necesita marcar las casillas **Ejecutar macros automáticas...** o **Procesar eventos**.
5. Haga clic en **Aplicar**.

En este punto en el menú **Herramientas | Macros** aparecen varios elementos nuevos.

#### Para ejecutar la macro "SearchPath":

1. Abra un proyecto de Authentic Desktop que contenga varios archivos (en este ejemplo, **C:\Users\\Documents\Altova\Authentic2023\AuthenticExamples\Examples.spp**).
2. En el menú Herramientas haga clic en Macros y después haga clic en **Ruta de búsqueda (SearchPath)**.
3. Teclee el término de búsqueda (en este ejemplo, ".xml").



En la imagen se ve cómo aparecen todos los nombres de archivos que contienen el término de búsqueda. Para mostrar uno de esos elementos en el editor principal haga clic en cualquiera de ellos y después haga clic en **Abrir**.

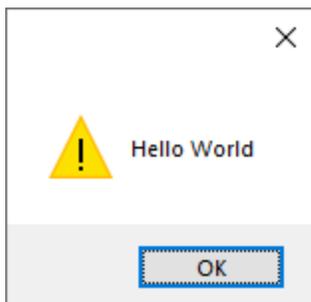
## 15.1.2 Comandos integrados

En este apartado se explican todos los comandos que puede usar el el Editor de scripts de Authentic Desktop.

- [alert](#) <sup>322</sup>
- [confirm](#) <sup>322</sup>
- [CLR.Create](#) <sup>323</sup>
- [CLR.Import](#) <sup>324</sup>
- [CLR.LoadAssembly](#) <sup>324</sup>
- [CLR.ShowImports](#) <sup>325</sup>
- [CLR.ShowLoadedAssemblies](#) <sup>326</sup>
- [CLR.Static](#) <sup>327</sup>
- [CreateForm](#) <sup>327</sup>
- [doevents](#) <sup>328</sup>
- [lastform](#) <sup>329</sup>
- [prompt](#) <sup>329</sup>
- [ShowForm](#) <sup>330</sup>
- [watchdog](#) <sup>331</sup>

### 15.1.2.1 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
```

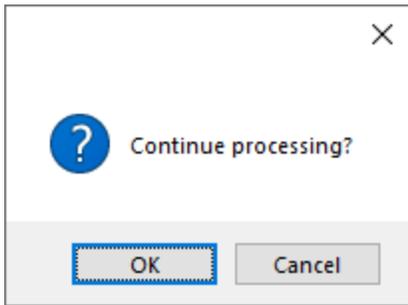
#### Ejemplo

Est código JScript muestra una caja de mensaje con el texto "Hello, World".

```
alert("Hello World");
```

### 15.1.2.2 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
```

## 15.1.2.3 CLR.Create

Creará 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 >>" );
```

### 15.1.2.4 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](#)<sup>324</sup>.

#### Firma

```
CLR.Import(strNamespaceCLR : String) -> void
```

#### Ejemplo

En lugar de tener que usar espacios de nombres completos, como aquí:

```
if ( ShowForm( "FormName" ) == CLR.Static( "System.Windows.Forms.DialogResult" ).OK )
{
    var sName = lastform.textboxFirstName.Text + " " + lastform.textboxLastName.Text;
    CLR.Static( "System.Windows.Forms.MessageBox" ).Show( "Hola " + sName );
}
```

Puede importar espacios de nombres primero y después usar la forma abreviada:

```
CLR.Import( "System.Windows.Forms" );

if ( ShowForm( "FormName" ) == CLR.Static( "DialogResult" ).OK )
{
    var sName = lastform.textboxFirstName.Text + " " + lastform.textboxLastName.Text;
    CLR.Static( "MessageBox" ).Show( "Hola " + sName );
}
```

### 15.1.2.5 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" );
}
```

### 15.1.2.6 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.

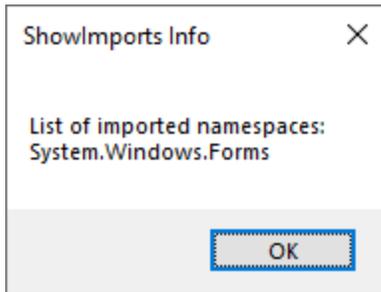
## Firma

```
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();
```



### 15.1.2.7 CLR.ShowLoadedAssemblies

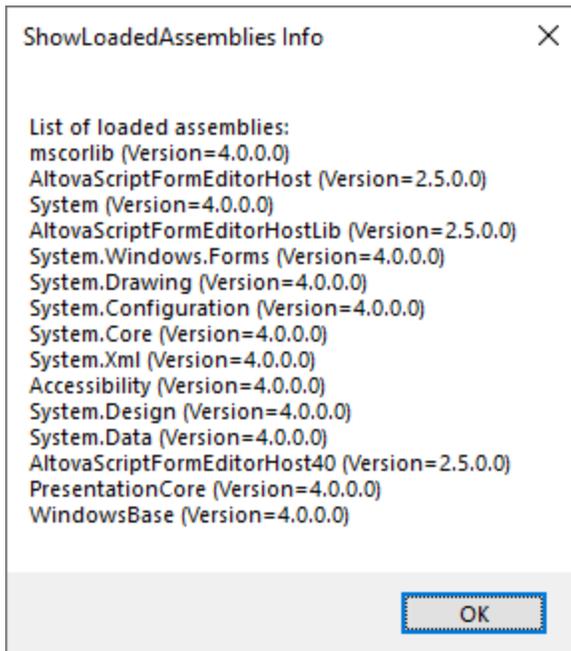
Abre un cuadro de mensaje que muestra los ensamblados cargados en ese momento. El usuario tiene que hacer clic en "Aceptar" para continuar.

#### Firma

```
CLR.ShowLoadedAssemblies() -> void
```

#### Ejemplo

```
CLR.ShowLoadedAssemblies();
```



### 15.1.2.8 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" );
```

### 15.1.2.9 CreateForm

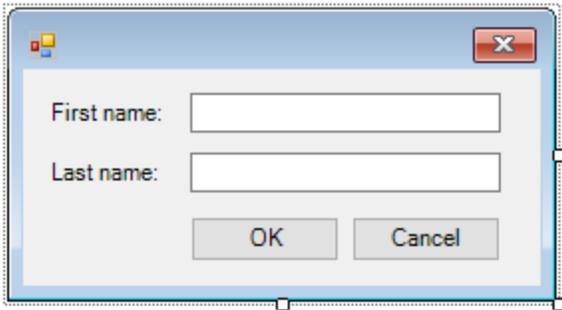
Instancia el objeto formulario `Form` identificado por el nombre dado como argumento. El formulario debe existir en la carpeta "Formularios" del proyecto de scripting. Devuelve el objeto formulario (`System.Windows.Forms.Form`) que corresponde al nombre dato o `null` si no existe ningún formulario con 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").

### 15.1.2.10 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
}
```

### 15.1.2.11 lastform

Este es un campo global que devuelve una referencia al último objeto formulario que se creó con `CreateForm()` o `ShowForm()`.

#### Firma

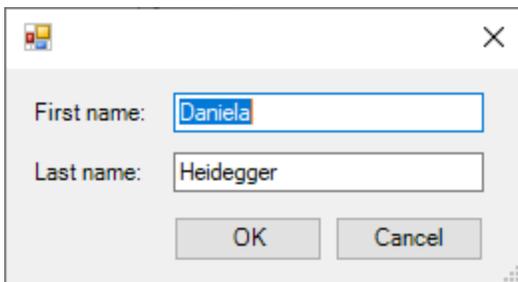
```
lastform -> formObj : System.Windows.Forms.Form
```

#### 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`.



### 15.1.2.12 prompt

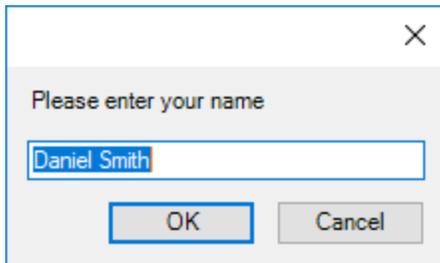
Abre un cuadro de diálogo en el que aparece un mensaje y un control de campo de texto con una respuesta predeterminada. Esto permite al usuario introducir un valor simple de cadena de texto. El valor de retorno es una cadena que contiene el valor del campo de texto o `null` si el usuario seleccionó "Cancelar".

#### Firma

```
prompt(strMessage : String, strDefault : String) -> val : String
```

## Ejemplo

```
var name = prompt( "Please enter your name", "Daniel Smith" );  
if ( name != null )  
    alert( ";Hola, " + name + "!" );
```



### 15.1.2.13 ShowForm

Instancia un objeto formulario nuevo a partir del nombre de formulario dado y lo muestra de inmediato como cuadro de diálogo. El valor de retorno es un número entero que representa el resultado generado:

`DialogResult` (`System.Windows.Forms.DialogResult`). Para ver la lista de posibles valores consulte la documentación de `DialogResult Enum` (<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.dialogresult?view=netframework-4.8>).

## Firma

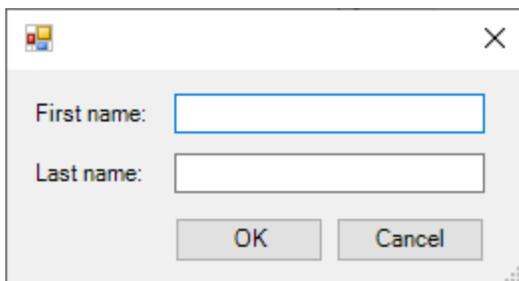
```
ShowForm(strFormName : String) -> result : Integer
```

## Ejemplo

Este código JScript

```
var dialogResult = ShowForm( "FormName" );
```

muestra el formulario "FormName" como cuadro de diálogo:



Así, `DialogResult` se puede seguir evaluando, por ejemplo:

```
if ( dialogResult == CLR.Static( "System.Windows.Forms.DialogResult" ).OK )
    alert( "ok" );
else
    alert( "cancel" );
```

**Nota:** el código de la imagen anterior solamente funciona si la propiedad `DialogResult` de los botones "Aceptar" y "Cancelar" está configurada correctamente en el panel *Propiedades* (por ejemplo, debe ser `OK` para el botón "Aceptar").

### 15.1.2.14 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
```

## 15.1.3 Habilitar scripts y macros

Una vez ha completado y probado un proyecto de scripting puede usarlo de varias maneras:

1. Como proyecto global de scripting para Authentic Desktop. Esto significa que Authentic Desktop puede usar todos los scripts y las macros del proyecto de scripting.
2. A nivel del proyecto. Esto significa que junto al proyecto de Authentic Desktop se guarda también una referencia al archivo `.asprj`. Cuando se abre el proyecto de Authentic Desktop también se puede llamar a los scripts y las macros asociados.

#### Para definir un proyecto de scripting como global:

1. En el menú **Herramientas** haga clic en **Opciones**.
2. Haga clic en la pestaña **Script/Scripting**.

3. Marque la casilla *Activar script* y navegue hasta el archivo `.asprj` para usarlo como proyecto global de scripting.

**Scripting**

Activación

Activar script

Archivo de proyecto de scripting global:

ntic2020\AuthenticExamples\SampleScripts.asprj

Procesamiento de scripts automático

Ejecutar macros automáticas cuando se inicie Authentic Desktop

Procesar eventos

También puede habilitar estas otras opciones de procesamiento de scripts:

<b>Ejecutar macros automáticas cuando se inicie Authentic Desktop</b>	Si marca esta casilla, cualquier macro que esté definida como "automática" en el proyecto se desencadena automáticamente cuando se inicia Authentic Desktop.
<b>Procesar eventos</b>	Marque esta casilla si sus scripts están vinculados a algún evento de aplicación. Desmárquela para evitar que los scripts reaccionen con los eventos.

#### Para habilitar el proyecto de scripting a nivel del proyecto:

1. Abra el proyecto.
2. En el menú **Proyecto** haga clic en **Configuración del proyecto**.
3. Marque la casilla **Activar scripts de proyecto** y navegue hasta el archivo `.asprj`.

La casilla *Ejecutar macros automáticas* funciona como hemos explicado en el punto anterior.

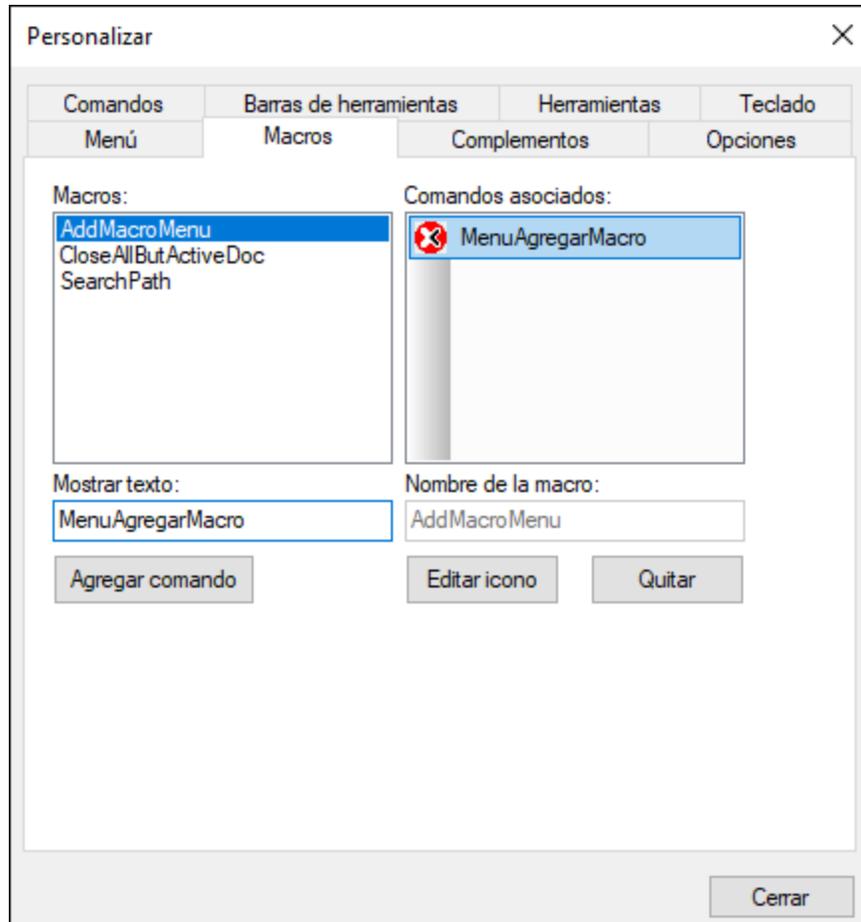
### 15.1.3.1 Ejecutar macros

Cuando un proyecto de scripting está activo en Authentic Desktop, cualquier macro que esté disponible en ese proyecto aparece en el menú **Herramientas | Macros**, por lo que puede ejecutar una macro en cualquier momento con el comando de menú correspondiente, como por ejemplo **Herramientas | Macros | <UnaMacro>**.

Las macros que se hayan configurado como automáticas se ejecutan automáticamente al iniciarse Authentic Desktop, siempre que se haya habilitado este comportamiento en las opciones, tal y como se describe en el apartado [Habilitar scripts y macros](#)<sup>331</sup>.

Para poder acceder más fácilmente a las macros puede crear botones para la barra de herramientas que activen esas macros:

1. En el menú **Herramientas** haga clic en **Personalizar**.
2. Haga clic en la pestaña *Macros*. En la lista aparecen todas las macros disponibles a nivel de la aplicación (en el proyecto global de scripting).
3. Haga clic en **Agregar comando**.



4. También puede hacer clic en **Editar icono** y seleccionar un icono nuevo para esa macro. Para asignar un atajo de teclado para esa macro, vaya a la pestaña *Teclado*.
5. Arrastre la macro desde el panel *Comandos asociados* hasta la barra de herramientas en la que quiere que aparezca.

#### Para eliminar una macro de una barra de herramientas:

1. En el menú **Herramientas** haga clic en **Personalizar**.
2. Haga clic en la pestaña *Macros*.
3. Arrastre la macro desde la barra de herramientas en la que aparece de vuelta hasta el panel *Comandos asociados*.

## 15.2 Complementos de Authentic Desktop para entornos IDE

Authentic Desktop permite crear complementos para entornos IDE e integrarlos en Authentic Desktop.

Puede usar estos complementos para:

- Configurar su versión de Authentic Desktop y agregar comandos a través de menús, iconos, botones, etc.
- Reaccionar a eventos desde Authentic Desktop.
- Ejecutar su propio código dentro de Authentic Desktop con acceso a toda la API de Authentic Desktop

Authentic Desktop espera que el complemento implemente la interfaz [IXMLSpyPlugIn](#)<sup>344</sup>. Con el paquete de instalación vienen ejemplos en VB.NET, C# y C++, los lenguajes compatibles con Authentic Desktop, que se instalan en la carpeta `Authentic2023\AuthenticExamples\IDEPlugin` de la carpeta de instalación:

Windows 7, 8, 10, 10, 11	C:/Users/<usuario>/Documentos
--------------------------	-------------------------------

Consulte el apartado [Archivos de muestra ATL](#)<sup>338</sup> para ver un ejemplo con C++.

### 15.2.1 Cómo registrar complementos para entornos IDE

Authentic Desktop mantiene una clave en el registro en la que almacena todos los complementos registrados para entornos IDE:

```
HKEY_CURRENT_USER\Software\Altova\XML Spy\PlugIns
```

Todos los valores de esta clave se tratan como referencias a los complementos registrados y deben tener este formato:

Nombre del valor:	ProgID del complemento
Tipo de valor:	debe ser REG_SZ
Datos del valor:	CLSID del componente

Cada vez que se inicia la aplicación, se examinan los valores de la clave `PlugIns` y se cargan los complementos que estén registrados.

#### Registrar el complemento a mano

Para registrar un complemento a mano utilice el cuadro de diálogo "Personalizar" del menú **Herramientas** de Authentic Desktop. Con el botón **Agregar complemento...** puede especificar qué DLL implementa el complemento. Authentic Desktop registra el DLL como servidor de COM y añade la correspondiente entrada en su clave `PlugIns`.

Si tiene problemas durante el registro manual, compruebe si el CLSID del complemento coincide con el de la clave `PlugIns`. Si es así, puede que el nombre del DLL del complemento no fuera lo suficientemente único. Utilice otro nombre o registre el complemento directamente (ver más abajo).

## Registrar el complemento directamente

Los complementos se pueden registrar directamente: para ello debe registrar el DLL primero y después agregar el valor que le corresponda en la clave `PlugIns` de Authentic Desktop durante la configuración del complemento, por ejemplo. El nuevo complemento se activará la próxima vez que se inicie Authentic Desktop.

## Crear complementos

En la carpeta `Examples\IDEPlugin` de la carpeta [\(Mis\) Documentos](#)<sup>12</sup> de la aplicación encontrará código fuente para crear ejemplos de complementos en Authentic Desktop. Siga estas instrucciones para crear un complemento a partir de este código fuente:

1. Abra la solución que desea crear como complemento en Visual Studio.
2. Cree el complemento con el comando del menú **Compilación**.
3. El archivo DLL del complemento se creará en la carpeta `Bin` o `Debug`. Este archivo DLL es el que se debe añadir como complemento (ver instrucciones más arriba).

**Nota:** actualmente los lenguajes compatibles son VB.NET, C# y C++.

## 15.2.2 Controles ActiveX

Los controles ActiveX son compatibles y los complementos para entornos IDE que también sean control ActiveX aparecerán en una barra de control de diálogo. Además, en la carpeta `IDEPlugin` de la carpeta `Examples` de la aplicación hay un complemento de muestra que también es un control ActiveX.

## 15.2.3 Configuración XML

El complemento para entornos IDE permite modificar la interfaz de usuario de Authentic Desktop. Esto se hace describiendo cada una de las modificaciones por separado mediante secuencias de datos XML. La configuración XML se pasa a Authentic Desktop con el método [GetUIModifications](#)<sup>349</sup> de la interfaz `IXMLSpyPlugIn`.

El archivo XML que contiene las modificaciones de la interfaz de usuario para el complemento debe tener esta estructura:

```
<ConfigurationData>
  <ImageFile>ruta de acceso del archivo de imagen</ImageFile>
  <Modifications>
    <Modification>
      ...
    </Modification>
    ...
  </Modifications>
</ConfigurationData>
```

Puede definir iconos o botones de barras de herramientas para los nuevos comandos de menú que el complemento añade a la interfaz gráfica de Authentic Desktop. La ruta de acceso del archivo que contiene las imágenes se establece con el elemento `ImageFile`. Cada imagen debe tener 16 x 16 píxels y usar 256 colores

como máximo. Las referencias a la imagen deben organizarse de izquierda a derecha en una sola línea (<ImageFile>...). El valor de índice de la imagen situada más a la derecha es cero.

El elemento `Modifications` puede tener un número indefinido de elementos secundarios `Modification`. Cada elemento `Modification` define un cambio concreto en la interfaz gráfica estándar de Authentic Desktop. A partir de la versión 4.3 también se pueden quitar elementos de la interfaz de Authentic Desktop.

## Estructura de los elementos `Modification`

Todos los elementos `Modification` están compuestos por dos elementos secundarios:

```
<Modification>
  <Action>tipo de acción</Action>
  <UIElement Type="tipo de elemento de interfaz gráfica">
  </UIElement>
</Modification>
```

Estos son los valores válidos para el elemento `Action`:

`Add`: para agregar el elemento a la interfaz de Authentic Desktop

`Hide`: para ocultar el elemento en la interfaz de Authentic Desktop

`Remove`: para quitar el elemento del cuadro de lista *Comandos* del cuadro de diálogo "Personalizar".

También puede combinar varios valores en el elemento `Action` (p. ej. `Hide Remove`).

El elemento `UIElement` describe los elementos de la interfaz gráfica de Authentic Desktop, nuevos o ya existentes. Se trata de estos elementos: barras de herramientas nuevas, botones nuevos, menús o comandos de menús. El atributo **type** define qué elemento es descrito por el elemento XML.

## Elementos secundarios frecuentes de `UIElement`

Los elementos `ID` y `Name` son válidos para todos los tipos diferentes de fragmentos XML `UIElement`. Sin embargo, se puede ignorar uno de los valores para un tipo concreto de `UIElement` (p. ej. `Name` se ignora para un separador).

```
<ID></ID>
<Name></Name>
```

Si `UIElement` describe un elemento actual de la interfaz gráfica, el valor del elemento `ID` viene predefinido en Authentic Desktop. Lo normal es que estos valores del elemento `ID` no sean conocidos. Si el fragmento XML describe una parte nueva de la interfaz gráfica, entonces el `ID` es aleatorio y su valor debería ser menor a 1000.

El elemento `Name` establece el valor textual. Los elementos actuales de la interfaz gráfica se pueden identificar con su nombre (p. ej. menús y elementos de menús). En el caso de elementos nuevos de la interfaz, el elemento `Name` establece el título (p. ej. el título de una barra de herramientas o el texto de un comando de menú).

## Barras de herramientas y menús

Para definir una barra de herramienta hace falta especificar el `ID` y el nombre de la barra de herramientas. Para las barras de herramientas ya existentes se puede indicar el nombre solamente o el `ID`, si se conoce. Para crear una barra de herramientas nueva es necesario indicar ambos valores. El atributo **type** debe ser igual a "ToolBar".

```
<UIElement Type="ToolBar">
  <ID>1</ID>
  <Name>TestPlugIn</Name>
</UIElement>
```

Para especificar un menú de Authentic Desktop son necesarios dos parámetros:

- El ID de la barra de menú que contiene el menú. Si en la ventana principal no hay ningún documento XML abierto, el ID de la barra de menú es 128. Si hay algún documento XML abierto, el ID de la barra de menú es 129.
- El nombre del menú. Los menús no tienen un valor ID asociado. En el fragmento siguiente, por ejemplo, se define el menú Edición de la barra de menú que está activa cuando hay un documento XML abierto como mínimo:

```
<UIElement Type="Menu">
  <ID>129</ID>
  <Name>Edición</Name>
</UIElement>
```

Si quiere crear un menú nuevo hace falta usar un elemento más: el elemento `Place`, que define la posición del nuevo menú en la barra de menús:

```
<UIElement Type="Menu">
  <ID>129</ID>
  <Name>Menú PlugIn</Name>
  <Place>12</Place>
</UIElement>
```

El valor `-1` para el elemento `Place` coloca el botón o menú nuevo al final de la barra de herramientas o del menú respectivamente.

## Comandos

Si añade un comando nuevo, a través de una barra de herramientas o de un comando de menú, el fragmento `UIElement` puede contener cualquiera de estos subelementos:

```
<MacroName></MacroName>
<Info></Info>
<ImageID></ImageID>
```

Si se especifica el elemento `MacroName`, Authentic Desktop busca una macro que tenga el mismo nombre en el entorno de scripting y la ejecuta cada vez que se procese este comando.

El elemento `Info` contiene una breve descripción que aparece en la barra de estado cuando se pasa el puntero del ratón por encima del comando asociado (tanto por el botón como por el comando de menú).

`ImageID` define el índice del icono del archivo de imagen. Recuerde que todos los iconos se almacenan en un solo archivo de imagen.

Para definir un botón de barra de herramientas debe crear un elemento `UIElement` con esta estructura:

```
<UIElement Type="ToolBarItem">
  <!--no volver a utilizar ID locales incluso si los comandos hacen lo mismo-->
```

```

        <ID>5</ID>
        <Name>Abrir archivo del repositorio...</Name>
        <!--Establecer posición en -1 si se trata del primer botón que se debe
insertar-->
        <Place>-1</Place>
        <ImageID>0</ImageID>
        <ToolBarID>1</ToolBarID>
        <!--en lugar del ID de la barra de herramientas también se puede usar el
nombre de la barra de herramientas-->
        <ToolBarName>TestPlugIn</ToolBarName>
    </UIElement>

```

Otros elementos que se pueden utilizar para declarar un botón de barra de herramientas son `Place`, `ToolBarID` y `ToolBarName`. Los elementos `ToolBarID` y `ToolBarName` sirven para identificar la barra de herramientas que contiene el botón nuevo o el botón actual. El valor textual de `ToolBarName` distingue entre mayúsculas y minúsculas. El atributo **type** (del elemento `UIElement`) debe ser igual a "ToolBarItem".

Para definir un comando de menú, además de los elementos estándar, también están los elementos `MenuID`, `Place` y `Parent`. El elemento `MenuID` puede ser 128 o 129. Para más información sobre estos valores consulte el párrafo anterior *Barras de herramientas y menús*.

El elemento `Parent` sirve para identificar el **menú** donde se debe insertar el comando de menú nuevo. Como los comandos de los submenús no tienen un ID único de Windows, necesitamos identificar de otra manera el primario del comando de menú. El valor del elemento `Parent` es una ruta de acceso al comando de menú.

El valor textual del elemento `Parent` debe equivaler al nombre del **menú primario del submenú** y el nombre del submenú debe ir separado por dos puntos. Si el menú no es un primario (porque no es un submenú) añada dos puntos al principio del nombre. El atributo **type** debe tener el valor "MenuItem". Por ejemplo, este elemento `UIElement` define un comando de menú:

```

<UIElement Type="MenuItem">
    <!--este elemento es un ID de comando local-->
    <ID>3</ID>
    <Name>Abrir archivo de repositorio...</Name>
    <Place>-1</Place>
    <MenuID>129</MenuID>
    <Parent>:Menú PlugIn</Parent>
    <ImageID>0</ImageID>
</UIElement>

```

Authentic Desktop permite añadir separadores de barras de herramientas y de menús si el valor del elemento `ID` es 0.

## 15.2.4 Archivos de muestra ATL

Esta sección explica cómo crear un sencillo DLL para el complemento de Authentic Desktop para entornos IDE usando la biblioteca ATL. Para generar el DLL es necesario saber trabajar con la biblioteca ATL, conocer los asistentes que generan objetos ATL nuevos y tener conocimientos de MS Visual Studio.

Para acceder a la API la implementación importa la biblioteca de tipos de Authentic Desktop. El código lee varias propiedades y llama a métodos usando los punteros inteligentes que ofrece la instrucción `#import`.

Además el código de muestra usa la clase MFC `CString` y macros de conversión ATL como `W2T`.

En general, estos son los pasos necesarios para crear un DLL con la biblioteca ATL:

1. Abra VisualStudio y seleccione Nuevo en el menú Archivo.
2. Seleccione la pestaña Proyectos.
3. Seleccione Asistente para proyectos ATL y escriba el nombre del proyecto.
4. Si quiere usar clases MFC o quiere crear un proyecto para el código de muestra, seleccione Compatibilidad para MFC.

Una vez creados los archivos de proyecto puede añadir un objeto ATL para implementar la interfaz `IXMLSpyPlugIn`:

1. En el menú Insertar seleccione Objeto ATL nuevo.
2. En el asistente seleccione Objeto simple y haga clic en Siguiente.
3. Escriba un nombre para el objeto.
4. En la pestaña Atributos seleccione Personal para el tipo de interfaz y deshabilite la opción Agregación.

El resultado es un código base para la implementación de la interfaz del complemento para entornos IDE. Consulte el resto de apartados de esta sección para modificar el código y añadir funciones básicas.

### 15.2.4.1 Descripción de interfaz (IDL)

El IDL del objeto ATL recién creado contiene una declaración para una interfaz COM.

- Esta declaración de interfaz debe reemplazarse con la declaración de `IXMLSpyPlugIn` tal y como aparece más abajo.
- El IDL también debe incluir la definición de la enumeración `SPYUpdateAction`.
- Reemplace el nombre de interfaz predeterminado generado (creado con el asistente) con el nombre `IXMLSpyPlugIn` en la declaración `coclass`. El IDL se debería parecer al código del ejemplo siguiente:

Tras crear el objeto ATL, es necesario implementar la interfaz del complemento de Authentic Desktop para entornos IDE:

```
import "oaidl.idl";
import "ocidl.idl";

// ----- inserte este bloque en su archivo IDL -----
typedef enum {
    spyEnable = 1,
    spyDisable = 2,
    spyCheck = 4,
    spyUncheck = 8
} SPYUpdateAction;
```

```
// ----- fin del bloque insertado -----

// ----- Por, ej. entrada de interfaz generada automáticamente por el asistente ATL -----
// [
//     object,
//     uuid(AB7CD86A-8145-429A-A1F3-270692E08AFC),

//     helpstring("IXMLSpyPlugIn Interface")
//     pointer_default(unique)
// ]
// interface IXMLSpyPlugIn : IUnknown
// {
// };

// ----- fin de la entrada de interfaz generada automáticamente

// ----- reemplazar la entrada de interfaz (ver más arriba) generada por el asistente ATL
// con este bloque -----

[
    odl,
    uuid(88F2A622-4B7E-42CD-8D04-3C0E5389DD85),
    helpstring("IXMLSpyPlugIn Interface")
]
interface IXMLSpyPlugIn : IUnknown
{
    HRESULT _stdcall OnCommand([in] long nID, [in] IDispatch* pXMLSpy);

    HRESULT _stdcall OnUpdateCommand([in] long nID, [in] IDispatch*
pXMLSpy, [out, retval] SPYUpdateAction* pAction);

    HRESULT _stdcall OnEvent([in] long nEventID, [in] SAFEARRAY(VARIANT)*
arrayParameters, [in] IDispatch* pXMLSpy, [out, retval] VARIANT* pReturnValue);

    HRESULT _stdcall GetUIModifications([out, retval] BSTR*
pModificationsXML);

    HRESULT _stdcall GetDescription([out, retval] BSTR* pDescription);
};

// ----- fin del bloque -----

// ----- El código que aparece ahora lo genera automáticamente el asistente ATL y puede ser
// diferente en cada caso -----

[
    uuid(24FE0D1B-3FC0-494E-B36E-1D4CE412B014),
    version(1.0),
    helpstring("XMLSpyIDEPlugInDLL 1.0 Type Library")
]
```

```

library XMLSPYIDEPLUGINDLLLib
{
    importlib("stdole32.tlb");
    importlib("stdole2.tlb");

    [
        uuid(3800E791-7F6B-4ACD-9E32-2AC184444501),
        helpstring("XMLSpyIDEPlugIn Class")
    ]
    coclass XMLSpyIDEPlugIn
    {
        [default] interface IXMLSpyPlugIn; // ----- definir IXMLSpyPlugIn como
interfaz predeterminada -----
    };
};

```

### 15.2.4.2 Definición de clase

En la definición de clase del objeto ATL deben realizarse varios cambios. La clase debe derivarse de IXMLSpyPlugIn, el *mapa de la interfaz* necesita una entrada para IXMLSpyPlugIn y los métodos de la interfaz del complemento para entornos IDE debe declararse:

```

#ifndef __XMLSPYIDEPLUGIN_H_
#define __XMLSPYIDEPLUGIN_H_

#include "resource.h" // símbolos principales

////////////////////////////////////
// CXMLSpyIDEPlugIn
class ATL_NO_VTABLE CXMLSpyIDEPlugIn :
public CComObjectRootEx<CComSingleThreadModel>,
public CComCoClass<CXMLSpyIDEPlugIn, &CLSID_XMLSpyIDEPlugIn>,
public IXMLSpyPlugIn
{
public:
    CXMLSpyIDEPlugIn ()
    {
    }

DECLARE_REGISTRY_RESOURCEID(IDR_XMLSPYIDEPLUGIN)
DECLARE_NOT_AGGREGATABLE(CXMLSpyIDEPlugIn)

DECLARE_PROTECT_FINAL_CONSTRUCT()

BEGIN_COM_MAP(CXMLSpyIDEPlugIn)
    COM_INTERFACE_ENTRY(IXMLSpyPlugIn)
END_COM_MAP()

// IXMLSpyIDEPlugIn
public:
    virtual HRESULT STDMETHODCALLTYPE OnCommand(long nID, IDispatch* pXMLSpy);

```

```

    virtual HRESULT _stdcall OnUpdateCommand(long nID, IDispatch* pXMLSpy,
    SPYUpdateAction* pAction);

    virtual HRESULT _stdcall OnEvent(long nEventID, SAFEARRAY **arrayParameters,
    IDispatch* pXMLSpy, VARIANT* pReturnValue);

    virtual HRESULT _stdcall GetUIModifications(BSTR* pModificationsXML);

    virtual HRESULT _stdcall GetDescription(BSTR* pDescription);
};

#endif // __XMLSPYIDEPLUGIN_H_

```

### 15.2.4.3 Implementación

El código de este apartado muestra una sencilla implementación de un complemento de Authentic Desktop para entornos IDE. Añade un comando de menú y un separador (disponible con Authentic Desktop) al menú **Herramientas**. Dentro del método OnUpdateCommand() el comando nuevo solamente se habilita cuando el documento activo se abre en la vista **Cuadrícula**. El comando busca el elemento XML que está resaltado y abre cualquier dirección URL empezando con "http://" a partir del valor textual del elemento.

```

////////////////////////////////////
// CXMLSpyIDEPlugIn

#import "XMLSpy.tlb"
using namespace XMLSpyLib;

HRESULT CXMLSpyIDEPlugIn::OnCommand(long nID, IDispatch* pXMLSpy)
{
    USES_CONVERSION;

    if(nID == 1) {
        IApplicationPtr ipSpyApp;

        if(pXMLSpy) {
            if(SUCCEEDED(pXMLSpy->QueryInterface(__uuidof(IApplication), (void **)
&ipSpyApp))) {
                IDocumentPtr ipDocPtr = ipSpyApp->ActiveDocument;

                // imaginamos que la vista Cuadrícula está activa
                if(ipDocPtr) {
                    IGridViewPtr ipGridPtr = ipDocPtr->GridView;

                    if(ipGridPtr) {
                        IXMLDataPtr ipXMLData = ipGridPtr->
>CurrentFocus;

                        CString strValue = W2T(ipXMLData->
>TextValue);

```

```

        if(!strValue.IsEmpty() && (strValue.Left(7) ==
_T("http://")))
::ShellExecute(NULL, _T("open"), W2T(ipXMLData->TextValue), NULL, NULL, SW_SHOWNORMAL);
    }
}
}
return S_OK;
}

HRESULT CXMLSpyIDEPlugIn::OnUpdateCommand(long nID, IDispatch* pXMLSpy, SPYUpdateAction*
pAction)
{
    *pAction = spyDisable;

    if(nID == 1) {
        IApplicationPtr ipSpyApp;

        if(pXMLSpy) {
            if(SUCCEEDED(pXMLSpy->QueryInterface(__uuidof(IApplication), (void **)
&ipSpyApp)) {
                IDocumentPtr ipDocPtr = ipSpyApp->ActiveDocument;

                // habilitar solamente si la vista Cuadrícula está activa
                if((ipDocPtr != NULL) && (ipDocPtr->CurrentViewMode ==
spyViewGrid))
                    *pAction = spyEnable;
            }
        }
    }

    return S_OK;
}

HRESULT CXMLSpyIDEPlugIn::OnEvent(long nEventID, SAFEARRAY **arrayParameters, IDispatch*
pXMLSpy, VARIANT* pReturnValue)
{
    return S_OK;
}

HRESULT CXMLSpyIDEPlugIn::GetUIModifications(BSTR* pModificationsXML)
{
    CComBSTR bstrMods = _T(" \
        <ConfigurationData> \
            <Modifications> ");

    // agregar "Abrir URL..." al menú Herramientas
    bstrMods.Append (_T(" \

```

```

        <Modification> \
            <Action>Add</Action> \
            <UIElement type=\"MenuItem\"> \
                <ID>1</ID> \
                <Name>Abrir URL...</Name> \
                <Place>0</Place> \
                <MenuID>129</MenuID> \
                <Parent>:Herramientas</Parent> \
            </UIElement> \
        </Modification>      "));
// agregar separador al menú Herramientas
bstrMods.Append (_T(" \
        <Modification> \
            <Action>Add</Action> \
            <UIElement type=\"MenuItem\"> \
                <ID>0</ID> \
                <Place>1</Place> \
                <MenuID>129</MenuID> \
                <Parent>:Herramientas</Parent> \
            </UIElement> \
        </Modification>      "));
// finalizar descripción de la modificación
bstrMods.Append (_T(" \
        </Modifications> \
        </ConfigurationData>"));

return bstrMods.CopyTo(pModificationsXML);
}

HRESULT CXMLSpyIDEPlugIn::GetDescription(BSTR* pDescription)
{
    CComBSTR    bstrDescr = _T("ATL C++ XMLSpy IDE PlugIn;Este complemento demuestra la
implementación de un DLL ATL sencillo como complemento de XMLSpy para entornos IDE.");
    return bstrDescr.CopyTo(pDescription);
}

```

## 15.2.5 IXMLSpyPlugIn

### Métodos

[OnCommand](#) <sup>345</sup>

[OnUpdateCommand](#) <sup>346</sup>

[OnEvent](#) <sup>346</sup>

[GetUIModifications](#) <sup>349</sup>

[GetDescription](#) <sup>350</sup>

### Descripción

Si se añade un DLL a Authentic Desktop como complemento para entornos IDE, debe registrar un componente COM que responda a una interfaz IXMLSpyPlugIn con el uuid(88F2A622-4B7E-42CD-8D04-3C0E5389DD85) reservado. De lo contrario no se reconocerá que es un complemento.

## 15.2.5.1 OnCommand

**Declaración:** `OnCommand (nID as long, pXMLSpy as IDispatch)`

### Descripción

Al método `OnCommand()` de la implementación de la interfaz se le llama cada vez que se procesa un comando añadido por el complemento, ya sea un comando de menú o un botón de barra de herramientas. `nID` almacena el ID del comando definido por el elemento `ID` del correspondiente elemento `UIElement`.

`pXMLSpy` almacena una referencia a la interfaz de envío del objeto `Application` de Authentic Desktop.

### Ejemplo

```
Public Sub IXMLSpyPlugIn_OnCommand(ByVal nID As Long, ByVal pXMLSpy As Object)
    If (Not (pXMLSpy Is Nothing)) Then
        Dim objDlg
        Dim objDoc As XMLSpyLib.Document
        Dim objSpy As XMLSpyLib.Application
        Set objSpy = pXMLSpy

        If nID = 3 Or nID = 5 Then
            Set objDlg = CreateObject("MSComDlg.CommonDialog")
            objDlg.Filter = "XML Files (*.xml)|*.xml|All Files (*.*)|*.*||"
            objDlg.FilterIndex = 1
            objDlg.ShowOpen

            If Len(objDlg.FileName) > 0 Then
                Set objDoc = objSpy.Documents.OpenFile(objDlg.FileName, False)
                Set objDoc = Nothing
            End If
        End If

        If nID = 4 Or nID = 6 Then
            Set objDlg = CreateObject("MSComDlg.CommonDialog")
            objDlg.Filter = "All Files (*.*)|*.*||"
            objDlg.Flags = cdIOFNPathMustExist
            objDlg.ShowSave

            If Len(objDlg.FileName) > 0 Then
                Set objDoc = objSpy.ActiveDocument

                If Not (objDoc Is Nothing) Then
                    objDoc.SetPathName objDlg.FileName
                    objDoc.Save
                    Set objDoc = Nothing
                End If
            End If
        End If

        Set objSpy = Nothing
    End If
End Sub
```

End Sub

## 15.2.5.2 OnUpdateCommand

**Declaración:** `OnUpdateCommand(nID as long, pXMLSpy as IDispatch) as SPYUpdateAction`

### Descripción

Al método `OnUpdateCommand()` se le llama cada vez que se debe establecer el estado visible de un botón o de un comando de menú. `nID` almacena el ID del comando definido por el elemento `ID` del correspondiente elemento `UIElement`

`pXMLSpy` almacena una referencia a la interfaz de envío del objeto `Application`.

Los valores devueltos posibles para establecer el estado de actualización son:

```
spyEnable           = 1
spyDisable          = 2
spyCheck            = 4
spyUncheck          = 8
```

### Ejemplo

```
Public Function IXMLSpyPlugIn_OnUpdateCommand(ByVal nID As Long, ByVal pXMLSpy As Object)
As SPYUpdateAction
    IXMLSpyPlugIn_OnUpdateCommand = spyDisable

    If (Not (pXMLSpy Is Nothing)) Then
        Dim objSpy As XMLSpyLib.Application
        Set objSpy = pXMLSpy

        If nID = 3 Or nID = 5 Then
            IXMLSpyPlugIn_OnUpdateCommand = spyEnable
        End If
        If nID = 4 Or nID = 6 Then
            If objSpy.Documents.Count > 0 Then
                IXMLSpyPlugIn_OnUpdateCommand = spyEnable
            Else
                IXMLSpyPlugIn_OnUpdateCommand = spyDisable
            End If
        End If
    End If
End Function
```

## 15.2.5.3 OnEvent

**Declaración:** `OnEvent(nEventID as long, arrayParameters as SAFEARRAY(VARIANT), pXMLSpy as IDispatch) as VARIANT`

**Descripción**

Al método `OnEvent()` se le llama cada vez que se genera un evento desde Authentic Desktop.

Los valores posibles para `nEventID` son:

<code>On_BeforeStartEditing</code>	= 1
<code>On_EditingFinished</code>	= 2
<code>On_FocusChanged</code>	= 3
<code>On_Beforedrag</code>	= 4
<code>On_BeforeDrop</code>	= 5
<code>On_OpenProject</code>	= 6
<code>On_OpenDocument</code>	= 7
<code>On_CloseDocument</code>	= 8
<code>On_SaveDocument</code>	= 9

Eventos disponibles desde la versión 4r4 de Authentic Desktop:

<code>On_DocEditDragOver</code>	= 10
<code>On_DocEditDrop</code>	= 11
<code>On_DocEditKeyDown</code>	= 12
<code>On_DocEditKeyUp</code>	= 13
<code>On_DocEditKeyPressed</code>	= 14
<code>On_DocEditMouseMove</code>	= 15
<code>On_DocEditButtonUp</code>	= 16
<code>On_DocEditButtonDown</code>	= 17
<code>On_DocEditContextMenu</code>	= 18
<code>On_DocEditPaste</code>	= 19
<code>On_DocEditCut</code>	= 20
<code>On_DocEditCopy</code>	= 21
<code>On_DocEditClear</code>	= 22
<code>On_DocEditSelectionChanged</code>	= 23

Eventos disponibles desde la versión 2004 de Authentic Desktop:

<code>On_DocEditDragOver</code>	= 10
---------------------------------	------

Eventos disponibles desde la versión 2004r4 de Authentic Desktop (biblioteca de tipos versión 1.4):

On_BeforeOpenProject	= 25
On_BeforeOpenDocument	= 26
On_BeforeSaveDocument	= 27
On_BeforeCloseDocument	= 28
On_ViewActivation	= 29
On_DocEditKeyboardEvent	= 30
On_DocEditMouseEvent	= 31

Eventos disponibles desde la versión 2006 SP1 de Authentic Desktop (biblioteca de tipos versión 1.5):

On_BeforeValidate	= 32
-------------------	------

Eventos disponibles desde la versión 2007 de Authentic Desktop (biblioteca de tipos versión 1.6):

On_BeforeShowSuggestions	= 33
On_ProjectOpened	= 34
On_Char	= 35

Eventos disponibles desde la versión 2009 de Authentic Desktop (biblioteca de tipos versión 2.2):

On_Initialize	= 36
On_Running	= 37
On_Shutdown	= 38

Eventos disponibles desde la versión 2012 de Authentic Desktop (biblioteca de tipos versión 2.8):

On_AuthenticBeforeSave	= 39
On_AuthenticContextMenuActivated	= 40
On_AuthenticLoad	= 41
On_AuthenticToolBarButtonClicked	= 42
On_AuthenticToolBarButtonExecuted	= 43
On_AuthenticUserAddedXMLNode	= 44

Los eventos se llaman igual que en el entorno de scripting de Authentic Desktop. Para los complementos para entornos IDE los nombres utilizados son irrelevantes. Los eventos se identifican con el valor `ID`.

`arrayParameters` es una matriz que se rellena con los parámetros del evento generado actualmente. El orden, el tipo y el significado de cada parámetro está disponible en el entorno de scripting de Authentic Desktop. El módulo Eventos de un proyecto de scripting contiene funciones predefinidas para todos los eventos de todas las versiones anteriores a la versión 4.4. Los parámetros que se pasan a las funciones predefinidas son idénticos a los elementos de matriz del parámetro `arrayParameters`.

Los eventos generados desde la vista Authentic de Authentic Desktop no pasan ningún parámetro directamente. En su lugar se usan objetos de *evento*. A los objetos de evento se puede acceder desde el objeto *Document* del documento activo.

`pXMLSpy` almacena una referencia a la interfaz de envío del objeto `Application` de Authentic Desktop.

Si el valor devuelto de `OnEvent()` está definido, entonces el complemento para entornos IDE no obtendrá este evento después. Tampoco un controlador de eventos dentro del entorno de scripting. Recuerde que todos los complementos para entornos IDE obtienen/procesan el evento antes que el entorno de scripting.

### 15.2.5.4 GetUIModifications

**Declaración:** `GetUIModifications()` as `String`

#### Descripción

Al método `GetUIModifications()` se le llama durante la inicialización del complemento para obtener los datos de configuración XML que definen los cambios en la interfaz gráfica de Authentic Desktop. Al método se le llama cuando el complemento se carga por primera vez y cada vez que se inicia Authentic Desktop.

Consulte el apartado [Configuración XML](#)<sup>335</sup> para ver cómo se modifica la interfaz gráfica.

#### Ejemplo

```
Public Function IXMLSpyPlugIn_GetUIModifications() As String
    ' GetUIModifications() obtiene el archivo XML con las modificaciones de la interfaz
    ' gráfica
    ' del archivo config.xml de la carpeta del complemento
    Dim strPath As String
    strPath = App.Path

    If Len(strPath) > 0 Then
        Dim fso As New FileSystemObject
        Dim file As file

        Set file = fso.GetFile(strPath & "\config.xml")

        If (Not (file Is Nothing)) Then
            Dim stream As TextStream
            Set stream = file.OpenAsTextStream(ForReading)

            ' esto reemplaza el símbolo '**path**' de archivo XML
            ' con la ruta de acceso de instalación del complemento para obtener el archivo
            ' de imagen
            Dim strMods As String
            strMods = stream.ReadAll
            strMods = Replace(strMods, "**path**", strPath)

            IXMLSpyPlugIn_GetUIModifications = strMods
        Else
            IXMLSpyPlugIn_GetUIModifications = ""
        End If
    End If
End Function
```

### 15.2.5.5 GetDescription

**Declaración:** `GetDescription()` as `String`

#### Descripción

`GetDescription()` sirve para definir la cadena *Description* para las entradas del complemento que aparecen en el cuadro de diálogo "Personalizar".

#### Ejemplo

```
Public Function IXMLSpyPlugIn_GetDescription() As String
    IXMLSpyPlugIn_GetDescription = "Complemento de muestra para XMLSpy;Este complemento
demuestra la implementación de un DLL Visual Basic sencillo como complemento para XMLSpy."
End Function
```

## 15.3 Application API

The COM-based API of Authentic Desktop (also called the Application API from now on) enables other applications to use the functionality of Authentic Desktop. As a result, it is possible to automate a wide range of tasks, from validating an XML file to modifying complex XML content (with the [XMLData](#)<sup>593</sup> interface).

Authentic Desktop 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](#)<sup>351</sup> and in the [Overview](#)<sup>352</sup> part of this section)
- From within the built-in Scripting Editor of Authentic Desktop. For a description of the scripting environment, see the section, [Scripting Editor](#)<sup>306</sup>.
- Authentic Desktop allows you to create and integrate your own plug-ins into the application using a special interface for plug-ins. A description of how to create plug-ins is given in the section [IDE Plug-ins](#)<sup>334</sup>.
- Via an ActiveX Control, which is available if the [integration package](#)<sup>618</sup> is installed. For more information, see the section [ActiveX Integration](#)<sup>618</sup>.

### External programs

In the [Overview](#)<sup>352</sup> part of this section, we describe how the functionality of Authentic Desktop can be accessed and automated from external programs.

Using the Application API from outside Authentic Desktop requires an instance of Authentic Desktop to be started first. How this is done depends on the programming language used. See the section, [Programming Languages](#)<sup>353</sup>, for information about individual languages.

Essentially, Authentic Desktop will be started via its COM registration. Then the `Application` object associated with the Authentic Desktop instance is returned. Depending on the COM settings, an object associated with an already running Authentic Desktop can be returned. Any programming language that supports creation and invocation of COM objects can be used. The most common of these are listed below.

- JScript and [VBScript](#)<sup>359</sup> 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#](#)<sup>362</sup> 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](#)<sup>373</sup>: 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 Authentic Desktop may still remain in the system.
- Don't hold references to objects in memory longer than you need them, especially those from the `XMLData` interface. If the user interacts between two calls of your client, then there is no guarantee that these references are still valid.
- Don't forget to disable dialogs if the user interface is not visible.
- See [Error handling in JScript](#) <sup>357</sup> (and in [C#](#) <sup>371</sup> and [Java](#) <sup>380</sup>) for details of how to avoid annoying error messages.
- Free references explicitly if you are using C# or C++.

## This documentation

This documentation section about the Application API is broadly divided into two parts.

- The first part consists of an [Overview](#) <sup>352</sup>, which describes the object model for the API and explains how the API is accessed via various [programming languages](#) <sup>353</sup>.
- The second part is a reference section ([Interfaces](#) <sup>381</sup> and [Enumerations](#) <sup>604</sup>) that contains descriptions of the interface objects of the Application API.

## 15.3.1 Overview

This overview of the Application API is organized as follows:

- [The Object Model](#) <sup>352</sup> describes the relationships between the objects of the Application API.
- [Programming Languages](#) <sup>353</sup> 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.

### 15.3.1.1 Object Model

The starting point for every application which uses the Application API is the [Application](#) <sup>382</sup> 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](#) <sup>353</sup> section.

### XMLSpy.Application or AuthenticDesktop.Application

Authentic Desktop installs a TypeLibrary containing the XMLSpyLib. If this TypeLibrary has been added to the development environment (VB development environment, for example) then an object of the `Application` type can be created with:

```
Set objSpy = New XMLSpyLib.Application
```

If only Authentic Desktop is installed (and not XMLSpy), then

```
Set objSpy = GetObject("", "XMLSpy.Application")
```

does not work, because there won't be any object registered in the Registry with a ProgID of XMLSpy.Application. In this case, the registered object is AuthenticDesktop.Application.

The code listings in this documentation assume that both Authentic Desktop and XMLSpy have been installed. If, however, only Authentic Desktop has been installed, then please modify code fragments to take account of this difference.

The application object consists of the following parts:

1. Document collection and reference to the active document.
2. Reference to current project and methods for creating and opening projects.
3. Methods to support the export to and import from databases, text files, and Word documents.
4. URL management.
5. Methods for macro menu items.

Once you have created an Application object you can start using the functionality of Authentic Desktop. In most cases, you either open a project and access the documents from there or you directly open a document via the [Documents](#)<sup>497</sup> interface.

### 15.3.1.2 Programming Languages

Programming languages differ in the way they support COM access. A few examples for the most frequently used languages (*links below*) will help you get started. The code listings in this section show how basic functionality can be accessed. The files in the API subfolder of the Examples folder can be used to test this functionality:

Windows 7, Windows 8, Windows 10, Windows 11	C:\Users\<<usuario>\Documentos\ Altova\Authentic\2023\%APPNAME%\Examples
---	---

## JScript

The JScript listings demonstrate the following basic functionality:

- [Start application or attach to a running instance](#)<sup>355</sup>
- [Simple document access](#)<sup>356</sup>
- [Iteration](#)<sup>357</sup>
- [Error handling](#)<sup>357</sup>
- [Events](#)<sup>358</sup>

## VBScript

VBScript is different than JScript only syntactically; otherwise it works in the same way. The listings below describe is an example of how VBScript can be used. For more information, refer to the [JScript examples](#)<sup>354</sup>.

- [Events](#)<sup>359</sup>: Shows how events are handled using VBScript.

## C#

C# can be used to access the Application API functionality. The code listings show how to access the API for certain basic functionality.

- [Start Authentic Desktop](#)<sup>368</sup>: Starts Authentic Desktop, which is registered as an automation server, or activates the application if it is already running.
- [Open OrgChart.pxf](#)<sup>369</sup>: Locates one of the example documents installed with Authentic Desktop and opens it. If this document is already open it becomes the active document.
- [OnDocumentOpened Event On/Off](#)<sup>372</sup>: Shows how to listen to Authentic Desktop events. When turned on, a message box will pop up after a document has been opened.
- [Open ExpReport.xml](#)<sup>369</sup>: Opens another example document.
- [Toggle View Mode](#)<sup>370</sup>: Changes the view of all open documents between Browser View and Authentic View. The code shows how to iterate through open documents.
- [Validate](#)<sup>371</sup>: Validates the active document and shows the result in a message box. The code shows how to handle errors and COM output parameters.
- [Shutdown Authentic Desktop](#)<sup>368</sup>: Stops Authentic Desktop.

## Java

The Authentic Desktop API can be accessed from Java code. [The Java sub-section of this section](#)<sup>373</sup> explains how some basic Authentic Desktop functionality can be accessed from Java code. It is organized into the following sub-sections:

- [Mapping Rules for the Java Wrapper](#)<sup>373</sup>
- [Example Java Project](#)<sup>375</sup>
- [Application Startup and Shutdown](#)<sup>378</sup>
- [Simple Document Access](#)<sup>379</sup>
- [Iterations](#)<sup>379</sup>
- [Use of Out-Parameters](#)<sup>380</sup>
- [Event Handlers](#)<sup>380</sup>

### 15.3.1.2.1 JScript

This section contains listings of JScript code that demonstrate the following basic functionality:

- [Start application or attach to a running instance](#)<sup>355</sup>
- [Simple document access](#)<sup>356</sup>
- [Iteration](#)<sup>357</sup>
- [Error handling](#)<sup>357</sup>
- [Events](#)<sup>358</sup>

## 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\Authentic\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>).

### 15.3.1.2.1 Start Application

The JScript below starts the application and shuts it down. If the COM object of the 32-bit Authentic Desktop 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 Authentic Desktop, el nombre registrado o identificador programático (ProgId) del objeto COM es `AuthenticDesktop.Application`. Para la versión de 64 bits de Authentic Desktop, el nombre es `AuthenticDesktop_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 Authentic Desktop. Por eso, si tiene instaladas tanto la versión de 32 bits como la versión de 64 bits de Authentic Desktop 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\Authentic2023\AuthenticExamples\API\JScript\start.js"
```

```
// Initialize application's COM object. This will start a new instance of the application and  
// return its main COM object. Depending on COM settings, the main COM object of an already  
// running application might be returned.
```

```

try {   objAuthentic = WScript.GetObject("", "AuthenticDesktop.Application");   }
catch(err) {}

if( typeof( objAuthentic ) == "undefined" )
{
    try   {   objAuthentic = WScript.GetObject("", "AuthenticDesktop_x64.Application")   }
    catch(err)
    {
        WScript.Echo( "Can't access or create AuthenticDesktop.Application" );
        WScript.Quit();
    }
}

// if newly started, the application will start without its UI visible. Set it to
// visible.
objAuthentic.Visible = true;

WScript.Echo(objAuthentic.Edition + " has successfully started. ");

objAuthentic.Visible = false; // will shutdown application if it has no more COM
//connections
//objAuthentic.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](#) <sup>355</sup>).

### 15.3.1.2.1.2 Simple Document Access

After you have started the application as shown in [Start Application](#) <sup>355</sup>, 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 Authentic Desktop Examples folder and set one of them as the active document.

```

// Locate examples via USERPROFILE shell variable. The path needs to be adapted to major
// release versions.
objWshShell = WScript.CreateObject("WScript.Shell");
majorVersionYear = objAuthentic.MajorVersion + 1998
strExampleFolder = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\\My
Documents\\Altova\\Authentic" + majorVersionYear + "\\AuthenticExamples\\";

// Tell Authentic to open two documents. No dialogs
objDoc1 = objAuthentic.Documents.OpenFile(strExampleFolder + "OrgChart.pxf", false);
objAuthentic.Documents.OpenFile(strExampleFolder + "ExpReport.xml", false);

// The document currently active can be easily located.
objDoc2 = objAuthentic.ActiveDocument;

// Let us make sure that the document is shown in Authentic view.
objDoc2.SwitchViewMode(5); // SPYViewModes.spyViewAuthentic = 5

// Now switch back to the document opened first
objDoc1.SetActiveDocument();

```

The JScript code listed above is available in the sample file `DocumentAccess.js` (see [Example Files](#) <sup>355</sup>).

### 15.3.1.2.1.3 Iteration

The JScript listing below shows how to iterate through the open documents. It is assumed that you have already started the application and opened some documents as shown in the previous sections.

```
// go through all open documents using a JScript Enumerator
bRequiresSaving = false;
for (var iterDocs = new Enumerator(objAuthentic.Documents); !iterDocs.atEnd();
iterDocs.moveNext())
{
    if (iterDocs.item().IsModified)
        bRequiresSaving = true;

    var strErrorText = new Array(1);
    var nErrorNumber = new Array(1);
    var errorData = new Array(1);

    if (!iterDocs.item().IsValid(strErrorText, nErrorNumber, errorData))
    {
        var text = strErrorText;
        // access that XMLData object only if filled in
        if (errorData[0] != null)
            text += "(" + errorData[0].Name + "/" + errorData[0].TextValue + ")";

        WScript.Echo("Document \"" + iterDocs.item().Name + "\" validation error[" +
nErrorNumber + "]: " + text);
    }
    else
    {
        // The COM call succeeded and the document is valid.
        WScript.Echo("Document \"" + iterDocs.item().Name + "\" is valid.");
    }
}

// go through all open documents using index-based access to the document collection
for (i = objAuthentic.Documents.Count; i > 0; i--)
    objAuthentic.Documents.Item(i).Close(false);
```

The JScript code listed above is available in the sample file `DocumentAccess.js` (see [Example Files](#) <sup>355</sup>).

### 15.3.1.2.1.4 Error Handling

The Application API returns errors in two different ways:

- The `HRESULT` returned by every API method
- The `IErrorInfo` interface of the Application API

Every API method returns an `HRESULT`. This return value gives the caller information about errors during execution of the method. If the call was successful, the return value is `S_OK`. The `HRESULT` option is commonly used in C/C++ programs.

However, programming languages such as VisualBasic and scripting languages (and other high-level development environments) don't give the programmer access to the `HRESULT` return of a COM call. Such languages use the `IErrorInfo` interface, which is also supported by the Application API. If an error occurs, the Application API creates a new object that implements the `IErrorInfo` interface. The information provided by the `IErrorInfo` interface is imported by the development environment into its own error-handling mechanism.

For example, the JScript code listing below causes an error to be thrown by incorrectly declaring an array. Additional information about the error object is provided by its properties `number` and `description`.

```
try {
    var arr = new Array(-1);
}
catch (err) {
    WScript.Echo("Error : (" + (err.number & 0xffff) + ") " + err.description);
}
```

### 15.3.1.2.1.5 Events

COM specifies that a client must register itself at a server for callbacks using the connection point mechanism. The automation interface for XMLSpy defines the necessary event interfaces. The way to connect to those events depends on the programming language you use in your client. The following code listing shows how this is done using JScript.

The method `WScript.ConnectObject` is used to receive events.

```
// The event-handler function
function DocEvent_OnBeforeCloseDocument(objDocument)
{
    WScript.Echo("Received event - before closing document");
}

// Create or connect to XMLSpy (or Authentic Desktop)
try
{
    // Create the environment and XMLSpy (or Authentic Desktop)
    objWshShell = WScript.CreateObject("WScript.Shell");
    objFSO = WScript.CreateObject("Scripting.FileSystemObject");
    objSpy = WScript.GetObject("", "XMLSpy.Application");

    // If only Authentic Desktop is installed (and XMLSpy is not installed) use:
    // objSpy = WScript.GetObject("", "AuthenticDesktop.Application")
}
```

```

catch(err)
  { WScript.Echo ("Can't create WScript.Shell object or XMLSpy"); }

// Create document object and connect to its events
objSpy.Visible = true;
majorVersionYear = objSpy.MajorVersion + 1998
docPath = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\\Documents\\Altova\
XMLSpy" + majorVersionYear + "\\Examples\\ExpReport.xml";
objDoc = objSpy.Documents.OpenFile (docPath, false);
WScript.ConnectObject(objDoc, "DocEvent_");

// Keep running while waiting for the event
// In the meanwhile close this document in XMLSpy (or Authentic Desktop) manually
WScript.Echo ("Sleeping for 10 seconds ...");
WScript.Sleep (10000);

objDoc = null;
WScript.Echo ("Stopped listening for event");
objSpy.Quit();

```

### 15.3.1.2.2 VBScript

VBScript is syntactically different than JScript but works in the same way. This section contains a listing showing [how events are used with VBScript](#)<sup>359</sup> and an [example](#)<sup>361</sup>.

For information about other functionality, refer to the JScript examples listed below:

- [Start application or attach to a running instance](#)<sup>355</sup>
- [Simple document access](#)<sup>356</sup>
- [Iteration](#)<sup>357</sup>
- [Error handling](#)<sup>357</sup>

#### 15.3.1.2.2.1 Events

COM specifies that a client must register itself at a server for callbacks using the connection point mechanism. The automation interface for XMLSpy defines the necessary event interfaces. The way to connect to those events depends on the programming language you use in your client. The following code listing shows how this is done using VBScript.

The method `WScript.ConnectObject` is used to receive events.

To run this code, paste it into a file with `.vbs` extension, and either double-click in Windows Explorer, or run it from a command prompt.

```

' the event handler function
Function DocEvent_OnBeforeCloseDocument(objDocument)
  Call WScript.Echo("received event - before closing document")

```

**End Function**

```

' create or connect to XmlSpy
Set objWshShell = WScript.CreateObject("WScript.Shell")
Set objFSO = WScript.CreateObject("Scripting.FileSystemObject")
Set objSpy = WScript.GetObject("", "XMLSpy.Application")
' If only Authentic is installed (and XMLSpy is not installed) use:
' Set objSpy = WScript.GetObject("", "AuthenticDesktop.Application")
' If only XMLSpy 64-bit is installed, use:
' Set objSpy = WScript.GetObject("", "XMLSpy_x64.Application")

' create document object and connect to its events
objSpy.Visible = True

' Find out user's personal folder and locate one of the installed examples.
personalFolder = objWshShell.ExpandEnvironmentStrings("%UserProfile%")
majorVersionYear = objSpy.MajorVersion + 1998
xmlspyExamplesFolder = personalFolder & "\Documents\Altova\XMLSpy" & majorVersionYear &
"\Examples\"
docPath = xmlspyExamplesFolder & "ExpReport.xml"

' open a document
Set objDoc = objSpy.Documents.OpenFile (docPath, False)
Call WScript.ConnectObject(objDoc, "DocEvent_")

' keep running while waiting on the event
' in the meantime close the document in XMLSPY manually
Call WScript.Echo ("sleeping for 10 seconds ...")
Call WScript.Sleep (10000)

Set objDoc = Nothing
Call WScript.Echo ("stopped listening for event")
Call objSpy.Quit

```

**Nota:** en la versión de 32 bits de Authentic Desktop, el nombre registrado o identificador programático (ProgId) del objeto COM es `AuthenticDesktop.Application`. Para la versión de 64 bits de Authentic Desktop, el nombre es `AuthenticDesktop_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 Authentic Desktop. Por eso, si tiene instaladas tanto la versión de 32 bits como la versión de 64 bits de Authentic Desktop 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\Authentic2023\AuthenticExamples\API\JScript\start.js"

```

### 15.3.1.2.2.2 Example: Using Events

Authentic View supports event connection on a per-object basis. Implementation of this feature is based on COM connection points and is available in environments that support this mechanism.

The following example is a VBScript code example that shows how to use events from within a VBScript project.

```
' -----
' VBScript example that demonstrates how to use events.
' -----

' Event handler for OnSelectionChanged event of AuthenticView
Function AuthenticViewEvent_OnSelectionChanged(objAuthenticRange)
    If objAuthenticRange.FirstTextPosition <> objAuthenticRange.LastTextPosition Then
        Call WScript.Echo("Selection: " & objAuthenticRange.Text & vbNewLine & vbNewLine
& "Close this dialog.")
    Else
        Call WScript.Echo("Cursor position: " & objAuthenticRange.FirstTextPosition &
vbNewLine & vbNewLine & "Close this dialog.")
    End If
End Function

' Start/access XMLSpy and connect to its automation interface.
Set WshShell = WScript.CreateObject("WScript.Shell")
Set objSpy = GetObject("", "XMLSpy.Application")
' Make the UI of XMLSpy visible.
objSpy.Visible = True

' Find out user's personal folder and locate one of the installed XMLSpy examples.
personalFolder = WshShell.ExpandEnvironmentStrings("%UserProfile%")
majorVersionYear = objSpy.MajorVersion + 1998
xmlspyExamplesFolder = personalFolder & "\Documents\Altova\XMLSpy" & majorVersionYear &
"\Examples\"
docPath = xmlspyExamplesFolder & "ExpReport.xml"

' Create object to access windows file system and test if the our document exists.
Set fso = CreateObject("Scripting.FileSystemObject")
If fso.FileExists(docPath) Then
    ' open the document
    Call objSpy.Documents.OpenFile(docPath, False)
    set objDoc = objSpy.ActiveDocument

    ' switch active document to authentic view
    objDoc.SwitchViewMode 4 ' spyViewAuthentic

    ' Register for connection point events on the authentic view of the active document.
    ' Any function with a valid event name prefixed with "AuthenticViewEvent_" will
    ' be called when the corresponding event gets triggered on the specified object.
    set objView = objDoc.AuthenticView
    Call WScript.ConnectObject(objView, "AuthenticViewEvent_")
    Call WScript.Echo("Events are connected." & vbNewLine & vbNewLine & "Now set or move
```

```

the cursor in XMLSpy." & vbNewLine & vbNewLine & "Close this dialog to shut down
XMLSpy.")

' To disconnect from the events delete the reference to the object.
set objView = Nothing
Else
Call WScript.Echo("The file " & docPath & " does not exist.")
End If

' shut down XMLSpy when this script ends
objSpy.Visible = False

```

### 15.3.1.2.3 C#

The C# programming language can be used to access the Application API functionality. You could use Visual Studio 2012/2013/2015/2017/2019/2022 to create the C# code, saving it in a Visual Studio project. Create the project as follows:

1. In Microsoft Visual Studio, add a new project using **File | New | Project**.
2. Add a reference to the Authentic Desktop Type Library by clicking **Project | Add Reference**. The Add Reference dialog appears. Browse for the Authentic Desktop Type Library component, which is located in the Authentic Desktop application folder, and add it.
3. Enter the code you want.
4. Compile the code and run it.

### Example C# project

Your Authentic Desktop 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\Authentic\2023\%APPNAME%\Examples
---	---

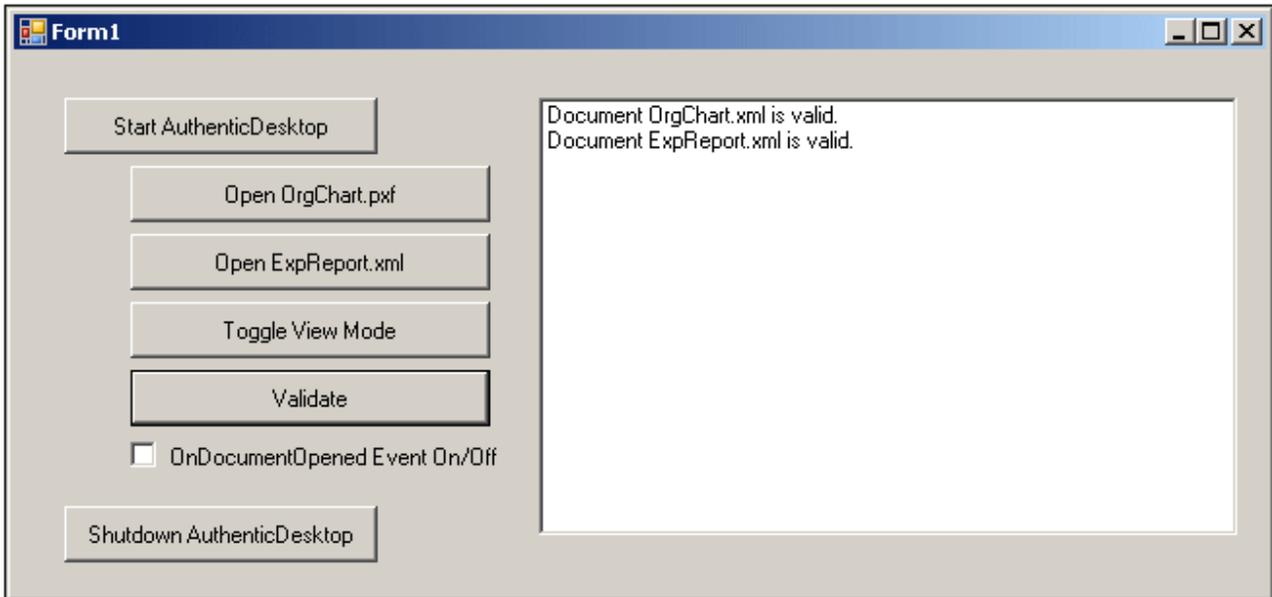
You can compile and run the project from within Visual Studio 2012/2013/2015/2017/2019/2022. The code listing below shows how basic application functionality can be used. This code is similar to the example C# project in the API Examples folder of your application package, but might differ slightly.

### Platform configuration

If you have a 64-bit operating system and are using a 32-bit installation of Authentic Desktop, 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 Authentic Desktop operations:



- [Start Authentic Desktop](#)<sup>368</sup>: Starts Authentic Desktop, which is registered as an automation server, or activates the application if it is already running.
- [Open OrgChart.pxf](#)<sup>369</sup>: Locates one of the example documents installed with Authentic Desktop and opens it. If this document is already open it becomes the active document.
- [Open ExpReport.xml](#)<sup>369</sup>: Opens another example document.
- [Toggle View Mode](#)<sup>370</sup>: Changes the view of all open documents between Text View and Authentic View. The code shows how to iterate through open documents.
- [Validate](#)<sup>371</sup>: Validates the active document and shows the result in a message box. The code shows how to handle errors and COM output parameters.
- [Shut down Authentic Desktop](#)<sup>368</sup>: Stops Authentic Desktop.

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 `AutomateAuthenticDesktop_VS2008.sln` or the file `AutomateAuthenticDesktop_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 `AutomateAuthenticDesktop` 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 AuthenticDesktop is accessed via its automation interface.
        XMLSpyLib.Application AuthenticDesktop;

        // Location of examples installed with AuthenticDesktop
        String strExamplesFolder;

        private void Form1_Load(object sender, EventArgs e)
        {
            // Locate examples installed with AuthenticDesktop.
            // REMARK: You might need to adapt this if you have a different major version
of the product.
            strExamplesFolder = Environment.GetEnvironmentVariable("USERPROFILE") + "\\My
Documents\\Altova\\Authentic2012\\AuthenticExamples\\";
        }

        // Handler for the "Start AuthenticDesktop" button
        private void StartAuthenticDesktop_Click(object sender, EventArgs e)
        {
            if (AuthenticDesktop == null)
            {
                Cursor.Current = Cursors.WaitCursor;

                // If there is no AuthenticDesktop instance, create one and make it
visible.
                AuthenticDesktop = new XMLSpyLib.Application();
                AuthenticDesktop.Visible = true;

                Cursor.Current = Cursors.Default;
            }
            else
            {
                // If an AuthenticDesktop instance is already running, make sure it's
visible.
                if (!AuthenticDesktop.Visible)
                    AuthenticDesktop.Visible = true;
            }
        }

        // Handler for the "Open OrgChart.pxf" button
        private void openOrgChart_Click(object sender, EventArgs e)

```

```
    {
        // Make sure there's a running Authentic Desktop instance, and that it's
visible
        StartAuthenticDesktop_Click(null, null);

        // Open a sample file installed with the product.
        AuthenticDesktop.Documents.OpenFile(strExamplesFolder + "OrgChart.pxf", false);
    }

    // Handler for the "Open ExpReport.xml" button
    private void openExpReport_Click(object sender, EventArgs e)
    {
visible
        // Make sure there's a running Authentic Desktop instance, and that it's
        StartAuthenticDesktop_Click(null, null);

        // Open a sample file installed with the product.
        AuthenticDesktop.Documents.OpenFile(strExamplesFolder + "ExpReport.xml",
false);
    }

    // Handler for the "Toggle View Mode" button
    private void toggleView_Click(object sender, EventArgs e)
    {
visible
        // Make sure there's a running Authentic Desktop instance, and that it's
        StartAuthenticDesktop_Click(null, null);

        // Iterate through all open documents and toggle the current view between Text
View and Authentic View.
        foreach (XMLSpyLib.Document doc in AuthenticDesktop.Documents)
            if (doc.CurrentViewMode == XMLSpyLib.SPYViewModes.spyViewAuthentic)
                doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewBrowser);
            else
                doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewAuthentic);
    }

    // Handler for the "Shut down AuthenticDesktop" button
    // Shut down application instance by explicitly releasing the COM object.
    private void shutdownAuthenticDesktop_Click(object sender, EventArgs e)
    {
        if (AuthenticDesktop != null)
        {
            // Allow shut down of AuthenticDesktop by releasing the UI
            AuthenticDesktop.Visible = false;

            // Explicitly release the COM object
            try
            {
                while
(System.Runtime.InteropServices.Marshal.ReleaseComObject(AuthenticDesktop) > 0) ;
            }
            finally
            {
                // Avoid subsequent access to this object.
                AuthenticDesktop = null;
            }
        }
    }
}
```

```

    }
}

// Handler for the "Validate" button
private void validate_Click(object sender, EventArgs e)
{
    // COM errors get returned to C# as exceptions. Use a try/catch block to handle
them.
    try
    {
        // Method 'IsValid' is one of the few functions that use output parameters.
        // Use 'object' type for these parameters.
        object strErrorText = "";
        object nErrorNumber = 0;
        object errorData = null;

        if (!AuthenticDesktop.ActiveDocument.IsValid(ref strErrorText, ref
nErrorNumber, ref errorData))
        {
            // The COM call succeeds but the document is not valid.
            // A detailed description of the problem is returned in strErrorText,
nErrorNumber and errorData.
            listBoxMessages.Items.Add("Document " +
AuthenticDesktop.ActiveDocument.Name + " is not valid.");
            listBoxMessages.Items.Add("\tErrorText : " + strErrorText);
            listBoxMessages.Items.Add("\tErrorNumber: " + nErrorNumber);
            listBoxMessages.Items.Add("\tElement : " + (errorData != null ?
((XMLSpyLib.XMLData)errorData).TextValue : "null"));
        }
        else
        {
            // The COM call succeeds and the document is valid.
            listBoxMessages.Items.Add("Document " +
AuthenticDesktop.ActiveDocument.Name + " is valid.");
        }
    }
    catch (Exception ex)
    {
        // The COM call was not successful.
        // Probably no application instance has been started or no document is
open.
        listBoxMessages.Items.Add("Error validating active document: " +
ex.Message);
    }

    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 OnDocumentOpened event
    private void handleOnDocumentOpened(XMLSpyLib.Document i_ipDocument)
    {
        String sText = "";

        if (i_ipDocument.Name.Length > 0)
            sText = "Document " + i_ipDocument.Name + " was opened!";
        else
            sText = "An empty document was created.";

        // 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 checkBoxEventOnOff_CheckedChanged(object sender, EventArgs e)
    {
        if (AuthenticDesktop != null)
        {
            if (checkBoxEventOnOff.Checked)
                AuthenticDesktop.OnDocumentOpened += new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);
            else
                AuthenticDesktop.OnDocumentOpened -= new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);
        }
    }
}

```

### 15.3.1.2.3.1 Add Reference to Authentic Desktop 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 `Authentic.tlb`, and is located in the Authentic Desktop application folder.

Then declare a variable to access the Authentic Desktop API:

```
// An instance of Authentic Desktop is accessed via its automation interface.
XMLSpyLib.Application Authentic Desktop;
```

### 15.3.1.2.3.2 Application Startup and Shutdown

In the code snippets below, the methods `StartAuthenticDesktop_Click` and `ShutdownAuthenticDesktop_Click` are those assigned to buttons in the [AutomateAuthenticDesktop example](#)<sup>362</sup> 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\Authentic\2023\%APPNAME%\Examples
---	---

You can compile and run the project from within Visual Studio 2012/2013/2015/2017/2019/2022.

### Starting Authentic Desktop

The following code snippet from the [AutomateAuthenticDesktop example](#)<sup>362</sup> shows how to start up the application.

```
// Handler for the "Start AuthenticDesktop" button
private void StartAuthenticDesktop_Click(object sender, EventArgs e)
{
    if (AuthenticDesktop == null)
    {
        Cursor.Current = Cursors.WaitCursor;

        // If there is no AuthenticDesktop instance, create one and make it
visible.
        AuthenticDesktop = new XMLSpyLib.Application();
        AuthenticDesktop.Visible = true;

        Cursor.Current = Cursors.Default;
    }
    else
    {
        // If an instance of Authentic Desktop is already running, make sure it's
visible
        if (!AuthenticDesktop.Visible)
            AuthenticDesktop.Visible = true;
    }
}
```

### Shutting down Authentic Desktop

The following code snippet from the [AutomateAuthenticDesktop example](#)<sup>362</sup> shows how to shut down the application.

```

// Handler for the "Shut down AuthenticDesktop" button
// Shut down application instance by explicitly releasing the COM object.
private void shutdownAuthenticDesktop_Click(object sender, EventArgs e)
{
    if (AuthenticDesktop != null)
    {
        // Allow shut down of AuthenticDesktop by releasing the UI
        AuthenticDesktop.Visible = false;

        // Explicitly release the COM object
        try
        {
            while
(System.Runtime.InteropServices.Marshal.ReleaseComObject(AuthenticDesktop) > 0) ;
        }
        finally
        {
            // Avoid subsequent access to this object.
            AuthenticDesktop = null;
        }
    }
}

```

### 15.3.1.2.3.3 Opening Documents

The code snippets below (from the [AutomateAuthenticDesktop example](#)<sup>362</sup>) 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.OpenFile\(string, boolean\)](#)<sup>500</sup>.

The [AutomateAuthenticDesktop example](#)<sup>362</sup> (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\Authentic\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 Authentic Desktop instance, and that it's
visible
    StartAuthenticDesktop_Click(null, null);

    // Open a sample file installed with the product.
    AuthenticDesktop.Documents.OpenFile(strExamplesFolder + "OrgChart.pxf", false);
}

```

```

    }

    // Handler for the "Open ExpReport.xml" button
    private void openExpReport_Click(object sender, EventArgs e)
    {
        // Make sure there's a running Authentic Desktop instance, and that it's
visible
        StartAuthenticDesktop_Click(null, null);

        // Open a sample file installed with the product.
        AuthenticDesktop.Documents.OpenFile(strExamplesFolder + "ExpReport.xml",
false);
    }

```

The file opened last will be the active file.

#### 15.3.1.2.3.4 Iterating through Open Documents

The code snippet below (from the [AutomateAuthenticDesktop example](#)<sup>362</sup>; see the file *Form1.cs*) shows how to iterate through open documents. A condition is then tested within the iteration loop, and the document view is switched between Browser View and Authentic View.

```

// Handler for the "Toggle View Mode" button
private void toggleView_Click(object sender, EventArgs e)
{
    // Make sure there's a running Authentic Desktop instance, and that it's
visible
    StartAuthenticDesktop_Click(null, null);

    // Iterate through all open documents and toggle the current view between
Browser View and Authentic View.
    foreach (XMLSpyLib.Document doc in AuthenticDesktop.Documents)
        if (doc.CurrentViewMode == XMLSpyLib.SPYViewModes.spyViewAuthentic)
            doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewBrowser);
        else
            doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewAuthentic);
}

```

The [AutomateAuthenticDesktop example](#)<sup>362</sup> example is located in the C# subfolder of the API Examples folder:

Windows 7, Windows 8, Windows 10, Windows 11	C:\Users\<<usuario>\Documentos\ Altova\Authentic\2023\%APPNAME%\Examples
---	---

You can compile and run the project from within Visual Studio 2012/2013/2015/2017/2019/2022.

### 15.3.1.2.3.5 Errors and COM Output Parameters

The code snippet below (from the [AutomateAuthenticDesktop example](#)<sup>362</sup>) shows how to handle errors and COM output parameters. The method [AuthenticDesktop.ActiveDocument.IsValid\(ref strErrorText, ref nErrorNumber, ref errorData\)](#)<sup>487</sup> uses output parameters that are used, in the code snippet below, to generate an error-message text.

The [AutomateAuthenticDesktop example](#)<sup>362</sup> (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\Authentic\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 "Validate" button
private void validate_Click(object sender, EventArgs e)
{
    // COM errors get returned to C# as exceptions. Use a try/catch block to handle
    them.
    try
    {
        // Method 'IsValid' is one of the few functions that use output parameters.
        // Use 'object' type for these parameters.
        object strErrorText = "";
        object nErrorNumber = 0;
        object errorData = null;

        if (!AuthenticDesktop.ActiveDocument.IsValid(ref strErrorText, ref
nErrorNumber, ref errorData))
        {
            // The COM call succeeds but the document is not valid.
            // A detailed description of the problem is returned in strErrorText,
nErrorNumber and errorData.
            listBoxMessages.Items.Add("Document " +
AuthenticDesktop.ActiveDocument.Name + " is not valid.");
            listBoxMessages.Items.Add("\tErrorText : " + strErrorText);
            listBoxMessages.Items.Add("\tErrorNumber: " + nErrorNumber);
            listBoxMessages.Items.Add("\tElement      : " + (errorData != null ?
((XMLSpyLib.XMLData)errorData).TextValue : "null"));
        }
        else
        {
            // The COM call succeeds and the document is valid.
            listBoxMessages.Items.Add("Document " +
AuthenticDesktop.ActiveDocument.Name + " is valid.");
        }
    }
}
```

```

        catch (Exception ex)
        {
            // The COM call was not successful.
            // Probably no application instance has been started or no document is
open.
            listBoxMessages.Items.Add("Error validating active document: " +
ex.Message);
        }
    }
}

```

### 15.3.1.2.3.6 Events

The code snippet below (from the [AutomateAuthenticDesktop example](#)<sup>362</sup>) lists the code for two event handlers. The [AutomateAuthenticDesktop example](#)<sup>362</sup> (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\Authentic\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 OnDocumentOpened event
private void handleOnDocumentOpened(XMLSpyLib.Document i_ipDocument)
{
    String sText = "";

    if (i_ipDocument.Name.Length > 0)
        sText = "Document " + i_ipDocument.Name + " was opened!";
    else

```

```

        sText = "An empty document was created.";

        // 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 checkBoxEventOnOff_CheckedChanged(object sender, EventArgs e)
    {
        if (AuthenticDesktop != null)
        {
            if (checkBoxEventOnOff.Checked)
                AuthenticDesktop.OnDocumentOpened += new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);
            else
                AuthenticDesktop.OnDocumentOpened -= new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);
        }
    }

```

#### 15.3.1.2.4 Java

The Application API can be accessed from Java code. To allow accessing the Authentic Desktop automation server directly from Java code, the libraries listed below must reside in the `classpath`. They are installed in the folder: `JavaAPI` in the Authentic Desktop 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
- `AuthenticAPI.jar`: Java classes that wrap the Authentic Desktop automation interface
- `AuthenticAPI_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](#)<sup>375</sup>.

#### 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 Authentic Desktop 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 <code>SetEncoding</code>	<code>setFileEncoding</code>
AuthenticView, method <code>Goto</code>	<code>gotoElement</code>
AuthenticRange, method <code>Goto</code>	<code>gotoElement</code>
AuthenticRange, method <code>Clone</code>	<code>cloneRange</code>

## This section

This section explains how some basic Authentic Desktop functionality can be accessed from Java code. It is organized into the following sub-sections:

- [Example Java Project](#) <sup>375</sup>
- [Application Startup and Shutdown](#) <sup>378</sup>
- [Simple Document Access](#) <sup>379</sup>
- [Iterations](#) <sup>379</sup>
- [Use of Out-Parameters](#) <sup>380</sup>
- [Event Handlers](#) <sup>380</sup>

### 15.3.1.2.4.1 Example Java Project

The Authentic Desktop 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\Authentic\2023\%APPNAME%\Examples
---	---

This folder contains Java examples for the Authentic Desktop 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)`.

<code>AltovaAutomation(_x64).dll</code>	Java-COM bridge: DLL part
<code>AltovaAutomation.jar</code>	Java-COM bridge: Java library part
<code>AuthenticAPI.jar</code>	Java classes of the Authentic Desktop API
<code>RunAuthenticDesktop.java</code>	Java example source code
<code>BuildAndRun.bat</code>	Batch file to compile and run example code from the command line prompt. Expects folder where Java Virtual Machine resides as parameter.
<code>.classpath</code>	Eclipse project helper file
<code>.project</code>	Eclipse project file
<code>Authentic_JavaDoc.zip</code>	Javadoc file containing help documentation for the Java API

#### What the example does

The example starts up Authentic Desktop and performs a few operations, including opening and closing documents. When done, Authentic Desktop stays open. You must close it manually.

- [Start Authentic Desktop](#)<sup>378</sup>: Starts Authentic Desktop, which is registered as an automation server, or activates Authentic Desktop if it is already running.
- [Open example files](#)<sup>379</sup>: Locates example documents installed with Authentic Desktop and opens them.
- [Iteration and Changing the View Mode](#)<sup>379</sup>: Changes the view of all open documents to Browser View. The code also shows how to iterate through open documents.
- [Iteration, validation, output parameters](#)<sup>380</sup>: Validates the active document and shows the result in a message box. The code shows how to use output parameters.
- [Event Handling](#)<sup>380</sup>: Shows how to handle Authentic Desktop events.
- [Shut down Authentic Desktop](#)<sup>378</sup>: Shuts down Authentic Desktop.

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 `RunAuthenticDesktop.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 `RunAuthenticDesktop` 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 `RunAuthenticDesktop.java` is listed below with comments.

```
01 // Access general JAVA-COM bridge classes
02 import com.altova.automation.libs.*;
03
04 // Access AuthenticDesktop Java-COM bridge
05 import com.altova.automation.AuthenticDesktop.*;
06 import com.altova.automation.AuthenticDesktop.Enums.SPYViewModes;
07
08 /**
09  * A simple example that starts AuthenticDesktop COM server and performs a view
10  * operations on it.
11  * Feel free to extend.
12  */
13 public class RunAuthenticDesktop
14 {
15     public static void main(String[] args)
16     {
17         // An instance of the application.
18         Application authenticDesktop = null;
19
20         // Instead of COM error-handling, use Java exception mechanism.
21         try
22         {
23             // Start AuthenticDesktop as COM server.
24             authenticDesktop = new Application();
25             // COM servers start up invisible so we make it visible
26             authenticDesktop.setVisible(true);
27
28             // Locate samples installed with the product.
```

```
28     String strExamplesFolder = System.getenv("USERPROFILE") + "\\My Documents\\Altova\\
\\Authentic2012\\AuthenticExamples\\";
29
30     // Open two files from the product samples.
31     authenticDesktop.getDocuments().openFile(strExamplesFolder + "OrgChart.pxf",
false);
32     authenticDesktop.getDocuments().openFile(strExamplesFolder + "ExpReport.xml",
false);
33
34     // Iterate through all open documents and set the View Mode to 'Text'.
35     for (Document doc:authenticDesktop.getDocuments())
36         if ( doc.getCurrentViewMode() != SPYViewModes.spyViewText)
37             doc.switchViewMode(SPYViewModes.spyViewText);
38
39     // An alternative iteration mode is index-based. COM indices are typically zero-
based.
40     Documents documents = authenticDesktop.getDocuments();
41     for (int i = 1; i <= documents.getCount(); i++)
42     {
43         Document doc = documents.getItem(i);
44
45         // Validation is one of the few methods that have output parameters.
46         // The class JVariant is the correct type for parameters in these cases.
47         // To get values back mark them with the by-reference flag.
48         JVariant validationErrorText = new JVariant.JStringVariant("");
validationErrorText.setByRefFlag();
49         JVariant validationErrorCount = new JVariant.JIntVariant(0);
validationErrorCount.setByRefFlag();
50         JVariant validationErrorXMLData = new JVariant.JIDispatchVariant(0);
validationErrorXMLData.setByRefFlag();
51         if (!doc.isValid(validationErrorText, validationErrorCount,
validationErrorXMLData))
52             System.out.println("Document " + doc.getName() + " is not wellformed - " +
validationErrorText.getStringValue());
53         else
54             System.out.println("Document " + doc.getName() + " is wellformed.");
55     }
56
57     // The following lines attach to the document events using a default
implementation
58     // for the events and override one of its methods.
59     // If you want to override all document events it is better to derive your
listener class
60     // from DocumentEvents and implement all methods of this interface.
61     Document doc = authenticDesktop.getActiveDocument();
62     doc.addListener(new DocumentEventsDefaultHandler()
63     {
64         @Override
65         public boolean onBeforeCloseDocument(Document i_ipDoc) throws
AutomationException
66         {
67             System.out.println("Document " + i_ipDoc.getName() + " requested closing.");
68
69             // Allow closing of document
70             return true;
71         }
    }
```

```
72     });
73     doc.close(true);
74     doc = null;
75
76     System.out.println("Watch AuthenticDesktop!");
77 }
78 catch (AutomationException e)
79 {
80     // e.printStackTrace();
81 }
82 finally
83 {
84     // Make sure that AuthenticDesktop can shut down properly.
85     if (authenticDesktop != null)
86         authenticDesktop.dispose();
87
88     // Since the COM server was made visible and still is visible, it will keep
running
89     // and needs to be closed manually.
90     System.out.println("Now close AuthenticDesktop!");
91 }
92 }
93 }
```

#### 15.3.1.2.4.2 Application Startup and Shutdown

The code listings below show how the application can be started up and shut down.

#### Application startup

Before starting up the application, the appropriate classes must be imported (*see below*).

```
01 // Access general JAVA-COM bridge classes
02 import com.altova.automation.libs.*;
03
04 // Access AuthenticDesktop Java-COM bridge
05 import com.altova.automation.AuthenticDesktop.*;
06 import com.altova.automation.AuthenticDesktop.Enums.SPYViewModes;
07
08 /**
09  * A simple example that starts AuthenticDesktop COM server and performs a view
operations on it.
10  * Feel free to extend.
11  */
12 public class RunAuthenticDesktop
13 {
14     public static void main(String[] args)
15     {
16         // An instance of the application.
17         Application authenticDesktop = null;
18
19         // Instead of COM error-handling, use Java exception mechanism.
20         try
```

```
21     {
22         // Start AuthenticDesktop as COM server.
23         authenticDesktop = new Application();
24         // COM servers start up invisible so we make it visible
25         authenticDesktop.setVisible(true);
26
27     ...
28     }
29 }
30 }
```

## Application shutdown

The application can be shut down as shown below.

```
1 {
2     // Make sure that AuthenticDesktop can shut down properly.
3     if (authenticDesktop != null)
4         authenticDesktop.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 AuthenticDesktop!");
9 }
```

### 15.3.1.2.4.3 Simple Document Access

The code listing below shows how to open a document.

```
1 // Locate samples installed with the product.
2 String strExamplesFolder = System.getenv("USERPROFILE") + "\\My Documents\\Altova\\
\\Authentic2012\\AuthenticExamples\\";
3
4 // Open two files from the product samples.
5 authenticDesktop.getDocuments().openFile(strExamplesFolder + "OrgChart.pxf", false);
6 authenticDesktop.getDocuments().openFile(strExamplesFolder + "ExpReport.xml", false);
```

### 15.3.1.2.4.4 Iterations

The listing below shows how to iterate through open documents.

```
01 // Iterate through all open documents and set the View mode to 'Browser'.
02 for (Document doc:authenticDesktop.getDocuments())
03     if ( doc.getCurrentViewMode() != SPYViewModes.spyViewBrowser)
04         doc.switchViewMode(SPYViewModes.spyViewBrowser);
05
06 // An alternative iteration mode is index-based. COM indices are typically zero-based.
```

```
07 Documents documents = authenticDesktop.getDocuments();
08 for (int i = 1; i <= documents.getCount(); i++)
09 {
10     Document doc = documents.getItem(i);
11     ...
12 }
```

#### 15.3.1.2.4.5 Use of Out-Parameters

The code listing below iterates through open documents and validates each of them. For each validation, a message is generated using the output parameters of the Validation method.

```
01 // Iterate through all open documents and set the View mode to 'Text'.
02 for (Document doc:authenticDesktop.getDocuments())
03     if ( doc.getCurrentViewMode() != SPYViewModes.spyViewText)
04         doc.switchViewMode(SPYViewModes.spyViewText);
05
06 // An alternative iteration mode is index-based. COM indices are typically zero-based.
07 Documents documents = authenticDesktop.getDocuments();
08 for (int i = 1; i <= documents.getCount(); i++)
09 {
10     Document doc = documents.getItem(i);
11
12     // Validation is one of the few methods that have output parameters.
13     // The class JVariant is the correct type for parameters in these cases.
14     // To get values back, mark them with the by-reference flag.
15     JVariant validationErrorText = new JVariant.JStringVariant("");
validationErrorText.setByRefFlag();
16     JVariant validationErrorCount = new JVariant.JIntVariant(0);
validationErrorCount.setByRefFlag();
17     JVariant validationErrorXMLData = new JVariant.JIDispatchVariant(0);
validationErrorXMLData.setByRefFlag();
18     if (!doc.isValid(validationErrorText, validationErrorCount, validationErrorXMLData))
19         System.out.println("Document " + doc.getName() + " is not wellformed - " +
validationErrorText.getStringValue());
20     else
21         System.out.println("Document " + doc.getName() + " is wellformed.");
22 }
```

#### 15.3.1.2.4.6 Event Handlers

The listing below shows how to listen for and use events.

```
01 // The following lines attach to the document events using a default implementation
02 // for the events and override one of its methods.
03 // If you want to override all document events, it is better to derive your listener
class
04 // from DocumentEvents and implement all methods of this interface.
```

```

05 Document doc = authenticDesktop.getActiveDocument();
06 doc.addListener(new DocumentEventsDefaultHandler()
07 {
08     @Override
09     public boolean onBeforeCloseDocument(Document i_ipDoc) throws AutomationException
10     {
11         System.out.println("Document " + i_ipDoc.getName() + " requested closing.");
12
13         // allow closing of document
14         return true;
15     }
16 });
17 doc.close(true);
18 doc = null;

```

## 15.3.2 Interfaces

### Object Hierarchy

- [Application](#) <sup>382</sup>
- [SpyProject](#) <sup>545</sup>
  - [SpyProjectItems](#) <sup>549</sup>
    - [SpyProjectItem](#) <sup>547</sup>
  - [Documents](#) <sup>497</sup>
  - [Document](#) <sup>467</sup>
    - [GridView](#) <sup>529</sup>
      - [AuthenticView](#) <sup>435</sup>
      - [AuthenticRange](#) <sup>407</sup>
        - [AuthenticDataTransfer](#) <sup>403</sup> (previously DocEditDataTransfer)
        - [AuthenticDataTransfer](#) <sup>403</sup> (previously DocEditDataTransfer)
      - [TextView](#) <sup>554</sup>
      - [XMLData](#) <sup>593</sup>
    - [Dialogs](#) <sup>387</sup>
    - [CodeGeneratorDlg](#) <sup>452</sup>
      - [FileSelectionDlg](#) <sup>510</sup>
      - [SchemaDocumentationDlg](#) <sup>532</sup>
      - [GenerateSampleXMLDlg](#) <sup>523</sup>
      - [DTDSchemaGeneratorDlg](#) <sup>501</sup>
      - [FindInFilesDlg](#) <sup>511</sup>
      - [DatabaseConnection](#) <sup>457</sup>
      - [ExportSettings](#) <sup>507</sup>
      - [TextImportExportSettings](#) <sup>551</sup>
      - [ElementList](#) <sup>505</sup>
      - [ElementListItem](#) <sup>506</sup>

[Enumerations](#) <sup>604</sup>

### Description

This chapter contains the reference of the Authentic Desktop 1.5 Type Library.

Most of the given examples are written in VisualBasic. These code snippets assume that there is a variable defined and set, called **objSpy of type Application**. There are also some code samples written in JavaScript.

## 15.3.2.1 Application

### Methods

[GetDatabaseImportElementList](#) <sup>388</sup>

[GetDatabaseSettings](#) <sup>389</sup>

[GetDatabaseTables](#) <sup>389</sup>

[ImportFromDatabase](#) <sup>392</sup>

[CreateXMLSchemaFromDBStructure](#) <sup>387</sup>

[GetTextImportElementList](#) <sup>390</sup>

[GetTextImportExportSettings](#) <sup>391</sup>

[ImportFromText](#) <sup>394</sup>

[ImportFromWord](#) <sup>395</sup>

[ImportFromSchema](#) <sup>393</sup>

[GetExportSettings](#) <sup>390</sup>

[NewProject](#) <sup>396</sup>

[OpenProject](#) <sup>396</sup>

[AddMacroMenuItem](#) <sup>385</sup>

[ClearMacroMenu](#) <sup>386</sup>

[ShowForm](#) <sup>399</sup>

[ShowApplication](#) <sup>398</sup>

[URLDelete](#) <sup>400</sup>

[URLMakeDirectory](#) <sup>400</sup>

[AddXSLT\\_XQParameter](#) <sup>386</sup>

[GetXSLT\\_XQParameterCount](#) <sup>391</sup>

[GetXSLT\\_XQParameterName](#) <sup>392</sup>

[GetXSLT\\_XQParameterXPath](#) <sup>392</sup>

[RemoveXSLT\\_XQParameter](#) <sup>397</sup>

[FindInFiles](#) <sup>388</sup>

[Quit](#) <sup>397</sup>

### Properties

[Application](#) <sup>386</sup>

[Parent](#) <sup>396</sup>

[ActiveDocument](#) <sup>385</sup>

[Documents](#) <sup>388</sup>

[CurrentProject](#) <sup>387</sup>[Dialogs](#) <sup>387</sup>[WarningNumber](#) <sup>401</sup>[WarningText](#) <sup>401</sup>[Status](#) <sup>399</sup>[MajorVersion](#) <sup>395</sup>[MinorVersion](#) <sup>395</sup>[Edition](#) <sup>388</sup>[IsAPISupported](#) <sup>395</sup>[ServicePackVersion](#) <sup>398</sup>

## Description

Application is the root for all other objects. It is the only object you can create by CreateObject (VisualBasic) or other similar COM related functions.

## Example

```
Dim objSpy As Application
Set objSpy = CreateObject("XMLSpy.Application")
```

### 15.3.2.1.1 Events

#### 15.3.2.1.1.1 OnBeforeOpenDocument

**Event:** OnBeforeOpenDocument(*objDialog* as [FileSelectionDlg](#) <sup>510</sup>)

## Description

This event gets fired whenever a document gets opened via the OpenFile or OpenURL menu command. It is sent after a document file has been selected but before the document gets opened. The file selection dialog object is initialized with the name of the selected document file. You can modify this selection. To continue the opening of the document leave the [FileSelectionDlg.DialogAction](#) <sup>511</sup> property of *io\_objDialog* at its default value [spyDialogOK](#) <sup>607</sup>. To abort the opening of the document set this property to [spyDialogCancel](#) <sup>607</sup>.

## Examples

Given below are examples of how this event can be scripted.

### **XMLSpy scripting environment - VBScript:**

```
Function On_BeforeOpenDocument(objDialog)
End Function
```

### **XMLSpy scripting environment - JScript:**

```
function On_BeforeOpenDocument(objDialog)
{
}
```

### **XMLSpy IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (26, ...) // nEventId = 26
```

### 15.3.2.1.1.2 *OnBeforeOpenProject*

**Event:** OnBeforeOpenProject(*objDialog* as [FileSelectionDlg](#)<sup>510</sup>)

#### Description

This event gets fired after a project file has been selected but before the project gets opened. The file selection dialog object is initialized with the name of the selected project file. You can modify this selection. To continue the opening of the project leave the [FileSelectionDlg.DialogAction](#)<sup>511</sup> property of *io\_objDialog* at its default value [spyDialogOK](#)<sup>607</sup>. To abort the opening of the project set this property to [spyDialogCancel](#)<sup>607</sup>.

#### Examples

Given below are examples of how this event can be scripted.

##### **XMLSpy scripting environment - VBScript:**

```
Function On_BeforeOpenProject(objDialog)  
End Function
```

##### **XMLSpy scripting environment - JScript:**

```
function On_BeforeOpenProject(objDialog)  
{  
}  
}
```

##### **XMLSpy IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (25, ...) // nEventId = 25
```

### 15.3.2.1.1.3 *OnDocumentOpened*

**Event:** OnDocumentOpened(*objDocument* as [Document](#)<sup>467</sup>)

#### Description

This event gets fired whenever a document opens in Authentic Desktop. This can happen due to opening a file with the OpenFile or OpenURL dialog, creating a new file or dropping a file onto Authentic Desktop. The new document gets passed as parameter. The operation cannot be canceled.

#### Examples

Given below are examples of how this event can be scripted.

##### **XMLSpy scripting environment - VBScript:**

```
Function On_OpenDocument(objDocument)  
End Function
```

##### **XMLSpy scripting environment - JScript:**

```
function On_OpenDocument(objDocument)  
{  
}  
}
```

**XMLSpy IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (7, ...) // nEventId = 7
```

#### 15.3.2.1.1.4 OnProjectOpened

**Event:** OnProjectOpened(*objProject* as [SpyProject](#)<sup>545</sup>)

**Description**

This event gets fired whenever a project gets opened in Authentic Desktop. The new project gets passed as parameter.

**Examples**

Given below are examples of how this event can be scripted.

**XMLSpy scripting environment - VBScript:**

```
Function On_OpenProject(objProject)  
End Function
```

**XMLSpy scripting environment - JScript:**

```
function On_OpenProject(objProject)  
{  
}  
}
```

**XMLSpy IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (6, ...) // nEventId = 6
```

#### 15.3.2.1.2 ActiveDocument

**Property:** ActiveDocument as [Document](#)<sup>467</sup>

**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.

#### 15.3.2.1.3 AddMacroMenuItem

**Method:** AddMacroMenuItem(*strMacro* as String, *strDisplayText* as String)

**Description**

Adds a menu item to the **Tools** menu. This new menu item invokes the macro defined by *strMacro*. See also [Example Scripting Project](#)<sup>319</sup>.

**Errors**

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1108 Number of macro items is limited to 16 items.

### 15.3.2.1.4 AddXSLT\_XQParameter

**Method:** AddXSLT\_XQParameter(name as String, XPath as String)

**Description**

Adds an XSLT or XQuery parameter. The parameter's name and value are the two arguments of the method.

**Errors**

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.
- 1124 The XPath expression is not set.
- 1125 Not a QName.
- 1126 The specified XPath is not valid. Reason for invalidity appended.
- 1127 A parameter with the submitted name already exists.

### 15.3.2.1.5 Application

**Property:** Application as [Application](#)<sup>382</sup> (read-only)

**Description**

Accesses the Authentic Desktop application object.

**Errors**

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

### 15.3.2.1.6 ClearMacroMenu

**Method:** ClearMacroMenu()

**Return Value**

None

**Description**

Removes from the **Tools** menu those menu items that were added by calling [AddMacroMenuItem](#)<sup>385</sup>. See also [Example Scripting Project](#)<sup>319</sup>.

**Errors**

1111 The application object is no longer valid.

### 15.3.2.1.7 CreateXMLSchemaFromDBStructure

**Method:** CreateXMLSchemaFromDBStructure(pImportSettings as [DatabaseConnection](#)<sup>457</sup>, pTables as [ElementList](#)<sup>505</sup>)

#### Description

CreateXMLSchemaFromDBStructure creates from a database specified in pImportSettings for the defined tables in pTables new XML Schema document(s) describing the database tables structure.

The parameter pTables specifies which table structures the XML Schema document should contain. This parameter can be NULL, specifying that all table structures will be exported.

See also [GetDataBaseTables](#)<sup>389</sup>.

#### Errors

- 1112 Invalid database specified.
- 1120 Database import failed.

### 15.3.2.1.8 CurrentProject

**Property:** CurrentProject as [SpyProject](#)<sup>545</sup>

#### Description

Reference to the active document. If no project is open, CurrentProject is null (nothing).

#### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

### 15.3.2.1.9 Dialogs

**Property:** Dialogs as [Dialogs](#)<sup>464</sup> (read-only)

#### Description

Access the built-in dialogs of Authentic Desktop.

#### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

### 15.3.2.1.10 Documents

**Property:** Documents as [Documents](#) <sup>497</sup>

#### Description

Collection of all open documents.

#### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

### 15.3.2.1.11 Edition

**Property:** Edition as String

#### Description

Returns the edition of the application, for example `Altova Authentic Desktop Enterprise Edition` for the Enterprise edition.

#### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

### 15.3.2.1.12 FindInFiles

**Method:** FindInFiles(*pSettings* as [FindInFilesDlg](#) <sup>511</sup>) as [FindInFilesResults](#) <sup>522</sup>

#### Description

Returns a [FindInFilesResults](#) <sup>522</sup> object containing information about the files that matched the specified settings.

#### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

### 15.3.2.1.13 GetDatabaseImportElementList

**Method:** GetDatabaseImportElementList(*pImportSettings* as [DatabaseConnection](#) <sup>457</sup>) as [ElementList](#) <sup>505</sup>

#### Description

The function returns a collection of `ElementListItem`s where the properties [ElementListItem.Name](#) <sup>507</sup> contain the names of the fields that can be selected for import and the properties [ElementListItem.ElementKind](#) <sup>507</sup> are initialized either to `spyXMLDataAttr` or `spyXMLDataElement`, depending on the value passed in [DatabaseConnection.AsAttributes](#) <sup>459</sup>. This list serves as a filter to what finally gets imported by a future call to [ImportFromDatabase](#) <sup>392</sup>. Use [ElementList.RemoveElement](#) <sup>506</sup> to exclude fields from import.

Properties mandatory to be filled out for the database connection are one of [DatabaseConnection.File](#) <sup>460</sup>, [DatabaseConnection.ADOConnection](#) <sup>458</sup> and [DatabaseConnection.ODBCConnection](#) <sup>462</sup>, as well as [DatabaseConnection.SQLSelect](#) <sup>463</sup>. Use the property [DatabaseConnection.AsAttributes](#) <sup>459</sup> to initialize [ElementListItem.ElementKind](#) <sup>507</sup> of the resulting element list to either *spyXMLDataAttr* or *spyXMLDataElement*, respectively.

### Example

See example at [ImportFromDatabase](#) <sup>392</sup>.

### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1107 Import from database failed.
- 1112 Invalid database specified.
- 1114 Select statement is missing.
- 1119 database element list import failed.

#### 15.3.2.1.14 GetDatabaseSettings

**Method:** [GetDatabaseSettings\(\)](#) as [DatabaseConnection](#) <sup>457</sup>

### Description

[GetDatabaseSettings](#) creates a new object of database settings. The object is used to specify database connection parameters for the methods [GetDatabaseTables](#) <sup>389</sup>, [GetDatabaseImportElementList](#) <sup>388</sup>, [ImportFromDatabase](#) <sup>392</sup>, [ImportFromSchema](#) <sup>393</sup> and [ExportToDatabase](#) <sup>479</sup>.

### Example

See example of [ImportFromDatabase](#) <sup>392</sup>.

### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

#### 15.3.2.1.15 GetDatabaseTables

**Method:** [GetDatabaseTables\(pImportSettings\)](#) as [DatabaseConnection](#) <sup>382</sup>) as [ElementList](#) <sup>505</sup>

### Description

[GetDatabaseTables](#) reads the table names from the database specified in *pImportSettings*. Properties mandatory to be filled out for the database connection are one of [DatabaseConnection.File](#) <sup>460</sup>, [DatabaseConnection.ADOConnection](#) <sup>458</sup> and [DatabaseConnection.ODBCConnection](#) <sup>462</sup>. All other properties are ignored.

The function returns a collection of [ElementListItem](#)s where the properties [ElementListItem.Name](#) <sup>507</sup> contain the names of tables stored in the specified database. The remaining properties of [ElementListItem](#) <sup>506</sup> are unused.

### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.

- 1112 Invalid database specified.
- 1113 Error while reading database table information.
- 1118 Database table query failed.

### Example

```

Dim objImpSettings As DatabaseConnection
Set objImpSettings = objSpy.GetDatabaseSettings
objImpSettings.ADOConnection = TxtADO.Text

'store table names in list box
ListTables.Clear

Dim objList As ElementList
Dim objItem As ElementListItem
On Error GoTo ErrorHandler
Set objList = objSpy.GetDatabaseTables(objImpSettings)

    For Each objItem In objList
        ListTables.AddItem objItem.Name
    Next

```

#### 15.3.2.1.16 GetExportSettings

**Method:** GetExportSettings() as [ExportSettings](#)<sup>507</sup> (read-only)

##### Description

GetExportSettings creates a new object of common export settings. This object is used to pass the parameters to the export functions and defines the behaviour of the export calls. See also the export functions from [Document](#)<sup>467</sup>.

##### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

#### 15.3.2.1.17 GetTextImportElementList

**Method:** GetTextImportElementList(plmportSettings as [TextImportExportSettings](#)<sup>551</sup>) as [ElementList](#)<sup>505</sup>

##### Description

GetTextImportElementList retrieves importing information about the text-file as specified in plmportSettings. The function returns a collection of ElementListItems where the properties [ElementListItem.Name](#)<sup>507</sup> contain the names of the fields found in the file. The values of remaining properties are undefined.

If the text-file does not contain a column header, set pImportSettings.[HeaderRow](#)<sup>553</sup> to false. The resulting element list will contain general column names like 'Field1' and so on.

##### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.

- 1107 Import from database failed.
- 1115 Error during text element list import. Cannot create parser for import file.
- 1116 Error during text element list import.

### Example

```
'-----  
' VBA client code fragment - import selected fields from text file  
'-----  
    Dim objImpSettings As TextImportExportSettings  
    Set objImpSettings = objSpy.GetTextImportExportSettings  
  
    objImpSettings.ImportFile = "C:\ImportMe.txt"  
    objImpSettings.HeaderRow = False  
  
    Dim objList As ElementList  
    Set objList = objSpy.GetTextImportElementList(objImpSettings)  
  
    'exclude first column  
    objList.RemoveItem 1  
  
    Dim objImpDoc As Document  
    On Error Resume Next  
    Set objImpDoc = objSpy.ImportFromText(objImpSettings, objList)  
    CheckForError
```

#### 15.3.2.1.18 GetTextImportExportSettings

**Method:** GetTextImportExportSettings() as [TextImportExportSettings](#)<sup>551</sup> (read-only)

#### Description

GetTextImportExportSettings creates a new object of common import and export settings for text files. See also the example for [Application.GetTextImportElementList](#)<sup>390</sup>.

#### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

#### 15.3.2.1.19 GetXSLT\_XQParameterCount

**Method:** GetXSLT\_XQParameterCount() as Long

#### Description

Returns the number of XSLT and XQuery parameters.

#### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

### 15.3.2.1.20 GetXSLT\_XQParameterName

**Method:** GetXSLT\_XQParameterName(index as Long) as String

#### Description

Returns the name of the XSLT or XQuery parameter identified by the supplied index.

#### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

### 15.3.2.1.21 GetXSLT\_XQParameterXPath

**Method:** GetXSLT\_XQParameterXPath(index as Long) as String

#### Description

Returns the XPath expression of the XSLT or XQuery parameter identified by the supplied index.

#### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

### 15.3.2.1.22 ImportFromDatabase

**Method:** ImportFromDatabase(plmportSettings as [DatabaseConnection](#)<sup>457</sup>, pElementList as [ElementList](#)<sup>505</sup>) as [Document](#)<sup>467</sup>

#### Return Value

Creates a new document containing the data imported from the database.

#### Description

ImportFromDatabase imports data from a database as specified in plmportSettings and creates a new document containing the data imported from the database. Properties mandatory to be filled out are one of [DatabaseConnection.File](#)<sup>460</sup>, [DatabaseConnection.ADOConnection](#)<sup>458</sup> or [DatabaseConnection.ODBCConnection](#)<sup>462</sup> and [DatabaseConnection.SQLSelect](#)<sup>463</sup>. Additionally, you can use [DatabaseConnection.AsAttributes](#)<sup>459</sup>, [DatabaseConnection.ExcludeKeys](#)<sup>460</sup>, [DatabaseConnection.IncludeEmptyElements](#)<sup>461</sup> and [NumberDateTimeFormat](#)<sup>462</sup> to further parameterize import.

The parameter pElementList specifies which fields of the selected data gets written into the newly created document, and which are created as elements and which as attributes. This parameter can be NULL, specifying that all selected fields will be imported as XML elements.

See [GetDatabaseSettings](#)<sup>389</sup> and [GetDatabaseImportElementList](#)<sup>388</sup> for necessary steps preceding any import of data from a database.

### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1107 Import from database failed.
- 1112 Invalid database specified.
- 1114 Select statement is missing.
- 1117 Transformation to XML failed.
- 1120 Database import failed.

### Example

```
Dim objImpSettings As DatabaseConnection
Set objImpSettings = objSpy.GetDatabaseSettings

objImpSettings.ADOConnection = strADOConnection
objImpSettings.SQLSelect = "SELECT * FROM MyTable"

Dim objDoc As Document
On Error Resume Next
Set objDoc = objSpy.ImportFromDatabase(objImpSettings,
objSpy.GetDatabaseImportElementList(objImpSettings))
' CheckForError here
```

#### 15.3.2.1.23 ImportFromSchema

**Method:** `ImportFromSchema(pImportSettings as DatabaseConnection457, strTable as String, pSchemaDoc as Document467) as Document467`

### Return Value

Creates a new document filled with data from the specified database as specified by the schema definition in *pSchemaDoc*.

### Description

`ImportFromSchema` imports data from a database specified in *pImportSettings*. Properties mandatory to be filled out are one of [DatabaseConnection.File](#)<sup>460</sup>, [DatabaseConnection.ADOConnection](#)<sup>458</sup> or [DatabaseConnection.ODBCConnection](#)<sup>462</sup>. Additionally, you can use [DatabaseConnection.AsAttributes](#)<sup>459</sup>, [DatabaseConnection.ExcludeKeys](#)<sup>460</sup> and [NumberDateTimeFormat](#)<sup>462</sup> to further parameterize import. All other properties get ignored.

`ImportFromSchema` does not use an explicit SQL statement to select the data. Instead, it expects a structure definition of the document to create in form of an XML schema document in *pSchemaDoc*. From this definition the database select statement is automatically deduced. Specify in *strTable* the table name of the import root that will become the root node in the new document.

See [GetDatabaseSettings](#)<sup>389</sup> and [GetDatabaseTables](#)<sup>389</sup> for necessary steps preceding an import from a database based on a schema definition. To create the schema definition file use command 'create database schema' from the 'convert' menu of Authentic Desktop.

**Errors**

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1107 Import from database failed.
- 1112 Invalid database specified.
- 1120 Database import failed.
- 1121 Could not create validator for the specified schema.
- 1122 Failed parsing schema for database import.

### 15.3.2.1.24 ImportFromText

**Method:** ImportFromText(*pImportSettings* as [TextImportExportSettings](#)<sup>551</sup>, *pElementList* as [ElementList](#)<sup>505</sup>) as [Document](#)<sup>467</sup>

**Description**

ImportFromText imports the text file as specified in *pImportSettings*. The parameter *pElementList* can be used as import filter. Either pass the list returned by a previous call to [GetTextImportElementList](#)<sup>390</sup> or null to import all columns. To avoid import of unnecessary columns use [ElementList.RemoveElement](#)<sup>506</sup> to remove the corresponding field names from *pElementList* before calling `ImportFromText`. The method returns the newly created document containing the imported data. This document is the same as the active document of Authentic Desktop.

**Errors**

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1107 Import from text file failed.
- 1117 Transformation to XML failed.

**Example**

```
'-----  
' VBA client code fragment - import from text file  
'-----  
    Dim objImpSettings As TextImportExportSettings  
    Set objImpSettings = objSpy.GetTextImportExportSettings  
  
    objImpSettings.ImportFile = strFileName  
    objImpSettings.HeaderRow = False  
  
    Dim objImpDoc As Document  
    On Error Resume Next  
    Set objImpDoc = objSpy.ImportFromText(objImpSettings,  
objSpy.GetTextImportElementList(objImpSettings))  
  
    CheckForError
```

### 15.3.2.1.25 ImportFromWord

**Method:** ImportFromWord(*strFile* as String) as [Document](#)<sup>467</sup>

#### Description

ImportFromWord imports the MS-Word Document *strFile* into a new XML document.

#### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.  
Import from document failed.

### 15.3.2.1.26 IsAPISupported

**Property:** IsAPISupported as Boolean

#### Description

Returns whether the API is supported in this version or not.

#### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

### 15.3.2.1.27 MajorVersion

**Property:** MajorVersion as Integer

#### Description

Returns the application version's major number, for example 15 for 2013 versions, and 16 for 2014 versions..

#### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

### 15.3.2.1.28 MinorVersion

**Property:** MinorVersion as Integer

#### Description

Returns the application version's minor number.

#### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

### 15.3.2.1.29 NewProject

**Method:** `NewProject(strPath as String, bDiscardCurrent as Boolean)`

#### Description

`NewProject` creates a new project.

If there is already a project open that has been modified and `bDiscardCurrent` is false, then `NewProject()` fails.

#### Errors

- 1111 The application object is no longer valid.
- 1102 A project is already open but *bDiscardCurrent* is true.
- 1103 Creation of new project failed.

### 15.3.2.1.30 OpenProject

**Method:** `OpenProject(strPath as String, bDiscardCurrent as Boolean, bDialog as Boolean)`

#### Parameters

`strPath`

Path and file name of the project to open. Can be empty if `bDialog` is true.

`bDiscardCurrent`

Discard currently open project and possibly lose changes.

`bDialog`

Show dialogs for user input.

#### Return Value

None

#### Description

`OpenProject` opens an existing project. If there is already a project open that has been modified and `bDiscardCurrent` is false, then `OpenProject()` fails.

#### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1101 Cannot open specified project.
- 1102 A project is already open but *bDiscardCurrent* is true.

### 15.3.2.1.31 Parent

**Property:** `Parent` as [Application](#)<sup>382</sup> (read-only)

#### Description

Accesses the Authentic Desktop application object.

**Errors**

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

### 15.3.2.1.32 Quit

**Method:** Quit()

**Return Value**

None

**Description**

This method terminates Authentic Desktop. All modified documents will be closed without saving the changes. This is also true for an open project.

If Authentic Desktop 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.

### 15.3.2.1.33 ReloadSettings

**Method:** ReloadSettings

**Return Value****Description**

The application settings are reloaded from the registry.

Available with TypeLibrary version 1.5

**Errors**

- 1111 The application object is no longer valid.

### 15.3.2.1.34 RemoveXSLT\_XQParameter

**Method:** RemoveXSLT\_XQParameter(index as Long)

**Description**

Removes the XSLT or XQuery parameter identified by the supplied index.

**Errors**

- 1111 The application object is no longer valid.

1100 Invalid address for the return parameter was specified.

### 15.3.2.1.35 RunMacro

**Method:** RunMacro(*strMacro* as String)

#### Return Value

#### Description

Calls the specified macro either from the project scripts (if present) or from the global scripts.

Available with TypeLibrary version 1.5

#### Errors

1111 The application object is no longer valid.

### 15.3.2.1.36 ScriptingEnvironment

**Property:** ScriptingEnvironment as IUnknown (read-only)

#### Description

Reference to any active scripting environment. This property makes it possible to access the TypeLibrary of the XMLSpyFormEditor.exe application which is used as the current scripting environment.

Available with TypeLibrary version 1.5

#### Errors

1111 The application object is no longer valid.  
1100 Invalid address for the return parameter was specified.

### 15.3.2.1.37 ServicePackVersion

**Property:** ServicePackVersion as Long

#### Description

Returns the Service Pack version number of the application. Eg: 1 for 2010 R2 SP1

#### Errors

1111 The application object is no longer valid.  
1100 Invalid address for the return parameter was specified.

### 15.3.2.1.38 ShowApplication

**Method:** ShowApplication(*bShow* as Boolean)

#### Return Value

None

**Description**

The method shows (bShow = True) or hides (bShow = False) Authentic Desktop.

**Errors**

1110 The application object is no longer valid.

### 15.3.2.1.39 ShowFindInFiles

**Method:** ShowFindInFiles(pSettings as [FindInFilesDlg](#)<sup>511</sup>) as Boolean

**Return Value**

Returns false if the user pressed the Cancel button, true otherwise.

**Description**

Displays the FindInFiles dialog preset with the given settings. The user modifications of the settings are stored in the passed dialog object.

**Errors**

1111 The application object is no longer valid.

1100 Invalid parameter or invalid address for the return parameter was specified.

### 15.3.2.1.40 ShowForm

**Method:** ShowForm(strFormName as String) as Long

**Return Value**

Returns zero if the user pressed a Cancel button or the form calls TheView.Cancel().

**Description**

Displays the form strFormName.

Forms, event handlers and macros can be created with the Scripting Environment. Select "Switch to scripting environment" from the **Tools** menu to invoke the Scripting Environment.

**Errors**

1111 The application object is no longer valid.

1100 Invalid parameter or invalid address for the return parameter was specified.

### 15.3.2.1.41 Status

**Property:** Status as [ENUMApplicationStatus](#)<sup>604</sup>

**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.

#### 15.3.2.1.42 URLDelete

**Method:** URLDelete(*strURL* as String,*strUser* as String,*strPassword* as String)

##### Return Value

None

##### Description

The method deletes the file at the URL *strURL*.

##### Errors

- 1111 The application object is no longer valid.
- 1109 Error deleting file at specified URL.

#### 15.3.2.1.43 URLMakeDirectory

**Method:** URLMakeDirectory(*strURL* as String,*strUser* as String,*strPassword* as String)

##### Return Value

None

##### Description

The method creates a new directory at the URL *strURL*.

##### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter specified.

#### 15.3.2.1.44 Visible

**Property:** Visible as VARIANT\_BOOL

##### Description

Sets or gets the `visibility` attribute of Authentic Desktop. This standard automation property makes usage of [ShowApplication](#) <sup>398</sup> obsolete.

##### Errors

- 1110 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

### 15.3.2.1.45 WarningNumber

**Property:** WarningNumber as integer

#### Description

Some methods fill the property WarningNumber with additional information if an error occurs.

Currently just [Documents.OpenFile](#)<sup>500</sup> fills this property.

#### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

### 15.3.2.1.46 WarningText

**Property:** WarningText as String

#### Description

Some methods fill the property WarningText with additional information if an error occurs.

Currently just [Documents.OpenFile](#)<sup>500</sup> fills this property.

#### Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

## 15.3.2.2 AuthenticContextMenu

The context menu interface provides the means for the user to customize the context menus shown in Authentic. The interface has the methods listed in this section.

### 15.3.2.2.1 CountItems

**Method:** CountItems () nItems as long

#### Return Value

Returns the number of menu items.

#### Errors

- 2501 Invalid object.

### 15.3.2.2.2 DeleteItem

**Method:** `DeleteItem(IndexPosition as long)`

**Return Value**

Deletes the menu item that has the index position submitted in the first parameter.

**Errors**

- 2501 Invalid object
- 2502 Invalid index

### 15.3.2.2.3 GetItemText

**Method:** `GetItemText(IndexPosition as long) MenuItemName as string`

**Return Value**

Gets the name of the menu item located at the index position submitted in the first parameter.

**Errors**

- 2501 Invalid object
- 2502 Invalid index

### 15.3.2.2.4 InsertItem

**Method:** `InsertItem(IndexPosition as long, MenuItemName as string, MacroName as string)`

**Return Value**

Inserts a user-defined menu item at the position in the menu specified in the first parameter and having the name submitted in the second parameter. The menu item will start a macro, so a valid macro name must be submitted.

**Errors**

- 2501 Invalid object
- 2502 Invalid index
- 2503 No such macro
- 2504 Internal error

### 15.3.2.2.5 SetItemText

**Method:** `SetItemText(IndexPosition as long, MenuItemName as string)`

**Return Value**

Sets the name of the menu item located at the index position submitted in the first parameter.

**Errors**

2501 Invalid object

2502 Invalid index

### 15.3.2.3 AuthenticDataTransfer

**Renamed from DocEditDataTransfer to AuthenticDataTransfer**

The DocEditView object is renamed to OldAuthenticView.  
DocEditSelection is renamed to AuthenticSelection.  
DocEditEvent is renamed to AuthenticEvent.  
DocEditDataTransfer is renamed to AuthenticDataTransfer.

Their usage—except for AuthenticDataTransfer—is no longer recommended. We will continue to support existing functionality for a yet undefined period of time but no new features will be added to these interfaces.

For examples on migrating from DocEdit to Authentic see the description of the different methods and properties of the different DocEdit objects.

**Methods**

[getData](#) <sup>404</sup>

**Properties**

[dropEffect](#) <sup>403</sup>

[ownDrag](#) <sup>404</sup>

[type](#) <sup>404</sup>

**Description**

The events OnDragOver and OnBeforeDrop provide information about the object being dragged with an instance of type AuthenticDataTransfer. It contains a description of the dragged object and its content. The latter is available either as string or a pointer to a COM object supporting the IUnkown interface.

#### 15.3.2.3.1 dropEffect

**Property:** dropEffect as long

**Description**

The property stores the drop effect from the default event handler. You can set the drop effect if you change this value and return TRUE for the event handler.

**Errors**

2101 Invalid address for the return parameter was specified.

### 15.3.2.3.2 `getData`

**Method:** `getData()` as Variant

**Description**

Retrieve the data associated with the dragged object. Depending on [AuthenticDataTransfer.type](#)<sup>404</sup>, that data is either a string or a COM interface pointer of type IUnknown.

**Errors**

2101 Invalid address for the return parameter was specified.

### 15.3.2.3.3 `ownDrag`

**Property:** `ownDrag` as Boolean (read-only)

**Description**

The property is TRUE if the current dragging source comes from inside Authentic View.

**Errors**

2101 Invalid address for the return parameter was specified.

### 15.3.2.3.4 `type`

**Property:** `type` as String (read-only)

**Description**

Holds the type of data you get with the [DocEditDataTransfer.getData](#)<sup>404</sup> method.

Currently supported data types are:

OWN	data from Authentic View itself
TEXT	plain text
UNICODETEXT	plain text as UNICODE

**Errors**

2101 Invalid address for the return parameter was specified.

## 15.3.2.4 `AuthenticEventContext`

The `EventContext` interface gives access to many properties of the context in which a macro is executed.

### 15.3.2.4.1 EvaluateXPath

**Method:** EvaluateXPath (strExpression as string) as strValue as string

#### Return Value

The method evaluates the XPath expression in the context of the node within which the event was triggered and returns a string.

#### Description

EvaluateXPath() executes an XPath expression with the given event context. The result is returned as a string, in the case of a sequence it is a space-separated string.

#### Errors

- 2201 Invalid object.
- 2202 No context.
- 2209 Invalid parameter.
- 2210 Internal error.
- 2211 XPath error.

### 15.3.2.4.2 GetEventContextType

**Method:** GetEventContextType () Type as AuthenticEventContextType enumeration

#### Return Value

Returns the context node type.

#### Description

GetEventContextType allows the user to determine whether the macro is in an XML node or in an XPath atomic item context. The enumeration AuthenticEventContextType is defined as follows:

authenticEventContextXML,  
authenticEventContextAtomicItem,  
authenticEventContextOther

If the context is a normal XML node, the GetXMLNode () function gives access to it (returns NULL if not).

#### Errors

- 2201 Invalid object.
- 2202 No context.
- 2209 Invalid parameter.

### 15.3.2.4.3 GetNormalizedTextValue

**Method:** GetNormalizedTextValue () strValue as string

#### Return Value

Returns the value of the current node as string

**Errors**

- 2201 Invalid object.
- 2202 No context.
- 2203 Invalid context
- 2209 Invalid parameter.

#### 15.3.2.4.4 GetVariableValue

**Method:** `GetVariableValue(strName as string) strValue as string`

**Return Value**

Gets the value of the variable submitted as the parameter.

**Description**

`GetVariableValue` gets the variable's value in the scope of the context.

```
nZoom = parseInt( AuthenticView.EventContext.GetVariableValue( 'Zoom' ) );
if ( nZoom > 1 )
{
    AuthenticView.EventContext.SetVariableValue( 'Zoom', nZoom - 1 );
}
```

**Errors**

- 2201 Invalid object.
- 2202 No context.
- 2204 No such variable in scope
- 2205 Variable cannot be evaluated
- 2206 Variable returns sequence
- 2209 Invalid parameter

#### 15.3.2.4.5 GetXMLNode

**Method:** `GetXMLNode ( ) Node as XMLData object`

**Return Value**

Returns the context XML node or `NULL`

**Errors**

- 2201 Invalid object.
- 2202 No context.
- 2203 Invalid context
- 2209 Invalid parameter.

### 15.3.2.4.6 IsAvailable

**Method:** IsAvailable () as Boolean

#### Return Value

Returns true if `EventContext` is set, false otherwise.

#### Errors

2201 Invalid object.

### 15.3.2.4.7 SetVariableValue

**Method:** SetVariableValue (strName as string, strValue as string)

#### Return Value

Sets the value (second parameter) of the variable submitted in the first parameter.

#### Description

`SetVariableValue` sets the variable's value in the scope of the context.

```
nZoom = parseInt( AuthenticView.EventContext.GetVariableValue( 'Zoom' ) );
if ( nZoom > 1 )
{
    AuthenticView.EventContext.SetVariableValue( 'Zoom', nZoom - 1 );
}
```

#### Errors

2201 Invalid object.  
 2202 No context.  
 2204 No such variable in scope  
 2205 Variable cannot be evaluated  
 2206 Variable returns sequence  
 2207 Variable read-only  
 2208 No modification allowed

## 15.3.2.5 AuthenticRange

The first table lists the properties and methods of `AuthenticRange` that can be used to navigate through the document and select specific portions.

Properties	Methods	
<a href="#">Application</a> <sup>410</sup>	<a href="#">Clone</a> <sup>411</sup>	<a href="#">MoveBegin</a> <sup>428</sup>

<a href="#">FirstTextPosition</a> <sup>414</sup>	<a href="#">CollapsToBegin</a> <sup>411</sup>	<a href="#">MoveEnd</a> <sup>429</sup>
<a href="#">FirstXMLData</a> <sup>415</sup>	<a href="#">CollapsToEnd</a> <sup>412</sup>	<a href="#">NextCursorPosition</a> <sup>420</sup>
<a href="#">FirstXMLDataOffset</a> <sup>416</sup>	<a href="#">ExpandTo</a> <sup>414</sup>	<a href="#">PreviousCursorPosition</a> <sup>421</sup>
<a href="#">LastTextPosition</a> <sup>426</sup>	<a href="#">Goto</a> <sup>419</sup>	<a href="#">Select</a> <sup>431</sup>
<a href="#">LastXMLData</a> <sup>427</sup>	<a href="#">GotoNext</a> <sup>420</sup>	<a href="#">SelectNext</a> <sup>432</sup>
<a href="#">LastXMLDataOffset</a> <sup>427</sup>	<a href="#">GotoPrevious</a> <sup>421</sup>	<a href="#">SelectPrevious</a> <sup>432</sup>
<a href="#">Parent</a> <sup>430</sup>	<a href="#">IsEmpty</a> <sup>424</sup>	<a href="#">SetFromRange</a> <sup>434</sup>
	<a href="#">IsEqual</a> <sup>424</sup>	

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
<a href="#">Text</a> <sup>435</sup>	<a href="#">Copy</a> <sup>412</sup>	<a href="#">AppendRow</a> <sup>409</sup>
	<a href="#">Cut</a> <sup>412</sup>	<a href="#">DeleteRow</a> <sup>413</sup>
	<a href="#">Delete</a> <sup>412</sup>	<a href="#">DuplicateRow</a> <sup>413</sup>
	<a href="#">IsCopyEnabled</a> <sup>423</sup>	<a href="#">InsertRow</a> <sup>423</sup>
	<a href="#">IsCutEnabled</a> <sup>423</sup>	<a href="#">IsFirstRow</a> <sup>424</sup>
	<a href="#">IsDeleteEnabled</a> <sup>424</sup>	<a href="#">IsInDynamicTable</a> <sup>425</sup>
	<a href="#">IsPasteEnabled</a> <sup>425</sup>	<a href="#">IsLastRow</a> <sup>425</sup>
	<a href="#">Paste</a> <sup>430</sup>	<a href="#">MoveRowDown</a> <sup>429</sup>
		<a href="#">MoveRowUp</a> <sup>429</sup>

The following methods provide the functionality of the Authentic entry helper windows for range objects.

Operations of the entry helper windows		
Elements	Attributes	Entities
<a href="#">CanPerformActionWith</a> <sup>411</sup>	<a href="#">GetElementAttributeValue</a> <sup>418</sup>	<a href="#">GetEntityNames</a> <sup>418</sup>
<a href="#">CanPerformAction</a> <sup>410</sup>	<a href="#">GetElementAttributeNames</a> <sup>417</sup>	<a href="#">InsertEntity</a> <sup>422</sup>
<a href="#">PerformAction</a> <sup>430</sup>	<a href="#">GetElementHierarchy</a> <sup>418</sup>	
	<a href="#">HasElementAttribute</a> <sup>422</sup>	
	<a href="#">IsTextStateApplied</a> <sup>426</sup>	

	<a href="#">SetElementAttributeValue</a> <sup>433</sup>	
--	---	--

### 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](#) <sup>450</sup>, 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](#) <sup>445</sup>, [AuthenticView.DocumentEnd](#) <sup>445</sup>, and [AuthenticView.WholeDocument](#) <sup>452</sup>. The most flexible method is [AuthenticView.Goto](#) <sup>447</sup>, 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](#) <sup>414</sup> and [LastTextPosition](#) <sup>426</sup>. 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](#) <sup>415</sup>, [FirstXMLDataOffset](#) <sup>416</sup>, [LastXMLData](#) <sup>427</sup>, and [LastXMLDataOffset](#) <sup>427</sup>. 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](#) <sup>435</sup> 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](#) <sup>412</sup>, [Copy](#) <sup>412</sup>, [Paste](#) <sup>430</sup> and [Delete](#) <sup>412</sup>.
- Table operations for tables that can grow dynamically.
- Methods that map the functionality of the Authentic entry helper windows.
- Access to the [XMLData](#) <sup>593</sup> objects of the underlying document to modify them directly.

#### 15.3.2.5.1 AppendRow

**Method:** AppendRow() as Boolean

### Description

If the beginning of the range is inside a dynamic table, this method inserts a new row at the end of the selected table. The selection of the range is modified to point to the beginning of the new row. The function returns *true* if the append operation was successful, otherwise *false*.

**Errors**

- 2001 The authentic range object or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

**Examples**

```
'-----
' Scripting environment - VBScript
' Append row at end of current dynamically growable table
'-----
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' check if we can insert something
If objRange.IsInDynamicTable Then
    objRange.AppendRow
    ' objRange points to beginning of new row
    objRange.Select
End If
```

### 15.3.2.5.2 Application

**Property:** Application as [Application](#)<sup>382</sup> (read-only)

**Description**

Accesses the Authentic Desktop 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.

### 15.3.2.5.3 CanPerformAction

**Method:** CanPerformAction (*eAction* as [SPYAuthenticActions](#)<sup>605</sup>, *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](#)<sup>411</sup>.

**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](#)<sup>430</sup>.

#### 15.3.2.5.4 CanPerformActionWith

**Method:** CanPerformActionWith (eAction as [SPYAuthenticActions](#)<sup>605</sup>, 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](#)<sup>411</sup>.

##### 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](#)<sup>430</sup>.

#### 15.3.2.5.5 Clone

**Method:** Clone() as [AuthenticRange](#)<sup>407</sup>

##### 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.

#### 15.3.2.5.6 CollapsToBegin

**Method:** CollapsToBegin() as [AuthenticRange](#)<sup>407</sup>

##### 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.

### 15.3.2.5.7 CollapsToEnd

**Method:** CollapsToEnd() as [AuthenticRange](#) <sup>407</sup>

#### 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.

### 15.3.2.5.8 Copy

**Method:** Copy() as Boolean

#### Description

Returns *False* if the range contains no portions of the document that may be copied.

Returns *True* if text, and in case of fully selected XML elements the elements as well, has been copied to the copy/paste buffer.

#### Errors

- 2001 The authentic range object or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

### 15.3.2.5.9 Cut

**Method:** Cut() as Boolean

#### Description

Returns *False* if the range contains portions of the document that may not be deleted.

Returns *True* after text, and in case of fully selected XML elements the elements as well, has been deleted from the document and saved in the copy/paste buffer.

#### Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

### 15.3.2.5.10 Delete

**Method:** Delete() as Boolean

#### Description

Returns *False* if the range contains portions of the document that may not be deleted.

Returns *True* after text, and in case of fully selected XML elements the elements as well, has been deleted from the document.

**Errors**

- 2001 The authentic range object or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

### 15.3.2.5.11 DeleteRow

**Method:** DeleteRow() as Boolean

**Description**

If the beginning of the range is inside a dynamic table, this method deletes the selected row. The selection of the range gets modified to point to the next element after the deleted row. The function returns *true*, if the delete operation was successful, otherwise *false*.

**Errors**

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

**Examples**

```
' _____  
' Scripting environment - VBScript  
' Delete selected row from dynamically growing table  
' _____  
  
Dim objRange  
' we assume that the active document is open in authentic view mode  
Set objRange = Application.ActiveDocument.AuthenticView.Selection  
  
' check if we are in a table  
If objRange.IsInDynamicTable Then  
    objRange.DeleteRow  
End If
```

### 15.3.2.5.12 DuplicateRow

**Method:** DuplicateRow() as Boolean

**Description**

If the beginning of the range is inside a dynamic table, this method inserts a duplicate of the current row after the selected one. The selection of the range gets modified to point to the beginning of the new row. The function returns *true* if the duplicate operation was successful, otherwise *false*.

**Errors**

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

**Examples**

```
' _____  
' Scripting environment - VBScript  
' duplicate row in current dynamically growable table  
' _____  
  
Dim objRange
```

```
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' check if we can insert something
If objRange.IsInDynamicTable Then
    objRange.DuplicateRow
    ' objRange points to beginning of new row
    objRange.Select
End If
```

### 15.3.2.5.13 EvaluateXPath

**Method:** EvaluateXPath (strExpression as string) strValue as string

#### Return Value

The method returns a string

#### Description

EvaluateXPath() executes an XPath expression with the context node being the beginning of the range selection. The result is returned as a string, in the case of a sequence it is a space-separated string. If XML context node is irrelevant, the user may provide any node, like `AuthenticView.XMLDataRoot`.

#### Errors

- 2001 Invalid object
- 2005 Invalid parameter
- 2008 Internal error
- 2202 Missing context node
- 2211 XPath error

### 15.3.2.5.14 ExpandTo

**Method:** ExpandTo (eKind as [SPYAuthenticElementKind](#)<sup>605</sup>), as [AuthenticRange](#)<sup>407</sup>

#### 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.

### 15.3.2.5.15 FirstTextPosition

**Property:** FirstTextPosition as Long

#### Description

Set or get the left-most text position index of the range object. This index is always less or equal to [LastTextPosition](#)<sup>426</sup>. 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](#)<sup>426</sup>, [LastTextPosition](#)<sup>426</sup> 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
```

#### 15.3.2.5.16 FirstXMLData

**Property:** FirstXMLData as [XMLData](#)<sup>593</sup>

### 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](#)<sup>416</sup> attribute.

Whenever you set FirstXMLData to a new data object, [FirstXMLDataOffset](#)<sup>416</sup> 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](#)<sup>416</sup> selects a position greater than the current [LastXMLData](#)<sup>427</sup> / [LastXMLDataOffset](#)<sup>427</sup>, the latter gets moved to the new start position.

HINT: You can use the [FirstXMLData](#)<sup>415</sup> and [LastXMLData](#)<sup>427</sup> properties to directly access and manipulate the underlying XML document in those cases where the methods available with the [AuthenticRange](#)<sup>407</sup> 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
```

#### 15.3.2.5.17 FirstXMLDataOffset

**Property:** FirstXMLDataOffset as Long

#### Description

Set or get the cursor position offset inside [FirstXMLData](#)<sup>415</sup> element for the beginning of the range. Offset positions are based on the characters returned by the [Text](#)<sup>435</sup> property, and start with 0. When setting a new offset, use -1 to set the offset to the last possible position in the element. The following cases require specific attention:

- The textual form of entries in Combo Boxes, Check Boxes and similar controls can be different from what you see on screen. Although the data offset is based on this text, there only two valid offset positions, one at the beginning and one at the end of the entry. An attempt to set the offset to somewhere in the middle of the entry, will result in the offset being set to the end.
- The textual form of XML Entities might differ in length from their representation on the screen. The offset is based on this textual form.

If FirstXMLData / [FirstXMLDataOffset](#)<sup>416</sup> selects a position after the current [LastXMLData](#)<sup>427</sup> / [LastXMLDataOffset](#)<sup>427</sup>, 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
```

#### 15.3.2.5.18 GetElementAttributeNames

**Method:** GetElementAttributeNames (*strElementName* as String, *out\_arrAttributeNames* as Variant)

### Description

Retrieve the names of all attributes for the enclosing element with the specified name. Use the element/attribute pairs, to set or get the attribute value with the methods [GetElementAttributeValue](#)<sup>418</sup> and [SetElementAttributeValue](#)<sup>433</sup>.

### 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](#)<sup>433</sup>.

### 15.3.2.5.19 GetElementAttributeValue

**Method:** GetElementAttributeValue (*strElementName* as String, *strAttributeName* as String) as String

**Description**

Retrieve the value of the attribute specified in *strAttributeName*, for the element identified with *strElementName*. If the attribute is supported but has no value assigned, the empty string is returned. To find out the names of attributes supported by an element, use [GetElementAttributeNames](#)<sup>417</sup>, or [HasElementAttribute](#)<sup>422</sup>.

**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](#)<sup>433</sup>.

### 15.3.2.5.20 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](#)<sup>417</sup>, 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.

### 15.3.2.5.21 GetEntityNames

**Method:** GetEntityNames (*out\_arrEntityNames* as Variant)

**Description**

Retrieve the names of all defined entities. The list of retrieved entities is independent of the current selection, or location. Use one of these names with the [InsertEntity](#)<sup>422</sup> 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](#)<sup>418</sup> and [InsertEntity](#)<sup>422</sup>.

### 15.3.2.5.22 GetVariableValue

**Method:** `GetVariableValue (strName as string) strVal as string`

**Return Value**

Gets the value of the variable named as the method's parameter.

**Errors**

- 2001 Invalid object.
- 2202 No context.
- 2204 No such variable in scope
- 2205 Variable cannot be evaluated
- 2206 Variable returns sequence
- 2209 Invalid parameter

### 15.3.2.5.23 Goto

**Method:** `Goto (eKind as SPYAuthenticElementKind605, nCount as Long, eFrom as SPYAuthenticDocumentPosition605) as AuthenticRange407`

**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.

### 15.3.2.5.24 GotoNext

**Method:** GotoNext (*eKind* as [SPYAuthenticElementKind](#)<sup>605</sup>) as [AuthenticRange](#)<sup>407</sup>

#### 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
```

### 15.3.2.5.25 GotoNextCursorPosition

**Method:** GotoNextCursorPosition() as [AuthenticRange](#)<sup>407</sup>

#### 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.

### 15.3.2.5.26 GotoPrevious

**Method:** GotoPrevious (eKind as [SPYAuthenticElementKind](#)<sup>605</sup>) as [AuthenticRange](#)<sup>407</sup>

#### 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
```

### 15.3.2.5.27 GotoPreviousCursorPosition

**Method:** GotoPreviousCursorPosition() as [AuthenticRange](#)<sup>407</sup>

#### 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.

### 15.3.2.5.28 HasElementAttribute

**Method:** HasElementAttribute (*strElementName* as String, *strAttributeName* as String) as Boolean

#### Description

Tests if the enclosing element with name *strElementName*, supports the attribute specified in *strAttributeName*.

#### Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid element name was specified.  
Invalid address for the return parameter was specified.

### 15.3.2.5.29 InsertEntity

**Method:** InsertEntity (*strEntityName* as String)

#### Description

Replace the ranges selection with the specified entity. The specified entity must be one of the entity names returned by [GetEntityNames](#)<sup>418</sup>.

#### 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
```

### 15.3.2.5.30 InsertRow

**Method:** InsertRow() as Boolean

#### Description

If the beginning of the range is inside a dynamic table, this method inserts a new row before the current one. The selection of the range gets modified to point to the beginning of the newly inserted row. The function returns *true* if the insert operation was successful, otherwise *false*.

#### Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

#### Examples

```
'-----  
' Scripting environment - VBScript  
' Insert row at beginning of current dynamically growing table  
'-----  
Dim objRange  
' we assume that the active document is open in authentic view mode  
Set objRange = Application.ActiveDocument.AuthenticView.Selection  
  
' check if we can insert something  
If objRange.IsInDynamicTable Then  
    objRange.InsertRow  
    ' objRange points to beginning of new row  
    objRange.Select  
End If
```

### 15.3.2.5.31 IsCopyEnabled

**Property:** IsCopyEnabled as Boolean (read-only)

#### Description

Checks if the copy operation is supported for this range.

#### Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

### 15.3.2.5.32 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.

### 15.3.2.5.33 IsDeleteEnabled

**Property:** IsDeleteEnabled as Boolean (read-only)

#### Description

Checks if the delete operation is supported for this range.

#### Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

### 15.3.2.5.34 IsEmpty

**Method:** IsEmpty() as Boolean

#### Description

Tests if the first and last position of the range are equal.

#### Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

### 15.3.2.5.35 IsEqual

**Method:** IsEqual (*objCmpRange* as [AuthenticRange](#)<sup>407</sup>) 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.

### 15.3.2.5.36 IsFirstRow

**Property:** IsFirstRow as Boolean (read-only)

#### Description

Test if the range is in the first row of a table. Which table is taken into consideration depends on the extent of the range. If the selection exceeds a single row of a table, the check is if this table is the first element in an embedding table. See the entry helpers of the user manual for more information.

#### Errors

- 2001 The authentic range object, or its related view object is no longer valid.

2005 Invalid address for the return parameter was specified.

### 15.3.2.5.37 IsInDynamicTable

**Method:** IsInDynamicTable() as Boolean

#### Description

Test if the whole range is inside a table that supports the different row operations like 'insert', 'append', duplicate, etc.

#### Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

### 15.3.2.5.38 IsLastRow

**Property:** IsLastRow as Boolean (read-only)

#### Description

Test if the range is in the last row of a table. Which table is taken into consideration depends on the extent of the range. If the selection exceeds a single row of a table, the check is if this table is the last element in an embedding table. See the entry helpers of the user manual for more information.

#### Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

### 15.3.2.5.39 IsPasteEnabled

**Property:** IsPasteEnabled as Boolean (read-only)

#### Description

Checks if the paste operation is supported for this range.

#### Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

### 15.3.2.5.40 IsSelected

**Property:** IsSelected as Boolean

#### Description

Returns true() if selection is present. The selection range still can be empty: that happens when e.g. only the cursor is set.

#### 15.3.2.5.41 IsTextStateApplied

**Method:** IsTextStateApplied (*i\_strElementName* as String) as Boolean

##### Description

Checks if all the selected text is embedded into an XML Element with name *i\_strElementName*. Common examples for the parameter *i\_strElementName* are "strong", "bold" or "italic".

##### Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

#### 15.3.2.5.42 LastTextPosition

**Property:** LastTextPosition as Long

##### Description

Set or get the rightmost text position index of the range object. This index is always greater or equal to [FirstTextPosition](#)<sup>414</sup>. Indexing starts with 0 at the document beginning, and increments with every different position that the text cursor can occupy. Incrementing the text position by 1, has the same effect as the cursor-right key. Decreasing the text position by 1 has the same effect as the cursor-left key.

If you set LastTextPosition to a value less than the current [FirstTextPosition](#)<sup>414</sup>, [FirstTextPosition](#)<sup>414</sup> gets set to the new LastTextPosition.

HINT: Use text cursor positions with care, since this is a costly operation compared to XMLData based cursor positioning.

##### Errors

- 2001 The authentic range object, or its related view object is not valid.
- 2005 Invalid address for the return parameter was specified.
- 2006 A text position outside the document was specified.

##### Examples

```
' _____
' Scripting environment - VBScript
' _____

Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

nDocStartPosition = objAuthenticView.DocumentBegin.FirstTextPosition
nDocEndPosition = objAuthenticView.DocumentEnd.FirstTextPosition

' let's create a range that selects the whole document
' in an inefficient way
```

```

Dim objRange
' we need to get a (any) range object first
Set objRange = objAuthenticView.DocumentBegin
objRange.FirstTextPosition = nDocStartPosition
objRange.LastTextPosition = nDocEndPosition

' let's check if we got it right
If objRange.IsEqual(objAuthenticView.WholeDocument) Then
    MsgBox "Test using direct text cursor positioning was ok"
Else
    MsgBox "Oops!"
End If

```

### 15.3.2.5.43 LastXMLData

**Property:** LastXMLData as [XMLData](#)<sup>593</sup>

#### 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](#)<sup>427</sup> attribute.

Whenever you set LastXMLData to a new data object, [LastXMLDataOffset](#)<sup>427</sup> 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](#)<sup>427</sup>, select a position less than the current [FirstXMLData](#)<sup>415</sup> / [FirstXMLDataOffset](#)<sup>416</sup>, the latter gets moved to the new end position.

HINT: You can use the [FirstXMLData](#)<sup>415</sup> and [LastXMLData](#)<sup>427</sup> properties to directly access and manipulate the underlying XML document in those cases, where the methods available with the [AuthenticRange](#)<sup>407</sup> 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.

### 15.3.2.5.44 LastXMLDataOffset

**Property:** LastXMLDataOffset as Long

#### Description

Set or get the cursor position inside [LastXMLData](#)<sup>427</sup> element for the end of the range.

Offset positions are based on the characters returned by the [Text](#)<sup>435</sup> 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](#)<sup>427</sup> / [LastXMLDataOffset](#)<sup>427</sup> selects a position before [FirstXMLData](#)<sup>415</sup> / [FirstXMLDataOffset](#)<sup>416</sup>, 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
```

#### 15.3.2.5.45 MoveBegin

**Method:** MoveBegin (eKind as [SPYAuthenticElementKind](#)<sup>605</sup>, nCount as Long) as [AuthenticRange](#)<sup>407</sup>

#### Description

Move the beginning of the range to the beginning of the nCount element of type eKind. Counting starts at the current beginning of the range object.

Use positive numbers for `nCount` to move towards the document end, use negative numbers to move towards document beginning. The end of the range stays unmoved, unless the new beginning would be larger than it. In this case, the end is moved to the new beginning. The method returns the modified range object.

**Errors**

- 2001 The authentic range object, or its related view object is no longer valid.
- 2003 Target lies after end of document.
- 2004 Target lies before beginning of document.
- 2005 Invalid element kind specified.  
Invalid address for the return parameter was specified.

### 15.3.2.5.46 MoveEnd

**Method:** `MoveEnd` (*eKind* as [SPYAuthenticElementKind](#)<sup>605</sup>, *nCount* as Long) as [AuthenticRange](#)<sup>407</sup>

**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.

### 15.3.2.5.47 MoveRowDown

**Method:** `MoveRowDown`() as Boolean

**Description**

If the beginning of the range is inside a dynamic table and selects a row which is not the last row in this table, this method swaps this row with the row immediately below. The selection of the range moves with the row, but does not otherwise change. The function returns *true* if the move operation was successful, otherwise *false*.

**Errors**

- 2001 The authentic range object or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

### 15.3.2.5.48 MoveRowUp

**Method:** `MoveRowUp`() as Boolean

**Description**

If the beginning of the range is inside a dynamic table and selects a row which is not the first row in this table, this method swaps this row with the row above. The selection of the range moves with the row, but does not change otherwise. The function returns *true* if the move operation was successful, otherwise *false*.

**Errors**

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

### 15.3.2.5.49 Parent

**Property:** Parent as [AuthenticView](#)<sup>435</sup> (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.

### 15.3.2.5.50 Paste

**Method:** Paste() as Boolean

**Description**

Returns *False* if the copy/paste buffer is empty, or its content cannot replace the current selection.

Otherwise, deletes the current selection, inserts the content of the copy/paste buffer, and returns *True*.

**Errors**

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

### 15.3.2.5.51 PerformAction

**Method:** PerformAction (*eAction* as [SPYAuthenticActions](#)<sup>605</sup>, *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](#)<sup>411</sup>.

**Errors**

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

2007 Invalid action was specified.

### Examples

```
' -----
' Scripting environment - VBScript
' Insert the innermost element
' -----
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' we determine the elements that can be inserted at the current position
Dim arrElements()
objRange.CanPerformActionWith spyAuthenticInsertBefore, arrElements

' we insert the first (innermost) element
If UBound(arrElements) >= 0 Then
    objRange.PerformAction spyAuthenticInsertBefore, arrElements(0)
    ' objRange now points to the beginning of the inserted element
    ' we set a default value and position at its end
    objRange.Text = "Hello"
    objRange.ExpandTo(spyAuthenticTag).CollapsToEnd().Select
Else
    MsgBox "Can't insert any elements at current position"
End If
```

### 15.3.2.5.52 Select

**Method:** Select()

#### Description

Makes this range the current user interface selection. You can achieve the same result using:  
'objRange.Parent.Selection = objRange'

#### Errors

2001 The authentic range object or its related view object is no longer valid.

### Examples

```
' -----
' Scripting environment - VBScript
' -----
Dim objAuthenticView
' 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()
```

### 15.3.2.5.53 SelectNext

**Method:** SelectNext (eKind as [SPYAuthenticElementKind](#)<sup>605</sup>) as [AuthenticRange](#)<sup>407</sup>

#### 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
```

### 15.3.2.5.54 SelectPrevious

**Method:** GotoPrevious (eKind as [SPYAuthenticElementKind](#)<sup>605</sup>) as [AuthenticRange](#)<sup>407</sup>

#### Description

Selects the element of type eKind before the current beginning of the range. The method returns the modified range object.

#### Errors

- 2001 The authentic range object, or its related view object is no longer valid.

- 2004 Target lies before begin of document.
- 2005 Invalid element kind specified.  
Invalid address for the return parameter was specified.

### Examples

```
' -----
' Scripting environment - VBScript
' Scan through the whole document tag-by-tag
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentEnd
Dim bBeginOfDocument
bBeginOfDocument = False

On Error Resume Next
While Not bBeginOfDocument
    objRange.SelectPrevious(spyAuthenticTag).Select
    If ((Err.number - vbObjecterror) = 2004) Then
        bBeginOfDocument = True
        Err.Clear
    ElseIf (Err.number <> 0) Then
        Err.Raise ' forward error
    End If
Wend
```

#### 15.3.2.5.55 SetElementAttributeValue

**Method:** SetElementAttributeValue (*strElementName* as String, *strAttributeName* as String, *strAttributeValue* as String)

#### Description

Set the value of the attribute specified in *strAttributeName* for the element identified with *strElementName*. If the attribute is supported but has no value assigned, the empty string is returned. To find out the names of attributes supported by an element, use [GetElementAttributeNames](#)<sup>417</sup>, or [HasElementAttribute](#)<sup>422</sup>.

#### 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

```

### 15.3.2.5.56 SetFromRange

**Method:** SetFromRange (*objSrcRange* as [AuthenticRange](#)<sup>407</sup>)

#### 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.

### 15.3.2.5.57 SetVariableValue

**Method:** SetVariableValue (*strName* as string, *strValue* as string)

#### Return Value

Sets the value (second parameter) of the variable named in the first parameter.

**Errors**

- 2201 Invalid object.
- 2202 No context.
- 2204 No such variable in scope
- 2205 Variable cannot be evaluated
- 2206 Variable returns sequence
- 2207 Variable read-only
- 2208 No modification allowed

### 15.3.2.5.58 Text

**Property:** Text as String

**Description**

Set or get the textual content selected by the range object.

The number of characters retrieved are not necessarily identical, as there are text cursor positions between the beginning and end of the selected range. Most document elements support an end cursor position different to the beginning cursor position of the following element. Drop-down lists maintain only one cursor position, but can select strings of any length. In the case of radio buttons and check boxes, the text property value holds the string of the corresponding XML element.

If the range selects more than one element, the text is the concatenation of the single texts. XML entities are expanded so that '&' is expected as '&amp;'.

Setting the text to the empty string, does not delete any XML elements. Use [Cut](#)<sup>412</sup>, [Delete](#)<sup>412</sup> or [PerformAction](#)<sup>430</sup> 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.

### 15.3.2.6 AuthenticView

**Properties**

[Application](#)<sup>443</sup>  
[AsXMLString](#)<sup>443</sup>  
[DocumentBegin](#)<sup>445</sup>  
[DocumentEnd](#)<sup>445</sup>  
[Event](#)<sup>446</sup>  
[MarkupVisibility](#)<sup>449</sup>  
[Parent](#)<sup>449</sup>  
[Selection](#)<sup>450</sup>  
[XMLDataRoot](#)<sup>452</sup>  
[WholeDocument](#)<sup>452</sup>

**Methods**

[Goto](#)<sup>447</sup>  
[IsRedoEnabled](#)<sup>448</sup>  
[IsUndoEnabled](#)<sup>448</sup>  
[Print](#)<sup>449</sup>  
[Redo](#)<sup>449</sup>  
[Undo](#)<sup>451</sup>  
[UpdateXMLInstanceEntities](#)<sup>451</sup>

**Events**

[OnBeforeCopy](#)<sup>436</sup>  
[OnBeforeCut](#)<sup>436</sup>  
[OnBeforeDelete](#)<sup>437</sup>  
[OnBeforeDrop](#)<sup>437</sup>  
[OnBeforePaste](#)<sup>438</sup>  
[OnDragOver](#)<sup>439</sup>  
[OnKeyboardEvent](#)<sup>439</sup>  
[OnMouseEvent](#)<sup>440</sup>  
[OnSelectionChanged](#)<sup>441</sup>

## Description

AuthenticView and its child objects [AuthenticRange](#)<sup>407</sup> and AuthenticDataTransfer provide you with an interface for **Authentic View**, 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](#)<sup>407</sup>)

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](#)<sup>407</sup> to make navigation inside the document straight-forward, and to allow for the flexible selection of logical text elements. Use the properties [DocumentBegin](#)<sup>445</sup>, [DocumentEnd](#)<sup>445</sup>, or [WholeDocument](#)<sup>452</sup> for simple selections, while using the [Goto](#)<sup>447</sup> method for more complex selections. To navigate relative to a given document range, see the methods and properties of the [AuthenticRange](#)<sup>407</sup> object.

### 15.3.2.6.1 Events

#### 15.3.2.6.1.1 OnBeforeCopy

**Event:** OnBeforeCopy() as Boolean

##### **Scripting environment - VBScript:**

```
Function On_AuthenticBeforeCopy()
    ' On_AuthenticBeforeCopy = False ' to disable operation
End Function
```

##### **Scripting environment - JScript:**

```
function On_AuthenticBeforeCopy()
{
    // return false; /* to disable operation */
}
```

##### **IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (21, ...) // nEventId = 21
```

## Description

This event gets triggered before a copy operation gets performed on the document. Return *True* (or nothing) to allow copy operation. Return *False* to disable copying.

#### 15.3.2.6.1.2 OnBeforeCut

**Event:** OnBeforeCut() as Boolean

**Scripting environment - VBScript:**

```
Function On_AuthenticBeforeCut()
    ' On_AuthenticBeforeCut = False ' to disable operation
End Function
```

**Scripting environment - JScript:**

```
function On_AuthenticBeforeCut()
{
    // return false; /* to disable operation */
}
```

**IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (20, ...) // nEventId = 20
```

**Description**

This event gets triggered before a cut operation gets performed on the document. Return *True* (or nothing) to allow cut operation. Return *False* to disable operation.

**15.3.2.6.1.3 OnBeforeDelete**

**Event:** OnBeforeDelete() as Boolean

**Scripting environment - VBScript:**

```
Function On_AuthenticBeforeDelete()
    ' On_AuthenticBeforeDelete = False ' to disable operation
End Function
```

**Scripting environment - JScript:**

```
function On_AuthenticBeforeDelete()
{
    // return false; /* to disable operation */
}
```

**IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (22, ...) // nEventId = 22
```

**Description**

This event gets triggered before a delete operation gets performed on the document. Return *True* (or nothing) to allow delete operation. Return *False* to disable operation.

**15.3.2.6.1.4 OnBeforeDrop**

**Event:** OnBeforeDrop (*i\_nXPos* as Long, *i\_nYPos* as Long, *i\_ipRange* as [AuthenticRange](#)<sup>407</sup>, *i\_ipData* as cancelBoolean

**Scripting environment - VBScript:**

```
Function On_AuthenticBeforeDrop(nXPos, nYPos, objRange, objData)
    ' On_AuthenticBeforeDrop = False ' to disable operation
End Function
```

**Scripting environment - JScript:**

```
function On_AuthenticBeforeDrop(nXPos, nYPos, objRange, objData)
{
    // return false; /* to disable operation */
}
```

**IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (11, ...) // nEventId = 11
```

**Description**

This event gets triggered whenever a previously dragged object gets dropped inside the application window. All event related information gets passed as parameters.

The first two parameters specify the mouse position at the time when the event occurred. The parameter *objRange* passes a range object that selects the XML element below the mouse position. The value of this parameter might be *NULL*. Be sure to check before you access the range object. The parameter *objData* allows to access information about the object being dragged.

Return *False* to cancel the drop operation. Return *True* (or nothing) to continue normal operation.

### 15.3.2.6.1.5 OnBeforePaste

**Event:** OnBeforePaste (*objData* as Variant, *strType* as String) as Boolean

**Scripting environment - VBScript:**

```
Function On_AuthenticBeforePaste(objData, strType)
    ' On_AuthenticBeforePaste = False ' to disable operation
End Function
```

**Scripting environment - JScript:**

```
function On_AuthenticBeforePaste(objData, strType)
{
    // return false; /* to disable operation */
}
```

**IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (19, ...) // nEventId = 19
```

**Description**

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.

### 15.3.2.6.1.6 OnBeforeSave

**Event:** OnBeforeSave (SaveAs flag) as Boolean

**Description:** `OnBeforeSave` gives the opportunity to e.g. warn the user about overwriting the existing XML document, or to make the document read-only when specific circumstances are not met. The event will be fired before the file dialog is shown.

### 15.3.2.6.1.7 `OnDragOver`

**Event:** `OnDragOver` (`nXPos` as Long, `nYPos` as Long, `eMouseEvent` as [SPYMouseEvent](#)<sup>610</sup>, `objRange` as [AuthenticRange](#)<sup>407</sup>, `objData` as `AuthenticDataTransfer`) as Boolean

**Scripting environment - VBScript:**

```
Function On_AuthenticDragOver(nXPos, nYPos, eMouseEvent, objRange, objData)
    ' On_AuthenticDragOver = False ' to disable operation
End Function
```

**Scripting environment - JScript:**

```
function On_AuthenticDragOver(nXPos, nYPos, eMouseEvent, objRange, objData)
{
    // return false; /* to disable operation */
}
```

**IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (10, ...) // nEventId = 10
```

**Description**

This event gets triggered whenever an object from within or outside of Authentic View gets dragged with the mouse over the application window. All event related information gets passed as parameters.

The first three parameters specify the mouse position, the mouse button status and the status of the virtual keys at the time when the event occurred. The parameter `objRange` passes a range object that selects the XML element below the mouse position. The value of this parameter might be `NULL`. Be sure to check before you access the range object. The parameter `objData` allows to access information about the object being dragged.

Return `False` to cancel the drag operation. Return `True` (or nothing) to continue normal operation.

### 15.3.2.6.1.8 `OnKeyboardEvent`

**Event:** `OnKeyboardEvent` (`eKeyEvent` as [SPYKeyEvent](#)<sup>609</sup>, `nKeyCode` as Long, `nVirtualKeyStatus` as Long) as Boolean

**Scripting environment - VBScript:**

```
Function On_AuthenticKeyboardEvent(eKeyEvent, nKeyCode, nVirtualKeyStatus)
    ' On_AuthenticKeyboardEvent = True ' to cancel bubbling of event
End Function
```

**Scripting environment - JScript:**

```
function On_AuthenticKeyboardEvent(eKeyEvent, nKeyCode, nVirtualKeyStatus)
{
    // return true; /* to cancel bubbling of event */
}
```

```
}

```

**IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (30, ...) // nEventId = 30

```

**Description**

This event gets triggered for *WM\_KEYDOWN*, *WM\_KEYUP* and *WM\_CHAR* Windows messages.

The actual message type is available in the *eKeyEvent* parameter. The status of virtual keys is combined in the parameter *nVirtualKeyStatus*. Use the bit-masks defined in the enumeration datatype [SPYVirtualKeyMask](#)<sup>616</sup>, to test for the different keys or their combinations.

### 15.3.2.6.1.9 OnLoad

**Event:** OnLoad ()

**Description:** OnLoad can be used e.g. to restrict some AuthenticView functionality, as shown in the example below:

```
function On_AuthenticLoad( )
{
    // We are disabling all entry helpers in order to prevent user from manipulating XML tree
    AuthenticView.DisableElementEntryHelper();
    AuthenticView.DisableAttributeEntryHelper();

    // We are also disabling the markup buttons for the same purpose
    AuthenticView.SetToolBarButtonState( 'AuthenticMarkupSmall', authenticToolBarButtonDisabled );
    AuthenticView.SetToolBarButtonState( 'AuthenticMarkupLarge', authenticToolBarButtonDisabled );
    AuthenticView.SetToolBarButtonState( 'AuthenticMarkupMixed', authenticToolBarButtonDisabled );
}

```

In the example the status of the Markup Small, Markup Large, Markup Mixed toolbar buttons are manipulated with the help of button identifiers. See [complete list](#)<sup>441</sup>.

### 15.3.2.6.1.10 OnMouseEvent

**Event:** OnMouseEvent (*nXPos* as Long, *nYPos* as Long, *eMouseEvent* as [SPYMouseEvent](#)<sup>610</sup>, *objRange* as [AuthenticRange](#)<sup>407</sup>) as Boolean

**Scripting environment - VBScript:**

```
Function On_AuthenticMouseEvent(nXPos, nYPos, eMouseEvent, objRange)
    ' On_AuthenticMouseEvent = True ' to cancel bubbling of event
End Function

```

**Scripting environment - JScript:**

```
function On_AuthenticMouseEvent(nXPos, nYPos, eMouseEvent, objRange)
{
    // return true; /* to cancel bubbling of event */
}

```

**IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (31, ...) // nEventId = 31
```

**Description**

This event gets triggered for every mouse movement and mouse button Windows message.

The actual message type and the mouse buttons status, is available in the *eMouseEvent* parameter. Use the bit-masks defined in the enumeration datatype [SPYMouseEvent](#)<sup>610</sup> 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*.

**15.3.2.6.1.11 OnSelectionChanged**

**Event:** OnSelectionChanged (*objNewSelection* as [AuthenticRange](#)<sup>407</sup>)

**Scripting environment - VBScript:**

```
Function On_AuthenticSelectionChanged (objNewSelection)
End Function
```

**Scripting environment - JScript:**

```
function On_AuthenticSelectionChanged (objNewSelection)
{
}
}
```

**IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (23, ...) // nEventId = 23
```

**Description**

This event gets triggered whenever the selection in the user interface changes.

**15.3.2.6.1.12 OnToolbarButtonClicked**

**Event:** OnToolbarButtonClicked (Button identifier)

**Description:** *OnToolbarButtonClicked* is fired when a toolbar button was clicked by user. The parameter button identifier helps to determine which button was clicked. The list of predefined button identifiers is below:

- AuthenticPrint
- AuthenticPrintPreview
- AuthenticUndo
- AuthenticRedo
- AuthenticCut
- AuthenticCopy
- AuthenticPaste
- AuthenticClear
- AuthenticMarkupHide

- AuthenticMarkupLarge
- AuthenticMarkupMixed
- AuthenticMarkupSmall
- AuthenticValidate
- AuthenticChangeWorkingDBXMLCell
- AuthenticSave
- AuthenticSaveAs
- AuthenticReload
- AuthenticTableInsertRow
- AuthenticTableAppendRow
- AuthenticTableDeleteRow
- AuthenticTableInsertCol
- AuthenticTableAppendCol
- AuthenticTableDeleteCol
- AuthenticTableJoinCellRight
- AuthenticTableJoinCellLeft
- AuthenticTableJoinCellAbove
- AuthenticTableJoinCellBelow
- AuthenticTableSplitCellHorizontally
- AuthenticTableSplitCellVertically
- AuthenticTableAlignCellContentTop
- AuthenticTableCenterCellVertically
- AuthenticTableAlignCellContentBottom
- AuthenticTableAlignCellContentLeft
- AuthenticTableCenterCellContent
- AuthenticTableAlignCellContentRight
- AuthenticTableJustifyCellContent
- AuthenticTableInsertTable
- AuthenticTableDeleteTable
- AuthenticTableProperties
- AuthenticAppendRow
- AuthenticInsertRow
- AuthenticDuplicateRow
- AuthenticMoveRowUp
- AuthenticMoveRowDown
- AuthenticDeleteRow
- AuthenticDefineEntities
- AuthenticXMLSignature

For custom buttons the user might add his own identifiers. Please, note that the user must take care, as the identifiers are not checked for uniqueness. The same identifiers can be used to identify buttons in the `Set/GetToolBarState()` COM API calls. By adding code for different buttons, the user is in the position to completely redefine the `AuthenticView` toolbar behavior, adding own methods for table manipulation, etc.

#### 15.3.2.6.1.13 *OnToolBarButtonExecuted*

**Event:** `OnToolBarButtonExecuted` (Button identifier)

**Description:** `OnToolBarButtonClicked` is fired when a toolbar button was clicked by user. The parameter button identifier helps to determine which button was clicked. See the list of [predefined button identifiers](#)<sup>441</sup>.

`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.

#### 15.3.2.6.1.14 OnUserAddedXMLNode

**Event:** `OnUserAddedXMLNode` (XML node)

**Description:** `OnUserAddedXMLNode` will be fired when the user adds an XML node as a primary action. This happens in the situations, where the user clicks on

- auto-add hyperlinks (see example `OnUserAddedXMLNode.sps`)
- the Insert..., Insert After..., Insert Before... context menu items
- Append row, Insert row toolbar buttons
- Insert After..., Insert Before... actions in element entry helper (outside `StyleVision`)

The event doesn't get fired on Duplicate row, or when the node was added externally (e.g. via COM API), or on Apply (e.g. Text State Icons), or when in XML table operations or in DB operations.

The event parameter is the XML node object, which was added giving the user an opportunity to manipulate the XML node added. An elaborate example for an event handler can be found in the `OnUserAddedXMLNode.sps` file.

#### 15.3.2.6.2 Application

**Property:** `Application` as [Application](#)<sup>382</sup> (read-only)

##### Description

Accesses the Authentic Desktop application object.

##### Errors

- 2000 The authentic view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

#### 15.3.2.6.3 AsXMLString

**Property:** `AsXMLString` as String

##### Description

Returns or sets the document content as an XML string. Setting the content to a new value does not change the schema file or sps file in use. If the new XMLString does not match the actual schema file error 2011 gets returned.

**Errors**

- 2000 The authentic view object is no longer valid.
- 2011 AsXMLString was set to a value which is no valid XML for the current schema file.

### 15.3.2.6.4 ContextMenu

**Property:** ContextMenu() as ContextMenu

**Description**

The property `ContextMenu` gives access to customize the context menu. The best place to do it is in the event handler `OnContextMenuActivated`.

**Errors**

- 2000 Invalid object.
- 2005 Invalid parameter.

### 15.3.2.6.5 CreateXMLNode

**Method:** CreateXMLNode (*nKind* as [SPYXMLDataKind](#)<sup>617</sup>) as [XMLData](#)<sup>593</sup>

**Return Value**

The method returns the new [XMLData](#)<sup>593</sup> object.

**Description**

To create a new XMLData object use the CreateXMLNode() method.

**Errors**

- 2000 Invalid object.
- 2012 Cannot create XML node.

### 15.3.2.6.6 DisableAttributeEntryHelper

**Method:** DisableAttributeEntryHelper()

**Description**

DisableAttributeEntryHelper() disables the attribute entry helper in XMLSpy, Authentic Desktop and Authentic Browser plug-in.

**Errors**

- 2000 Invalid object.

### 15.3.2.6.7 DisableElementEntryHelper

**Method:** DisableElementEntryHelper ()

**Description**

DisableElementEntryHelper () disables the element entry helper in XMLSpy, Authentic Desktop and Authentic Browser plug-in.

**Errors**

2000 Invalid object.

### 15.3.2.6.8 DisableEntityEntryHelper

**Method:** DisableEntityEntryHelper ()

**Description**

DisableEntityEntryHelper () disables the entity entry helper in XMLSpy, Authentic Desktop and Authentic Browser plug-in.

**Errors**

2000 Invalid object.

### 15.3.2.6.9 DocumentBegin

**Property:** DocumentBegin as [AuthenticRange](#)<sup>407</sup> (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.

### 15.3.2.6.10 DocumentEnd

**Property:** DocumentEnd as [AuthenticRange](#)<sup>407</sup> (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.

### 15.3.2.6.11 DoNotPerformStandardAction

**Method:** DoNotPerformStandardAction ()

#### Description

DoNotPerformStandardAction() serves as cancel bubble for macros, and stops further execution after macro has finished.

#### Errors

2000 Invalid object.

### 15.3.2.6.12 EvaluateXPath

**Method:** EvaluateXPath (XMLData as [XMLData](#)<sup>593</sup>, 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.

### 15.3.2.6.13 Event

**Property:** Event as AuthenticEvent (read-only)

#### Description

This property gives access to parameters of the last event in the same way as OldAuthenticView.event does. Since all events for the scripting environment and external clients are now available with parameters this Event property should only be used from within IDE-Plugins.

#### Errors

2000 The authentic view object is no longer valid.  
2005 Invalid address for the return parameter was specified.

### 15.3.2.6.14 EventContext

**Property:** EventContext() as EventContext

#### Description

`EventContext` property gives access to the running macros context. See the [EventContext](#)<sup>404</sup> interface description for more details.

#### Errors

2000 Invalid object.

### 15.3.2.6.15 GetToolbarButtonState

**Method:** GetToolbarButtonState (`ButtonIdentifier` as string) as `AuthenticToolbarButtonState`

#### Return Value

The method returns `AuthenticToolbarButtonState`

#### Description

`Get/SetToolbarButtonState` queries the status of a toolbar button, and lets the user disable or enable the button, identified via its button identifier ([see list above](#)<sup>441</sup>). 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](#)<sup>606</sup>.

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.

### 15.3.2.6.16 Goto

**Method:** Goto (`eKind` as [SPYAuthenticElementKind](#)<sup>605</sup>, `nCount` as Long, `eFrom` as [SPYAuthenticDocumentPosition](#)<sup>605</sup>) as [AuthenticRange](#)<sup>407</sup>

#### 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
```

**15.3.2.6.17 IsRedoEnabled**

**Property:** IsRedoEnabled as Boolean (read-only)

**Description**

True if redo steps are available and [Redo](#)<sup>449</sup> is possible.

**Errors**

- 2000 The authentic view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

**15.3.2.6.18 IsUndoEnabled**

**Property:** IsUndoEnabled as Boolean (read-only)

**Description**

True if undo steps are available and [Undo](#)<sup>451</sup> is possible.

**Errors**

- 2000 The authentic view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

### 15.3.2.6.19 MarkupVisibility

**Property:** MarkupVisibility as [SPYAuthenticMarkupVisibility](#) <sup>606</sup>

#### 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.

### 15.3.2.6.20 Parent

**Property:** Parent as [Document](#) <sup>467</sup> (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.

### 15.3.2.6.21 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.

### 15.3.2.6.22 Redo

**Method:** Redo() as Boolean

#### Description

Redo the modification undone by the last undo command.

**Errors**

- 2000 The authentic view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

**15.3.2.6.23 Selection**

**Property:** Selection as [AuthenticRange](#) <sup>407</sup>

**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
```

**15.3.2.6.24 SetToolBarButtonState**

**Method:** SetToolBarButtonState (ButtonIdentifier as string, AuthenticToolBarButtonState state)

**Description**

Get/SetToolBarButtonState queries the status of a toolbar button, and lets the user disable or enable the button, identified via its button identifier ([see list above](#) <sup>441</sup>). 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](#) <sup>606</sup>.

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.

### 15.3.2.6.25 Undo

**Method:** Undo() as Boolean

#### Description

Undo the last modification of the document from within this view.

#### Errors

- 2000 The authentic view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

### 15.3.2.6.26 UpdateXMLInstanceEntities

**Method:** UpdateXMLInstanceEntities()

#### Description

Updates the internal representation of the declared entities, and refills the entry helper. In addition, the validator is reloaded, allowing the XML file to validate correctly. Please note that this may also cause schema files to be reloaded.

#### Errors

The method never returns an error.

#### Example

```
// -----  
// 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();  
    }  
}
```

### 15.3.2.6.27 WholeDocument

**Property:** WholeDocument as [AuthenticRange](#)<sup>407</sup> (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.

### 15.3.2.6.28 XMLDataRoot

**Property:** XMLDataRoot as [XMLData](#)<sup>593</sup> (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.

## 15.3.2.7 CodeGeneratorDlg

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

### Properties and Methods

Standard automation properties

[Application](#)<sup>453</sup>

[Parent](#)<sup>456</sup>

Programming language selection properties

[ProgrammingLanguage](#)<sup>456</sup>

[TemplateFileName](#)<sup>457</sup>

Settings for C++ code

[CPPSettings\\_DOMType](#)<sup>453</sup>

[CPPSettings\\_LibraryType](#)<sup>454</sup>

[CPPSettings\\_UseMFC](#)<sup>455</sup>

[CPPSettings\\_GenerateVC6ProjectFile](#)<sup>453</sup>

[CPPSettings\\_GenerateVSProjectFile](#)<sup>454</sup>

Settings for C# code

[CSharpSettings\\_ProjectType](#)<sup>455</sup>

Dialog handling for above code generation properties

[PropertySheetDialogAction](#) <sup>457</sup>

Output path selection properties

[OutputPath](#) <sup>455</sup>

[OutputPathDialogAction](#) <sup>456</sup>

Presentation of result

[OutputResultDialogAction](#) <sup>456</sup>

### Description

Use this object to configure the generation of program code for schema files. The method [GenerateProgramCode](#) <sup>483</sup> expects a CodeGeneratorDlg as parameter to configure code generation as well as the associated user interactions.

### 15.3.2.7.1 Application

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

**Property:** Application as [Application](#) <sup>382</sup> (read-only)

### Description

Access the Authentic Desktop application object.

### Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

### 15.3.2.7.2 CPPSettings\_DOMType

**Property:** CPPSettings\_DOMType as [SPYDOMType](#) <sup>607</sup>

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

### Description

Defines one of the settings that configure generation of C++ code.

### Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

### 15.3.2.7.3 CPPSettings\_GenerateVC6ProjectFile

**Property:** CPPSettings\_GenerateVC6ProjectFile as Boolean

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

**Description**

Defines one of the settings that configure generation of C++ code.

**Errors**

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

#### 15.3.2.7.4 CPPSettings\_GenerateGCCMakefile

**Property:** CPPSettings\_GenerateGCCMakefile as Boolean

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

**Description**

Creates makefiles to compile the generated code under Linux with GCC.

**Errors**

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

#### 15.3.2.7.5 CPPSettings\_GenerateVSProjectFile

**Property:** CSharpSettings\_GenerateVSProjectFile as [SPYProjectType](#)<sup>611</sup>

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

**Description**

Defines one of the settings that configure generation of C++ code. Only `spyVisualStudio2005Project (=4)` and `spyVisualStudio2008Project (=5)` and `spyVisualStudio2010Project (=6)` are valid project types.

**Errors**

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

#### 15.3.2.7.6 CPPSettings\_LibraryType

**Property:** CPPSettings\_LibraryType as [SPYLibType](#)<sup>609</sup>

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

**Description**

Defines one of the settings that configure generation of C++ code.

**Errors**

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

### 15.3.2.7.7 CPPSettings\_UseMFC

**Property:** CPPSettings\_UseMFC as Boolean

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

**Description**

Defines one of the settings that configure generation of C++ code.

**Errors**

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

### 15.3.2.7.8 CSharpSettings\_ProjectType

**Property:** CSharpSettings\_ProjectType as [SPYProjectType](#)<sup>611</sup>

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

**Description**

Defines the only setting to configure generation of C# code.

**Errors**

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

### 15.3.2.7.9 OutputPath

**Property:** OutputPath as String

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

**Description**

Selects the base directory for all generated code.

**Errors**

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

### 15.3.2.7.10 OutputPathDialogAction

**Property:** OutputPathDialogAction as [SPYDialogAction](#) <sup>607</sup>

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

#### Description

Defines how the sub-dialog for selecting the code generation output path gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog with the current value of the [OutputPath](#) <sup>455</sup> property as default. Use *spyDialogOK(0)* to hide the dialog from the user.

#### Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

### 15.3.2.7.11 OutputResultDialogAction

**Property:** OutputResultDialogAction as [SPYDialogAction](#) <sup>607</sup>

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

#### Description

Defines how the sub-dialog that asks to show the result of the code generation process gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog. Use *spyDialogOK(0)* to hide the dialog from the user.

#### Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

### 15.3.2.7.12 Parent

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

**Property:** Parent as [Dialogs](#) <sup>464</sup> (read-only)

#### Description

Access the parent of the object.

#### Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

### 15.3.2.7.13 ProgrammingLanguage

**Property:** ProgrammingLanguage as [ProgrammingLanguage](#) <sup>611</sup>

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

**Description**

Selects the output language for the code to be generated.

CAUTION: Setting this property to one of C++, C# or Java, changes the property [TemplateFileName](#)<sup>457</sup> to the appropriate template file delivered with Authentic Desktop as well. If you want to generate C++, C# or Java code based on your own templates, set first the programming language and then select your template file.

**Errors**

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

### 15.3.2.7.14 PropertySheetDialogAction

**Property:** PropertySheetDialogAction as [SPYDialogAction](#)<sup>607</sup>

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

**Description**

Defines how the sub-dialog that configures the code generation process gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog with the current values as defaults. Use *spyDialogOK(0)* to hide the dialog from the user.

**Errors**

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

### 15.3.2.7.15 TemplateFileName

**Property:** TemplateFileName as String

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

**Description**

Selects the code generation template file. Authentic Desktop comes with template files for C++, C# or Java in the SPL folder of your installation directory.

Setting this property to one of the code generation template files of your Authentic Desktop installation automatically sets the [ProgrammingLanguage](#)<sup>456</sup> property to its appropriate value.

**Errors**

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

## 15.3.2.8 DatabaseConnection

**Properties for import and export**

[File](#)<sup>460</sup> or

[ADOConnection](#) <sup>458</sup> or  
[ODBCConnection](#) <sup>462</sup>

### Properties for import only

[DatabaseKind](#) <sup>460</sup>  
[SQLSelect](#) <sup>463</sup>  
[AsAttributes](#) <sup>459</sup>  
[ExcludeKeys](#) <sup>460</sup>  
[IncludeEmptyElements](#) <sup>461</sup>  
[NumberDateTimeFormat](#) <sup>462</sup>  
[NullReplacement](#) <sup>461</sup>  
[CommentIncluded](#) <sup>459</sup>

### Properties for export only

[CreateMissingTables](#) <sup>459</sup>  
[CreateNew](#) <sup>459</sup>  
[TextFieldLen](#) <sup>463</sup>  
[DatabaseSchema](#) <sup>460</sup>

### Properties for XML Schema from DB Structure generation

[PrimaryKeys](#) <sup>462</sup>  
[ForeignKeys](#) <sup>461</sup>  
[UniqueKeys](#) <sup>463</sup>  
[SchemaExtensionType](#) <sup>462</sup>  
[SchemaFormat](#) <sup>463</sup>  
[ImportColumnsType](#) <sup>461</sup>

### Description

DatabaseConnection specifies the parameters for the database connection.

Please note that the properties of the DatabaseConnection interface are referring to the settings of the import and export dialogs of Authentic Desktop.

#### 15.3.2.8.1 ADOConnection

**Property:** ADOConnection as String

### Description

The property ADOConnection contains a connection string. Either use this property or [ODBCConnection](#) <sup>462</sup> or [File](#) <sup>460</sup> to refer to a database.

### Errors

No error codes are returned.

### Example

```
Dim objSpyConn As DatabaseConnection  
Set objSpyConn = objSpy.GetDatabaseSettings
```

```
Dim objADO As DataLinks  
Set objADO = CreateObject("DataLinks")
```

```
If Not (objADO Is Nothing) Then
    Dim objConn As Connection
    Set objConn = objADO.PromptNew
    objSpyConn.ADOConnection = objConn.ConnectionString
End If
```

### 15.3.2.8.2 AsAttributes

**Property:** AsAttributes as Boolean

#### Description

Set AsAttributes to true if you want to initialize all import fields to be imported as attributes. Default is false and will initialize all fields to be imported as elements. This property is used only in calls to [Application.GetDatabaseImportElementList](#)<sup>388</sup>.

#### Errors

No error codes are returned.

### 15.3.2.8.3 CommentIncluded

**Property:** CommentIncluded as Boolean

#### Description

This property tells whether additional comments are added to the generated XML. Default is true. This property is used only when importing from databases.

#### Errors

No error codes are returned.

### 15.3.2.8.4 CreateMissingTables

**Property:** CreateMissingTables as Boolean

#### Description

If CreateMissingTables is true, tables which are not already defined in the export database will be created during export. Default is true. This property is used only when exporting to databases.

#### Errors

No error codes are returned.

### 15.3.2.8.5 CreateNew

**Property:** CreateNew as Boolean

#### Description

Set CreateNew true if you want to create a new database on export. Any existing database will be overwritten. See also [DatabaseConnection.File](#)<sup>460</sup>. Default is false. This property is used only when exporting to databases.

**Errors**

No error codes are returned.

### 15.3.2.8.6 DatabaseKind

**Property:** DatabaseKind as [SPYDatabaseKind](#)<sup>606</sup>

**Description**

Select the kind of database that gets access. The default value is spyDB\_Unspecified(7) and is sufficient in most cases. This property is used only when importing from databases.

**Errors**

No error codes are returned.

### 15.3.2.8.7 DatabaseSchema

**Property:** DatabaseSchema as String

**Description**

This property specifies the Schema used for export in Schema aware databases. Default is "". This property is used only when exporting to databases.

**Errors**

No error codes are returned.

### 15.3.2.8.8 ExcludeKeys

**Property:** ExcludeKeys as Boolean

**Description**

Set ExcludeKeys to true if you want to exclude all key columns from the import data. Default is false. This property is used only when importing from databases.

**Errors**

No error codes are returned.

### 15.3.2.8.9 File

**Property:** File as String

**Description**

The property File sets the path for the database during export or import. This property can only be used in conjunction with a Microsoft Access database. Either use this property or [ODBCConnection](#)<sup>462</sup> or [ADOConnection](#)<sup>458</sup> to refer to the database.

**Errors**

No error codes are returned.

### 15.3.2.8.10 ForeignKeys

**Property:** ForeignKeys as Boolean

**Description**

Specifies whether the Foreign Keys constraint is created or not. Default is true. This property is used only when creating a XML Schema from a DB structure.

**Errors**

No error codes are returned.

### 15.3.2.8.11 ImportColumnsType

**Property:** ImportColumnsType as [SPYImportColumnsType](#)<sup>608</sup>

**Description**

Defines if column information from the DB is saved as element or attribute in the XML Schema. Default is as element. This property is used only when creating a XML Schema from a DB structure.

**Errors**

No error codes are returned.

### 15.3.2.8.12 IncludeEmptyElements

**Property:** IncludeEmptyElements as Boolean

**Description**

Set IncludeEmptyElements to false if you want to exclude all empty elements. Default is true. This property is used only when importing from databases.

**Errors**

No error codes are returned.

### 15.3.2.8.13 NullReplacement

**Property:** NullReplacement as String

**Description**

This property contains the text value that is used during import for empty elements (null values). Default is "". This property is used only when importing from databases.

**Errors**

No error codes are returned.

#### 15.3.2.8.14 NumberDateTimeFormat

**Property:** NumberDateTimeFormat as [SPYNumberDateTimeFormat](#) <sup>610</sup>

**Description**

The property NumberDateTimeFormat sets the format of numbers and date- and time-values. Default is [spySystemLocale](#) <sup>610</sup>. This property is used only when importing from databases.

**Errors**

No error codes are returned.

#### 15.3.2.8.15 ODBCConnection

**Property:** ODBCConnection as String

**Description**

The property ODBCConnection contains a ODBC connection string. Either use this property or [ADOConnection](#) <sup>458</sup> or [File](#) <sup>460</sup> to refer to a database.

**Errors**

No error codes are returned.

#### 15.3.2.8.16 PrimaryKeys

**Property:** PrimaryKeys as Boolean

**Description**

Specifies whether the Primary Keys constraint is created or not. Default is true. This property is used only when creating a XML Schema from a DB structure.

**Errors**

No error codes are returned.

#### 15.3.2.8.17 SchemaExtensionType

**Property:** SchemaExtensionType as [SPYSchemaExtensionType](#) <sup>614</sup>

**Description**

Defines the Schema extension type used during the Schema generation. This property is used only when creating a XML Schema from a DB structure.

**Errors**

No error codes are returned.

### 15.3.2.8.18 SchemaFormat

**Property:** SchemaFormat as [SPYSchemaFormat](#) <sup>614</sup>

#### Description

Defines the Schema format used during the Schema generation. This property is used only when creating a XML Schema from a DB structure.

#### Errors

No error codes are returned.

### 15.3.2.8.19 SQLSelect

**Property:** SQLSelect as String

#### Description

The SQL query for the import is stored in the property SQLSelect. This property is used only when importing from databases.

#### Errors

No error codes are returned.

### 15.3.2.8.20 TextFieldLen

**Property:** TextFieldLen as long

#### Description

The property TextFieldLen sets the length for created text fields during the export. Default is 255. This property is used only when exporting to databases.

#### Errors

No error codes are returned.

### 15.3.2.8.21 UniqueKeys

**Property:** UniqueKeys as Boolean

#### Description

Specifies whether the Unique Keys constraint is created or not. Default is true. This property is used only when creating a XML Schema from a DB structure.

#### Errors

No error codes are returned.

## 15.3.2.9 Dialogs

### Properties and Methods

Standard automation properties

[Application](#)<sup>464</sup>

[Parent](#)<sup>465</sup>

Various dialog objects

[CodeGeneratorDlg](#)<sup>464</sup>

[FileSelectionDlg](#)<sup>465</sup>

[SchemaDocumentationDlg](#)<sup>465</sup>

[GenerateSampleXMLDlg](#)<sup>465</sup>

[DTDSchemaGeneratorDlg](#)<sup>466</sup>

[FindInFilesDlg](#)<sup>466</sup>

### Description

The Dialogs object provides access to different built-in dialogs of Authentic Desktop. These dialog objects allow to initialize the fields of user dialogs before they get presented to the user or allow to simulate complete user input by your program.

#### 15.3.2.9.1 Application

**Property:** Application as [Application](#)<sup>382</sup> (read-only)

### Description

Access the Authentic Desktop application object.

### Errors

- 2300 The object is no longer valid.
- 2301 Invalid address for the return parameter was specified.

#### 15.3.2.9.2 CodeGeneratorDlg

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

**Property:** CodeGeneratorDlg as [CodeGeneratorDlg](#)<sup>452</sup> (read-only)

### Description

Get a new instance of a code generation dialog object. You will need this object to pass the necessary parameters to the code generation methods. Initial values are taken from last usage of the code generation dialog.

### Errors

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

### 15.3.2.9.3 FileSelectionDlg

**Property:** FileSelectionDlg as [FileSelectionDlg](#)<sup>510</sup> (read-only)

#### Description

Get a new instance of a file selection dialog object.

File selection dialog objects are passed to you with the some events that signal opening or saving of documents and projects.

#### Errors

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

### 15.3.2.9.4 Parent

**Property:** Parent as [Application](#)<sup>382</sup> (read-only)

#### Description

Access the Authentic Desktop application object.

#### Errors

- 2300 The object is no longer valid.
- 2301 Invalid address for the return parameter was specified.

### 15.3.2.9.5 SchemaDocumentationDlg

**Property:** SchemaDocumentationDlg as [SchemaDocumentationDlg](#)<sup>552</sup> (read-only)

#### Description

Get a new instance of a dialog object that parameterizes generation of schema documentation. See [Document.GenerateSchemaDocumentation](#)<sup>483</sup> for its usage.

#### Errors

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

### 15.3.2.9.6 GenerateSampleXMLDlg

**Property:** GenerateSampleXMLDlg as [GenerateSampleXMLDlg](#)<sup>523</sup> (read-only)

#### Description

Get a new instance of a dialog object that parameterizes generation of a sample XML based on a W3C schema or DTD. See [GenerateSampleXML](#)<sup>483</sup> for its usage.

#### Errors

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

### 15.3.2.9.7 DTDSchemaGeneratorDlg

**Property:** DTDSchemaGeneratorDlg as [DTDSchemaGeneratorDlg](#)<sup>501</sup> (read-only)

#### Description

Get a new instance of a dialog object that parameterizes generation of a schema or DTD. See [Document.GenerateDTDOrSchemaEx](#)<sup>482</sup> for its usage.

#### Errors

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

### 15.3.2.9.8 FindInFilesDlg

**Property:** FindInFilesDlg as [FindInFilesDlg](#)<sup>511</sup> (read-only)

#### Description

Get a new instance of a dialog object that parameterizes the search (or replacement) of strings in files. See [Application.FindInFiles](#)<sup>388</sup> for its usage.

#### Errors

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

### 15.3.2.9.9 WSDLDocumentationDlg

**Property:** WSDLDocumentationDlg as [WSDLDocumentationDlg](#)<sup>559</sup> (read-only)

#### Description

Get a new instance of a dialog object that parameterizes generation of WSDL documentation. See [Document.GenerateWSDLDocumentation](#)<sup>484</sup> for its usage.

#### Errors

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

### 15.3.2.9.10 WSDL20DocumentationDlg

**Property:** WSDL20DocumentationDlg as [WSDL20DocumentationDlg](#)<sup>571</sup> (read-only)

#### Description

Get a new instance of a dialog object that parameterizes generation of WSDL 2.0 documentation. See [Document.GenerateWSDL20Documentation](#)<sup>484</sup> for its usage.

**Errors**

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

### 15.3.2.9.11 XBRLDocumentationDlg

**Property:** XBRLDocumentationDlg as [XBRLDocumentationDlg](#)<sup>582</sup> (read-only)

**Description**

Get a new instance of a dialog object that parameterizes generation of XBRL documentation. See [Document.GenerateXBRLDocumentation](#)<sup>484</sup> for its usage.

**Errors**

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

## 15.3.2.10 Document

**Properties and Methods**

Standard automation properties

[Application](#)<sup>472</sup>  
[Parent](#)<sup>490</sup>

Various document properties and methods

[SetActiveDocument](#)<sup>492</sup>  
[Encoding](#)<sup>478</sup>  
[SetEncoding \(obsolete\)](#)<sup>492</sup>  
[Suggestions](#)<sup>494</sup>

XML validation

[IsValid](#)<sup>487</sup>  
[SetExternalsValid](#)<sup>493</sup>

Document conversion and transformation

[AssignDTD](#)<sup>472</sup>  
[AssignSchema](#)<sup>472</sup>  
[AssignXSL](#)<sup>472</sup>  
[AssignXSLFO](#)<sup>473</sup>  
[ConvertDTDOrSchema](#)<sup>474</sup>  
[ConvertDTDOrSchemaEx](#)<sup>475</sup>  
[GenerateDTDOrSchema](#)<sup>482</sup>  
[GenerateDTDOrSchemaEx](#)<sup>482</sup>  
[FlattenDTDOrSchema](#)<sup>481</sup>  
[CreateSchemaDiagram](#)<sup>477</sup>  
[ExecuteXQuery](#)<sup>479</sup>  
[TransformXSL](#)<sup>495</sup>  
[TransformXSLEx](#)<sup>495</sup>

[TransformXSLFO](#) <sup>495</sup>  
[GenerateProgramCode](#) <sup>483</sup> (Enterprise Edition only)  
[GenerateSchemaDocumentation](#) <sup>483</sup>  
[GenerateSampleXML](#) <sup>483</sup>  
[ConvertToWSDL20](#) <sup>475</sup>

#### Document export

[GetExportElementList](#) <sup>485</sup>  
[ExportToText](#) <sup>480</sup>  
[ExportToDatabase](#) <sup>479</sup>  
[CreateDBStructureFromXMLSchema](#) <sup>476</sup>  
[GetDBStructureList](#) <sup>485</sup>

#### File saving and naming

[FullName](#) <sup>481</sup>  
[Name](#) <sup>489</sup>  
[Path](#) <sup>490</sup>  
[GetPathName \(obsolete\)](#) <sup>486</sup>  
[SetPathName \(obsolete\)](#) <sup>493</sup>  
[Title](#) <sup>495</sup>  
[IsModified](#) <sup>486</sup>  
[Saved](#) <sup>491</sup>  
[SaveAs](#) <sup>491</sup>  
[Save](#) <sup>490</sup>  
[SaveInString](#) <sup>491</sup>  
[SaveToURL](#) <sup>492</sup>  
[Close](#) <sup>474</sup>

#### View access

[CurrentViewMode](#) <sup>477</sup>  
[SwitchViewMode](#) <sup>494</sup>  
[AuthenticView](#) <sup>473</sup>  
[GridView](#) <sup>486</sup>  
[DocEditView \(obsolete\)](#) <sup>477</sup>

#### Access to XMLData

[RootElement](#) <sup>490</sup>  
[DataRoot](#) <sup>477</sup>  
[CreateChild](#) <sup>476</sup>  
[UpdateViews](#) <sup>496</sup>  
[StartChanges](#) <sup>494</sup>  
[EndChanges](#) <sup>478</sup>  
[UpdateXMLData](#) <sup>496</sup>

#### Description

Document objects represent XML documents opened in Authentic Desktop.

Use one of the following properties to access documents that are already open in Authentic Desktop:

[Application.ActiveDocument](#) <sup>385</sup>  
[Application.Documents](#) <sup>388</sup>

Use one of the following methods to open a new document in Authentic Desktop:

[Documents.OpenFile](#) <sup>500</sup>

[Documents.OpenURL](#) <sup>500</sup>  
[Documents.OpenURLDialog](#) <sup>501</sup>  
[Documents.NewFile](#) <sup>498</sup>  
[Documents.NewFileFromText](#) <sup>499</sup>  
[SpyProjectItem.Open](#) <sup>548</sup>  
[Application.ImportFromDatabase](#) <sup>392</sup>  
[Application.ImportFromSchema](#) <sup>393</sup>  
[Application.ImportFromText](#) <sup>394</sup>  
[Application.ImportFromWord](#) <sup>395</sup>  
[Document.ConvertDTDOrSchema](#) <sup>474</sup>  
[Document.GenerateDTDOrSchema](#) <sup>482</sup>

### 15.3.2.10.1 Events

#### 15.3.2.10.1.1 OnBeforeSaveDocument

**Event:** OnBeforeSaveDocument(*objDocument* as [Document](#) <sup>467</sup>, *objDialog* as [FileSelectionDlg](#) <sup>510</sup>)

##### **XMLSpy scripting environment - VBScript:**

```
Function On_BeforeSaveDocument(objDocument, objDialog)
End Function
```

```
' old handler - now obsolete
' return string to save to new file name
' return empty string to cancel save operation
' return nothing to save to original name
Function On_SaveDocument(objDocument, strFilePath)
End Function
```

##### **XMLSpy scripting environment - JScript:**

```
function On_BeforeSaveDocument(objDocument, objDialog)
{
}
```

```
// old handler - now obsolete
// return string to save to new file name
// return empty string to cancel save operation
// return nothing to save to original name
function On_SaveDocument(objDocument, strFilePath)
{
}
```

##### **XMLSpy IDE Plugin:**

```
IXMLSpyPlugin.OnEvent (27, ...) // nEventId = 27
```

##### **Description**

This event gets fired on any attempt to save a document. The file selection dialog object is initialized with the name chosen for the document file. You can modify this selection. To continue saving the document leave the

[FileSelectionDlg.DialogAction](#)<sup>511</sup> property of *io\_objDialog* at its default value [spyDialogOK](#)<sup>607</sup>. To abort saving of the document set this property to [spyDialogCancel](#)<sup>607</sup>.

### 15.3.2.10.1.2 OnBeforeCloseDocument

**Event:** OnBeforeCloseDocument(*objDocument* as [Document](#)<sup>467</sup>) as Boolean

#### **XMLSpy scripting environment - VBScript:**

```
Function On_BeforeCloseDocument(objDocument)
    ' On_BeforeCloseDocument = False ' to prohibit closing of document
End Function
```

#### **XMLSpy scripting environment - JScript:**

```
function On_BeforeCloseDocument(objDocument)
{
    // return false; /* to prohibit closing of document */
}
```

#### **XMLSpy IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (28, ...) // nEventId = 28
```

#### **Description**

This event gets fired on any attempt to close a document. To prevent the document from being closed return false.

### 15.3.2.10.1.3 OnBeforeValidate

**Event:** OnBeforeValidate(*objDocument* as [Document](#)<sup>467</sup>, *bOnLoading* as Boolean, *bOnCommand* as Boolean) as Boolean

#### **XMLSpy scripting environment - VBScript:**

```
Function On_BeforeValidate(objDocument, bOnLoading, bOnCommand)
    On_BeforeValidate = bCancelDefaultValidation 'set by the script if necessary
End Function
```

#### **XMLSpy scripting environment - JScript:**

```
function On_BeforeValidate(objDocument, bOnLoading, bOnCommand)
{
    return bCancelDefaultValidation //set by the script if necessary
}
```

#### **XMLSpy IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (32, ...) // nEventId = 32
```

#### **Description**

This event gets fired before the document is validated. It is possible to suppress the default validation by returning false from the event handler. In this case the script should also set the validation result using the [SetExternallsValid](#)<sup>493</sup> method.

bOnLoading is true if the event is raised on the initial validation on loading the document.

bOnCommand is true whenever the user selected the Validate command from the Toolbar or menu.

Available with TypeLibrary version 1.5

#### 15.3.2.10.1.4 OnCloseDocument

**Event:** OnCloseDocument(*objDocument* as [Document](#)<sup>467</sup>)

**XMLSpy scripting environment - VBScript:**

```
Function On_Close Document(objDocument)  
End Function
```

**XMLSpy scripting environment - JScript:**

```
function On_Close Document(objDocument)  
{  
}  
}
```

**XMLSpy IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (8, ...) // nEventId = 8
```

**Description**

This event gets fired as a result of closing a document. Do not modify the document from within this event.

#### 15.3.2.10.1.5 OnViewActivation

**Event:** OnViewActivation(*objDocument* as [Document](#)<sup>467</sup>, *eViewMode* as [SPYViewModes](#)<sup>616</sup>, *bActivated* as Boolean)

**XMLSpy scripting environment - VBScript:**

```
Function On_ViewActivation(objDocument, eViewMode, bActivated)  
End Function
```

**XMLSpy scripting environment - JScript:**

```
function On_ViewActivation(objDocument, eViewMode, bActivated)  
{  
}  
}
```

**XMLSpy IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (29, ...) // nEventId = 29
```

**Description**

This event gets fired whenever a view of a document becomes visible (i.e. becomes the active view) or invisible (i.e. another view becomes the active view or the document gets closed). However, the first view activation event after a document gets opened cannot be received, since there is no document object to get the event from. Use the [Application.OnDocumentOpened](#)<sup>384</sup> event instead.

### 15.3.2.10.2 Application

**Property:** Application as [Application](#)<sup>382</sup> (read-only)

#### Description

Accesses the Authentic Desktop application object.

#### Errors

- 1400 The object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

### 15.3.2.10.3 AssignDTD

**Method:** AssignDTD(*strDTDFile* as String, *bDialog* as Boolean)

#### Description

The method places a reference to the DTD file "strDTDFile" into the document. Note that no error occurs if the file does not exist, or is not accessible. If *bDialog* is true Authentic Desktop presents a dialog to set the file.

#### Errors

- 1400 The object is no longer valid.
- 1409 You are not allowed to assign a DTD to the document.

### 15.3.2.10.4 AssignSchema

**Method:** AssignSchema (*strSchemaFile* as String, *bDialog* as Boolean)

#### Description

The method places a reference to the schema file "strSchemaFile" into the document. Note that no error occurs if the file does not exist or is not accessible. If *bDialog* is true Authentic Desktop presents a dialog to set the file.

#### Errors

- 1400 The object is no longer valid.
- 1409 You are not allowed to assign a schema file to the document.

### 15.3.2.10.5 AssignXSL

**Method:** AssignXSL (*strXSLFile* as String, *bDialog* as Boolean)

#### Description

The method places a reference to the XSL file "strXSLFile" into the document. Note that no error occurs if the file does not exist or is not accessible. If *bDialog* is true Authentic Desktop presents a dialog to set the file.

#### Errors

- 1400 The object is no longer valid.
- 1409 You are not allowed to assign an XSL file to the document.

### 15.3.2.10.6 AssignXSLFO

**Method:** AssignXSLFO (*strXSLFOFile* as String, *bDialog* as Boolean)

#### Description

The method places a reference to the XSLFO file "strXSLFile" into the document. Note that no error occurs if the file does not exist or is not accessible. If *bDialog* is true Authentic Desktop presents a dialog to set the file.

#### Errors

- 1400 The object is no longer valid.
- 1409 You are not allowed to assign an XSL file to the document.

### 15.3.2.10.7 AsXMLString

**Property:** AsXMLString as String

#### Description

This property can be used to get or set the document content.

#### Errors

- 1400 The document object is no longer valid.
- 1404 Cannot create XMLData object.
- 1407 View mode cannot be switched.

### 15.3.2.10.8 AuthenticView

**Method:** AuthenticView as [AuthenticView](#)<sup>435</sup> (read-only)

#### Description

Returns an object that gives access to properties and methods specific to Authentic view. The object returned is only valid if the current document is opened in Authentic view mode. The lifetime of an object ends with the next view switch. Any attempt to access objects or any of its children afterwards will result in an error indicating that the object is invalid.

#### Errors

- 1400 The object is no longer valid.
- 1417 Document needs to be open in authentic view mode.

#### Examples

```
' _____
' XMLSpy scripting environment - VBScript
' secure access to authentic view object
' _____

Dim objDocument
Set objDocument = Application.ActiveDocument
If (Not objDocument Is Nothing) Then
    ' we have an active document, now check for view mode
    If (objDocument.CurrentViewMode <> spyViewAuthentic) Then
```

```

        If (Not objDocument.SwitchViewMode (spyViewAuthentic)) Then
            MsgBox "Active document does not support authentic view mode"
        Else
            ' now it is safe to access the authentic view object
            Dim objAuthenticView
            Set objAuthenticView = objDocument.AuthenticView
            ' now use the authentic view object

        End If
    End If
Else
    MsgBox "No document is open"
End If

```

### 15.3.2.10.9 Close

**Method:** Close (*bDiscardChanges* as Boolean)

#### Description

To close the document call this method. If *bDiscardChanges* is true and the document is modified, the document will be closed but not saved.

#### Errors

- 1400 The object is no longer valid.
- 1401 Document needs to be saved first.

### 15.3.2.10.10 ConvertDTDOrSchema

**Method:** ConvertDTDOrSchema (*nFormat* as [SPYDTDSchemaFormat](#)<sup>607</sup>, *nFrequentElements* as [SPYFrequentElements](#)<sup>608</sup>)

#### Parameters

*nFormat*

Sets the schema output format to DTD or W3C.

*nFrequentElements*

Create complex elements as elements or complex types.

#### Description

ConvertDTDOrSchema takes an existing schema format and converts it into a different format.

For a finer tuning of DTD/XSD conversion, use [ConvertDTDOrSchemaEx](#)<sup>475</sup>.

#### Errors

- 1400 The object is no longer valid.
- 1412 Error during conversion. In the case of DTD to DTD or XSD to XSD conversion, the following errors are returned: *DTD to DTD conversion is not supported. Please use function FlattenDTDOrSchema instead and*

*Schema to schema conversion is not supported. Please use function FlattenDTDOOrSchema instead.*

### 15.3.2.10.11 ConvertDTDOOrSchemaEx

**Method:** ConvertDTDOOrSchemaEx (*nFormat* as [SPYDTDSchemaFormat](#)<sup>607</sup>, *nFrequentElements* as [SPYFrequentElements](#)<sup>608</sup>, *sOutputPath* as String, *nOutputPathDialogAction* as [SPYDialogAction](#)<sup>607</sup>)

#### Parameters

*nFormat*

Sets the schema output format to DTD, or W3C.

*nFrequentElements*

Create complex elements as elements or complex types.

*sOutputPath*

The file path for the newly generated file.

*nOutputPathDialogAction*

Defines the dialog interaction for this call.

#### Description

`ConvertDTDOOrSchemaEx` takes an existing schema format and converts it into a different format.

#### Errors

1400 The object is no longer valid.

1412 Error during conversion. In the case of DTD to DTD or XSD to XSD conversion, the following errors are returned: *DTD to DTD conversion is not supported. Please use function FlattenDTDOOrSchema instead* and *Schema to schema conversion is not supported. Please use function FlattenDTDOOrSchema instead*.

### 15.3.2.10.12 ConvertToWSDL20

**Method:** ConvertToWSDL20 (*sFilePath* as String, *bShowDialogs* as Boolean)

#### Parameters

*sFilePath*

This specifies the file name of the converted WSDL. In case the source WSDL includes files which also must be converted, then only the directory part of the given path is used and the file names are generated automatically.

*bShowDialogs*

Defines whether file/folder selection dialogs are shown.

**Description**

Converts the WSDL 1.1 document to a WSDL 2.0 file. It will also convert any referenced WSDL files that are referenced from within this document. Note that this functionality is limited to WSDL View only. See [Document.CurrentViewMode](#)<sup>477</sup> and [SPYViewModes](#)<sup>616</sup>.

**Errors**

- 1400 The document object is no longer valid.
- 1407 Invalid parameters have been passed or an empty file name has been specified as output target.
- 1417 The document is not opened in WSDL view, maybe it is not an '.wsdl' file.
- 1421 Feature is not available in this edition.
- 1433 WSDL 1.1 to WSDL 2.0 conversion failed.

### 15.3.2.10.13 CreateChild

**Method:** CreateChild (*nKind* as [SPYXMLDataKind](#)<sup>617</sup>) as [XMLData](#)<sup>593</sup>

**Return Value**

The method returns the new XMLData object.

**Description**

To create a new XMLData object use the CreateChild() method.

**Errors**

- 1400 The object is no longer valid.
- 1404 Cannot create XMLData object.
- 1407 Invalid address for the return parameter was specified.

### 15.3.2.10.14 CreateDBStructureFromXMLSchema

**Method:** CreateDBStructureFromXMLSchema (*pDatabase* as [DatabaseConnection](#)<sup>457</sup>, *pTables* as [ElementList](#)<sup>505</sup>, *bDropTableWithExistingName* as Boolean) as String

**Description**

CreateDBStructureFromXMLSchema exports the given tables to the specified database. The function returns the SQL statements that were necessary to perform the changes.

See also [GetDBStructureList](#)<sup>485</sup>.

**Errors**

- 1429 Database selection missing.
- 1430 Document export failed.

### 15.3.2.10.15 CreateSchemaDiagram

**Method:** CreateSchemaDiagram (*nKind* as [SPYSchemaDefKind](#)<sup>613</sup>, *strName* as String, *strFile* as String)

#### Return Value

None.

#### Description

The method creates a diagram of the schema type *strName* of kind *nKind* and saves the output file into *strFile*. Note that this functionality is limited to Schema View only. See [Document.CurrentViewMode](#)<sup>477</sup> and [SPYViewModes](#)<sup>616</sup>.

#### Errors

- 1400 The object is no longer valid.
- 1414 Failed to save diagram.
- 1415 Invalid schema definition type specified.

### 15.3.2.10.16 CurrentViewMode

**Method:** CurrentViewMode as [SPYViewModes](#)<sup>616</sup>

#### Description

The property holds the current view mode of the document. See also [Document.SwitchViewMode](#)<sup>494</sup>.

#### Errors

- 1400 The object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

### 15.3.2.10.17 DataRoot

**Property:** DataRoot as [XMLData](#)<sup>593</sup> (read-only)

#### Description

This property provides access to the document's first XMLData object of type *spyXMLDataElement*. This is typically the root element for all document content data. See [XMLSpyDocument.RootElement](#)<sup>490</sup> to get the root element of the whole document including XML prolog data. If the [CurrentViewMode](#)<sup>477</sup> is not *spyViewGrid* or *spyViewAuthentic* an [UpdateXMLData](#)<sup>496</sup> may be necessary to get access to the latest [XMLData](#)<sup>593</sup>.

#### Errors

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

### 15.3.2.10.18 DocEditView

**Method:** DocEditView as DocEditView

#### Description

Holds a reference to the current Authentic View object.

**Errors**

- 1400 The object is no longer valid.
- 1407 Invalid address for the return parameter was specified.
- 1417 Document needs to be open in authentic view mode.

### 15.3.2.10.19 Encoding

**Property:** Encoding as String

**Description**

This property provides access to the document's encoding value. However, this property can only be accessed when the document is opened in *spyViewGrid*, *spyViewText* or *spyViewAuthentic*. See [CurrentViewMode](#)<sup>477</sup> on how to detect a document's actual view mode.

This property makes the method [SetEncoding](#)<sup>492</sup> obsolete.

Possible values are, for example:

- 8859-1,
- 8859-2,
- ASCII, ISO-646,
- 850,
- 1252,
- 1255,
- SHIFT-JIS, MS-KANJI,
- BIG5, FIVE,
- UTF-7,
- UTF-8,
- UTF-16

**Errors**

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.
- 1416 Operation not supported in current view mode.

### 15.3.2.10.20 EndChanges

**Method:** EndChanges()

**Description**

Use the method EndChanges to display all changes since the call to [Document.StartChanges](#)<sup>494</sup>.

**Errors**

- 1400 The object is no longer valid.

### 15.3.2.10.21 ExecuteXQuery

**Method:** ExecuteXQuery (*strXMLFileName* as String)

#### Description

Execute the XQuery statements contained in the document of the document object. Either an XQuery execution or an XQuery Update is performed depending on the file extension of the document. Use the XML file specified in the argument as the XML target document that the XQuery document processes.

- If the document has an XQuery file extension as defined in the Options dialog of Authentic Desktop, then an XQuery execution is performed. By default: `.xq`, `.xql`, and `.xquery` are set as XQuery file extensions in Authentic Desktop.
- If the document has an XQuery Update file extension as defined in the Options dialog of Authentic Desktop, then an XQuery Update action is performed. By default: `.xqu` is set as an XQuery Update file extension in Authentic Desktop.

If your XQuery script does not use an XML source, set the parameter `strXMLFileName` to an empty string.

#### Errors

- 1400 The document object is no longer valid.
- 1423 XQuery transformation error.
- 1424 Not all files required for operation could be loaded. Most likely, the file specified in `strXMLFileName` does not exist or is not valid.

### 15.3.2.10.22 ExportToDatabase

**Method:** ExportToDatabase (*pFromChild* as [XMLData](#)<sup>593</sup>, *pExportSettings* as [ExportSettings](#)<sup>507</sup>, *pDatabase* as [DatabaseConnection](#)<sup>457</sup>)

#### Description

ExportToDatabase exports the XML document starting with the element `pFromChild`. The parameter `pExportSettings` defines the behaviour of the export (see [Application.GetExportSettings](#)<sup>390</sup>). The parameter `pDatabase` specifies the destination of the export (see [Application.GetDatabaseSettings](#)<sup>389</sup>). [UpdateXMLData\(\)](#)<sup>496</sup> might be indirectly needed as you have to pass the [XMLData](#)<sup>593</sup> as parameter to this function.

#### Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.
- 1416 Error during export.
- 1429 Database selection missing.
- 1430 Document export failed.

#### Example

```
Dim objDoc As Document
```

```

Set objDoc = objSpy.ActiveDocument

'set the behaviour of the export with ExportSettings
Dim objExpSettings As ExportSettings
Set objExpSettings = objSpy.GetExportSettings

'set the destination with DatabaseConnection
Dim objDB As DatabaseConnection
Set objDB = objSpy.GetDatabaseSettings

objDB.CreateMissingTables = True
objDB.CreateNew = True
objDB.File = "C:\Export.mdb"

objDoc.ExportToDatabase objDoc.RootElement, objExpSettings, objDB
If Err.Number <> 0 Then
    a = MsgBox("Error: " & (Err.Number - vbObjectError) & Chr(13) &
        "Description: " & Err.Description)
End If

```

### 15.3.2.10.23 ExportToText

**Method:** ExportToText (pFromChild as [XMLData](#)<sup>593</sup>, pExportSettings as [ExportSettings](#)<sup>507</sup>, pTextSettings as [TextImportExportSettings](#)<sup>551</sup>)

#### Description

ExportToText exports tabular information from the document starting at pFromChild into one or many text files. Columns of the resulting tables are generated in alphabetical order of the column header names. Use [GetExportElementList](#)<sup>485</sup> to learn about the data that will be exported. The parameter pExportSettings defines the specifics for the export. Set the property [ExportSettings.ElementList](#)<sup>508</sup> to the - possibly modified - list returned by [GetExportElementList](#)<sup>485</sup> to avoid exporting all contained tables. The parameter pTextSettings defines the options specific to text export and import. You need to set the property [TextImportExportSettings.DestinationFolder](#)<sup>552</sup> before you call ExportToText. [UpdateXMLData\(\)](#)<sup>496</sup> might be indirectly needed as you have to pass the [XMLData](#)<sup>593</sup> as parameter to this function.

#### Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.
- 1416 Error during export.
- 1430 Document export failed.

#### Example

```

'-----
' VBA client code fragment - export document to text files
'-----

Dim objDoc As Document
Set objDoc = objSpy.ActiveDocument

Dim objExpSettings As ExportSettings
Set objExpSettings = objSpy.GetExportSettings

```

```
objExpSettings.ElementList = objDoc.GetExportElementList(  
                                                                    objDoc.RootElement,  
                                                                    objExpSettings)  
  
Dim objTextExp As TextImportExportSettings  
Set objTextExp = objSpy.GetTextImportExportSettings  
objTextExp.HeaderRow = True  
objTextExp.DestinationFolder = "C:\Exports"  
  
On Error Resume Next  
objDoc.ExportToText objDoc.RootElement, objExpSettings, objTextExp  
  
If Err.Number <> 0 Then  
    a = MsgBox("Error: " & (Err.Number - vbObjectError) & Chr(13) & "Description: "  
              & Err.Description)  
End If
```

#### 15.3.2.10.24 FlattenDTDOrSchema

**Method:** FlattenDTDOrSchema (sOutputPath as String, nOutputPathDialogAction as [SPYDialogAction](#)<sup>607</sup>)

##### Parameters

sOutputPath

The file path for the newly generated file.

nOutputPathDialogAction

Defines the dialog interaction for this call.

##### Description

FlattenDTDOrSchema takes an existing DTD or schema, generates a flattened file, and saves the generated file at the specified location. In the case of DTDs, flattening removes parameter entities and produces a single DTD from a collection of modules; sections marked `IGNORE` are suppressed and unused parameter entities are deleted. When an XML Schema is flattened, (i) the components of all included schemas are added as global components of the active schema, and (ii) included schemas are deleted.

##### Errors

- 1400 The object is no longer valid.
- 1412 Error during conversion.

#### 15.3.2.10.25 FullName

**Property:** FullName as String

##### Description

This property can be used to get or set the full file name - including the path - to where the document gets saved. The validity of the name is not verified before the next save operation.

This property makes the methods [GetPathName](#)<sup>486</sup> and [SetPathName](#)<sup>493</sup> obsolete.

#### Errors

- 1400 The document object is no longer valid.
- 1402 Empty string has been specified as full file name.

### 15.3.2.10.26 GenerateDTDOrSchema

**Method:** GenerateDTDOrSchema (*nFormat* as [SPYDTDSchemaFormat](#)<sup>607</sup>, *nValuesList* as integer, *nDetection* as [SPYTypeDetection](#)<sup>615</sup>, *nFrequentElements* as [SPYFrequentElements](#)<sup>608</sup>)

#### Parameters

*nFormat*

Sets the schema output format to DTD, or W3C.

*nValuesList*

Generate not more than this amount of enumeration-facets per type. Set to -1 for unlimited.

*nDetection*

Specifies granularity of simple type detection.

*nFrequentElements*

Shall the types for all elements be defined as global? Use the value *spyGlobalComplexType* to define them on global scope. Otherwise, use the value *spyGlobalElements*.

#### Description

Use this method to automatically generate a DTD or schema for the current XML document.

For a finer tuning of DTD / schema generation, use [GenerateDTDOrSchemaEx](#)<sup>482</sup>.

Note that this functionality is not available in ZIP View only. See [Document.CurrentViewMode](#)<sup>477</sup> and [SPYViewModes](#)<sup>616</sup>.

#### Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.

### 15.3.2.10.27 GenerateDTDOrSchemaEx

**Method:** GenerateDTDOrSchemaEx (*objDlg* as [DTDSchemaGeneratorDlg](#)<sup>501</sup>) as [Document](#)<sup>467</sup>

#### Description

Use this method to automatically generate a DTD or schema for the current XML document. A

[DTDSchemaGeneratorDlg](#)<sup>501</sup> object is used to pass information to the schema/DTD generator. The generation process can be configured to allow user interaction or run without further user input.

Note that this functionality is not available in ZIP View only. See [Document.CurrentViewMode](#)<sup>477</sup> and [SPYViewModes](#)<sup>616</sup>.

**Errors**

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.

### 15.3.2.10.28 GenerateProgramCode

**Method:** GenerateProgramCode (*objDlg* as [CodeGeneratorDlg](#)<sup>452</sup>)

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

**Description**

Generate Java, C++ or C# class files from the XML Schema definitions in your document. A [CodeGeneratorDlg](#)<sup>452</sup> object is used to pass information to the code generator. The generation process can be configured to allow user interaction or run without further user input.

**Errors**

- 1400 The document object is no longer valid.
- 1407 An empty file name has been specified.
- 1421 Feature not available in this edition

### 15.3.2.10.29 GenerateSampleXML

**Method:** GenerateSampleXML (*objDlg* as [GenerateSampleXMLDlg](#)<sup>523</sup>) as [Document](#)<sup>467</sup>

**Description**

Generates a sample XML if the document is a schema or DTD. Use [Dialogs.GenerateSampleXMLDlg](#)<sup>465</sup> to get an initialized set of options.

Available with TypeLibrary version 1.5

**Errors**

- 1400 The document object is no longer valid.

### 15.3.2.10.30 GenerateSchemaDocumentation

**Method:** GenerateSchemaDocumentation (*objDlg* as [SchemaDocumentationDlg](#)<sup>532</sup>)

**Description**

Generate documentation for a schema definition file in HTML, MS-Word, or RTF format. The parameter *objDlg* is used to parameterize the generation process. Use [Dialogs.SchemaDocumentationDlg](#)<sup>465</sup> to get an initialized set of options. As a minimum, you will need to set the property [SchemaDocumentationDlg.OutputFile](#)<sup>539</sup> before starting the generation process. Note that this functionality is limited to Schema View only. See [Document.CurrentViewMode](#)<sup>477</sup> and [SPYViewModes](#)<sup>616</sup>.

**Errors**

- 1400 The document object is no longer valid.

- 1407 Invalid parameters have been passed or an empty file name has been specified as output target.
- 1417 The document is not opened in schema view, maybe it is not an '.xsd' file.
- 1421 Feature is not available in this edition.
- 1422 Error during generation

### 15.3.2.10.31 GenerateWSDL20Documentation

**Method:** GenerateWSDL20Documentation (*objDlg* as [WSDL20DocumentationDlg](#)<sup>571</sup>)

#### Description

Generate documentation for a WSDL definition file in HTML, MS-Word, or RTF format. The parameter *objDlg* is used to parameterize the generation process. Use [Dialogs.WSDL20DocumentationDlg](#)<sup>466</sup> to get an initialized set of options. As a minimum, you will need to set the property [WSDL20DocumentationDlg.OutputFile](#)<sup>576</sup> before starting the generation process. Note that this functionality is limited to WSDL View only. See [Document.CurrentViewMode](#)<sup>477</sup> and [SPYViewModes](#)<sup>616</sup>.

#### Errors

- 1400 The document object is no longer valid.
- 1407 Invalid parameters have been passed or an empty file name has been specified as output target.
- 1417 The document is not opened in schema view, maybe it is not an '.xsd' file.
- 1421 Feature is not available in this edition.
- 1422 Error during generation

### 15.3.2.10.32 GenerateWSDLDocumentation

**Method:** GenerateWSDLDocumentation (*objDlg* as [WSDLDocumentationDlg](#)<sup>559</sup>)

#### Description

Generate documentation for a WSDL definition file in HTML, MS-Word, or RTF format. The parameter *objDlg* is used to parameterize the generation process. Use [Dialogs.WSDLDocumentationDlg](#)<sup>466</sup> to get an initialized set of options. As a minimum, you will need to set the property [WSDLDocumentationDlg.OutputFile](#)<sup>565</sup> before starting the generation process. Note that this functionality is limited to WSDL View only. See [Document.CurrentViewMode](#)<sup>477</sup> and [SPYViewModes](#)<sup>616</sup>.

#### Errors

- 1400 The document object is no longer valid.
- 1407 Invalid parameters have been passed or an empty file name has been specified as output target.
- 1417 The document is not opened in schema view, maybe it is not an '.xsd' file.
- 1421 Feature is not available in this edition.
- 1422 Error during generation

### 15.3.2.10.33 GenerateXBRLDocumentation

**Method:** GenerateXBRLDocumentation (*objDlg* as [XBRLDocumentationDlg](#)<sup>582</sup>)

#### Description

Generate documentation for an XBRL file in HTML, MS-Word, or RTF format. The parameter *objDlg* is used to parameterize the generation process. Use [Dialogs.XBRLDocumentationDlg](#)<sup>467</sup> to get an initialized set of options. As a minimum, you will need to set the property [XBRLDocumentationDlg.OutputFile](#)<sup>587</sup> before starting the generation process. Note that this functionality is limited to XBRL View only. See [Document.CurrentViewMode](#)<sup>477</sup> and [SPYViewModes](#)<sup>616</sup>.

#### Errors

- 1400 The document object is no longer valid.
- 1407 Invalid parameters have been passed or an empty file name has been specified as output target.
- 1417 The document is not opened in schema view, maybe it is not an '.xsd' file.
- 1421 Feature is not available in this edition.
- 1422 Error during generation

### 15.3.2.10.34 GetDBStructureList

**Method:** GetDBStructureList (*pDatabase* as [DatabaseConnection](#)<sup>457</sup>) as [ElementList](#)<sup>505</sup>

#### Description

GetDBStructureList creates a collection of elements from the Schema document for which tables in the specified database are created. The function returns a collection of [ElementListItem](#)s where the properties [ElementListItem.Name](#)<sup>507</sup> contain the names of the tables.

See also [CreateDBStructureFromXMLSchema](#)<sup>476</sup>.

#### Errors

- 1400 The object is no longer valid.
- 1427 Failed creating parser for the specified XML.
- 1428 Export of element list failed.
- 1429 Database selection missing.

### 15.3.2.10.35 GetExportElementList

**Method:** GetExportElementList (*pFromChild* as [XMLData](#)<sup>593</sup>, *pExportSettings* as [ExportSettings](#)<sup>507</sup>) as [ElementList](#)<sup>505</sup>

#### Description

GetExportElementList creates a collection of elements to export from the document, depending on the settings in *pExportSettings* and starting from the element *pFromChild*. The function returns a collection of

ElementListItems where the properties [ElementListItem.Name](#)<sup>507</sup> contain the names of the tables that can be exported from the document. The property [ElementListItem.FieldCount](#)<sup>507</sup> contains the number of columns in the table. The property [ElementListItem.RecordCount](#)<sup>507</sup> contains the number of records in the table. The property [ElementListItem.ElementKind](#)<sup>507</sup> is unused. [UpdateXMLData\(\)](#)<sup>496</sup> might be indirectly needed as you have to pass the [XMLData](#)<sup>593</sup> as parameter to this function.

#### Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.
- 1427 Failed creating parser for the specified XML.
- 1428 Export of element list failed.

### 15.3.2.10.36 GetPathName (obsolete)

Superseded by [Document.FullName](#)<sup>481</sup>

```
// ---- 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](#)<sup>493</sup> (obsolete).

### 15.3.2.10.37 GridView

**Property:** GridView as [GridView](#)<sup>529</sup>

#### Description

This property provides access to the grid view functionality of the document.

#### Errors

- 1400 The object is no longer valid.
- 1407 Invalid address for the return parameter was specified.
- 1417 Document needs to be open in enhanced grid view mode.

### 15.3.2.10.38 IsModified

**Property:** IsModified as Boolean

**Description**

True if the document is modified.

**Errors**

- 1400 The object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

**15.3.2.10.39 IsValid**

**Method:** HRESULT IsValid([in, out] VARIANT \*strError, [in, out] VARIANT \*nErrorPos, [in, out] VARIANT \*pBadData, [out,retval] VARIANT\_BOOL \*bValid);

**Return Value**

True if the document is valid, false if not. To call `IsValid()`, the application GUI must be visible. (If you wish to validate without the GUI being visible, please use [Altova RaptorXML Server](#).)

**Description**

`IsValid` validates the document against its associated schema or DTD. `strError` gives you the same error message as when you validate the file within the GUI.

**Errors**

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.
- 1408 Unable to validate file.

**Example**

The following C++ code snippet provides an example of how to use the `IsValid` method.

```
#import "XMLSpy.tlb"

CComPtr< XMLSpyLib::IDocument12> ipDoc = ipXMLSpy->GetActiveDocument();
if ( ipDoc )
{
    // prepare in/out parameters for IsValid call
    CComVariant variantError;
    CComVariant variantErrorPos;
    CComVariant variantBadData;
    // IsValid always shows a dialog with the validation result. This cannot be turned
off.
    bool bIsValid = ipDoc->IsValid( &variantError, &variantErrorPos, &variantBadData )
== VARIANT_TRUE;

    if ( !bIsValid )
    {
        // retrieve values from out parameters
        CString strError = (V_VT( &variantError ) == VT_BSTR ?
V_BSTR( &variantError ) : _T( "" ));
```

```

        long npos = (V_VT( &variantErrorPos ) == VT_I4 ? V_I4( &variantErrorPos ) : -
1);
        CComQIPtr< XMLSpyLib::IXMLData > ipXMLBadData = (V_VT( &variantBadData ) ==
VT_DISPATCH ? V_DISPATCH( &variantBadData ) : nullptr);

        if ( ipXMLBadData )
            strError += CString( _T("\n\n Node: ") ) + (LPCWSTR)ipXMLBadData-
>GetName();

        if ( !strError.IsEmpty() )
            AfxMessageBox( "Validation failed - " + strError );
    }
}

```

### 15.3.2.10.4 IsValidEx

**Method:** IsValidEx (nXSDVersion as [SPYValidateXSDVersion](#)<sup>615</sup>, nErrorLimit as int, nErrorFormat as [SPYValidateErrorFormat](#)<sup>615</sup>, out strError as Variant) as Boolean

#### Return Value

True if the document is valid, false if not.

#### Description

IsValidEx validates the document against its associated schema or DTD.

#### In parameters:

nXSDVersion which is an enumeration value of [SPYValidateXSDVersion](#)<sup>615</sup> that selects the XSD version to validate against.

nErrorLimit which is an integer. Values must be 1 to 999.

nErrorFormat which is an enumeration value of [SPYValidateErrorFormat](#)<sup>615</sup> that selects the XSD version to validate against.

#### Out parameter:

strError is the error message, and is the same as that received when validating the file within the GUI.

#### Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.
- 1408 Unable to validate file.

#### Example

The following C++ code snippet provides an example of how to use the IsValidEx method.

```
#import "XMLSpy.tlb"
```

```
CComPtr< XMLSpyLib::IDocument12> ipDoc = ipXMLSpy->GetActiveDocument();
```

```

if ( ipDoc )
{
    CComVariant variantErrorEx;
    bool bIsValidEx = ipDoc->IsValidEx( XMLSpyLib::spyValidateXSDVersion_AutoDetect,
100, XMLSpyLib::SPYValidateErrorFormat_LongXML, &variantErrorEx ) == VARIANT_TRUE;

    // // retrieve values from out parameters
    CString strErrorEx = (V_VT( &variantErrorEx ) == VT_BSTR ?
V_BSTR( &variantErrorEx ) : _T( "" ));
    if ( !strErrorEx.IsEmpty() )
        AfxMessageBox( "Validation failed - " + strErrorEx );
}

```

#### 15.3.2.10.41 IsWellFormed

**Method:** IsWellFormed (*pData* as XMLData, *bWithChildren* as Boolean, *strError* as Variant, *nErrorPos* as Variant, *pBadXMLData* as Variant) as Boolean

#### Return Value

True if the document is well formed.

#### Description

IsWellFormed checks the document for well-formedness starting at the element *pData*.

If the document is not well formed, *strError* contains an error message, *nErrorPos* the position in the file and *pBadXMLData* holds a reference to the element which breaks the well-formedness. These out-parameters are defined as VARIANTS to support scripting languages like VBScript.

#### Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.

#### Example

See IsValid.

#### 15.3.2.10.42 Name

**Property:** Name as String (read-only)

#### Description

Use this property to retrieve the name - not including the path - of the document file. To change the file name for a document use the property [FullName](#)<sup>481</sup>.

#### Errors

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

### 15.3.2.10.43 Parent

**Property:** Parent as [Documents](#)<sup>497</sup> (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](#)<sup>382</sup> (read-only)

### 15.3.2.10.44 Path

**Property:** Path as String (read-only)

#### Description

Use this property to retrieve the path - not including the file name - of the document file. To change the file name and path for a document use the property [FullName](#)<sup>481</sup>.

#### Errors

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

### 15.3.2.10.45 RootElement

**Property:** RootElement as [XMLData](#)<sup>593</sup> (read-only)

#### Description

The property RootElement provides access to the root element of the XML structure of the document including the XML prolog data. To access the first element of a document's content navigate to the first child of kind *spyXMLDataElement* or use the [Document.DataRoot](#)<sup>477</sup> property. If the [CurrentViewMode](#)<sup>477</sup> is not *spyViewGrid* or *spyViewAuthentic* an [UpdateXMLData](#)<sup>496</sup> may be necessary to get access to the latest [XMLData](#)<sup>593</sup>.

#### Errors

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

### 15.3.2.10.46 Save

**Method:** Save()

#### Description

The method writes any modifications of the document to the associated file. See also [Document.FullName](#)<sup>481</sup>.

**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.

### 15.3.2.10.47 SaveAs

**Method:** SaveAs (*strFileName* as String)

**Description**

Save the document to the file specified. If saving was successful, the [FullName](#)<sup>481</sup> property gets set to the specified file name.

**Errors**

- 1400 The document object is no longer valid.
- 1407 An empty file name has been specified.
- 1403 Error when saving file, probably the file name is invalid.

### 15.3.2.10.48 Saved

**Property:** Saved as Boolean (read-only)

**Description**

This property can be used to check if the document has been saved after the last modifications. It returns the negation of [IsModified](#)<sup>486</sup>.

**Errors**

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

### 15.3.2.10.49 SaveInString

**Method:** SaveInString (*pData* as [XMLData](#)<sup>593</sup>, *bMarked* as Boolean) as String

**Parameters**

*pData*

XMLData element to start. Set *pData* to [Document.RootElement](#)<sup>490</sup> if you want to copy the complete file.

*bMarked*

If *bMarked* is true, only the elements selected in the grid view are copied.

**Return Value**

Returns a string with the XML data.

**Description**

SaveInString starts at the element pData and converts the XMLData objects to a string representation. [UpdateXMLData\(\)](#)<sup>496</sup> might be indirectly needed as you have to pass the [XMLData](#)<sup>593</sup> as parameter to this function.

**Errors**

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.

### 15.3.2.10.50 SaveToURL

**Method:** SaveToURL (*strURL* as String, *strUser* as String, *strPassword* as String)

**Return Value****Description**

SaveToURL() writes the document to the URL strURL. This method does not set the permanent file path of the document.

**Errors**

- 1400 The object is no longer valid.
- 1402 Invalid URL specified.
- 1403 Error while saving to URL.

### 15.3.2.10.51 SetActiveDocument

**Method:** SetActiveDocument()

**Description**

The method sets the document as the active and brings it to the front.

**Errors**

- 1400 The object is no longer valid.

### 15.3.2.10.52 SetEncoding (obsolete)

**Superseded by** [Document.Encoding](#)<sup>478</sup>

```
// ---- javascript sample ----  
// instead of:  
// Application.ActiveDocument.SetEncoding("UTF-16");  
// use now:  
Application.ActiveDocument.Encoding = "UTF-16";
```

**Method:** SetEncoding (*strEncoding* as String)

**Description**

SetEncoding sets the encoding of the document like the menu item "File/Encoding..." in Authentic Desktop. Possible values for strEncoding are, for example:

```
8859-1,  
8859-2,  
ASCII, ISO-646,  
850,  
1252,  
1255,  
SHIFT-JIS, MS-KANJI,  
BIG5, FIVE,  
UTF-7,  
UTF-8,  
UTF-16
```

### 15.3.2.10.53 SetExternallsValid

**Method:** SetExternallsValid (*bValid* as Boolean)

#### Parameters

bValid

Sets the result of an external validation process.

#### Description

The internal information set by this method is only queried on cancelling the default validation in any [OnBeforeValidate](#)<sup>470</sup> handler.

Available with TypeLibrary version 1.5

#### Errors

1400 The object is no longer valid.

### 15.3.2.10.54 SetPathName (obsolete)

Superseded by [Document.FullName](#)<sup>481</sup>

```
// ---- 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.

### 15.3.2.10.55 StartChanges

**Method:** `StartChanges()`

#### Description

After `StartChanges` is executed Authentic Desktop will not update its editor windows until [Document.EndChanges](#)<sup>478</sup> is called. This increases performance of complex tasks to the XML structure.

#### Errors

1400 The object is no longer valid.

### 15.3.2.10.56 Suggestions

**Property:** `Suggestions` as Array

#### Description

This property contains the last valid user suggestions for this document. The XMLSpy generated suggestions can be modified before they are shown to the user in the [OnBeforeShowSuggestions](#)<sup>554</sup> event.

#### Errors

1400 The object is no longer valid.

1407 Invalid parameter or invalid address for the return parameter was specified.

### 15.3.2.10.57 SwitchViewMode

**Method:** `SwitchViewMode (nMode as SPYViewModes616)` as Boolean

#### Return value

Returns true if view mode is switched.

#### Description

The method sets the current view mode of the document in Authentic Desktop. See also [Document.CurrentViewMode](#)<sup>477</sup>.

#### Errors

1400 The object is no longer valid.

1407 Invalid address for the return parameter was specified.

1417 Invalid view mode specified.

### 15.3.2.10.58 TextView

**Property:** `TextView` as [TextView](#)<sup>554</sup>

#### Description

This property provides access to the text view functionality of the document.

**Errors**

- 1400 The object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

### 15.3.2.10.59 Title

**Property:** Title as String (read-only)

**Description**

Title contains the file name of the document. To get the path and filename of the file use [FullName](#)<sup>481</sup>.

**Errors**

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

### 15.3.2.10.60 TransformXSL

**Method:** TransformXSL()

**Description**

TransformXSL processes the XML document via the associated XSL file. See [Document.AssignXSL](#)<sup>472</sup> on how to place a reference to a XSL file into the document.

**Errors**

- 1400 The document object is no longer valid.
- 1411 Error during transformation process.

### 15.3.2.10.61 TransformXSLEx

**Method:** TransformXSLEx(*nAction* as [SPYDialogAction](#)<sup>607</sup>)

**Description**

TransformXSLEx processes the XML document via the associated XSL file. The parameter specifies whether a dialog asking for the result document name should pop up or not. See [Document.AssignXSL](#)<sup>472</sup> on how to place a reference to a XSL file into the document.

**Errors**

- 1400 The document object is no longer valid.
- 1411 Error during transformation process.

### 15.3.2.10.62 TransformXSLFO

**Method:** TransformXSLFO()

**Description**

TransformXSLFO processes the XML document via the associated XSLFO file. See [AssignXSLFO](#)<sup>473</sup> on how to place a reference to a XSLFO file into the document. You need to assign a FOP processor to Authentic Desktop before you can use this method.

**Errors**

- 1400 The document object is no longer valid.
- 1411 Error during transformation process.

### 15.3.2.10.63 TreatXBRLInconsistenciesAsErrors

**Property:** TreatXBRLInconsistenciesAsErrors as Boolean

**Description**

If this is set to `true` the `Document.IsValid()` method will return `false` for XBRL instances containing inconsistencies as defined by the XBRL Specification. The default value of this property is `false`.

**Errors**

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

### 15.3.2.10.64 UpdateViews

**Method:** UpdateViews()

**Description**

To redraw the Enhanced Grid View and the Tree View call UpdateViews. This can be important after you changed the XMLData structure of a document. This method does not redraw the text view of Authentic Desktop.

**Errors**

- 1400 The document object is no longer valid.

### 15.3.2.10.65 UpdateXMLData

**Method:** UpdateXMLData() as Boolean

**Description**

The [XMLData](#)<sup>593</sup> tree is updated from the current view. Please note that this can fail in case of the TextView if the current XML text is not well-formed. This is not necessary if [CurrentViewMode](#)<sup>477</sup> is `spyViewGrid` or `spyViewAuthentic` because these views keep the [XMLData](#)<sup>593</sup> updated.

Available with TypeLibrary version 1.5

**Errors**

- 1400 The document object is no longer valid.

## 15.3.2.11 Documents

### Properties

[Count](#) <sup>498</sup>

[Item](#) <sup>498</sup>

### Methods

[NewAuthenticFile](#) <sup>498</sup>

[NewFile](#) <sup>498</sup>

[NewFileFromText](#) <sup>499</sup>

[OpenAuthenticFile](#) <sup>499</sup>

[OpenFile](#) <sup>500</sup>

[OpenURL](#) <sup>500</sup>

[OpenURLDialog](#) <sup>501</sup>

### Description

This object represents the set of documents currently open in Authentic Desktop. Use this object to open further documents or iterate through already opened documents.

### Examples

```
' _____  
' XMLSpy scripting environment - VBScript  
' iterate through open documents  
' _____  
  
Dim objDocuments  
Set objDocuments = Application.Documents  
  
For Each objDoc In objDocuments  
    'do something useful with your document  
    objDoc.SetActiveDocument()  
Next  
  
// _____  
// XMLSpy scripting environment - JScript  
// close all open documents  
// _____  
for (var iter = new Enumerator (Application.Documents);  
    ! iter.atEnd();  
    iter.moveNext())  
{  
    // MsgBox ("Closing file " + iter.item().Name);  
    iter.item().Close (true);  
}
```

### 15.3.2.11.1 Count

**Property:** Count as long

#### Description

Count of open documents.

#### Errors

- 1600 Invalid Documents object
- 1601 Invalid input parameter

### 15.3.2.11.2 Item

**Method:** Item (*n* as long) as [Document](#)<sup>467</sup>

#### Description

Gets the document with the index *n* in this collection. Index is 1-based.

#### Errors

- 1600 Invalid Documents object
- 1601 Invalid input parameter

### 15.3.2.11.3 NewAuthenticFile

**Method:** NewAuthenticFile (*strSPSPath* as String, *strXMLPath* as String) as [Document](#)<sup>467</sup>

#### Parameters

*strSPSPath*

The path to the SPS document.

*strXMLPath*

The new XML document name.

#### Return Value

The method returns the new document.

#### Description

NewAuthenticFile creates a new XML file and opens it in Authentic View using SPS design *strSPSPath*.

### 15.3.2.11.4 NewFile

**Method:** NewFile (*strFile* as String, *strType* as String) as [Document](#)<sup>467</sup>

**Parameters**

strFile

Full path of new file.

strType

Type of new file as string (i.e. "xml", "xsd", ... )

**Return Value**

Returns the new file.

**Description**

NewFile creates a new file of type strType (i.e. "xml"). The newly created file is also the ActiveDocument.

### 15.3.2.11.5 NewFileFromText

**Method:** NewFileFromText (*strText* as String, *strType* as String) as [Document](#)<sup>467</sup>

**Parameters**

strText

The content of the new document in plain text.

strType

Type of the document to create (i.e. "xml").

**Return Value**

The method returns the new document.

**Description**

NewFileFromText creates a new document with strText as its content.

### 15.3.2.11.6 OpenAuthenticFile

**Method:** OpenAuthenticFile (*strSPSPPath* as String, *strXMLPath* as String) as [Document](#)<sup>467</sup>

**Parameters**

strSPSPPath

The path to the SPS document.

strXMLPath

The path to the XML document (can be empty).

**Return Value**

The method returns the new document.

**Description**

OpenAuthenticFile opens an XML file or database in Authentic View using SPS design strSPSPPath.

### 15.3.2.11.7 OpenFile

**Method:** OpenFile (*strPath* as String, *bDialog* as Boolean) as [Document](#)<sup>467</sup>

#### Parameters

*strPath*

Path and file name of file to open.

*bDialog*

Show dialogs for user input.

#### Return Value

Returns the opened file on success.

#### Description

OpenFile opens the file *strPath*. If *bDialog* is TRUE, a file-dialog will be displayed.

#### Example

```
Dim objDoc As Document
Set objDoc = objSpy.Documents.OpenFile(strFile, False)
```

### 15.3.2.11.8 OpenURL

**Method:** OpenURL (*strURL* as String, *nURLType* as [SPYURLTypes](#)<sup>615</sup>, *nLoading* as [SPYLoading](#)<sup>609</sup>, *strUser* as String, *strPassword* as String) as [Document](#)<sup>467</sup>

#### Parameters

*strURL*

URL to open as document.

*nURLType*

Type of document to open. Set to -1 for auto detection.

*nLoading*

Set *nLoading* to 0 (zero) if you want to load it from cache or proxy. Otherwise set *nLoading* to 1.

*strUser*

Name of the user if required. Can be empty.

*strPassword*

Password for authentication. Can be empty.

#### Return Value

The method returns the opened document.

#### Description

OpenURL opens the URL *strURL*.

### 15.3.2.11.9 OpenURLDialog

**Method:** OpenURLDialog (*strURL* as String, *nURLType* as [SPYURLTypes](#)<sup>615</sup>, *nLoading* as [SPYLoading](#)<sup>609</sup>, *strUser* as String, *strPassword* as String) as [Document](#)<sup>467</sup>

#### Parameters

*strURL*

URL to open as document.

*nURLType*

Type of document to open. Set to -1 for auto detection.

*nLoading*

Set *nLoading* to 0 (zero) if you want to load it from cache or proxy. Otherwise set *nLoading* to 1.

*strUser*

Name of the user if required. Can be empty.

*strPassword*

Password for authentication. Can be empty.

#### Return Value

The method returns the opened document.

#### Description

OpenURLDialog displays the "open URL" dialog to the user and presets the input fields with the given parameters.

### 15.3.2.12 DTDSchemaGeneratorDlg

#### Properties and Methods

Standard automation properties

[Application](#)<sup>502</sup>

[Parent](#)<sup>504</sup>

[DTDSchemaFormat](#)<sup>502</sup>

[ValueList](#)<sup>505</sup>

[TypeDetection](#)<sup>505</sup>

[FrequentElements](#)<sup>502</sup>

[MergeAllEqualNamed](#)<sup>503</sup>

[ResolveEntities](#)<sup>505</sup>

[AttributeTypeDefinition](#)<sup>502</sup>

[GlobalAttributes](#)<sup>503</sup>

[OnlyStringEnums](#)<sup>504</sup>

[MaxEnumLength](#)<sup>503</sup>

[OutputPath](#)<sup>504</sup>

[OutputPathDialogAction](#)<sup>504</sup>

#### Description

Use this object to configure the generation of a schema or DTD. The method [GenerateDTDOrSchemaEx](#)<sup>482</sup> expects a `DTDSchemaGeneratorDlg` as parameter to configure the generation as well as the associated user interactions.

### 15.3.2.12.1 Application

**Property:** Application as [Application](#)<sup>382</sup> (read-only)

#### Description

Access the Authentic Desktop application object.

#### Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

### 15.3.2.12.2 AttributeTypeDefinition

**Property:** AttributeTypeDefinition as [SPYAttributeTypeDefinition](#)<sup>604</sup>

#### Description

Specifies how attribute definitions get merged.

#### Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

### 15.3.2.12.3 DTDSchemaFormat

**Property:** DTDSchemaFormat as [SPYDTDSchemaFormat](#)<sup>607</sup>

#### Description

Sets the schema output format to DTD, or W3C.

#### Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

### 15.3.2.12.4 FrequentElements

**Property:** FrequentElements as [SPYFrequentElements](#)<sup>608</sup>

#### Description

Shall the types for all elements be defined as global? Use the value `spyGlobalComplexType` to define them on global scope. Otherwise, use the value `spyGlobalElements`.

**Errors**

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

### 15.3.2.12.5 GlobalAttributes

**Property:** GlobalAttributes as Boolean

**Description**

Shall attributes with same name and type be resolved globally?

**Errors**

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

### 15.3.2.12.6 MaxEnumLength

**Property:** MaxEnumLength as Integer

**Description**

Specifies the maximum number of characters allowed for enumeration names. If one value is longer than this, no enumeration will be generated.

**Errors**

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

### 15.3.2.12.7 MergeAllEqualNamed

**Property:** MergeAllEqualNamed as Boolean

**Description**

Shall types of all elements with the same name be merged into one type?

**Errors**

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

### 15.3.2.12.8 OnlyStringEnums

**Property:** OnlyStringEnums as Boolean

#### Description

Specifies if enumerations will be created only for plain strings or all types of values.

#### Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

### 15.3.2.12.9 OutputPath

**Property:** OutputPath as String

#### Description

Selects the file name for the generated schema/DTD.

#### Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

### 15.3.2.12.10 OutputPathDialogAction

**Property:** OutputPathDialogAction as [SPYDialogAction](#)<sup>607</sup>

#### Description

Defines how the sub-dialog for selecting the schema/DTD output path gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog with the current value of the [OutputPath](#)<sup>504</sup> property as default. Use *spyDialogOK(0)* to hide the dialog from the user.

#### Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

### 15.3.2.12.11 Parent

**Property:** Parent as [Dialogs](#)<sup>464</sup> (read-only)

#### Description

Access the parent of the object.

#### Errors

- 3000 The object is no longer valid.

3001 Invalid address for the return parameter was specified.

### 15.3.2.12.12 ResolveEntities

**Property:** ResolveEntities as Boolean

#### Description

Shall all entities be resolved before generation starts? If yes, an info-set will be built.

#### Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

### 15.3.2.12.13 TypeDetection

**Property:** TypeDetection as [SPYTypeDetection](#) <sup>615</sup>

#### Description

Specifies granularity of simple type detection.

#### Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

### 15.3.2.12.14 ValueList

**Property:** ValueList as Integer

#### Description

Generate not more than this amount of enumeration-facets per type. Set to -1 for unlimited.

#### Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

### 15.3.2.13 ElementList

#### Properties

[Count](#) <sup>506</sup>  
[Item](#) <sup>506</sup>

#### Methods

[RemoveElement](#) <sup>506</sup>

### Description

Element lists are used for different purposes during export and import of data. Depending on this purpose, different properties of [ElementListItem](#) <sup>506</sup> are used.

It can hold

- a list of table names returned by a call to [Application.GetDatabaseTables](#) <sup>389</sup>,
- a list of field names returned by a call to [Application.GetDatabaseImportElementList](#) <sup>388</sup> or [Application.GetTextImportElementList](#) <sup>390</sup>,
- a field name filter list used in [Application.ImportFromDatabase](#) <sup>392</sup> and [Application.ImportFromText](#) <sup>394</sup>,
- a list of table names and counts for their rows and columns as returned by calls to [GetExportElementList](#) <sup>485</sup> or
- a field name filter list used in [Document.ExportToDatabase](#) <sup>479</sup> and [Document.ExportToText](#) <sup>480</sup>.

#### 15.3.2.13.1 Count

**Property:** Count as long (read-only)

### Description

Count of elements in this collection.

#### 15.3.2.13.2 Item

**Method:** Item(n as long) as [ElementListItem](#) <sup>506</sup>

### Description

Gets the element with the index n from this collection. The first item has index 1.

#### 15.3.2.13.3 RemoveElement

**Method:** RemoveElement(Index as long)

### Description

RemoveElement removes the element Index from the collection. The first Item has index 1.

### 15.3.2.14 ElementListItem

#### Properties

[Name](#) <sup>507</sup>

[ElementKind](#) <sup>507</sup>

[FieldCount](#) <sup>507</sup>

[RecordCount](#) <sup>507</sup>

### Description

An element in an [ElementList](#) <sup>505</sup>. Usage of its properties depends on the purpose of the element list. For details see [ElementList](#) <sup>505</sup>.

### 15.3.2.14.1 ElementKind

**Property:** ElementKind as [SPYXMLDataKind](#) <sup>617</sup>

#### Description

Specifies if a field should be imported as XML element (data value of spyXMLDataElement) or attribute (data value of spyXMLDataAttr).

### 15.3.2.14.2 FieldCount

**Property:** FieldCount as long (read-only)

#### Description

Count of fields (i.e. columns) in the table described by this element. This property is only valid after a call to [Document.GetExportElementList](#) <sup>485</sup>.

### 15.3.2.14.3 Name

**Property:** Name as String (read-only)

#### Description

Name of the element. This is either the name of a table or a field, depending on the purpose of the element list.

### 15.3.2.14.4 RecordCount

**Property:** RecordCount as long (read-only)

#### Description

Count of records (i.e. rows) in the table described by this element. This property is only valid after a call to [Document.GetExportElementList](#) <sup>485</sup>.

## 15.3.2.15 ExportSettings

### Properties

[ElementList](#) <sup>508</sup>

[EntitiesToText](#) <sup>508</sup>

[ExportAllElements](#) <sup>508</sup>

[SubLevelLimit](#) <sup>510</sup>

[FromAttributes](#) <sup>509</sup>

[FromSingleSubElements](#) <sup>509</sup>

[FromTextValues](#) <sup>509</sup>

[CreateKeys](#) <sup>508</sup>[IndependentPrimaryKey](#) <sup>509</sup>[Namespace](#) <sup>509</sup>[ExportCompleteXML](#) <sup>509</sup>[StartFromElement](#) <sup>510</sup>**Description**

ExportSettings contains options used during export of XML data to a database or text file.

### 15.3.2.15.1 CreateKeys

**Property:** CreateKeys as Boolean

**Description**

This property turns creation of keys (i.e. primary key and foreign key) on or off. Default is True.

### 15.3.2.15.2 ElementList

**Property:** ElementList as [ElementList](#) <sup>505</sup>

**Description**

Default is empty list. This list of elements defines which fields will be exported. To get the list of available fields use [Document.GetExportElementList](#) <sup>485</sup>. It is possible to prevent exporting columns by removing elements from this list with [ElementList.RemoveElement](#) <sup>506</sup> before passing it to [Document.ExportToDatabase](#) <sup>479</sup> or [Document.ExportToText](#) <sup>480</sup>.

### 15.3.2.15.3 EntitiesToText

**Property:** EntitiesToText as Boolean

**Description**

Defines if XML entities should be converted to text or left as they are during export. Default is True.

### 15.3.2.15.4 ExportAllElements

**Property:** ExportAllElements as Boolean

**Description**

If set to true, all elements in the document will be exported. If set to false, then [ExportSettings.SubLevelLimit](#) <sup>510</sup> is used to restrict the number of sub levels to export. Default is true.

### 15.3.2.15.5 ExportCompleteXML

**Property:** ExportCompleteXML as Boolean

#### Description

Defines whether the complete XML is exported or only the element specified by [StartFromElement](#)<sup>510</sup> and its children. Default is True.

### 15.3.2.15.6 FromAttributes

**Property:** FromAttributes as Boolean

#### Description

Set FromAttributes to false if no export data should be created from attributes. Default is True.

### 15.3.2.15.7 FromSingleSubElements

**Property:** FromSingleSubElements as Boolean

#### Description

Set FromSingleSubElements to false if no export data should be created from elements. Default is True.

### 15.3.2.15.8 FromTextValues

**Property:** FromTextValues as Boolean

#### Description

Set FromTextValues to false if no export data should be created from text values. Default is True.

### 15.3.2.15.9 IndependentPrimaryKey

**Property:** IndependentPrimaryKey as Boolean

#### Description

Turns creation of independent primary key counter for every element on or off. If [ExportSettings.CreateKeys](#)<sup>508</sup> is False, this property will be ignored. Default is True.

### 15.3.2.15.10 Namespace

**Property:** Namespace as [SPYExportNamespace](#)<sup>608</sup>

#### Description

The default setting removes all namespace prefixes from the element names. In some database formats the colon is not a legal character. Default is spyNoNamespace.

### 15.3.2.15.11 StartFromElement

**Property:** StartFromElement as String

#### Description

Specifies the start element for the export. This property is only considered when [ExportCompleteXML](#)<sup>509</sup> is false.

### 15.3.2.15.12 SubLevelLimit

**Property:** SubLevelLimit as Integer

#### Description

Defines the number of sub levels to include for the export. Default is 0. This property is ignored if [ExportSettings.ExportAllElements](#)<sup>508</sup> is true.

## 15.3.2.16 FileSelectionDlg

### Properties and Methods

Standard automation properties

[Application](#)<sup>510</sup>

[Parent](#)<sup>511</sup>

Dialog properties

[FullName](#)<sup>511</sup>

Acceptance or cancellation of action that caused event

[DialogAction](#)<sup>511</sup>

#### Description

The dialog object allows you to receive information about an event and pass back information to the event handler in the same way as with a user dialog. Use the [FileSelectionDlg.FullName](#)<sup>511</sup> to select or modify the file path and set the [FileSelectionDlg.DialogAction](#)<sup>511</sup> property to cancel or agree with the action that caused the event.

### 15.3.2.16.1 Application

**Property:** Application as [Application](#)<sup>382</sup> (read-only)

#### Description

Access the Authentic Desktop application object.

#### Errors

- 2400 The object is no longer valid.
- 2401 Invalid address for the return parameter was specified.

### 15.3.2.16.2 DialogAction

**Property:** DialogAction as [SPYDialogAction](#) <sup>607</sup>

#### Description

If you want your script to perform the file selection operation without any user interaction necessary, simulate user interaction by either setting the property to *spyDialogOK(0)* or *spyDialogCancel(1)*.

To allow your script to fill in the default values but let the user see and react on the dialog, use the value *spyDialogUserInput(2)*. If you receive a FileSelectionDlg object in an event handler, *spyDialogUserInput(2)* is not supported and will be interpreted as *spyDialogOK(0)*.

#### Errors

- 2400 The object is no longer valid.
- 2401 Invalid value for dialog action or invalid address for the return parameter was specified.

### 15.3.2.16.3 FullName

**Property:** FullName as String

#### Description

Access the full path of the file the gets selected by the dialog. Most events that pass a FileSelectionDlg object to you allow you modify this value and thus influence the action that caused the event (e.g. load or save to a different location).

#### Errors

- 2400 The object is no longer valid.
- 2401 Invalid address for the return parameter was specified.

### 15.3.2.16.4 Parent

**Property:** Parent as [Dialogs](#) <sup>464</sup> (read-only)

#### Description

Access the parent of the object.

#### Errors

- 2400 The object is no longer valid.
- 2401 Invalid address for the return parameter was specified.

### 15.3.2.17 FindInFilesDlg

#### Properties and Methods

Standard automation properties

[Application](#) <sup>512</sup>

[Parent](#) <sup>514</sup>

[Find](#) <sup>513</sup>  
[RegularExpression](#) <sup>514</sup>  
[Replace](#) <sup>515</sup>  
[DoReplace](#) <sup>513</sup>  
[ReplaceOnDisk](#) <sup>515</sup>  
[MatchWholeWord](#) <sup>514</sup>  
[MatchCase](#) <sup>514</sup>  
[SearchLocation](#) <sup>515</sup>  
[StartFolder](#) <sup>516</sup>  
[IncludeSubfolders](#) <sup>513</sup>  
[SearchInProjectFilesDoExternal](#) <sup>515</sup>  
[FileExtension](#) <sup>513</sup>  
[AdvancedXMLSearch](#) <sup>512</sup>  
[XMLElementNames](#) <sup>517</sup>  
[XMLElementContents](#) <sup>517</sup>  
[XMLAttributeNames](#) <sup>516</sup>  
[XMLAttributeContents](#) <sup>516</sup>  
[XMLComments](#) <sup>517</sup>  
[XMLCDATA](#) <sup>517</sup>  
[XMLPI](#) <sup>518</sup>  
[XMLRest](#) <sup>518</sup>  
[ShowResult](#) <sup>516</sup>

### Description

Use this object to configure the search (or replacement) for strings in files. The method [FindInFiles](#) <sup>388</sup> expects a [FindInFilesDlg](#) as parameter.

#### 15.3.2.17.1 AdvancedXMLSearch

**Property:** [AdvancedXMLSearch](#) as Boolean

### Description

Specifies if the XML search properties ([XMLElementNames](#) <sup>517</sup>, [XMLElementContents](#) <sup>517</sup>, [XMLAttributeNames](#) <sup>516</sup>, [XMLAttributeContents](#) <sup>517</sup>, [XMLComments](#) <sup>517</sup>, [XMLCDATA](#) <sup>517</sup>, [XMLPI](#) <sup>518</sup> and [XMLRest](#) <sup>518</sup>) are considered. The default is false.

### Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

#### 15.3.2.17.2 Application

**Property:** [Application](#) as [Application](#) <sup>382</sup> (read-only)

### Description

Access the Authentic Desktop application object.

### Errors

- 3500 The object is no longer valid.

3501 Invalid address for the return parameter was specified.

### 15.3.2.17.3 DoReplace

**Property:** DoReplace as Boolean

#### Description

Specifies if the matched string is replaced by the string defined in [Replace](#)<sup>515</sup>. The default is false.

#### Errors

3500 The object is no longer valid.  
3501 Invalid address for the return parameter was specified.

### 15.3.2.17.4 FileExtension

**Property:** FileExtension as String

#### Description

Specifies the file filter of the files that should be considered during the search. Multiple file filters must be delimited with a semicolon (eg: \*.xml;\*.dtd;a\*.xsd). Use the wildcards \* and ? to define the file filter.

#### Errors

3500 The object is no longer valid.  
3501 Invalid address for the return parameter was specified.

### 15.3.2.17.5 Find

**Property:** Find as String

#### Description

Specifies the string to search for.

#### Errors

3500 The object is no longer valid.  
3501 Invalid address for the return parameter was specified.

### 15.3.2.17.6 IncludeSubfolders

**Property:** IncludeSubfolders as Boolean

#### Description

Specifies if subfolders are searched too. The default is true.

#### Errors

3500 The object is no longer valid.  
3501 Invalid address for the return parameter was specified.

### 15.3.2.17.7 MatchCase

**Property:** MatchCase as Boolean

#### Description

Specifies if the search is case sensitive. The default is true.

#### Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

### 15.3.2.17.8 MatchWholeWord

**Property:** MatchWholeWord as Boolean

#### Description

Specifies whether the whole word or just a part of it must match. The default is false.

#### Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

### 15.3.2.17.9 Parent

**Property:** Parent as [Dialogs](#)<sup>464</sup> (read-only)

#### Description

Access the parent of the object.

#### Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

### 15.3.2.17.10 RegularExpression

**Property:** RegularExpression as Boolean

#### Description

Specifies if [Find](#)<sup>513</sup> contains a regular expression. The default is false.

#### Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

### 15.3.2.17.11 Replace

**Property:** Replace as String

#### Description

Specifies the replacement string. The matched string is only replaced if [DoReplace](#)<sup>513</sup> is set true.

#### Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

### 15.3.2.17.12 ReplaceOnDisk

**Property:** ReplaceOnDisk as Boolean

#### Description

Specifies if the replacement is done directly on disk. The modified file is not opened. The default is false.

#### Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

### 15.3.2.17.13 SearchInProjectFilesDoExternal

**Property:** SearchInProjectFilesDoExternal as Boolean

#### Description

Specifies if the external folders in the open project are searched, when a project search is performed. The default is false.

#### Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

### 15.3.2.17.14 SearchLocation

**Property:** SearchLocation as [SPYFindInFilesSearchLocation](#)<sup>608</sup>

#### Description

Specifies the location of the search. The default is spyFindInFiles\_Documents.

#### Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

### 15.3.2.17.15 ShowResult

**Property:** ShowResult as Boolean

#### Description

Specifies if the result is displayed in the Find in Files output window. The default is false.

#### Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

### 15.3.2.17.16 StartFolder

**Property:** StartFolder as String

#### Description

Specifies the folder where the disk search starts.

#### Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

### 15.3.2.17.17 XMLAttributeContents

**Property:** XMLAttributeContents as Boolean

#### Description

Specifies if attribute contents are searched when [AdvancedXMLSearch](#)<sup>512</sup> is true. The default is true.

#### Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

### 15.3.2.17.18 XMLAttributeNames

**Property:** XMLAttributeNames as Boolean

#### Description

Specifies if attribute names are searched when [AdvancedXMLSearch](#)<sup>512</sup> is true. The default is true.

#### Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

### 15.3.2.17.19 XMLCDATA

**Property:** XMLCDATA as Boolean

#### Description

Specifies if CDATA tags are searched when [AdvancedXMLSearch](#)<sup>512</sup> is true. The default is true.

#### Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

### 15.3.2.17.20 XMLComments

**Property:** XMLComments as Boolean

#### Description

Specifies if comments are searched when [AdvancedXMLSearch](#)<sup>512</sup> is true. The default is true.

#### Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

### 15.3.2.17.21 XMLElementContents

**Property:** XMLElementContents as Boolean

#### Description

Specifies if element contents are searched when [AdvancedXMLSearch](#)<sup>512</sup> is true. The default is true.

#### Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

### 15.3.2.17.22 XMLElementNames

**Property:** XMLElementNames as Boolean

#### Description

Specifies if element names are searched when [AdvancedXMLSearch](#)<sup>512</sup> is true. The default is true.

#### Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

### 15.3.2.17.23 XMLPI

**Property:** XMLPI as Boolean

#### Description

Specifies if XML processing instructions are searched when [AdvancedXMLSearch](#)<sup>512</sup> is true. The default is true.

#### Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

### 15.3.2.17.24 XMLRest

**Property:** XMLRest as Boolean

#### Description

Specifies if the rest of the XML (which is not covered by the other XML search properties) is searched when [AdvancedXMLSearch](#)<sup>512</sup> is true. The default is true.

#### Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

## 15.3.2.18 FindInFilesResult

### Properties and Methods

Standard automation properties

[Application](#)<sup>518</sup>

[Parent](#)<sup>519</sup>

[Count](#)<sup>519</sup>

[Item](#)<sup>519</sup>

[Path](#)<sup>520</sup>

[Document](#)<sup>519</sup>

#### Description

This object represents a file that matched the search criteria. It contains a list of [FindInFilesResultMatch](#)<sup>520</sup> objects that describe the matching position.

### 15.3.2.18.1 Application

**Property:** Application as [Application](#)<sup>382</sup> (read-only)

#### Description

Access the Authentic Desktop application object.

**Errors**

- 3700 The object is no longer valid.
- 3701 Invalid address for the return parameter was specified.

### 15.3.2.18.2 Count

**Property:** Count as long (read-only)

**Description**

Count of elements in this collection.

### 15.3.2.18.3 Document

**Property:** Path as [Document](#)<sup>467</sup> (read-only)

**Description**

This property returns the [Document](#)<sup>467</sup> object if the matched file is already open in XMLSpy.

**Errors**

- 3700 The object is no longer valid.
- 3701 Invalid address for the return parameter was specified.

### 15.3.2.18.4 Item

**Method:** Item(n as long) as [FindInFilesResultMatch](#)<sup>520</sup>

**Description**

Gets the element with the index n from this collection. The first item has index 1.

### 15.3.2.18.5 Parent

**Property:** Parent as [FindInFilesResults](#)<sup>522</sup> (read-only)

**Description**

Access the parent of the object.

**Errors**

- 3700 The object is no longer valid.
- 3701 Invalid address for the return parameter was specified.

### 15.3.2.18.6 Path

**Property:** Path as String (read-only)

#### Description

Returns the path of the file that matched the search criteria.

#### Errors

- 3700 The object is no longer valid.
- 3701 Invalid address for the return parameter was specified.

## 15.3.2.19 FindInFilesResultMatch

### Properties and Methods

Standard automation properties

[Application](#) <sup>520</sup>

[Parent](#) <sup>521</sup>

[Line](#) <sup>521</sup>

[Position](#) <sup>521</sup>

[Length](#) <sup>520</sup>

[LineText](#) <sup>521</sup>

[Replaced](#) <sup>522</sup>

#### Description

Contains the exact position in the file of the matched string.

### 15.3.2.19.1 Application

**Property:** Application as [Application](#) <sup>382</sup> (read-only)

#### Description

Access the Authentic Desktop application object.

#### Errors

- 3800 The object is no longer valid.
- 3801 Invalid address for the return parameter was specified.

### 15.3.2.19.2 Length

**Property:** Length as Long (read-only)

#### Description

Returns the length of the matched string.

**Errors**

- 3800 The object is no longer valid.
- 3801 Invalid address for the return parameter was specified.

### 15.3.2.19.3 Line

**Property:** Line as Long (read-only)

**Description**

Returns the line number of the match. The line numbering starts with 0.

**Errors**

- 3800 The object is no longer valid.
- 3801 Invalid address for the return parameter was specified.

### 15.3.2.19.4 LineText

**Property:** LineText as String (read-only)

**Description**

Returns the text of the line.

**Errors**

- 3800 The object is no longer valid.
- 3801 Invalid address for the return parameter was specified.

### 15.3.2.19.5 Parent

**Property:** Parent as [FindInFilesResult](#)<sup>518</sup> (read-only)

**Description**

Access the parent of the object.

**Errors**

- 3800 The object is no longer valid.
- 3801 Invalid address for the return parameter was specified.

### 15.3.2.19.6 Position

**Property:** Position as Long (read-only)

**Description**

Returns the start position of the match in the line. The position numbering starts with 0.

**Errors**

- 3800 The object is no longer valid.
- 3801 Invalid address for the return parameter was specified.

### 15.3.2.19.7 Replaced

**Property:** Replaced as Boolean (read-only)

**Description**

True if the matched string was replaced.

**Errors**

- 3800 The object is no longer valid.
- 3801 Invalid address for the return parameter was specified.

## 15.3.2.20 FindInFilesResults

**Properties and Methods**

Standard automation properties

[Application](#) <sup>522</sup>

[Parent](#) <sup>523</sup>

[Count](#) <sup>523</sup>

[Item](#) <sup>523</sup>

**Description**

This is the result of the [FindInFiles](#) <sup>388</sup> method. It is a list of [FindInFilesResult](#) <sup>518</sup> objects.

### 15.3.2.20.1 Application

**Property:** Application as [Application](#) <sup>382</sup> (read-only)

**Description**

Access the Authentic Desktop application object.

**Errors**

- 3600 The object is no longer valid.
- 3601 Invalid address for the return parameter was specified.

### 15.3.2.20.2 Count

**Property:** Count as long (read-only)

#### Description

Count of elements in this collection.

### 15.3.2.20.3 Item

**Method:** Item(n as long) as [FindInFilesResult](#) <sup>518</sup>

#### Description

Gets the element with the index n from this collection. The first item has index 1.

### 15.3.2.20.4 Parent

**Property:** Parent as [Application](#) <sup>382</sup> (read-only)

#### Description

Access the parent of the object.

#### Errors

- 3600 The object is no longer valid.
- 3601 Invalid address for the return parameter was specified.

## 15.3.2.21 GenerateSampleXMLDlg

### Properties and Methods

Standard automation properties

[Application](#) <sup>524</sup>

[Parent](#) <sup>527</sup>

[NonMandatoryAttributes](#) <sup>526</sup>

[NonMandatoryElements](#) <sup>526</sup>

[RepeatCount](#) <sup>527</sup>

[FillAttributesWithSampleData](#) <sup>525</sup>

[FillElementsWithSampleData](#) <sup>525</sup>

[ContentOfNillableElementsIsNonMandatory](#) <sup>525</sup>

[TryToUseNonAbstractTypes](#) <sup>528</sup>

[SchemaOrDTDAssignment](#) <sup>528</sup>

[LocalNameOfRootElement](#) <sup>526</sup>

[NamespaceURIOfRootElement](#) <sup>526</sup>

[OptionsDialogAction](#) <sup>527</sup>

Properties that are no longer supported

[TakeFirstChoice - obsolete](#) <sup>528</sup>

[FillWithSampleData - obsolete](#) <sup>525</sup>  
[Optimization - obsolete](#) <sup>527</sup>

### Description

Used to set the parameters for the generation of sample XML instances based on a W3C schema or DTD.

#### 15.3.2.21.1 Application

**Property:** Application as [Application](#) <sup>382</sup> (read-only)

### Description

Access the Authentic Desktop application object.

### Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

#### 15.3.2.21.2 ChoiceMode

**Property:** ChoiceMode as [SPYSampleXMLGenerationChoiceMode](#) <sup>612</sup>

### Description

Specifies which elements will be generated.

### Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

#### 15.3.2.21.3 ConsiderSampleValueHints

**Property:** ConsiderSampleValueHints as Boolean

### Description

Selects whether to use [SampleValueHints](#) <sup>528</sup> or not.

### Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

#### 15.3.2.21.4 ContentOfNillableElementsIsNonMandatory

**Property:** ContentOfNillableElementsIsNonMandatory as Boolean

##### Description

If true, the contents of elements that are nillable will not be treated as mandatory.

##### Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

#### 15.3.2.21.5 FillAttributesWithSampleData

**Property:** FillAttributesWithSampleData as Boolean

##### Description

If true, attributes will have sample content.

##### Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

#### 15.3.2.21.6 FillElementsWithSampleData

**Property:** FillElementsWithSampleData as Boolean

##### Description

If true, elements will have sample content.

##### Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

#### 15.3.2.21.7 FillWithSampleData - obsolete

**Property:** FillWithSampleData as Boolean

##### Description

Do no longer access this property. Use [FillAttributesWithSampleData](#)<sup>525</sup> and [FillElementsWithSampleData](#)<sup>525</sup>, instead.

##### Errors

- 0001 The property is no longer accessible.

### 15.3.2.21.8 LocalNameOfRootElement

**Property:** LocalNameOfRootElement as String

**Description**

Specifies the local name of the root element for the generated sample XML.

**Errors**

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

### 15.3.2.21.9 NamespaceURIOfRootElement

**Property:** NamespaceURIOfRootElement as String

**Description**

Specifies the namespace URI of the root element for the generated sample XML.

**Errors**

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

### 15.3.2.21.10 NonMandatoryAttributes

**Property:** NonMandatoryAttributes as Boolean

**Description**

If true attributes which are not mandatory are created in the sample XML instance file.

**Errors**

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

### 15.3.2.21.11 NonMandatoryElements

**Property:** NonMandatoryElements as Boolean

**Description**

If true, elements which are not mandatory are created in the sample XML instance file.

**Errors**

- 2200 The object is no longer valid.
- 2201 Invalid address was specified for the return parameter.

### 15.3.2.21.12 Optimization - obsolete

**Property:** Optimization as [SPYSampleXMLGenerationOptimization](#)<sup>612</sup>

#### Description

Do not use this property any longer. Use ChoiceMode and NonMandatoryElements.

#### Errors

0001 The property is no longer accessible.

### 15.3.2.21.13 OptionsDialogAction

**Property:** OptionsDialogAction as [SPYDialogAction](#)<sup>607</sup>

#### Description

To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

#### Errors

2200 The object is no longer valid.  
2201 Invalid value has been used to set the property.  
Invalid address for the return parameter was specified.

### 15.3.2.21.14 Parent

**Property:** Parent as [Dialogs](#)<sup>464</sup> (read-only)

#### Description

Access the parent of the object.

#### Errors

2200 The object is no longer valid.  
2201 Invalid address for the return parameter was specified.

### 15.3.2.21.15 RepeatCount

**Property:** RepeatCount as long

#### Description

Number of elements to create for repeated types.

#### Errors

2200 The object is no longer valid.  
2201 Invalid address for the return parameter was specified.

### 15.3.2.21.16 SampleValueHints

**Property:** SampleValueHints as [SPYSampleXMLGenerationSampleValueHints](#) <sup>612</sup>

#### Description

Specifies how to select data for the generated sample file.

#### Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

### 15.3.2.21.17 SchemaOrDTDAssignment

**Property:** SchemaOrDTDAssignment as [SPYSampleXMLGenerationSchemaOrDTDAssignment](#) <sup>612</sup>

#### Description

Specifies in which way a reference to the related schema or DTD - which is this document - will be generated into the sample XML.

#### Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

### 15.3.2.21.18 TakeFirstChoice - obsolete

**Property:** TakeFirstChoice as Boolean

#### Description

Do no longer use this property.

#### Errors

- 0001 The property is no longer accessible.

### 15.3.2.21.19 TryToUseNonAbstractTypes

**Property:** TryToUseNonAbstractTypes as Boolean

#### Description

If true, tries to use a non-abstract type for xsi:type, if element has an abstract type.

#### Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

## 15.3.2.22 GridView

### Methods

[Deselect](#) <sup>531</sup>

[Select](#) <sup>532</sup>

[SetFocus](#) <sup>532</sup>

### Properties

[CurrentFocus](#) <sup>531</sup>

[IsVisible](#) <sup>532</sup>

### Description

GridView Class

### 15.3.2.22.1 Events

#### 15.3.2.22.1.1 OnBeforeDrag

**Event:** OnBeforeDrag() as Boolean

**XMLSpy scripting environment - VBScript:**

```
Function On_BeforeDrag()  
    ' On_BeforeStartEditing = False ' to prohibit dragging  
End Function
```

**XMLSpy scripting environment - JScript:**

```
function On_BeforeDrag()  
{  
    // return false; /* to prohibit dragging */  
}
```

**XMLSpy IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (4, ...) // nEventId = 4
```

### Description

This event gets fired on an attempt to drag an XMLData element on the grid view. Return *false* to prevent dragging the data element to a different position.

#### 15.3.2.22.1.2 OnBeforeDrop

**Event:** OnBeforeDrop(objXMLData as [XMLData](#) <sup>593</sup>) as Boolean

**XMLSpy scripting environment - VBScript:**

```
Function On_BeforeDrop(objXMLData)
```

```
' On_BeforeStartEditing = False ' to prohibit dropping
End Function
```

**XMLSpy scripting environment - JScript:**

```
function On_BeforeDrop(objXMLData)
{
    // return false; /* to prohibit dropping */
}
```

**XMLSpy IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (5, ...) // nEventId = 5
```

**Description**

This event gets fired on an attempt to drop a previously dragged XMLData element on the grid view. Return *false* to prevent the data element to be moved from its original position to the drop destination position.

### 15.3.2.22.1.3 OnBeforeStartEditing

**Event:** OnBeforeStartEditing(objXMLData as [XMLData](#)<sup>593</sup>, bEditingName as Boolean) as Boolean

**XMLSpy scripting environment - VBScript:**

```
Function On_BeforeStartEditing(objXMLData, bEditingName)
' On_BeforeStartEditing = False ' to prohibit editing the field
End Function
```

**XMLSpy scripting environment - JScript:**

```
function On_BeforeStartEditing(objXMLData, bEditingName)
{
    // return false; /* to prohibit editing the field */
}
```

**XMLSpy IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (1, ...) // nEventId = 1
```

**Description**

This event gets fired before the editing mode for a grid cell gets entered. If the parameter *bEditingName* is true, the name part of the element will be edited, if its value is false, the value part will be edited.

### 15.3.2.22.1.4 OnEditingFinished

**Event:** OnEditingFinished(objXMLData as [XMLData](#)<sup>593</sup>, bEditingName as Boolean)

**XMLSpy scripting environment - VBScript:**

```
Function On_EditingFinished(objXMLData, bEditingName)
End Function
```

**XMLSpy scripting environment - JScript:**

```
function On_EditingFinished(objXMLData, bEditingName)
{
}
```

**XMLSpy IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (2, ...) // nEventId = 2
```

**Description**

This event gets fired when the editing mode of a grid cell is exited. The parameter *bEditingName* specifies if the name part of the element has been edited.

### 15.3.2.22.1.5 OnFocusChanged

**Event:** OnFocusChanged(objXMLData as [XMLData](#)<sup>593</sup>, bSetFocus as Boolean, bEditingName as Boolean)

**XMLSpy scripting environment - VBScript:**

```
Function On_FocusChanged(objXMLData, bSetFocus, bEditingName)
End Function
```

**XMLSpy scripting environment - JScript:**

```
function On_FocusChanged(objXMLData, bSetFocus, bEditingName)
{
}
```

**XMLSpy IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (3, ...) // nEventId = 3
```

**Description**

This event gets fired whenever a grid cell receives or loses the cursor focus. If the parameter *bEditingName* is *true*, focus of the name part of the grid element has changed. Otherwise, focus of the value part has changed.

### 15.3.2.22.2 CurrentFocus

**Property:** CurrentFocus as [XMLData](#)<sup>593</sup>

**Description**

Holds the XML element with the current focus. This property is read-only.

### 15.3.2.22.3 Deselect

**Method:** Deselect(pData as [XMLData](#)<sup>593</sup>)

**Description**

Deselects the element pData in the grid view.

#### 15.3.2.22.4 IsVisible

**Property:** IsVisible as Boolean

##### Description

True if the grid view is the active view of the document. This property is read-only.

#### 15.3.2.22.5 Select

**Method:** Select (*pData* as [XMLData](#)<sup>593</sup>)

##### Description

Selects the XML element *pData* in the grid view.

#### 15.3.2.22.6 SetFocus

**Method:** SetFocus (*pFocusData* as [XMLData](#)<sup>593</sup>)

##### Description

Sets the focus to the element *pFocusData* in the grid view.

### 15.3.2.23 SchemaDocumentationDlg

#### Properties and Methods

Standard automation properties

[Application](#)<sup>534</sup>

[Parent](#)<sup>540</sup>

Interaction and visibility properties

[OutputFile](#)<sup>539</sup>

[OutputFileDialogAction](#)<sup>540</sup>

[OptionsDialogAction](#)<sup>539</sup>

[ShowProgressbar](#)<sup>543</sup>

[ShowResult](#)<sup>543</sup>

Document generation options and methods

[OutputFormat](#)<sup>540</sup>

[UseFixedDesign](#)<sup>545</sup>

[SPSFile](#)<sup>545</sup>

[EmbedDiagrams](#)<sup>535</sup>

[DiagramFormat](#)<sup>534</sup>

[MultipleOutputFiles](#)<sup>539</sup>

[EmbedCSSInHTML](#)<sup>534</sup>

[CreateDiagramsFolder](#)<sup>534</sup>

[GenerateRelativeLinks](#)<sup>535</sup>

[IncludeAll](#) <sup>535</sup>  
[IncludeIndex](#) <sup>537</sup>  
[IncludeGlobalAttributes](#) <sup>536</sup>  
[IncludeGlobalElements](#) <sup>536</sup>  
[IncludeLocalAttributes](#) <sup>537</sup>  
[IncludeLocalElements](#) <sup>538</sup>  
[IncludeGroups](#) <sup>537</sup>  
[IncludeComplexTypes](#) <sup>536</sup>  
[IncludeSimpleTypes](#) <sup>538</sup>  
[IncludeAttributeGroups](#) <sup>536</sup>  
[IncludeRedefines](#) <sup>538</sup>  
[IncludeReferencedSchemas](#) <sup>538</sup>

[AllDetails](#) <sup>533</sup>  
[ShowDiagram](#) <sup>541</sup>  
[ShowNamespace](#) <sup>542</sup>  
[ShowType](#) <sup>544</sup>  
[ShowChildren](#) <sup>541</sup>  
[ShowUsedBy](#) <sup>545</sup>  
[ShowProperties](#) <sup>543</sup>  
[ShowSingleFacets](#) <sup>544</sup>  
[ShowPatterns](#) <sup>543</sup>  
[ShowEnumerations](#) <sup>542</sup>  
[ShowAttributes](#) <sup>541</sup>  
[ShowIdentityConstraints](#) <sup>542</sup>  
[ShowAnnotations](#) <sup>540</sup>  
[ShowSourceCode](#) <sup>544</sup>

### Description

This object combines all options for schema document generation as they are available through user interface dialog boxes in Authentic Desktop. The document generation options are initialized with the values used during the last generation of schema documentation. However, before using the object you have to set the [SetOutputFile](#) <sup>539</sup> property to a valid file path. Use [OptionsDialogAction](#) <sup>539</sup>, [OutputFileDialogAction](#) <sup>540</sup> and [ShowProgressBar](#) <sup>543</sup> to specify the level of user interaction desired. You can use [IncludeAll](#) <sup>535</sup> and [AllDetails](#) <sup>533</sup> to set whole option groups at once or the individual properties to operate on a finer granularity.

#### 15.3.2.23.1 AllDetails

**Method:** AllDetails (i\_bDetailsOn as Boolean)

### Description

Use this method to turn all details options on or off.

### Errors

2900 The object is no longer valid.

### 15.3.2.23.2 Application

**Property:** Application as [Application](#)<sup>382</sup> (read-only)

#### Description

Access the Authentic Desktop application object.

#### Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.3 CreateDiagramsFolder

**Property:** CreateDiagramsFolder as Boolean

#### Description

Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The default for the first run is false.

#### Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.4 DiagramFormat

**Property:** DiagramFormat as [SPYImageKind](#)<sup>608</sup>

#### Description

This property specifies the generated diagram image type. This property is not available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is PNG.

#### Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.5 EmbedCSSInHTML

**Property:** EmbedCSSInHTML as Boolean

#### Description

Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The default for the first run is true.

**Errors**

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.6 EmbedDiagrams

**Property:** EmbedDiagrams as Boolean

**Description**

Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

**Errors**

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.7 GenerateRelativeLinks

**Property:** GenerateRelativeLinks as Boolean

**Description**

Set this property to true, to create relative paths to local files. This property is not available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is false.

**Errors**

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.8 IncludeAll

**Method:** IncludeAll (i\_bInclude as Boolean)

**Description**

Use this method to mark or unmark all include options.

**Errors**

- 2900 The object is no longer valid.

### 15.3.2.23.9 IncludeAttributeGroups

**Property:** IncludeAttributeGroups as Boolean

#### Description

Set this property to true, to include attribute groups in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

#### Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.10 IncludeComplexTypes

**Property:** IncludeComplexTypes as Boolean

#### Description

Set this property to true, to include complex types in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

#### Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.11 IncludeGlobalAttributes

**Property:** IncludeGlobalAttributes as Boolean

#### Description

Set this property to true, to include global attributes in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

#### Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.12 IncludeGlobalElements

**Property:** IncludeGlobalElements as Boolean

#### Description

Set this property to true, to include global elements in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

**Errors**

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.13 IncludeGroups

**Property:** IncludeGroups as Boolean

**Description**

Set this property to true, to include groups in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

**Errors**

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.14 IncludeIndex

**Property:** IncludeIndex as Boolean

**Description**

Set this property to true, to include an index in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

**Errors**

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.15 IncludeLocalAttributes

**Property:** IncludeLocalAttributes as Boolean

**Description**

Set this property to true, to include local attributes in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

**Errors**

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.16 IncludeLocalElements

**Property:** IncludeLocalElements as Boolean

#### Description

Set this property to true, to include local elements in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

#### Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.17 IncludeRedefines

**Property:** IncludeRedefines as Boolean

#### Description

Set this property to true, to include redefines in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

#### Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.18 IncludeReferencedSchemas

**Property:** IncludeReferencedSchemas as Boolean

#### Description

Set this property to true, to include referenced schemas in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

#### Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.19 IncludeSimpleTypes

**Property:** IncludeSimpleTypes as Boolean

#### Description

Set this property to true, to include simple types in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

**Errors**

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.20 MultipleOutputFiles

**Property:** MultipleOutputFiles as Boolean

**Description**

Set this property to true, to split the documentation files. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is false.

**Errors**

- 2900 The object is no longer valid.
- 2901 Invalid value has been used to set the property.  
Invalid address for the return parameter was specified.

### 15.3.2.23.21 OptionsDialogAction

**Property:** OptionsDialogAction as [SPYDialogAction](#)<sup>607</sup>

**Description**

To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

**Errors**

- 2900 The object is no longer valid.
- 2901 Invalid value has been used to set the property.  
Invalid address for the return parameter was specified.

### 15.3.2.23.22 OutputFile

**Property:** OutputFile as String

**Description**

Full path and name of the file that will contain the generated documentation. In case of HTML output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>.

**Errors**

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.23 OutputFileDialogAction

**Property:** OutputFileDialogAction as [SPYDialogAction](#)<sup>607</sup>

#### Description

To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in [OutputFile](#)<sup>539</sup> should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

#### Errors

- 2900 The object is no longer valid.
- 2901 Invalid value has been used to set the property.  
Invalid address for the return parameter was specified.

### 15.3.2.23.24 OutputFormat

**Property:** OutputFormat as [SPYSchemaDocumentationFormat](#)<sup>613</sup>

#### Description

Defines the kind of documentation that will be generated: HTML (value=0), MS-Word (value=1), or RTF (value=2). The property gets initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is HTML.

#### Errors

- 2900 The object is no longer valid.
- 2901 Invalid value has been used to set the property.  
Invalid address for the return parameter was specified.

### 15.3.2.23.25 Parent

**Property:** Parent as [Dialogs](#)<sup>464</sup> (read-only)

#### Description

Access the parent of the object.

#### Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.26 ShowAnnotations

**Property:** ShowAnnotations as Boolean

#### Description

Set this property to true, to show the annotations to a type definition in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

**Errors**

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.27 ShowAttributes

**Property:** ShowAttributes as Boolean

**Description**

Set this property to true, to show the type definitions attributes in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

**Errors**

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.28 ShowChildren

**Property:** ShowChildren as Boolean

**Description**

Set this property to true, to show the children of a type definition as links in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

**Errors**

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.29 ShowDiagram

**Property:** ShowDiagram as Boolean

**Description**

Set this property to true, to show type definitions as diagrams in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

**Errors**

- 2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

### 15.3.2.23.30 ShowEnumerations

**Property:** ShowEnumerations as Boolean

#### Description

Set this property to true, to show the enumerations contained in a type definition in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

#### Errors

2900 The object is no longer valid.  
2901 Invalid address for the return parameter was specified.

### 15.3.2.23.31 ShowIdentityConstraints

**Property:** ShowIdentityConstraints as Boolean

#### Description

Set this property to true, to show a type definitions identity constraints in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

#### Errors

2900 The object is no longer valid.  
2901 Invalid address for the return parameter was specified.

### 15.3.2.23.32 ShowNamespace

**Property:** ShowNamespace as Boolean

#### Description

Set this property to true, to show the namespace of type definitions in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

#### Errors

2900 The object is no longer valid.  
2901 Invalid address for the return parameter was specified.

### 15.3.2.23.33 ShowPatterns

**Property:** ShowPatterns as Boolean

#### Description

Set this property to true, to show the patterns of a type definition in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

#### Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.34 ShowProgressBar

**Property:** ShowProgressBar as Boolean

#### Description

Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

#### Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.35 ShowProperties

**Property:** ShowProperties as Boolean

#### Description

Set this property to true, to show the type definition properties in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

#### Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.36 ShowResult

**Property:** ShowResult as Boolean

#### Description

Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in Authentic Desktop. To show Word documentation, MS-Word will be started.

The property gets initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

**Errors**

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.37 ShowSingleFacets

**Property:** ShowSingleFacets as Boolean

**Description**

Set this property to true, to show the facets of a type definition in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

**Errors**

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.38 ShowSourceCode

**Property:** ShowSourceCode as Boolean

**Description**

Set this property to true, to show the XML source code for type definitions in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

**Errors**

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.39 ShowType

**Property:** ShowType as Boolean

**Description**

Set this property to true, to show the type of type definitions in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

**Errors**

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.40 ShowUsedBy

**Property:** ShowUsedBy as Boolean

#### Description

Set this property to true, to show the used-by relation for type definitions in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#)<sup>483</sup>. The default for the first run is true.

#### Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.41 SPSFile

**Property:** SPSFile as String

#### Description

Full path and name of the SPS file that will be used to generate the documentation.

#### Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

### 15.3.2.23.42 UseFixedDesign

**Property:** UseFixedDesign as Boolean

#### Description

Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

#### Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

## 15.3.2.24 SpyProject

#### Methods

[CloseProject](#)<sup>546</sup>

[SaveProject](#)<sup>546</sup>

[SaveProjectAs](#)<sup>547</sup>

#### Properties

[RootItems](#) <sup>546</sup>[ProjectFile](#) <sup>546</sup>**Description**

SpyProject Class

### 15.3.2.24.1 CloseProject

**Declaration:** CloseProject(*bDiscardChanges* as Boolean, *bCloseFiles* as Boolean, *bDialog* as Boolean)**Parameters**

bDiscardChanges

Set bDiscardChanges to FALSE if you want to save the changes of the open project files and the project.

bCloseFiles

Set bCloseFiles to TRUE to close all open project files.

bDialog

Show dialogs for user input.

**Description**

CloseProject closes the current project.

### 15.3.2.24.2 ProjectFile

**Declaration:** ProjectFile as String**Description**

Path and filename of the project.

### 15.3.2.24.3 RootItems

**Declaration:** RootItems as [SpyProjectItems](#) <sup>549</sup>**Description**

Root level of collection of project items.

### 15.3.2.24.4 SaveProject

**Declaration:** SaveProject**Description**

SaveProject saves the current project.

### 15.3.2.24.5 SaveProjectAs

**Declaration:** SaveProjectAs (strPath as String, bDialog as Boolean)

#### Parameters

strPath

Full path with file name of new project file.

bDialog

If bDialog is TRUE, a file-dialog will be displayed.

#### Description

SaveProjectAs stores the project data into a new location.

### 15.3.2.25 SpyProjectItem

#### Methods

[Open](#) <sup>548</sup>

#### Properties

[ChildItems](#) <sup>547</sup>

[ParentItem](#) <sup>548</sup>

[FileExtensions](#) <sup>547</sup>

[ItemType](#) <sup>548</sup>

[Name](#) <sup>548</sup>

[Path](#) <sup>548</sup>

[ValidateWith](#) <sup>548</sup>

[XMLForXSLTransformation](#) <sup>549</sup>

[XSLForXMLTransformation](#) <sup>549</sup>

[XSLTransformationFileExtension](#) <sup>549</sup>

[XSLTransformationFolder](#) <sup>549</sup>

#### Description

SpyProjectItem Class

### 15.3.2.25.1 ChildItems

**Declaration:** ChildItems as [SpyProjectItems](#) <sup>549</sup>

#### Description

If the item is a folder, ChildItems is the collection of the folder content.

### 15.3.2.25.2 FileExtensions

**Declaration:** FileExtensions as String

#### Description

Used to set the file extensions if the project item is a folder.

### 15.3.2.25.3 ItemType

**Declaration:** ItemType as [SPYProjectItemTypes](#) <sup>611</sup>

#### Description

This property is read-only.

### 15.3.2.25.4 Name

**Declaration:** Name as String

#### Description

Name of the project item. This property is read-only.

### 15.3.2.25.5 Open

**Declaration:** Open as [Document](#) <sup>467</sup>

#### Return Value

The project item opened as document.

#### Description

Opens the project item.

### 15.3.2.25.6 ParentItem

**Declaration:** ParentItem as [SpyProjectItem](#) <sup>547</sup>

#### Description

Parent item of the current project item. Can be NULL (Nothing) if the project item is a top-level item.

### 15.3.2.25.7 Path

**Declaration:** Path as String

#### Description

Path of project item. This property is read-only.

### 15.3.2.25.8 ValidateWith

**Declaration:** ValidateWith as String

**Description**

Used to set the schema/DTD for validation.

### 15.3.2.25.9 XMLForXSLTransformation

**Declaration:** XMLForXSLTransformation as String

**Description**

Used to set the XML for XSL transformation.

### 15.3.2.25.10 XSLForXMLTransformation

**Declaration:** XSLForXMLTransformation as String

**Description**

Used to set the XSL for XML transformation.

### 15.3.2.25.11 XSLTransformationFileExtension

**Declaration:** XSLTransformationFileExtension as String

**Description**

Used to set the file extension for XSL transformation output files.

### 15.3.2.25.12 XSLTransformationFolder

**Declaration:** XSLTransformationFolder as String

**Description**

Used to set the destination folder for XSL transformation output files.

## 15.3.2.26 SpyProjectItems

**Methods**

[AddFile](#) <sup>550</sup>

[AddFolder](#) <sup>550</sup>

[AddURL](#) <sup>550</sup>

[RemoveItem](#) <sup>551</sup>

**Properties**

[Count](#) <sup>551</sup>

[Item](#) <sup>551</sup>

**Description**

SpyProjectItems Class

### 15.3.2.26.1 AddFile

**Declaration:** AddFile (*strPath* as String)

#### Parameters

*strPath*

Full path with file name of new project item

#### Description

The method adds a new file to the collection of project items.

### 15.3.2.26.2 AddFolder

**Declaration:** AddFolder (*strName* as String)

#### Parameters

*strName*

Name of the new folder.

#### Description

The method AddFolder adds a folder with the name *strName* to the collection of project items.

### 15.3.2.26.3 AddURL

**Declaration:** AddURL (*strURL* as String, *nURLType* as [SPYURLTypes](#)<sup>615</sup>, *strUser* as String, *strPassword* as String, *bSave* as Boolean)

#### Description

*strURL*

URL to open as document.

*nURLType*

Type of document to open. Set to -1 for auto detection.

*strUser*

Name of the user if required. Can be empty.

*strPassword*

Password for authentication. Can be empty.

*bSave*

Save user and password information.

#### Description

The method adds an URL item to the project collection.

#### 15.3.2.26.4 Count

**Declaration:** Count as long

##### Description

This property gets the count of project items in the collection. The property is read-only.

#### 15.3.2.26.5 Item

**Declaration:** Item (*n* as long) as [SpyProjectItem](#)<sup>547</sup>

##### Description

Retrieves the *n*-th element of the collection of project items. The first item has index 1.

#### 15.3.2.26.6 RemoveItem

**Declaration:** RemoveItem (*pltem* as [SpyProjectItem](#)<sup>547</sup>)

##### Description

RemoveItem deletes the item *pltem* from the collection of project items.

### 15.3.2.27 TextImportExportSettings

#### Properties for import only

[ImportFile](#)<sup>553</sup>

#### Properties for export only

[DestinationFolder](#)<sup>552</sup>

[FileExtension](#)<sup>553</sup>

[CommentIncluded](#)<sup>552</sup>

[RemoveDelimiter](#)<sup>553</sup>

[RemoveNewline](#)<sup>553</sup>

#### Properties for import and export

[HeaderRow](#)<sup>553</sup>

[FieldDelimiter](#)<sup>552</sup>

[EnclosingCharacter](#)<sup>552</sup>

[Encoding](#)<sup>552</sup>

[EncodingByteOrder](#)<sup>552</sup>

##### Description

TextImportExportSettings contains options common to text import and export functions.

### 15.3.2.27.1 CommentIncluded

**Property:** CommentIncluded as Boolean

#### Description

This property tells whether additional comments are added to the generated text file. Default is true. This property is used only when exporting to text files.

### 15.3.2.27.2 DestinationFolder

**Property:** DestinationFolder as String

#### Description

The property DestinationFolder sets the folder where the created files are saved during text export.

### 15.3.2.27.3 EnclosingCharacter

**Property:** EnclosingCharacter as [SPYTextEnclosing](#)<sup>614</sup>

#### Description

This property defines the character that encloses all field values for import and export. Default is [spyNoEnclosing](#)<sup>614</sup>.

### 15.3.2.27.4 Encoding

**Property:** Encoding as String

#### Description

The property Encoding sets the character encoding for the text files for importing and exporting.

### 15.3.2.27.5 EncodingByteOrder

**Property:** EncodingByteOrder as [SPYEncodingByteOrder](#)<sup>607</sup>

#### Description

The property EncodingByteOrder sets the byte order for Unicode characters. Default is [spyNONE](#)<sup>607</sup>.

### 15.3.2.27.6 FieldDelimiter

**Property:** FieldDelimiter as [SPYTextDelimiters](#)<sup>614</sup>

#### Description

The property `FieldDelimiter` defines the delimiter between the fields during import and export. Default is [spyTabulator<sup>614</sup>](#).

#### 15.3.2.27.7 FileExtension

**Property:** FileExtension as String

##### Description

This property sets the file extension for files created on text export.

#### 15.3.2.27.8 HeaderRow

**Property:** HeaderRow as Boolean

##### Description

The property `HeaderRow` is used during import and export. Set `HeaderRow` true on import, if the first line of the text file contains the names of the columns. Set `HeaderRow` true on export, if the first line in the created text files should contain the name of the columns. Default value is true.

#### 15.3.2.27.9 ImportFile

**Property:** ImportFile as String

##### Description

This property is used to set the text file for import. The string has to be a full qualified path.

#### 15.3.2.27.10 RemoveDelimiter

**Property:** RemoveDelimiter as Boolean

##### Description

The property `RemoveDelimiter` defines whether characters in the text that are equal to the delimiter character are removed. Default is false. This property is used only when exporting to text files.

#### 15.3.2.27.11 RemoveNewline

**Property:** RemoveNewline as Boolean

##### Description

The property `RemoveNewline` defines whether newline characters in the text are removed. Default is false. This property is used only when exporting to text files.

## 15.3.2.28 TextView

### Properties and Methods

[Application](#) <sup>555</sup>  
[Parent](#) <sup>557</sup>  
  
[LineFromPosition](#) <sup>557</sup>  
[PositionFromLine](#) <sup>558</sup>  
[LineLength](#) <sup>557</sup>  
[SelText](#) <sup>559</sup>  
[GetRangeText](#) <sup>556</sup>  
[ReplaceText](#) <sup>558</sup>  
[MoveCaret](#) <sup>557</sup>  
[GoToLineChar](#) <sup>556</sup>  
[SelectText](#) <sup>559</sup>  
[SelectionStart](#) <sup>558</sup>  
[SelectionEnd](#) <sup>558</sup>  
[Text](#) <sup>559</sup>  
[LineCount](#) <sup>556</sup>  
[Length](#) <sup>556</sup>

### Description

#### 15.3.2.28.1 Events

##### 15.3.2.28.1.1 OnBeforeShowSuggestions

**Event:** OnBeforeShowSuggestions() as Boolean

#### Description

This event gets fired before a suggestion window is shown. The [Document](#) <sup>467</sup> property [Suggestions](#) <sup>494</sup> contains a string array that is recommended to the user. It is possible to modify the displayed recommendations during this event. Before doing so you have to assign an empty array to the [Suggestions](#) <sup>494</sup> property. The best location for this is the [OnDocumentOpened](#) <sup>384</sup> event. To prevent the suggestion window to show up return false and true to continue its display.

#### Examples

Given below are examples of how this event can be scripted.

##### **XMLSpy scripting environment - VBScript:**

```
Function On_BeforeShowSuggestions()
End Function
```

##### **XMLSpy scripting environment - JScript:**

```
function On_BeforeShowSuggestions()
{
}
```

**XMLSpy IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (33, ...) // nEventId = 33
```

**15.3.2.28.1.2 OnChar**

**Event:** OnChar(nChar as Long, bExistSuggestion as Boolean) as Boolean

**Description**

This event gets fired on each key stroke. The parameter nChar is the key that was pressed and bExistSuggestions tells whether a Authentic Desktop generated suggestions window is displayed after this key. The [Document](#)<sup>467</sup> property [Suggestions](#)<sup>494</sup> contains a string array that is recommended to the user. It is possible to modify the displayed recommendations during this event. Before doing so you have to assign an empty array to the [Suggestions](#)<sup>494</sup> property. The best location for this is the [OnDocumentOpened](#)<sup>384</sup> event. To prevent the suggestion window to show up return false and true to continue its display. It is also possible to create a new suggestions window when none is provided by Authentic Desktop. Set the [Document](#)<sup>467</sup> property [Suggestions](#)<sup>494</sup> to a string array with your recommendations and return true. This event is fired before the [OnBeforeShowSuggestions](#)<sup>554</sup> event. If you prevent to show the suggestion window by returning false then [OnBeforeShowSuggestions](#)<sup>554</sup> is not fired.

**Examples**

Given below are examples of how this event can be scripted.

**XMLSpy scripting environment - VBScript:**

```
Function On_Char(nChar, bExistSuggestions)
End Function
```

**XMLSpy scripting environment - JScript:**

```
function On_Char(nChar, bExistSuggestions)
{
}
}
```

**XMLSpy IDE Plugin:**

```
IXMLSpyPlugIn.OnEvent (35, ...) // nEventId = 35
```

**15.3.2.28.2 Application**

**Property:** Application as [Application](#)<sup>382</sup> (read-only)

**Description**

Access the Authentic Desktop application object.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.28.3 GetRangeText

**Method:** GetRangeText(nStart as Long, nEnd as Long) as String

**Description**

Returns the text in the specified range.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.28.4 GoToLineChar

**Method:** GoToLineChar(nLine as Long, nChar as Long)

**Description**

Moves the caret to the specified line and character position.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.28.5 Length

**Property:** Length as Long

**Description**

Returns the character count of the document.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.28.6 LineCount

**Property:** LineCount as Long

**Description**

Returns the number of lines in the document.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.28.7 LineFromPosition

**Method:** LineFromPosition(nCharPos as Long) as Long

#### Description

Returns the line number of the character position.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.28.8 LineLength

**Method:** LineLength(nLine as Long) as Long

#### Description

Returns the length of the line.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.28.9 MoveCaret

**Method:** MoveCaret(nDiff as Long)

#### Description

Moves the caret nDiff characters.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.28.10 Parent

**Property:** Parent as [Document](#)<sup>467</sup> (read-only)

#### Description

Access the parent of the object.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.28.11 PositionFromLine

**Method:** PositionFromLine(nLine as Long) as Long

**Description**

Returns the start position of the line.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.28.12 ReplaceText

**Method:** ReplaceText(nPosFrom as Long, nPosTill as Long, sText as String)

**Description**

Replaces the text in the specified range.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.28.13 SelectionEnd

**Property:** SelectionEnd as Long

**Description**

Returns/sets the text selection end position.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.28.14 SelectionStart

**Property:** SelectionStart as Long

**Description**

Returns/sets the text selection start position.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.28.15 SelectText

**Method:** SelectText(nPosFrom as Long, nPosTill as Long)

#### Description

Selects the text in the specified range.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.28.16 SelText

**Property:** SelText as String

#### Description

Returns/sets the selected text.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.28.17 Text

**Property:** Text as String

#### Description

Returns/sets the document text.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

## 15.3.2.29 WSDLDocumentationDlg

### Properties and Methods

Standard automation properties

[Application](#) <sup>561</sup>

[Parent](#) <sup>566</sup>

Interaction and visibility properties

[GlobalElementsAndTypesOnly](#) <sup>562</sup>

[OptionsDialogAction](#) <sup>565</sup>

[OutputFile](#) <sup>565</sup>

[OutputFileDialogAction](#) <sup>566</sup>

[SeparateSchemaDocument](#) <sup>566</sup>

[ShowProgressBar](#) <sup>568</sup>  
[ShowResult](#) <sup>569</sup>

#### Document generation options and methods

[OutputFormat](#) <sup>566</sup>  
[UseFixedDesign](#) <sup>570</sup>  
[SPSFile](#) <sup>570</sup>  
[EmbedDiagrams](#) <sup>562</sup>  
[DiagramFormat](#) <sup>561</sup>  
[MultipleOutputFiles](#) <sup>565</sup>  
[EmbedCSSInHTML](#) <sup>562</sup>  
[CreateDiagramsFolder](#) <sup>561</sup>

[IncludeAll](#) <sup>562</sup>  
[IncludeBinding](#) <sup>563</sup>  
[IncludeImportedWSDLFiles](#) <sup>563</sup>  
[IncludeMessages](#) <sup>563</sup>  
[IncludeOverview](#) <sup>564</sup>  
[IncludePortType](#) <sup>564</sup>  
[IncludeService](#) <sup>564</sup>  
[IncludeTypes](#) <sup>564</sup>

[AllDetails](#) <sup>560</sup>  
[ShowBindingDiagram](#) <sup>567</sup>  
[ShowExtensibility](#) <sup>567</sup>  
[ShowMessageParts](#) <sup>567</sup>  
[ShowPort](#) <sup>568</sup>  
[ShowPortTypeDiagram](#) <sup>568</sup>  
[ShowPortTypeOperations](#) <sup>568</sup>  
[ShowServiceDiagram](#) <sup>569</sup>  
[ShowSourceCode](#) <sup>569</sup>  
[ShowTypesDiagram](#) <sup>570</sup>  
[ShowUsedBy](#) <sup>570</sup>

#### Description

This object combines all options for WSDL document generation as they are available through user interface dialog boxes in Authentic Desktop. The document generation options are initialized with the values used during the last generation of WSDL documentation. However, before using the object you have to set the [OutputFile](#) <sup>565</sup> property to a valid file path. Use [OptionsDialogAction](#) <sup>565</sup>, [OutputFileDialogAction](#) <sup>565</sup> and [ShowProgressBar](#) <sup>568</sup> to specify the level of user interaction desired. You can use [IncludeAll](#) <sup>562</sup> and [AllDetails](#) <sup>560</sup> to set whole option groups at once or the individual properties to operate on a finer granularity.

#### 15.3.2.29.1 AllDetails

**Method:** AllDetails (i\_bDetailsOn as Boolean)

#### Description

Use this method to turn all details options on or off.

**Errors**

4300 The object is no longer valid.

### 15.3.2.29.2 Application

**Property:** Application as [Application](#)<sup>382</sup> (read-only)

**Description**

Access the Authentic Desktop application object.

**Errors**

3900 The object is no longer valid.  
3901 Invalid address for the return parameter was specified.

### 15.3.2.29.3 CreateDiagramsFolder

**Property:** CreateDiagramsFolder as Boolean

**Description**

Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is false.

**Errors**

3900 The object is no longer valid.  
3901 Invalid address for the return parameter was specified.

### 15.3.2.29.4 DiagramFormat

**Property:** DiagramFormat as [SPYImageKind](#)<sup>608</sup>

**Description**

This property specifies the generated diagram image type. This property is not available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is PNG.

**Errors**

3900 The object is no longer valid.  
3901 Invalid address for the return parameter was specified.

### 15.3.2.29.5 EmbedCSSInHTML

**Property:** EmbedCSSInHTML as Boolean

#### Description

Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.6 EmbedDiagrams

**Property:** EmbedDiagrams as Boolean

#### Description

Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.7 GlobalElementsAndTypesOnly

**Property:** GlobalElementsAndTypesOnly as Boolean

#### Description

Returns/sets a value indicating whether a full Schema documentation is done or only Global Elements and Types are documented.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.8 IncludeAll

**Method:** IncludeAll (i\_bInclude as Boolean)

#### Description

Use this method to mark or unmark all include options.

**Errors**

4300 The object is no longer valid.

### 15.3.2.29.9 IncludeBinding

**Property:** IncludeBinding as Boolean

**Description**

Set this property to true, to include bindings in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

**Errors**

3900 The object is no longer valid.  
3901 Invalid address for the return parameter was specified.

### 15.3.2.29.10 IncludeImportedWSDLFiles

**Property:** IncludeImportedWSDLFiles as Boolean

**Description**

Set this property to true, to include imported WSDL files in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

**Errors**

3900 The object is no longer valid.  
3901 Invalid address for the return parameter was specified.

### 15.3.2.29.11 IncludeMessages

**Property:** IncludeMessages as Boolean

**Description**

Set this property to true, to include messages in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

**Errors**

3900 The object is no longer valid.  
3901 Invalid address for the return parameter was specified.

### 15.3.2.29.12 IncludeOverview

**Property:** IncludeOverview as Boolean

#### Description

Set this property to true, to include an overview in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.13 IncludePortType

**Property:** IncludePortType as Boolean

#### Description

Set this property to true, to include port types in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.14 IncludeService

**Property:** IncludeService as Boolean

#### Description

Set this property to true, to include services in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.15 IncludeTypes

**Property:** IncludeTypes as Boolean

#### Description

Set this property to true, to include types in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.16 MultipleOutputFiles

**Property:** MultipleOutputFiles as Boolean

**Description**

Set this property to true, to split the documentation files. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is false.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid value has been used to set the property.  
Invalid address for the return parameter was specified.

### 15.3.2.29.17 OptionsDialogAction

**Property:** OptionsDialogAction as [SPYDialogAction](#)<sup>607</sup>

**Description**

To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid value has been used to set the property.  
Invalid address for the return parameter was specified.

### 15.3.2.29.18 OutputFile

**Property:** OutputFile as String

**Description**

Full path and name of the file that will contain the generated documentation. In case of HTML output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.19 OutputFileDialogAction

**Property:** OutputFileDialogAction as [SPYDialogAction](#) <sup>607</sup>

#### Description

To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in [OutputFile](#) <sup>565</sup> should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid value has been used to set the property.  
Invalid address for the return parameter was specified.

### 15.3.2.29.20 OutputFormat

**Property:** OutputFormat as [SPYSchemaDocumentationFormat](#) <sup>613</sup>

#### Description

Defines the kind of documentation that will be generated: HTML (value=0), MS-Word (value=1), or RTF (value=2). The property gets initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#) <sup>484</sup>. The default for the first run is HTML.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid value has been used to set the property.  
Invalid address for the return parameter was specified.

### 15.3.2.29.21 Parent

**Property:** Parent as [Dialogs](#) <sup>464</sup> (read-only)

#### Description

Access the parent of the object.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.22 SeparateSchemaDocument

**Property:** SeparateSchemaDocument as Boolean

#### Description

Returns/sets a value indicating whether the Schema documentation should be placed in a separate document.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.23 ShowBindingDiagram

**Property:** ShowBindingDiagram as Boolean

**Description**

Set this property to true, to show binding diagrams in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.24 ShowExtensibility

**Property:** ShowExtensibility as Boolean

**Description**

Set this property to true, to show service and binding extensibilities in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.25 ShowMessageParts

**Property:** ShowMessageParts as Boolean

**Description**

Set this property to true, to show message parts of messages in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.26 ShowPort

**Property:** ShowPort as Boolean

#### Description

Set this property to true, to show service ports in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.27 ShowPortTypeDiagram

**Property:** ShowPortTypeDiagram as Boolean

#### Description

Set this property to true, to show port type diagrams in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.28 ShowPortTypeOperations

**Property:** ShowPortTypeOperations as Boolean

#### Description

Set this property to true, to show port type operations in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.29 ShowProgressBar

**Property:** ShowProgressBar as Boolean

#### Description

Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.30 ShowResult

**Property:** ShowResult as Boolean

**Description**

Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in Authentic Desktop. To show Word documentation, MS-Word will be started. The property gets initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.31 ShowServiceDiagram

**Property:** ShowServiceDiagram as Boolean

**Description**

Set this property to true, to show service diagrams in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.32 ShowSourceCode

**Property:** ShowSourceCode as Boolean

**Description**

Set this property to true, to show source code for the includes in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.33 ShowTypesDiagram

**Property:** ShowTypesDiagram as Boolean

#### Description

Set this property to true, to show type diagrams in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.34 ShowUsedBy

**Property:** ShowUsedBy as Boolean

#### Description

Set this property to true, to show the used-by relation for types, bindings and messages definitions in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#)<sup>484</sup>. The default for the first run is true.

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.35 UseFixedDesign

**Property:** UseFixedDesign as Boolean

#### Description

Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

#### Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.29.36 SPSFile

**Property:** SPSFile as String

#### Description

Full path and name of the SPS file that will be used to generate the documentation.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.30 WSDL20DocumentationDlg

**Properties and Methods**

## Standard automation properties

[Application](#) <sup>572</sup>  
[Parent](#) <sup>577</sup>

## Interaction and visibility properties

[GlobalElementsAndTypesOnly](#) <sup>573</sup>  
[OptionsDialogAction](#) <sup>576</sup>  
[OutputFile](#) <sup>576</sup>  
[OutputFileDialogAction](#) <sup>577</sup>  
[SeparateSchemaDocument](#) <sup>577</sup>  
[ShowProgressBar](#) <sup>579</sup>  
[ShowResult](#) <sup>580</sup>

## Document generation options and methods

[OutputFormat](#) <sup>577</sup>  
[UseFixedDesign](#) <sup>581</sup>  
[SPSFile](#) <sup>581</sup>  
[EmbedDiagrams](#) <sup>573</sup>  
[DiagramFormat](#) <sup>573</sup>  
[MultipleOutputFiles](#) <sup>576</sup>  
[EmbedCSSInHTML](#) <sup>573</sup>  
[CreateDiagramsFolder](#) <sup>572</sup>

[IncludeAll](#) <sup>574</sup>  
[IncludeBinding](#) <sup>574</sup>  
[IncludeImportedWSDLFiles](#) <sup>574</sup>  
[IncludeInterface](#) <sup>575</sup>  
[IncludeOverview](#) <sup>575</sup>  
[IncludeService](#) <sup>575</sup>  
[IncludeTypes](#) <sup>575</sup>

[AllDetails](#) <sup>572</sup>  
[ShowBindingDiagram](#) <sup>578</sup>  
[ShowExtensibility](#) <sup>578</sup>  
[ShowEndpoint](#) <sup>578</sup>  
[ShowFault](#) <sup>579</sup>  
[ShowInterfaceDiagram](#) <sup>579</sup>  
[ShowOperation](#) <sup>579</sup>  
[ShowServiceDiagram](#) <sup>580</sup>  
[ShowSourceCode](#) <sup>580</sup>  
[ShowTypesDiagram](#) <sup>581</sup>  
[ShowUsedBy](#) <sup>581</sup>

**Description**

This object combines all options for WSDL document generation as they are available through user interface dialog boxes in Authentic Desktop. The document generation options are initialized with the values used during the last generation of WSDL documentation. However, before using the object you have to set the [OutputFile](#)<sup>576</sup> property to a valid file path. Use [OptionsDialogAction](#)<sup>576</sup>, [OutputFileDialogAction](#)<sup>577</sup> and [ShowProgressBar](#)<sup>579</sup> to specify the level of user interaction desired. You can use [IncludeAll](#)<sup>574</sup> and [AllDetails](#)<sup>572</sup> to set whole option groups at once or the individual properties to operate on a finer granularity.

### 15.3.2.30.1 AllDetails

**Method:** AllDetails (i\_bDetailsOn as Boolean)

#### Description

Use this method to turn all details options on or off.

#### Errors

4300 The object is no longer valid.

### 15.3.2.30.2 Application

**Property:** Application as [Application](#)<sup>382</sup> (read-only)

#### Description

Access the Authentic Desktop application object.

#### Errors

4300 The object is no longer valid.  
4301 Invalid address for the return parameter was specified.

### 15.3.2.30.3 CreateDiagramsFolder

**Property:** CreateDiagramsFolder as Boolean

#### Description

Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is false.

#### Errors

4300 The object is no longer valid.  
4301 Invalid address for the return parameter was specified.

#### 15.3.2.30.4 DiagramFormat

**Property:** DiagramFormat as [SPYImageKind](#) <sup>608</sup>

##### Description

This property specifies the generated diagram image type. This property is not available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#) <sup>484</sup>. The default for the first run is PNG.

##### Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

#### 15.3.2.30.5 EmbedCSSInHTML

**Property:** EmbedCSSInHTML as Boolean

##### Description

Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#) <sup>484</sup>. The default for the first run is true.

##### Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

#### 15.3.2.30.6 EmbedDiagrams

**Property:** EmbedDiagrams as Boolean

##### Description

Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#) <sup>484</sup>. The default for the first run is true.

##### Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

#### 15.3.2.30.7 GlobalElementsAndTypesOnly

**Property:** GlobalElementsAndTypesOnly as Boolean

**Description**

Returns/sets a value indicating whether a full Schema documentation is done or only Global Elements and Types are documented.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.30.8 IncludeAll

**Method:** IncludeAll (i\_bInclude as Boolean)

**Description**

Use this method to mark or unmark all include options.

**Errors**

- 4300 The object is no longer valid.

### 15.3.2.30.9 IncludeBinding

**Property:** IncludeBinding as Boolean

**Description**

Set this property to true, to include bindings in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is true.

**Errors**

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.10 IncludeImportedWSDLFiles

**Property:** IncludeImportedWSDLFiles as Boolean

**Description**

Set this property to true, to include imported WSDL files in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is true.

**Errors**

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.11 IncludeInterface

**Property:** IncludeInterface as Boolean

#### Description

Set this property to true, to include interfaces in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is true.

#### Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.12 IncludeOverview

**Property:** IncludeOverview as Boolean

#### Description

Set this property to true, to include an overview in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is true.

#### Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.13 IncludeService

**Property:** IncludeService as Boolean

#### Description

Set this property to true, to include services in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is true.

#### Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.14 IncludeTypes

**Property:** IncludeTypes as Boolean

#### Description

Set this property to true, to include types in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is true.

**Errors**

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.15 MultipleOutputFiles

**Property:** MultipleOutputFiles as Boolean

**Description**

Set this property to true, to split the documentation files. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is false.

**Errors**

- 4300 The object is no longer valid.
- 4301 Invalid value has been used to set the property.  
Invalid address for the return parameter was specified.

### 15.3.2.30.16 OptionsDialogAction

**Property:** OptionsDialogAction as [SPYDialogAction](#)<sup>607</sup>

**Description**

To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

**Errors**

- 4300 The object is no longer valid.
- 4301 Invalid value has been used to set the property.  
Invalid address for the return parameter was specified.

### 15.3.2.30.17 OutputFile

**Property:** OutputFile as String

**Description**

Full path and name of the file that will contain the generated documentation. In case of HTML output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>.

**Errors**

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.18 OutputFileDialogAction

**Property:** OutputFileDialogAction as [SPYDialogAction](#)<sup>607</sup>

#### Description

To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in [OutputFile](#)<sup>576</sup> should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

#### Errors

- 4300 The object is no longer valid.
- 4301 Invalid value has been used to set the property.  
Invalid address for the return parameter was specified.

### 15.3.2.30.19 OutputFormat

**Property:** OutputFormat as [SPYSchemaDocumentationFormat](#)<sup>613</sup>

#### Description

Defines the kind of documentation that will be generated: HTML (value=0), MS-Word (value=1), or RTF (value=2). The property gets initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is HTML.

#### Errors

- 4300 The object is no longer valid.
- 4301 Invalid value has been used to set the property.  
Invalid address for the return parameter was specified.

### 15.3.2.30.20 Parent

**Property:** Parent as [Dialogs](#)<sup>464</sup> (read-only)

#### Description

Access the parent of the object.

#### Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.21 SeparateSchemaDocument

**Property:** SeparateSchemaDocument as Boolean

#### Description

Returns/sets a value indicating whether the Schema documentation should be placed in a separate document.

**Errors**

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

### 15.3.2.30.22 ShowBindingDiagram

**Property:** ShowBindingDiagram as Boolean

**Description**

Set this property to true, to show binding diagrams in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is true.

**Errors**

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.23 ShowEndpoint

**Property:** ShowEndpoint as Boolean

**Description**

Set this property to true, to show service endpoints in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is true.

**Errors**

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.24 ShowExtensibility

**Property:** ShowExtensibility as Boolean

**Description**

Set this property to true, to show service and binding extensibilities in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is true.

**Errors**

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.25 ShowFault

**Property:** ShowFault as Boolean

#### Description

Set this property to true, to show faults in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is true.

#### Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.26 ShowInterfaceDiagram

**Property:** ShowInterfaceDiagram as Boolean

#### Description

Set this property to true, to show interface diagrams in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is true.

#### Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.27 ShowOperation

**Property:** ShowOperation as Boolean

#### Description

Set this property to true, to show interface and binding operations in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is true.

#### Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.28 ShowProgressBar

**Property:** ShowProgressBar as Boolean

#### Description

Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

**Errors**

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.29 ShowResult

**Property:** ShowResult as Boolean

**Description**

Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in Authentic Desktop. To show Word documentation, MS-Word will be started. The property gets initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is true.

**Errors**

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.30 ShowServiceDiagram

**Property:** ShowServiceDiagram as Boolean

**Description**

Set this property to true, to show service diagrams in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is true.

**Errors**

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.31 ShowSourceCode

**Property:** ShowSourceCode as Boolean

**Description**

Set this property to true, to show source code for the includes in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is true.

**Errors**

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.32 ShowTypesDiagram

**Property:** ShowTypesDiagram as Boolean

#### Description

Set this property to true, to show type diagrams in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is true.

#### Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.33 ShowUsedBy

**Property:** ShowUsedBy as Boolean

#### Description

Set this property to true, to show the used-by relation for types, bindings and messages definitions in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#)<sup>484</sup>. The default for the first run is true.

#### Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.34 SPSFile

**Property:** SPSFile as String

#### Description

Full path and name of the SPS file that will be used to generate the documentation.

#### Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.30.35 UseFixedDesign

**Property:** UseFixedDesign as Boolean

#### Description

Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

**Errors**

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

### 15.3.2.31 XBRLDocumentationDlg

**Properties and Methods**

Standard automation properties

[Application](#) <sup>583</sup>  
[Parent](#) <sup>588</sup>

Interaction and visibility properties

[OptionsDialogAction](#) <sup>586</sup>  
[OutputFile](#) <sup>587</sup>  
[OutputFileDialogAction](#) <sup>587</sup>  
[ShowProgressBar](#) <sup>591</sup>  
[ShowResult](#) <sup>591</sup>

Document generation options and methods

[OutputFormat](#) <sup>587</sup>  
[UseFixedDesign](#) <sup>592</sup>  
[SPSFile](#) <sup>592</sup>  
[EmbedDiagrams](#) <sup>584</sup>  
[DiagramFormat](#) <sup>584</sup>  
[EmbedCSSInHTML](#) <sup>584</sup>  
[CreateDiagramsFolder](#) <sup>583</sup>

[IncludeAll](#) <sup>585</sup>  
[IncludeOverview](#) <sup>586</sup>  
[IncludeNamespacePrefixes](#) <sup>586</sup>  
[IncludeGlobalElements](#) <sup>585</sup>  
[IncludeDefinitionLinkroles](#) <sup>585</sup>  
[IncludePresentationLinkroles](#) <sup>586</sup>  
[IncludeCalculationLinkroles](#) <sup>585</sup>

[AllDetails](#) <sup>583</sup>  
[ShowDiagram](#) <sup>589</sup>  
[ShowSubstitutiongroup](#) <sup>592</sup>  
[ShowItemtype](#) <sup>589</sup>  
[ShowBalance](#) <sup>589</sup>  
[ShowPeriod](#) <sup>591</sup>  
[ShowAbstract](#) <sup>588</sup>  
[ShowNillable](#) <sup>590</sup>  
[ShowLabels](#) <sup>590</sup>  
[ShowReferences](#) <sup>591</sup>  
[ShowLinkbaseReferences](#) <sup>590</sup>

[ShortQualifiedName](#) <sup>588</sup>

[ShowImportedElements](#) <sup>589</sup>**Description**

This object combines all options for XBRL document generation as they are available through user interface dialog boxes in Authentic Desktop. The document generation options are initialized with the values used during the last generation of XBRL documentation. However, before using the object you have to set the [OutputFile](#) <sup>587</sup> property to a valid file path. Use [OptionsDialogAction](#) <sup>586</sup>, [OutputFileDialogAction](#) <sup>587</sup> and [ShowProgressBar](#) <sup>591</sup> to specify the level of user interaction desired. You can use [IncludeAll](#) <sup>585</sup> and [AllDetails](#) <sup>583</sup> to set whole option groups at once or the individual properties to operate on a finer granularity.

### 15.3.2.31.1 AllDetails

**Method:** AllDetails (i\_bDetailsOn as Boolean)

**Description**

Use this method to turn all details options on or off.

**Errors**

4400 The object is no longer valid.

### 15.3.2.31.2 Application

**Property:** Application as [Application](#) <sup>382</sup> (read-only)

**Description**

Access the Authentic Desktop application object.

**Errors**

4400 The object is no longer valid.  
4401 Invalid address for the return parameter was specified.

### 15.3.2.31.3 CreateDiagramsFolder

**Property:** CreateDiagramsFolder as Boolean

**Description**

Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#) <sup>485</sup>. The default for the first run is false.

**Errors**

4400 The object is no longer valid.

4401 Invalid address for the return parameter was specified.

#### 15.3.2.31.4 DiagramFormat

**Property:** DiagramFormat as [SPYImageKind](#)<sup>608</sup>

##### Description

This property specifies the generated diagram image type. This property is not available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is PNG.

##### Errors

4400 The object is no longer valid.  
4401 Invalid address for the return parameter was specified.

#### 15.3.2.31.5 EmbedCSSInHTML

**Property:** EmbedCSSInHTML as Boolean

##### Description

Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

##### Errors

4400 The object is no longer valid.  
4401 Invalid address for the return parameter was specified.

#### 15.3.2.31.6 EmbedDiagrams

**Property:** EmbedDiagrams as Boolean

##### Description

Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

##### Errors

4400 The object is no longer valid.  
4401 Invalid address for the return parameter was specified.

### 15.3.2.31.7 IncludeAll

**Method:** IncludeAll (i\_bInclude as Boolean)

#### Description

Use this method to mark or unmark all include options.

#### Errors

4400 The object is no longer valid.

### 15.3.2.31.8 IncludeCalculationLinkroles

**Property:** IncludeCalculationLinkroles as Boolean

#### Description

Set this property to true, to include calculation linkroles in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

#### Errors

4400 The object is no longer valid.  
4401 Invalid address for the return parameter was specified.

### 15.3.2.31.9 IncludeDefinitionLinkroles

**Property:** IncludeDefinitionLinkroles as Boolean

#### Description

Set this property to true, to include definition linkroles in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

#### Errors

4400 The object is no longer valid.  
4401 Invalid address for the return parameter was specified.

### 15.3.2.31.10 IncludeGlobalElements

**Property:** IncludeGlobalElements as Boolean

#### Description

Set this property to true, to include global elements in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

**Errors**

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.11 IncludeNamespacePrefixes

**Property:** IncludeNamespacePrefixes as Boolean

**Description**

Set this property to true, to include namespace prefixes in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

**Errors**

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.12 IncludeOverview

**Property:** IncludeOverview as Boolean

**Description**

Set this property to true, to include an overview in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

**Errors**

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.13 IncludePresentationLinkroles

**Property:** IncludePresentationLinkroles as Boolean

**Description**

Set this property to true, to include presentation linkroles in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

**Errors**

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.14 OptionsDialogAction

**Property:** OptionsDialogAction as [SPYDialogAction](#)<sup>607</sup>

**Description**

To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

**Errors**

- 4400 The object is no longer valid.
- 4401 Invalid value has been used to set the property.  
Invalid address for the return parameter was specified.

### 15.3.2.31.15 OutputFile

**Property:** OutputFile as String

**Description**

Full path and name of the file that will contain the generated documentation. In case of HTML output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>.

**Errors**

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.16 OutputFileDialogAction

**Property:** OutputFileDialogAction as [SPYDialogAction](#)<sup>607</sup>

**Description**

To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in [OutputFile](#)<sup>587</sup> should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

**Errors**

- 4400 The object is no longer valid.
- 4401 Invalid value has been used to set the property.  
Invalid address for the return parameter was specified.

### 15.3.2.31.17 OutputFormat

**Property:** OutputFormat as [SPYSchemaDocumentationFormat](#)<sup>613</sup>

**Description**

Defines the kind of documentation that will be generated: HTML (value=0), MS-Word (value=1), or RTF (value=2). The property gets initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is HTML.

**Errors**

- 4400 The object is no longer valid.
- 4401 Invalid value has been used to set the property.  
Invalid address for the return parameter was specified.

### 15.3.2.31.18 Parent

**Property:** Parent as [Dialogs](#)<sup>464</sup> (read-only)

#### Description

Access the parent of the object.

#### Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.19 ShortQualifiedName

**Property:** ShortQualifiedName as Boolean

#### Description

Set this property to true, to use short qualified names in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

#### Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.20 ShowAbstract

**Property:** ShowAbstract as Boolean

#### Description

Set this property to true, to show abstracts in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

#### Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.21 ShowBalance

**Property:** ShowBalance as Boolean

#### Description

Set this property to true, to show balances in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

#### Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.22 ShowDiagram

**Property:** ShowDiagram as Boolean

#### Description

Set this property to true, to show diagrams in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

#### Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.23 ShowImportedElements

**Property:** ShowImportedElements as Boolean

#### Description

Set this property to true, to show imported elements in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

#### Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.24 ShowItemtype

**Property:** ShowItemtype as Boolean

#### Description

Set this property to true, to show item types in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

**Errors**

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.25 ShowLabels

**Property:** ShowLabels as Boolean

**Description**

Set this property to true, to show labels in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

**Errors**

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.26 ShowLinkbaseReferences

**Property:** ShowLinkbaseReferences as Boolean

**Description**

Set this property to true, to show linkbase references in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

**Errors**

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.27 ShowNillable

**Property:** ShowNillable as Boolean

**Description**

Set this property to true, to show nillable properties in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

**Errors**

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.28 ShowPeriod

**Property:** ShowPeriod as Boolean

#### Description

Set this property to true, to show periods in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

#### Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.29 ShowProgressBar

**Property:** ShowProgressBar as Boolean

#### Description

Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

#### Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.30 ShowReferences

**Property:** ShowReferences as Boolean

#### Description

Set this property to true, to show references in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

#### Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.31 ShowResult

**Property:** ShowResult as Boolean

#### Description

Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in Authentic Desktop. To show Word documentation, MS-Word will be started. The property gets initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

**Errors**

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.32 ShowSubstitutiongroup

**Property:** ShowSubstitutiongroup as Boolean

**Description**

Set this property to true, to show substitution groups in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#)<sup>485</sup>. The default for the first run is true.

**Errors**

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.33 SPSFile

**Property:** SPSFile as String

**Description**

Full path and name of the SPS file that will be used to generate the documentation.

**Errors**

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.31.34 UseFixedDesign

**Property:** UseFixedDesign as Boolean

**Description**

Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

**Errors**

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

### 15.3.2.32 XMLData

#### Properties

[Kind](#)<sup>602</sup>

[Name](#)<sup>603</sup>

[TextValue](#)<sup>604</sup>

[HasChildren](#)<sup>600</sup>

[MayHaveChildren](#)<sup>602</sup>

[Parent](#)<sup>603</sup>

#### Methods

[GetFirstChild](#)<sup>598</sup>

[GetNextChild](#)<sup>599</sup>

[GetCurrentChild](#)<sup>598</sup>

[InsertChild](#)<sup>601</sup>

[InsertChildAfter](#)<sup>601</sup>

[InsertChildBefore](#)<sup>602</sup>

[AppendChild](#)<sup>594</sup>

[EraseAllChildren](#)<sup>595</sup>

[EraseChild](#)<sup>595</sup>

[EraseCurrentChild](#)<sup>596</sup>

[IsSameNode](#)<sup>602</sup>

[CountChildren](#)<sup>594</sup>

[CountChildrenKind](#)<sup>595</sup>

[GetChild](#)<sup>596</sup>

[GetChildAttribute](#)<sup>597</sup>

[GetChildElement](#)<sup>597</sup>

[GetChildKind](#)<sup>597</sup>

[GetNamespacePrefixForURI](#)<sup>599</sup>

[HasChildrenKind](#)<sup>601</sup>

[SetTextValueXMLEncoded](#)<sup>603</sup>

#### 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](#)<sup>603</sup> 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.

### 15.3.2.32.1 AppendChild

**Declaration:** AppendChild (pNewData as [XMLData](#)<sup>593</sup>)

#### 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
```

### 15.3.2.32.2 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.

### 15.3.2.32.3 CountChildrenKind

**Declaration:** CountChildrenKind (*nKind* as [SPYXMLDataKind](#)<sup>617</sup>) 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.

### 15.3.2.32.4 EraseAllChildren

**Declaration:** EraseAllChildren

#### Description

EraseAllChildren deletes all associated children of the XMLData object.

#### Errors

1500 The XMLData object is no longer valid.

1900 Document must not be modified

#### Example

The sample erases all elements of the active document.

```
Dim objCurrentParent As XMLData
```

```
Set objCurrentParent = objSpy.ActiveDocument.RootElement  
objCurrentParent.EraseAllChildren
```

### 15.3.2.32.5 EraseChild

**Method:** EraseChild (Child as [XMLData](#)<sup>593</sup>)

#### Description

Deletes the given child node.

#### Errors

1500 Invalid object.

1506 Invalid input xml

1510 Invalid parameter.

### 15.3.2.32.6 EraseCurrentChild

**Declaration:** EraseCurrentChild

#### Description

EraseCurrentChild deletes the current XMLData child object. Before you call EraseCurrentChild you must initialize an internal iterator with [XMLData.GetFirstChild](#)<sup>593</sup>. After deleting the current child, EraseCurrentChild increments the internal iterator of the XMLData element. No error is returned when the last child gets erased and the iterator is moved past the end of the child list. The next call to EraseCurrentChild however, will return error 1503.

#### Errors

- 1500 The XMLData object is no longer valid.
- 1503 No iterator is initialized for this XMLData object, or the iterator points past the last child.
- 1900 Document must not be modified

#### Examples

```
// -----
// XMLSpy scripting environment - JScript
// erase all children of XMLData
// -----
// let's get an XMLData element, we assume that the
// cursor selects the parent of a list in grid view
var objList = Application.ActiveDocument.GridView.CurrentFocus;

// the following line would be shorter, of course
//   objList.EraseAllChildren ();

// but we want to demonstrate the usage of EraseCurrentChild
if ((objList != null) && (objList.HasChildren))
{
    try
    {
        objEle = objList.GetFirstChild(-1);
        while (objEle != null)
            objList.EraseCurrentChild();
            // no need to call GetNextChild
    }
    catch (err)
        // 1503 - we reached end of child list
        { if ((err.number & 0xffff) != 1503) throw (err); }
}
```

### 15.3.2.32.7 GetChild

**Declaration:** GetChild (position as long) as [XMLData](#)<sup>593</sup>

#### 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.

### 15.3.2.32.8 GetChildAttribute

**Method:** GetChildAttribute (strName as string) child as XMLData object (NULL on error)

**Description**

Retrieves the attribute having the given name.

**Errors**

- 1500 Invalid object.
- 1510 Invalid parameter.

### 15.3.2.32.9 GetChildElement

**Method:** GetChildElement (strName as string, nIndex as long) child as XMLData object (NULL on error)

**Description**

Retrieves the Nth child element with the given name.

**Errors**

- 1500 Invalid object.
- 1510 Invalid parameter.

### 15.3.2.32.10 GetChildKind

**Declaration:** GetChildKind (position as long, nKind as [SPYXMLDataKind](#)<sup>617</sup>) as [XMLData](#)<sup>593</sup>

**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.

### 15.3.2.32.11 GetCurrentChild

**Declaration:** GetCurrentChild as [XMLData](#)<sup>593</sup>

#### 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](#)<sup>598</sup>.

#### 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.

### 15.3.2.32.12 GetFirstChild

**Declaration:** GetFirstChild (nKind as [SPYXMLDataKind](#)<sup>617</sup>) as [XMLData](#)<sup>593</sup>

#### 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](#)<sup>598</sup>, [GetNextChild](#)<sup>599</sup> or [EraseCurrentChild](#)<sup>596</sup>.

#### 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](#)<sup>599</sup>.

### 15.3.2.32.13 GetNamespacePrefixForURI

**Method:** GetNamespacePrefixForURI (strURI as string) strNS as string

#### Description

Returns the namespace prefix of the supplied URI.

#### Errors

- 1500 Invalid object.
- 1510 Invalid parameter.

### 15.3.2.32.14 GetNextChild

**Declaration:** GetNextChild as [XMLData](#) <sup>593</sup>

#### 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](#) <sup>598</sup>.

Check for the last child of the element as shown in the sample below.

#### Errors

- 1500 The XMLData object is no longer valid.
- 1503 No iterator is initialized for this XMLData object.
- 1510 Invalid address for the return parameter was specified.

#### Examples

```
' -----
' VBA code snippet - iterate XMLData children
' -----
On Error Resume Next
Set objParent = objSpy.ActiveDocument.RootElement

'get elements of all kinds
Set objCurrentChild = objParent.GetFirstChild(-1)

Do
    'do something useful with the child

    'step to next child
    Set objCurrentChild = objParent.GetNextChild
Loop Until (Err.Number - vbObjectError = 1503)

// -----
```

```
// XMLSpy scripting environment - JScript
// iterate through children of XMLData
// -----
try
{
    var objXMLData = ... // initialize somehow
    var objChild = objXMLData.GetFirstChild(-1);

    while (true)
    {
        // do something usefull with objChild

        objChild = objXMLData.GetNextChild();
    }
}
catch (err)
{
    if ((err.number & 0xffff) == 1504)
        ; // element has no children
    else if ((err.number & 0xffff) == 1503)
        ; // last child reached
    else
        throw (err);
}
```

### 15.3.2.32.15 GetTextValueXMLDecoded

**Method:** GetTextValueXMLDecoded ()as string

#### Description

Gets the decoded text value of the XML.

#### Errors

- 1500 Invalid object.
- 1510 Invalid parameter.

### 15.3.2.32.16 HasChildren

**Declaration:** HasChildren as Boolean

#### Description

The property is true if the object is the parent of other XMLData objects. This property is read-only.

#### Errors

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

### 15.3.2.32.17 HasChildrenKind

**Declaration:** HasChildrenKind (*nKind* as [SPYXMLDataKind](#)<sup>617</sup>) 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.

### 15.3.2.32.18 InsertChild

**Declaration:** InsertChild (*pNewData* as [XMLData](#)<sup>593</sup>)

#### Description

InsertChild inserts the new child before the current child (see also [XMLData.GetFirstChild](#)<sup>598</sup>, [XMLData.GetNextChild](#)<sup>599</sup> 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

### 15.3.2.32.19 InsertChildAfter

**Method:** InsertChildAfter (Node as XMLData, NewData as XMLData)

#### Description

Inserts a new XML node (supplied with the second parameter) after the specified node (first parameter).

#### Errors

- 1500 Invalid object.
- 1506 Invalid input xml
- 1507 No children allowed
- 1510 Invalid parameter.
- 1512 Child is already added
- 1514 Invalid kind at position

### 15.3.2.32.20 InsertChildBefore

**Method:** InsertChildBefore (Node as XMLData, NewData as XMLData)

#### Description

Inserts a new XML node (supplied with the second parameter) before the specified node (first parameter).

#### Errors

- 1500 Invalid object.
- 1506 Invalid input xml
- 1507 No children allowed
- 1510 Invalid parameter.
- 1512 Child is already added
- 1514 Invalid kind at position

### 15.3.2.32.21 IsSameNode

**Declaration:** IsSameNode (pNodeToCompare as [XMLData](#)<sup>593</sup>) 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.

### 15.3.2.32.22 Kind

**Declaration:** Kind as [SPYXMLDataKind](#)<sup>617</sup>

#### 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.

### 15.3.2.32.23 MayHaveChildren

**Declaration:** MayHaveChildren as Boolean

#### Description

Indicates whether it is allowed to add children to this XMLData object.  
This property is read-only.

**Errors**

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

### 15.3.2.32.24 Name

**Declaration:** Name as String

**Description**

Used to modify and to get the name of the XMLData object.

**Errors**

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

### 15.3.2.32.25 Parent

**Declaration:** Parent as [XMLData](#) <sup>593</sup>

**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.

### 15.3.2.32.26 SetTextValueXMLEncoded

**Method:** SetTextValueXMLEncoded ( *strVal* as [String](#) <sup>617</sup> )

**Description**

Sets the encoded text value of the XML.

**Errors**

- 1500 Invalid object.
- 1513 Modification not allowed.

### 15.3.2.32.27 TextValue

**Declaration:** TextValue as String

#### Description

Used to modify and to get the text value of this XMLData object.

#### Errors

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

## 15.3.3 Enumerations

This is a list of all enumerations used by the Authentic Desktop API. If your scripting environment does not support enumerations use the number-values instead.

### 15.3.3.1 ENUMApplicationStatus

#### Description

Enumeration to specify the current Application status.

#### Possible values:

eApplicationRunning	= 0
eApplicationAfterLicenseCheck	= 1
eApplicationBeforeLicenseCheck	= 2
eApplicationConcurrentLicenseCheckFailed	= 3
eApplicationProcessingCommandLine	= 4

### 15.3.3.2 SPYAttributeTypeDefinition

#### Description

Attribute type definition that can be selected for generation of Sample XML.

This type is used with the method [GenerateDTDOrSchema](#)<sup>482</sup> and [GenerateDTDOrSchemaEx](#)<sup>482</sup>.

#### Possible values:

spyMergedGlobal	= 0
spyDistinctGlobal	= 1
spyLocal	= 2

### 15.3.3.3 SPYAuthenticActions

**Description**

Actions that can be performed on [AuthenticRange](#)<sup>407</sup> objects.

**Possible values:**

spyAuthenticInsertAt	= 0
spyAuthenticApply	= 1
spyAuthenticClearSurr	= 2
spyAuthenticAppend	= 3
spyAuthenticInsertBefore	= 4
spyAuthenticRemove	= 5

### 15.3.3.4 SPYAuthenticDocumentPosition

**Description**

Relative and absolute positions used for navigating with [AuthenticRange](#)<sup>407</sup> objects.

**Possible values:**

spyAuthenticDocumentBegin	= 0
spyAuthenticDocumentEnd	= 1
spyAuthenticRangeBegin	= 2
spyAuthenticRangeEnd	= 3

### 15.3.3.5 SPYAuthenticElementActions

**Description**

Actions that can be used with the obsolete object `GetAllowedElements` (superseded by [AuthenticRange.CanPerformActionWith](#)<sup>411</sup>).

**Possible values:**

k_ActionInsertAt	= 0
k_ActionApply	= 1
k_ActionClearSurr	= 2
k_ActionAppend	= 3
k_ActionInsertBefore	= 4
k_ActionRemove	= 5

### 15.3.3.6 SPYAuthenticElementKind

**Description**

Enumeration of the different kinds of elements used for navigation and selection within the [AuthenticRange](#)<sup>407</sup> and [AuthenticView](#)<sup>435</sup> objects.

**Possible values:**

spyAuthenticChar	= 0
spyAuthenticWord	= 1
spyAuthenticLine	= 3
spyAuthenticParagraph	= 4
spyAuthenticTag	= 6
spyAuthenticDocument	= 8
spyAuthenticTable	= 9
spyAuthenticTableRow	= 10
spyAuthenticTableColumn	= 11

### 15.3.3.7 SPYAuthenticMarkupVisibility

#### Description

Enumeration values to customize the visibility of markup with [MarkupVisibility](#)<sup>449</sup>.

#### Possible values:

spyAuthenticMarkupHidden	= 0
spyAuthenticMarkupSmall	= 1
spyAuthenticMarkupLarge	= 2
spyAuthenticMarkupMixed	= 3

### 15.3.3.8 SPYAuthenticToolBarButtonState

#### Description

Authentic toolbar button states are given by the following enumeration:

#### Possible values:

authenticToolBarButtonDefault	= 0
authenticToolBarButtonEnabled	= 1
authenticToolBarButtonDisabled	= 2

### 15.3.3.9 SPYDatabaseKind

#### Description

Values to select different kinds of databases for import. See [DatabaseConnection.DatabaseKind](#)<sup>460</sup> for its use.

#### Possible values:

spyDB_Access	= 0
spyDB_SQLServer	= 1
spyDB_Oracle	= 2
spyDB_Sybase	= 3
spyDB_MySQL	= 4
spyDB_DB2	= 5
spyDB_Other	= 6
spyDB_Unspecified	= 7

```
spyDB_PostgreSQL = 8
spyDB_iSeries     = 9
```

### 15.3.3.10 SPYDialogAction

#### Description

Values to simulate different interactions on dialogs. See [Dialogs](#)<sup>464</sup> for all dialogs available.

#### Possible values:

```
spyDialogOK       = 0    // simulate click on OK button
spyDialogCancel   = 1    // simulate click on Cancel button
spyDialogUserInput = 2    // show dialog and allow user interaction
```

### 15.3.3.11 SPYDOMType

#### Description

Enumeration values to parameterize generation of C++ code from schema definitions.

#### Possible values:

```
spyDOMType_msxml4   = 0    Obsolete
spyDOMType_xerces   = 1
spyDOMType_xerces3  = 2
spyDOMType_msxml6   = 3
```

`spyDOMType_xerces` indicates Xerces 2.x usage; `spyDOMType_xerces3` indicates Xerces 3.x usage.

### 15.3.3.12 SPYDTDSchemaFormat

#### Description

Enumeration to identify the different schema formats.

#### Possible values:

```
spyDTD      = 0
spyW3C      = 1
```

### 15.3.3.13 SPYEncodingByteOrder

#### Description

Enumeration values to specify encoding byte ordering for text import and export.

#### Possible values:

```
spyNONE      = 0
spyLITTLE_ENDIAN = 1
```

spyBIG\_ENDIAN = 2

### 15.3.3.14 SPYExportNamespace

#### Description

Enumeration type to configure handling of namespace identifiers during export.

#### Possible values:

spyNoNamespace = 0  
spyReplaceColonWithUnderscore = 1

### 15.3.3.15 SPYFindInFilesSearchLocation

#### Description

The different locations where a search can be performed. This type is used with the [FindInFilesDlg](#)<sup>511</sup> dialog.

#### Possible values:

spyFindInFiles\_Documents = 0  
spyFindInFiles\_Project = 1  
spyFindInFiles\_Folder = 2

### 15.3.3.16 SPYFrequentElements

#### Description

Enumeration value to parameterize schema generation.

#### Possible values:

spyGlobalElements = 0  
spyGlobalComplexType = 1

### 15.3.3.17 SPYImageKind

#### Description

Enumeration values to parameterize image type of the generated documentation. These values are used in [SchemaDocumentationDialog.DiagramFormat](#)<sup>534</sup>.

#### Possible values:

spyImageType\_PNG = 0  
spyImageType\_EMF = 1

### 15.3.3.18 SPYImportColumnsType

#### Description

Enumeration to specify different Import columns types.

**Possible values:**

spyImportColumns_Element	= 0
spyImportColumns_Attribute	= 1

### 15.3.3.19 SPYKeyEvent

**Description**

Enumeration type to identify the different key events. These events correspond with the equally named windows messages.

**Possible values:**

spyKeyDown	= 0
spyKeyUp	= 1
spyKeyPressed	= 2

### 15.3.3.20 SPYKeyStatus

**Description**

Enumeration type to identify the key status.

**Possible values:**

spyLeftShiftKeyMask	= 1
spyRightShiftKeyMask	= 2
spyLeftCtrlKeyMask	= 4
spyRightCtrlKeyMask	= 8
spyLeftAltKeyMask	= 16
spyRightAltKeyMask	= 32

### 15.3.3.21 SPYLibType

**Description**

Enumeration values to parameterize generation of C++ code from schema definitions.

**Possible values:**

spyLibType_static	= 0
spyLibType_dll	= 1

### 15.3.3.22 SPYLoading

**Description**

Enumeration values to define loading behaviour of URL files.

**Possible values:**

spyUseCacheProxy	= 0
------------------	-----

```
spyReload = 1
```

### 15.3.3.23 SPYMouseEvent

#### Description

Enumeration type that defines the mouse status during a mouse event. Use the enumeration values as bitmasks rather than directly comparing with them.

#### Examples

```
' to check for ctrl-leftbutton-down in VB
If (i_eMouseEvent = (XMLSpyLib.spyLeftButtonDownMask Or XMLSpyLib.spyCtrlKeyDownMask)) Then
    ' react on ctrl-leftbutton-down
End If
```

```
' to check for double-click with any button in VBScript
If (((i_eMouseEvent And spyDoubleClickMask) <> 0) Then
    ' react on double-click
End If
```

#### Possible values:

spyNoButtonMask	= 0	
spyMouseMoveMask	= 1	
spyLeftButtonMask	= 2	
spyMiddleButtonMask	= 4	
spyRightButtonMask	= 8	
spyButtonUpMask	= 16	
spyButtonDownMask	= 32	
spyDoubleClickMask	= 64	
spyShiftKeyDownMask	= 128	
spyCtrlKeyDownMask	= 256	
spyLeftButtonDownMask	= 34	// spyLeftButtonMask   spyButtonDownMask
spyMiddleButtonDownMask	= 36	// spyMiddleButtonMask   spyButtonDownMask
spyRightButtonDownMask	= 40	// spyRightButtonMask   spyButtonDownMask
spyLeftButtonUpMask	= 18	// spyLeftButtonMask   spyButtonUpMask
spyMiddleButtonUpMask	= 20	// spyMiddleButtonMask   spyButtonUpMask
spyRightButtonUpMask	= 24	// spyRightButtonMask   spyButtonUpMask
spyLeftDoubleClickMask	= 66	// spyRightButtonMask   spyButtonUpMask
spyMiddleDoubleClickMask	= 68	// spyMiddleButtonMask   spyDoubleClickMask
spyRightDoubleClickMask	= 72	// spyRightButtonMask   spyDoubleClickMask

### 15.3.3.24 SPYNumberDateTimeFormat

#### Description

Enumeration value to configure database connections.

#### Possible values:

spySystemLocale	= 0
spySchemaCompatible	= 1

### 15.3.3.25 SPYProgrammingLanguage

**Description**

Enumeration values to select the programming language for code generation from schema definitions.

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

**Possible values:**

spyUndefinedLanguage	= -1
spyJava	= 0
spyCpp	= 1
spyCSharp	= 2

### 15.3.3.26 SPYProjectItemTypes

**Description**

Enumeration values to identify the different elements in project item lists. See [SpyProjectItem.ItemType](#)<sup>548</sup>.

**Possible values:**

spyUnknownItem	= 0
spyFileItem	= 1
spyFolderItem	= 2
spyURLItem	= 3

### 15.3.3.27 SPYProjectType

**Description**

Enumeration values to parameterize generation of C# from schema definitions.

**Possible values:**

spyVisualStudioProject	= 0	Obsolete
spyVisualStudio2003Project	= 1	Obsolete
spyBorlandProject	= 2	Obsolete
spyMonoMakefile	= 3	Obsolete
spyVisualStudio2005Project	= 4	Obsolete
spyVisualStudio2008Project	= 5	Obsolete
spyVisualStudio2010Project	= 6	For C++ code also
spyVisualStudio2013Project	= 7	For C++ code also
spyVisualStudio2015Project	= 8	For C++ code also
spyVisualStudio2017Project	= 9	For C++ code also
spyVisualStudio2019Project	= 10	For C++ code also

### 15.3.3.28 SpySampleXMLGenerationChoiceMode

#### Description

This enumeration is used in [GenerateSampleXMLDlg.ChoiceMode](#)<sup>524</sup>:

```
spySampleXMLGen_FirstBranch    = 0
spySampleXMLGen_AllBranches    = 1
spySampleXMLGen_ShortestBranch = 2
```

### 15.3.3.29 SPYSampleXMLGenerationOptimization (Obsolete)

**This enumeration is OBSOLETE since v2014.**

#### Description

Specify the elements that will be generated in the Sample XML.  
This enumeration is used in [GenerateSampleXMLDlg](#)<sup>523</sup>.

#### Possible values:

```
spySampleXMLGen_Optimized      = 0
spySampleXMLGen_NonMandatoryElements = 1
spySampleXMLGen_Everything     = 2
```

### 15.3.3.30 SpySampleXMLGenerationSampleValueHints

#### Description

This enumeration is used in [GenerateSampleXMLDlg.SampleValueHints](#)<sup>528</sup>

```
spySampleXMLGen_FirstFit      = 0
spySampleXMLGen_RandomFit     = 1
spySampleXMLGen_CycleThrough  = 2
```

### 15.3.3.31 SPYSampleXMLGenerationSchemaOrDTDAssignment

#### Description

Specifies what kind of reference to the schema/DTD should be added to the generated Sample XML.  
This enumeration is used in [GenerateSampleXMLDlg](#)<sup>523</sup>.

**Possible values:**

spySampleXMLGen_AssignRelatively	= 0
spySampleXMLGen_AssignAbsolutely	= 1
spySampleXMLGen_DoNotAssign	= 2

### 15.3.3.32 SPYSchemaDefKind

**Description**

Enumeration type to select schema diagram types.

**Possible values:**

spyKindElement	= 0
spyKindComplexType	= 1
spyKindSimpleType	= 2
spyKindGroup	= 3
spyKindModel	= 4
spyKindAny	= 5
spyKindAttr	= 6
spyKindAttrGroup	= 7
spyKindAttrAny	= 8
spyKindIdentityUnique	= 9
spyKindIdentityKey	= 10
spyKindIdentityKeyRef	= 11
spyKindIdentitySelector	= 12
spyKindIdentityField	= 13
spyKindNotation	= 14
spyKindInclude	= 15
spyKindImport	= 16
spyKindRedefine	= 17
spyKindFacet	= 18
spyKindSchema	= 19
spyKindCount	= 20

### 15.3.3.33 SPYSchemaDocumentationFormat

**Description**

Enumeration values to parameterize generation of schema documentation. These values are used in [SchemaDocumentationDialog.OutputFormat](#)<sup>540</sup>.

**Possible values:**

spySchemaDoc_HTML	= 0
spySchemaDoc_MSWord	= 1
spySchemaDoc_RTF	= 2
spySchemaDoc_PDF	= 3

### 15.3.3.34 SPYSchemaExtensionType

**Description**

Enumeration to specify different Schema Extension types.

**Possible values:**

spySchemaExtension_None	= 0
spySchemaExtension_SQL_XML	= 1
spySchemaExtension_MS_SQL_Server	= 2
spySchemaExtension_Oracle	= 3

### 15.3.3.35 SPYSchemaFormat

**Description**

Enumeration to specify different Schema Format types.

**Possible values:**

spySchemaFormat_Hierarchical	= 0
spySchemaFormat_Flat	= 1

### 15.3.3.36 SPYTextDelimiters

**Description**

Enumeration values to specify text delimiters for text export.

**Possible values:**

spyTabulator	= 0
spySemicolon	= 1
spyComma	= 2
spySpace	= 3

### 15.3.3.37 SPYTextEnclosing

**Description**

Enumeration value to specify text enclosing characters for text import and export.

**Possible values:**

spyNoEnclosing	= 0
spySingleQuote	= 1
spyDoubleQuote	= 2

### 15.3.3.38 SPYTypeDetection

#### Description

Enumeration to select how type detection works during [GenerateDTDOrSchema](#)<sup>482</sup> and [GenerateDTDOrSchemaEx](#)<sup>482</sup>.

#### Possible values:

spyBestPossible	= 0
spyNumbersOnly	= 1
spyNoDetection	= 2

### 15.3.3.39 SPYURLTypes

#### Description

Enumeration to specify different URL types.

#### Possible values:

spyURLTypeAuto	= -1
spyURLTypeXML	= 0
spyURLTypeDTD	= 1

### 15.3.3.40 SPYValidateXSDVersion

#### Description

Enumeration values that select what XSD version to use. The XSD version that is selected depends on both (i) the presence/absence—and, if present, the value—of the `/xs:schema/@vc:minVersion` attribute of the XSD document, and (ii) the value of this enumeration.

`spyValidateXSDVersion_1_0` selects XSD 1.0 if `vc:minVersion` is absent, or is present with any value.  
`spyValidateXSDVersion_1_1` selects XSD 1.1 if `vc:minVersion` is absent, or is present with any value.  
`spyValidateXSDVersion_AutoDetect` selects XSD 1.1 if `vc:minVersion=1.1`. If the `vc:minVersion` attribute is absent, or is present with a value other than 1.1, then XSD 1.0 is selected.

#### Possible values

spyValidateXSDVersion_AutoDetect	= 0
spyValidateXSDVersion_1_1	= 1
spyValidateXSDVersion_1_0	= 2

### 15.3.3.41 SPYValidateErrorFormat

#### Description

Enumeration values that select the format of the error message.

**Possible values**

```
spyValidateErrorFormat_Text           = 0
spyValidateErrorFormat_ShortXML       = 1
spyValidateErrorFormat_LongXML        = 2
```

### 15.3.3.42 SPYViewModes

**Description**

Enumeration values that define the different view modes for XML documents. The mode *spyViewAuthentic(4)* identifies the mode that was intermediately called DocEdit mode and is now called Authentic mode. The mode *spyViewJsonSchema* identifies a mode which is mapped to the Schema Design View on the GUI but is distinguished internally.

**Possible values:**

```
spyViewGrid           = 0
spyViewText           = 1
spyViewBrowser        = 2
spyViewSchema         = 3
spyViewContent        = 4    // obsolete
spyViewAuthentic      = 4
spyViewWSDL           = 5
spyViewZIP            = 6
spyViewEditionInfo    = 7
spyViewXBRL           = 8
spyViewJsonSchema     = 9
```

### 15.3.3.43 SPYVirtualKeyMask

**Description**

Enumeration type for the most frequently used key masks that identify the status of the virtual keys. Use these values as bitmasks rather than directly comparing with them. When necessary, you can create further masks by using the 'logical or' operator.

**Examples**

```
' VBScript sample: check if ctrl-key is pressed
If ((i_nVirtualKeyStatus And spyCtrlKeyMask) <> 0)) Then
    ' ctrl-key is pressed
End If
```

```
' VBScript sample: check if ONLY ctrl-key is pressed
If (i_nVirtualKeyStatus == spyCtrlKeyMask) Then
    ' exactly ctrl-key is pressed
End If
```

```
// JScript sample: check if any of the right virtual keys is pressed
if ((i_nVirtualKeyStatus & (spyRightShiftKeyMask | spyRightCtrlKeyMask | spyRightAltKeyMask)) != 0)
{
    ; ' right virtual key is pressed
}

```

**Possible values:**

spyNoVirtualKeyMask	= 0	
spyLeftShiftKeyMask	= 1	
spyRightShiftKeyMask	= 2	
spyLeftCtrlKeyMask	= 4	
spyRightCtrlKeyMask	= 8	
spyLeftAltKeyMask	= 16	
spyRightAltKeyMask	= 32	
spyShiftKeyMask	= 3	// spyLeftShiftKeyMask   spyRightShiftKeyMask
spyCtrlKeyMask	= 12	// spyLeftCtrlKeyMask   spyRightCtrlKeyMask
spyAltKeyMask	= 48	// spyLeftAltKeyMask   spyRightAltKeyMask

### 15.3.3.44 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

## 15.4 ActiveX Integration

The Authentic Desktop 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 Authentic Desktop Integration Package must be installed (see <https://www.altova.com/components/download>). Ensure that you install Authentic Desktop first, and then the Authentic Desktop Integration Package. Other prerequisites apply, depending on language and platform (see [Prerequisites](#)<sup>618</sup>).

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 Authentic Desktop (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 Authentic Desktop 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 Authentic Desktop graphical user interface according to your custom requirements.

Integration at document level means embedding Authentic Desktop into your own application piece-by-piece. This includes implementing not only the main Authentic Desktop 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](#)<sup>621</sup> and [Integration at Document Level](#)<sup>623</sup> describe the key steps at these respective levels. The [ActiveX Integration Examples](#)<sup>626</sup> section provides examples in C#, HTML, and Java. Looking through these examples will help you to make the right decisions quickly. The [Object Reference](#)<sup>646</sup> section describes all COM objects that can be used for integration, together with their properties and methods.

For information about using Authentic Desktop as a Visual Studio plug-in, see [Authentic Desktop in Visual Studio](#)<sup>166</sup>.

### 15.4.1 Prerequisites

To integrate the Authentic Desktop ActiveX control into a custom application, the following must be installed on your computer:

- Authentic Desktop
- The Authentic Desktop 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 Authentic Desktop and Authentic Desktop Integration Package. For applications developed under Microsoft .NET platform with Visual Studio, both the 32-

bit and 64-bit versions of Authentic Desktop and Authentic Desktop Integration Package must be installed, as explained below.

### Microsoft .NET (C#, VB.NET) with Visual Studio

To integrate the Authentic Desktop 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
- Authentic Desktop 32-bit and Authentic Desktop Integration Package 32-bit
- The ActiveX controls must be added to the Visual Studio toolbox (see [Adding the ActiveX Controls to the Toolbox](#)<sup>619</sup>).

If you want to integrate the 64-bit ActiveX control, the following prerequisites apply in addition to the ones above:

- Authentic Desktop 32-bit and Authentic Desktop 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)
- Authentic Desktop 64-bit and Authentic Desktop 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](#)<sup>626</sup>.

### Java

To integrate the Authentic Desktop 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
- Authentic Desktop and Authentic Desktop Integration Package

**Note:** To run the 64-bit version of the Authentic Desktop ActiveX control, use a 64-bit version of Eclipse, as well as the 64-bit version of Authentic Desktop and the Authentic Desktop Integration Package.

### Authentic Desktop 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):

- Authentic Desktop
- The Authentic Desktop Integration Package
- The custom integration code or application.

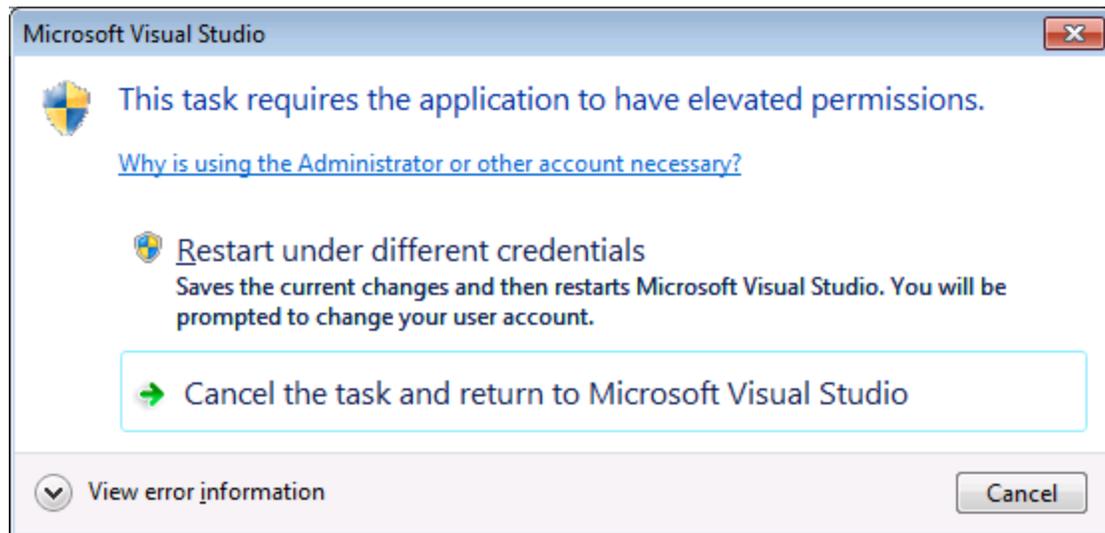
## 15.4.2 Adding the ActiveX Controls to the Toolbox

To use the Authentic Desktop ActiveX controls in an application developed with Visual Studio, the controls must first be added to the Visual Studio Toolbox, as follows:

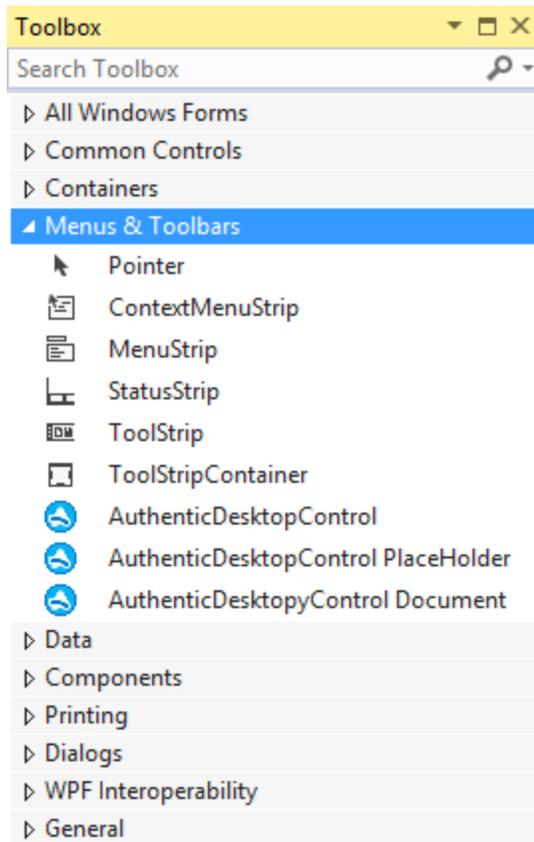
1. On the **Tools** menu of Visual Studio, click **Choose Toolbox Items**.
2. On the **COM Components** tab, select the check boxes next to the Authentic DesktopControl, Authentic DesktopControl Document, and Authentic DesktopControl Placeholder.

In case the controls above are not available, follow the steps below:

1. On the **COM Components** tab, click **Browse**, and select the **AuthenticControl.ocx** file from the Authentic Desktop installation folder. Remember that the Authentic Desktop Integration Package must be installed; otherwise, this file is not available, see [Prerequisites](#) <sup>618</sup>.
2. If prompted to restart Visual Studio with elevated permissions, click **Restart under different credentials**.



If the steps above were successful, the Authentic Desktop ActiveX controls become available in the Visual Studio Toolbox.



**Note:** For an application-level integration, only the **AuthenticDesktopControl** ActiveX control is used (see [Integration at Application Level](#)<sup>621</sup>). The **AuthenticDesktopControl Document** and **AuthenticDesktopControl Placeholder** controls are used for document-level integration (see [Integration at Document Level](#)<sup>623</sup>).

### 15.4.3 Integration at Application Level

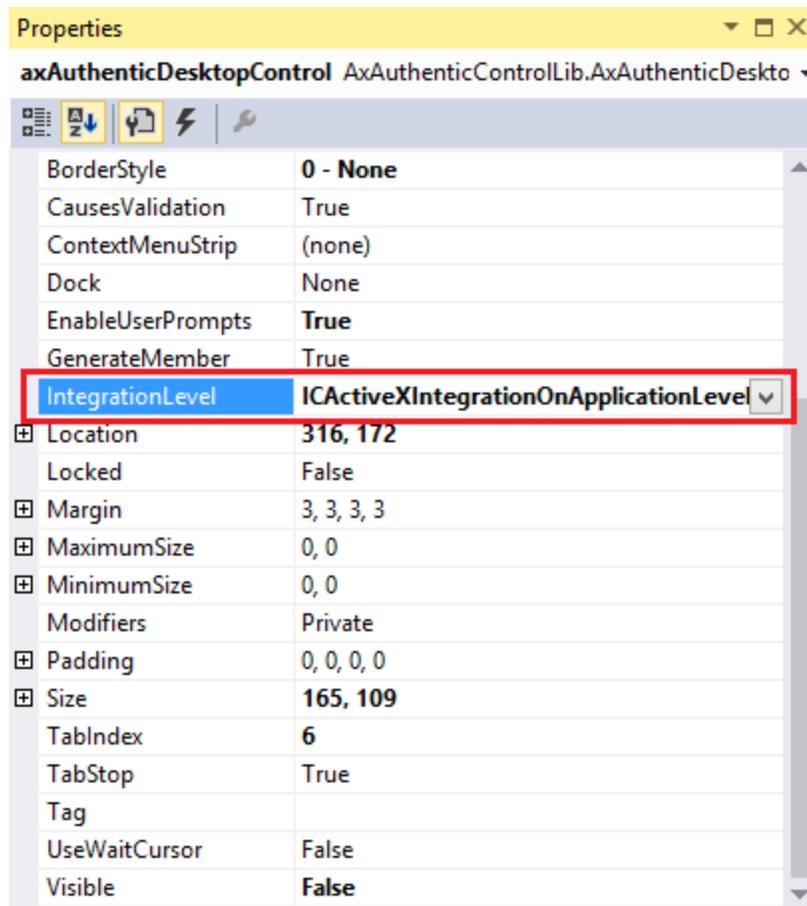
Integration at application level allows you to embed the complete interface of Authentic Desktop into a window of your application. With this type of integration, you get the whole user interface of Authentic Desktop, including all menus, toolbars, the status bar, document windows, and helper windows. Customization of the application's user interface is restricted to what Authentic Desktop provides. This includes rearrangement and resizing of helper windows and customization of menus and toolbars.

The only ActiveX control you need to integrate is [AuthenticDesktopControl](#)<sup>649</sup>. Do not instantiate or access [AuthenticDesktopControlDocument](#)<sup>657</sup> or [AuthenticDesktopControlPlaceholder](#)<sup>663</sup> ActiveX controls when integrating at application-level.

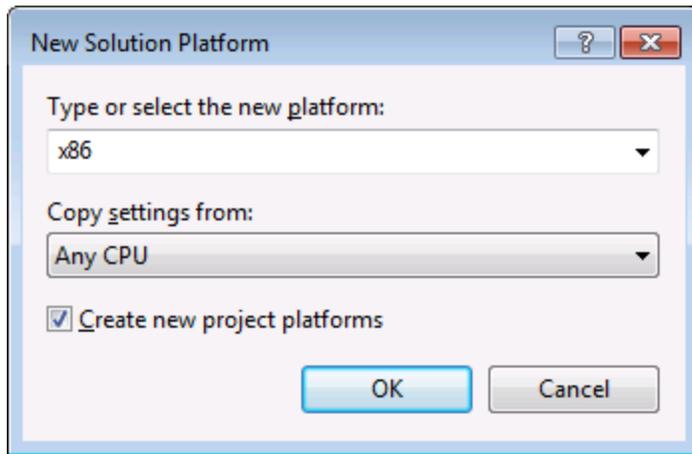
If you have any initialization to do or if you want to automate some behaviour of Authentic Desktop, use the properties, methods, and events described for [AuthenticDesktopControl](#)<sup>649</sup>. Consider using [AuthenticDesktopControl.Application](#)<sup>650</sup> for more complex access to Authentic Desktop functionality.

In C# or VB.NET with Visual Studio, the steps to create a basic, one-form application which integrates the Authentic Desktop ActiveX controls at application level are as follows:

1. Check that all prerequisites are met (see [Prerequisites](#) <sup>618</sup>).
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](#) <sup>619</sup>).
4. Drag the **AuthenticDesktopControl** from the toolbox onto your new form.
5. Select the **AuthenticDesktopControl** 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).

## 15.4.4 Integration at Document Level

Compared to integration at application level, integration at document level is a more complex, yet more flexible way to embed Authentic Desktop functionality into your application by means of ActiveX controls. With this approach, your code can access selectively the following parts of the Authentic Desktop user interface:

- Document editing window
- Project window
- Info window
- Messages window
- Entry helper windows (Elements, Attributes, Entities)
- Output window

As mentioned in [Integration at Application Level](#)<sup>621</sup>, for an ActiveX integration at application level, only one control is required, namely the **AuthenticDesktopControl**. However, for an ActiveX integration at document level, Authentic Desktop functionality is provided by the following ActiveX controls:

- [AuthenticDesktopControl](#)<sup>649</sup>
- [AuthenticDesktopControl Document](#)<sup>657</sup>
- [AuthenticDesktopControl Placeholder](#)<sup>663</sup>

These controls are supplied by the **AuthenticControl.ocx** file available in the application installation folder of Authentic Desktop. 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](#)<sup>619</sup>).

The basic steps to integrate the ActiveX controls at document level into your application are as follows:

1. First, instantiate **AuthenticDesktopControl** in your application. Instantiating this control is mandatory; it enables support for the **AuthenticDesktopControl Document** and **AuthenticDesktopControl Placeholder** controls mentioned above. It is important to set the [IntegrationLevel](#)<sup>651</sup> property to **ICActiveXIntegrationOnDocumentLevel** (or "1"). To hide the control from the user, set its **Visible** property to **False**.

**Note:** When integrating at document level, do not use the **Open** method of the **AuthenticDesktopControl**; this might lead to unexpected results. Use the corresponding open methods of **AuthenticDesktopControl Document** and **AuthenticDesktopControl Placeholder** instead.

2. Create at least one instance of **AuthenticDesktopControl Document** in your application. This control supplies the document editing window of Authentic Desktop 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 **AuthenticDesktopControl Placeholder** control for each additional window (other than the document window) that must be available to your application.

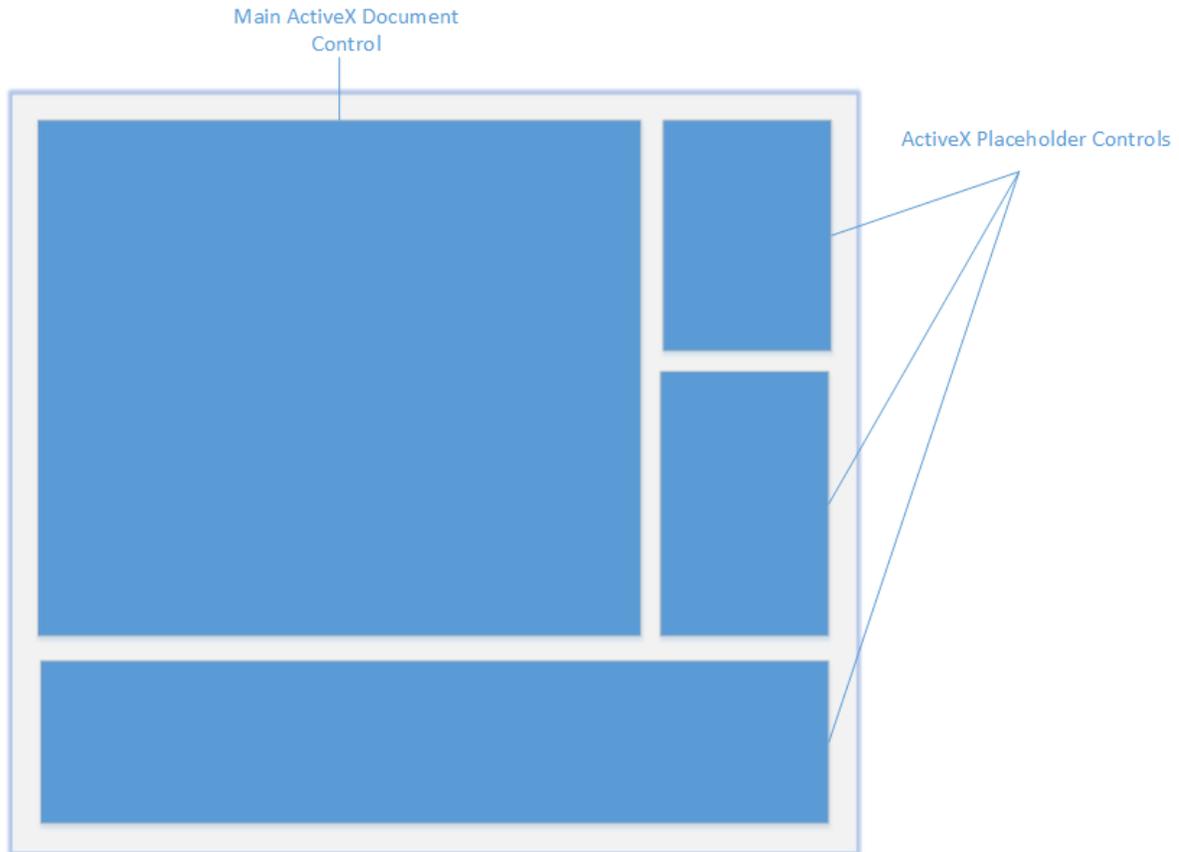
Instances of **AuthenticDesktopControl Placeholder** allow you to selectively embed additional windows of Authentic Desktop into your application. The window kind (for example, Project window) is defined by the property **PlaceholderWindowID**. Therefore, to set the window kind, set the property **PlaceholderWindowID**. For valid window identifiers, see [AuthenticDesktopControlPlaceholderWindow](#)<sup>666</sup>.

**Note:** Use only one **AuthenticDesktopControl Placeholder** for each window identifier.

For placeholder controls that select the Authentic Desktop project window, additional methods are available. Use **OpenProject** to load a Authentic Desktop project. Use the property **Project** and the methods and properties from the Authentic Desktop 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 Authentic Desktop 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](#)<sup>619</sup>).
3. Drag the [AuthenticDesktopControl](#)<sup>649</sup> from the toolbox onto your new form.
4. Set the **IntegrationLevel** property of the **AuthenticDesktopControl** to **ICActiveXIntegrationOnDocumentLevel**, and the **Visible** property to **False**. You can do this either from code or from the **Properties** window.
5. Drag the [AuthenticDesktopControl Document](#)<sup>657</sup> from the toolbox onto the form. This control provides the main document window of Authentic Desktop to your application, so you may need to resize it to a reasonable size for a document.
6. Optionally, add one or more [AuthenticDesktopControl Placeholder](#)<sup>663</sup> 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 **AuthenticDesktopControl Placeholder** control to a valid window identifier. For the list of valid values, see [AuthenticDesktopControlPlaceholderWindow](#)<sup>666</sup>.
8. Add commands to your application (at minimum, you will need to open, save and close documents), as shown below.

## Querying Authentic Desktop Commands

When you integrate at document level, no Authentic Desktop 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](#)<sup>651</sup> property of the **AuthenticDesktopControl**.
- To retrieve commands organized according to their menu structure, use the [MainMenu](#)<sup>652</sup> property.
- To retrieve commands organized by the toolbar in which they appear, use the [Toolbars](#)<sup>652</sup> property.
- To send commands to Authentic Desktop, use the [Exec](#)<sup>653</sup> method.
- To query if a command is currently enabled or disabled, use the [QueryStatus](#)<sup>654</sup> method.

This enables you to flexibly integrate Authentic Desktop commands into your application's menus and toolbars.

Your installation of Authentic Desktop also provides you with command label images used within Authentic Desktop. See the folder **<ApplicationFolder>\Examples\ActiveX\Images** of your Authentic Desktop

installation for icons in GIF format. The file names correspond to the command names as they are listed in the [Command Reference](#) <sup>639</sup> section.

## General considerations

To automate the behaviour of Authentic Desktop, use the properties, methods, and events described for the [AuthenticDesktopControl](#) <sup>649</sup>, [AuthenticDesktopControl Document](#) <sup>657</sup>, and [AuthenticDesktopControl Placeholder](#) <sup>663</sup>.

For more complex access to Authentic Desktop functionality, consider using the following properties:

- [AuthenticDesktopControl.Application](#) <sup>650</sup>
- [AuthenticDesktopControlDocument.Document](#) <sup>658</sup>
- [AuthenticDesktopControlPlaceholder.Project](#) <sup>664</sup>

These properties give you access to the Authentic Desktop automation interface (AuthenticDesktopAPI)

**Note:** To open a document, always use [AuthenticDesktopControlDocument.Open](#) <sup>660</sup> or [AuthenticDesktopControlDocument.New](#) <sup>660</sup> on the appropriate document control. To open a project, always use [AuthenticDesktopControlPlaceholder.OpenProject](#) <sup>665</sup> on a placeholder control embedding a Authentic Desktop project window.

For examples that show how to instantiate and access the necessary controls in different programming environments, see [ActiveX Integration Examples](#) <sup>626</sup>.

## 15.4.5 ActiveX Integration Examples

This section contains examples of Authentic Desktop 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 Authentic Desktop installation.

### 15.4.5.1 C#

A basic ActiveX integration example solution for C# and Visual Studio is available in the folder `<ApplicationFolder>\Examples\ActiveX\C#`. Before you compile the source code and run the sample, make sure that all prerequisites are met (see [Running the Sample C# Solution](#) <sup>626</sup>).

#### 15.4.5.1.1 Running the Sample C# Solution

The sample Visual Studio solution available in the folder `<ApplicationFolder>\Examples\ActiveX\C#` illustrates how to consume the Authentic Desktop 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](#) <sup>618</sup>.

## 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

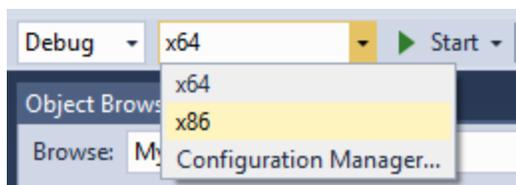
The sample application contains one instance of [AuthenticDesktopControlDocument](#)<sup>657</sup> and several instances of [AuthenticDesktopControlPlaceHolder](#)<sup>663</sup> controls. Double-check that the following properties of these controls are set as shown in the table below:

Control name	Property	Property value
axAuthenticDesktopControl	IntegrationLevel	ICActiveXIntegrationOnDocumentLevel
axAuthenticDesktopControlHelperWndEntities	PlaceholderWindowID	2
axAuthenticDesktopControlHelperWndAttributes	PlaceholderWindowID	1
axAuthenticDesktopControlHelperWndElements	PlaceholderWindowID	0
axAuthenticDesktopControlHelperWndInfo	PlaceholderWindowID	18
axAuthenticDesktopControlHelperWndProject	PlaceholderWindowID	4

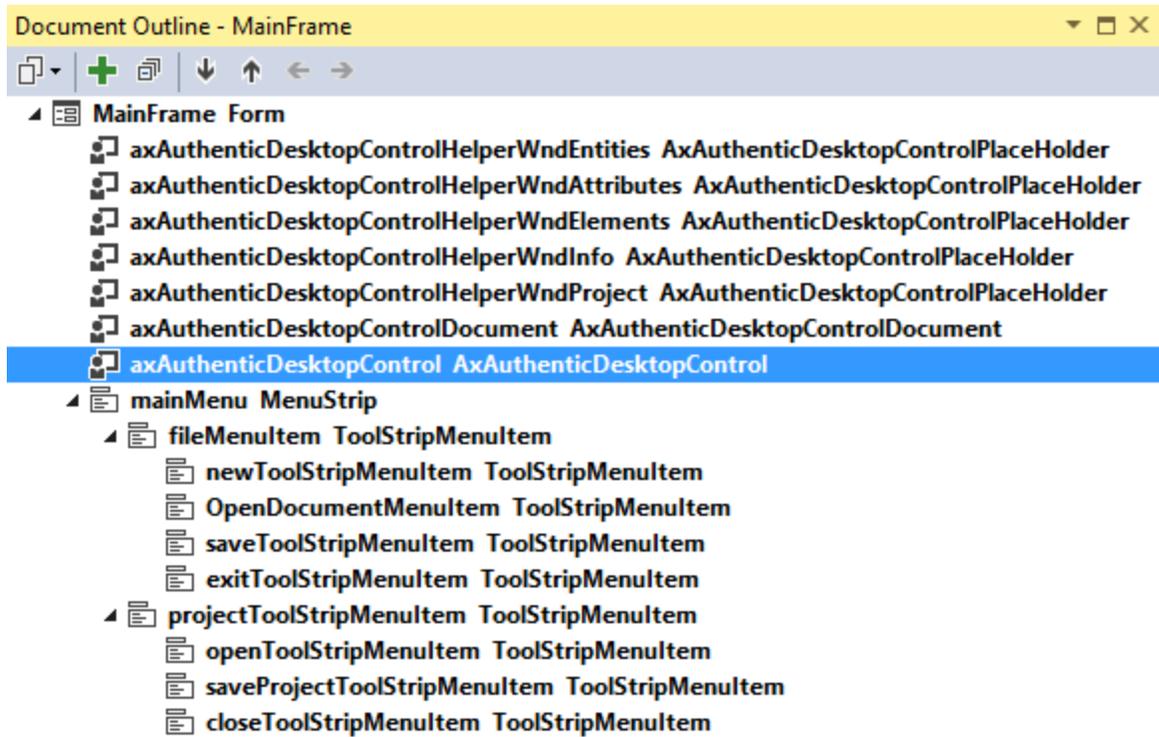
Here is how you can view or set the properties of an ActiveX control:

1. Open the **MDIMain.cs** form in the designer window.

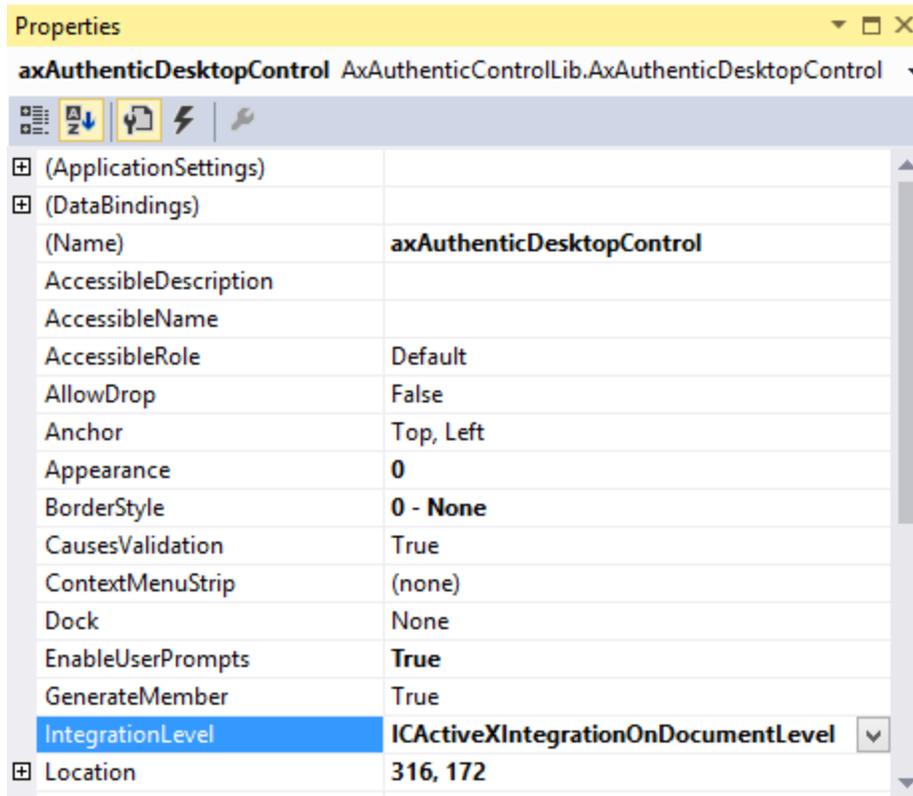
**Note:** On 64-bit Windows, it may be necessary to change the build configuration of the Visual Studio solution to "x86" **before** opening the designer window. If you need to build the sample as a 64-bit application, see [Prerequisites](#)<sup>618</sup>.



2. Open the **Document Outline** window of Visual Studio (On the **View** menu, click **Other Windows | Document Outline**).



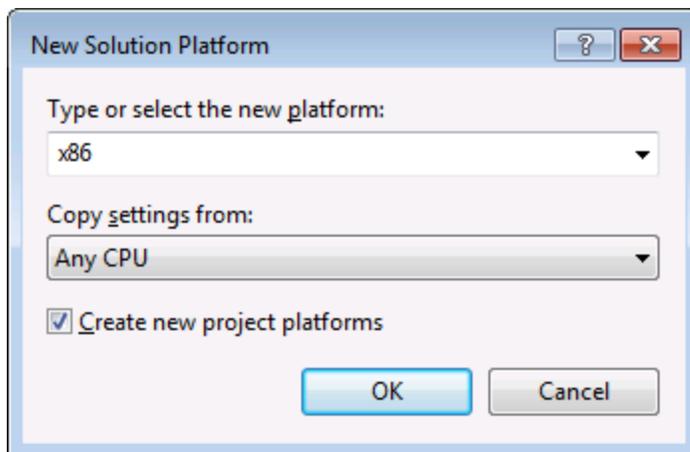
3. Click an ActiveX control in the **Document Outline** window, and edit its required property in the **Properties** window, for example:



### IntegrationLevel

#### 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**).



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.

## 15.4.5.2 Java

Authentic Desktop 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 Authentic Desktop installation, after you have installed both Authentic Desktop and the Authentic Desktop Integration Package (see also [Prerequisites](#)<sup>618</sup>).

- `AltovaAutomation.dll`: a JNI wrapper for Altova automation servers (in case of the 32-bit installation of Authentic Desktop)
- `AltovaAutomation_x64.dll`: a JNI wrapper for Altova automation servers (in case of the 64-bit installation of Authentic Desktop)
- `AltovaAutomation.jar`: Java classes to access Altova automation servers
- `AuthenticActiveX.jar`: Java classes that wrap the Authentic ActiveX interface
- `AuthenticActiveX_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](#)<sup>631</sup>.

### Rules for mapping the ActiveX Control names to Java

For the documentation of ActiveX controls, see [Object Reference](#)<sup>646</sup>. 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 Authentic Desktop 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 `AuthenticDesktopControl`, 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:

AuthenticDesktopControl: Java class to access the application

AuthenticDesktopControlEvents: Events interface for the AuthenticDesktopControl

AuthenticDesktopControlEventsDefaultHandler: Default handler for

AuthenticDesktopControlEvents

## Exceptions to mapping rules

There are some exceptions to the rules listed above. These are listed below:

Interface	Java name
AuthenticDesktopControlDocument, method New	newDocument
Document, method SetEncoding	setFileEncoding
AuthenticView, method Goto	gotoElement
AuthenticRange, method Goto	gotoElement
AuthenticRange, method Clone	cloneRange

## This section

This section shows how some basic Authentic Desktop ActiveX functionality can be accessed from Java code. It is organized into the following sub-sections:

- [Example Java Project](#) <sup>631</sup>
- [Creating the ActiveX Controls](#) <sup>633</sup>
- [Loading Data in the Controls](#) <sup>634</sup>
- [Basic Event Handling](#) <sup>634</sup>
- [Menus](#) <sup>635</sup>
- [UI Update Event Handling](#) <sup>637</sup>
- [Creating an XML Tree](#) <sup>637</sup>

### 15.4.5.2.1 Example Java Project

The Authentic Desktop 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 AuthenticDesktopControl 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
.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
AuthenticActiveX.jar	Java classes of the Authentic Desktop ActiveX control
AuthenticActiveX_JavaDoc.zip	Javadoc file containing help documentation for the Java API
AuthenticDesktopContainer.java	Java example source code
AuthenticDesktopContainerEventHandler.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.
XMLTreeDialog.java	Java example source code

### What the example does

The example places one AuthenticDesktop document editor window, the Project window, the Info window and an Authentic entry helper in an AWT frame window. It reads out the File menu defined for Authentic 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](#) <sup>633</sup>: Starts Authentic Desktop, which is registered as an automation server, or activates Authentic Desktop if it is already running.
- [Loading Data in the Controls](#) <sup>634</sup>: Locates one of the example documents installed with Authentic Desktop and opens it.
- [Basic Event Handling](#) <sup>634</sup>: Changes the view of all open documents to Browser View. The code also shows how to iterate through open documents.
- [Menus](#) <sup>635</sup>: Validates the active document and shows the result in a message box. The code shows how to use output parameters.
- [UI Update Event Handling](#) <sup>637</sup>: Shows how to handle Authentic Desktop events.
- [Creating an XML Tree](#) <sup>637</sup>: Shows how to create an XML tree and prepare it for modal activation.

### Updating the path to the Examples folder

Before running the provided sample, you may need to edit the **AuthenticDesktopContainer.java** file. Namely, check that the following path refers to the actual folder where the Authentic Desktop example files are stored on your operating system:

```
// Locate samples installed with the product.
final String strExamplesFolder = System.getenv( "USERPROFILE" ) + "\\Documents\\Altova\\
\\Authentic2023\\AuthenticExamples\\";
```

## Running the example from the command line

To run the example from the command line:

1. Check that all prerequisites are met (see [Prerequisites](#)<sup>618</sup>).
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 `AuthenticDesktopContainer.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](#)<sup>618</sup>).
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**.

### 15.4.5.2.2 Creating the ActiveX Controls

The code listing below show how ActiveX controls can be created. The constructors will create the Java wrapper objects. Adding these Canvas-derived objects to a panel or to a frame will trigger the creation of the wrapped ActiveX object.

```
01  /**
02   * Authentic Desktop manager control - always needed
03   */
04  public static AuthenticDesktopControl      authenticDesktopControl = null;
05
06  /**
07   * Authentic Desktop document editing control
08   */
09  public static AuthenticDesktopControlDocument  authenticDesktopDocument = null;
10
```

```

11  /**
12   * Tool windows - Authentic Desktop place-holder controls
13   */
14  private static AuthenticDesktopControlPlaceHolder    authenticDesktopInfoToolWindow =
null;
15  private static AuthenticDesktopControlPlaceHolder
authenticDesktopEHElementToolWindow = null;
16  private static AuthenticDesktopControlPlaceHolder    authenticDesktopProjectToolWindow
= null;
17
18  // Create the Authentic Desktop ActiveX control, The parameter determines that we want
// to place document controls and place-holder controls individually.
19  // It gives us full control over the menu, as well.
20  authenticDesktopControl = new AuthenticDesktopControl(
    ICActiveXIntegrationLevel.ICActiveXIntegrationOnDocumentLevel.getValue() );
21
22  authenticDesktopDocument = new AuthenticDesktopControlDocument();
23  authenticDesktopDocument.setPreferredSize( new Dimension ( 640, 480 ) );
24  frame.add( authenticDesktopDocument, BorderLayout.CENTER );
25
26  // Create a project window and open the sample project in it
27  authenticDesktopProjectToolWindow = new AuthenticDesktopControlPlaceHolder(
    XMLSpyControlPlaceholderWindow.XMLSpyControlProjectWindowToolWnd.getValue() );
28  authenticDesktopProjectToolWindow.setPreferredSize( new Dimension( 200, 200 ) );

```

### 15.4.5.2.3 Loading Data in the Controls

The code listing below show how data can be loaded in the ActiveX controls.

```

1  // Locate samples installed with the product.
2  final String strExamplesFolder = System.getenv( "USERPROFILE" ) +
    "\\Documents\\Altova\\Authentic2022\\AuthenticExamples\\";
3  authenticDesktopProjectToolWindow = new
AuthenticDesktopControlPlaceHolder( XMLSpyControlPlaceholderWindow.XMLSpyControlProjectWind
owToolWnd.getValue() );

```

### 15.4.5.2.4 Basic Event Handling

The code listing below shows how basic events can be handled. When calling the AuthenticDesktopControl'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 AuthenticDesktopDocumentControl'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  authenticDesktopControl.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                 AuthenticDesktopContainer.authenticDesktopDocument.open( i_strFileName );
19                 AuthenticDesktopContainer.authenticDesktopDocument.requestFocusInWindow();
20             } catch (Exception e) {
21                 e.printStackTrace();
22             }
23         }
24     }

```

### 15.4.5.2.5 Menus

The code listing below shows how menu items can be created. Each `Authentic DesktopCommand` 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 `AuthenticDesktopControl` object by calling its `exec` method. The `menuMap` object that is filled during menu creation is used later (see section [UI Update Event Handling](#)<sup>637</sup>).

```

01 // Load the file menu when the button is pressed
02 btnMenu.addActionListener( new ActionListener() {
03     public void actionPerformed(ActionEvent e) {
04         try {
05             // Create the menubar that will be attached to the frame
06             MenuBar mb = new MenuBar();
07             // Load the main menu's first item - the File menu
08             XMLSpyCommand xmlSpyMenu =
xmlSpyControl.getMainMenu().getSubCommands().getItem( 0 );
09             // Create Java menu items from the Commands objects
10             Menu fileMenu = new Menu();
11             handlerObject.fillMenu( fileMenu, xmlSpyMenu.getSubCommands() );
12             fileMenu.setLabel( xmlSpyMenu.getLabel().replace( "&", "" ) );
13             mb.add( fileMenu );
14             frame.setMenuBar( mb );
15             frame.validate();
16         } catch (AutomationException e1) {
17             e1.printStackTrace();
18         }
19         // Disable the button when the action has been performed
20         ((AbstractButton) e.getSource()).setEnabled( false );
21     }
22 } );
23 /** * Populates a menu with the commands and submenus contained in an XMLSpyCommands
object */
24     public void fillMenu(Menu newMenu, XMLSpyCommands xmlSpyMenu) throws
AutomationException
25     {

```

```
26 // For each command/submenu in the xmlSpyMenu
27 for ( int i = 0 ; i < xmlSpyMenu.getCount() ; ++i )
28 {
29     XMLSpyCommand xmlSpyCommand = xmlSpyMenu.getItem( i );
30     if ( xmlSpyCommand.getIsSeparator() )
31         newMenu.addSeparator();
32     else
33     {
34         XMLSpyCommands subCommands = xmlSpyCommand.getSubCommands();
35         // Is it a command (leaf), or a submenu?
36         if ( subCommands.isNull() || subCommands.getCount() == 0 )
37         {
38             // Command -> add it to the menu, set its ActionCommand to its ID and store it
in the menuMap
39             MenuItem mi = new MenuItem( xmlSpyCommand.getLabel().replace( "&", "" ) );
40             mi.setActionCommand( "" + xmlSpyCommand.getID() );
41             mi.addActionListener( this );
42             newMenu.add( mi );
43             menuMap.put( xmlSpyCommand.getID(), mi );
44         }
45         else
46         {
47             // Submenu -> create submenu and repeat recursively
48             Menu newSubMenu = new Menu();
49             fillMenu( newSubMenu, subCommands );
50             newSubMenu.setLabel( xmlSpyCommand.getLabel().replace( "&", "" ) );
51             newMenu.add( newSubMenu );
52         }
53     }
54 }
55 }
56
57 /**
58  * Action handler for the menu items
59  * Called when the user selects a menu item; the item's action command corresponds to
the command table for XMLSpy
60  */
61 public void actionPerformed( ActionEvent e )
62 {
63     try
64     {
65         int iCmd = Integer.parseInt( e.getActionCommand() );
66         // Handle explicitly the Close commands
67         switch ( iCmd )
68         {
69             case 57602: // Close
70             case 34050: // Close All
71                 AuthenticDesktopContainer.initXmlSpyDocument();
72                 break;
73             default:
74                 AuthenticDesktopContainer.xmlSpyControl.exec( iCmd );
75                 break;
76         }
77     }
78     catch ( Exception ex )
79     {
80         ex.printStackTrace();
81     }
82 }
```

```
83 }
```

### 15.4.5.2.6 UI Update Event Handling

The code listing below shows how a UI-Update event handler can be created.

```
01  /**
02   * Call-back from the XMLSpyControl.
03   * Called to enable/disable commands
04   */
05  @Override
06  public void onUpdateCmdUI() throws AutomationException
07  {
08      // A command should be enabled if the result of queryStatus contains the Supported
09      (1) and Enabled (2) flags
10      for ( java.util.Map.Entry<Integer, MenuItem> pair : menuMap.entrySet() )
11
12          pair.getValue().setEnabled( AuthenticDesktopContainer.authenticDesktopControl.querySt
13          atus( pair.getKey() ) > 2 );
14  }
15  /**
16   * Call-back from the XMLSpyControl.
17   * Usually called while enabling/disabling commands due to UI updates
18   */
19  @Override
20  public boolean onIsActiveEditor( String i_strFilePath ) throws AutomationException
21  {
22      try {
23          return
24          AuthenticDesktopContainer.authenticDesktopDocument.getDocument().getFullName().equalsIgnore
25          Case( i_strFilePath );
26      } catch ( Exception e ) {
27          return false;
28      }
29  }
```

### 15.4.5.2.7 Creating an XML Tree

The listing below loads an XML data object as nodes in a tree.

```
01 // access required XMLSpy Java-COM classes
02 import com.altova.automation.XMLSpy.XMLData;
03
04 // access AWT and Swing components
05 import java.awt.*;
06 import javax.swing.*;
07 import javax.swing.tree.*;
08
09 /**
10  * A simple example of a tree control loading the structure from an XMLData object.
11  * The class receives an XMLData object, loads its nodes in a JTree, and prepares
12  * for modal activation.
```

```

13 *
14 * Feel free to modify and extend this sample.
15 *
16 * @author Altova GmbH
17 */
18 class XMLTreeDialog extends JDialog
19 {
20     /**
21     * The tree control
22     */
23     private JTree myTree;
24
25     /**
26     * Root node of the tree control
27     */
28     private DefaultMutableTreeNode top ;
29
30     /**
31     * Constructor that prepares the modal dialog containing the filled tree control
32     * @param xml The data to be displayed in the tree
33     * @param parent Parent frame
34     */
35     public XMLTreeDialog( XMLData xml, Frame parent )
36     {
37         // Construct the modal dialog
38         super( parent, "XML tree", true );
39         // Arrange controls in the dialog
40         top = new DefaultMutableTreeNode("root");
41         myTree = new JTree(top);
42         setContentPane( new JScrollPane( myTree ) );
43         // Build up the tree
44         fillTree( top, xml );
45         myTree.expandRow( 0 );
46     }
47
48     /**
49     * Loads the nodes of an XML element under a given tree node
50     * @param node Target tree node
51     * @param elem Source XML element
52     */
53     private void fillTree( DefaultMutableTreeNode node, XMLData elem)
54     {
55         try
56         {
57             // There are several ways to iterate through child elements: either using the
58             // getFirstChild/getNextChild,
59             // or by incrementing an index up to countChildren and calling getChild [as shown
60             // below].
61             // If you only want to get children of one kind, you should use
62             // countChildrenKind/getChildKind,
63             // or provide a kind to the getFirstChild before iterating with the getNextChild.
64             int nSize = elem.countChildren() ;
65             for ( int i = 0 ; i < nSize ; ++i)
66             {
67                 // Create a new tree node for each child element, and continue recursively
68                 XMLData newElem = elem.getChild(i) ;
69                 DefaultMutableTreeNode newNode = new DefaultMutableTreeNode( newElem.getName() )
70
71             }
72         }
73     }
74 }

```

```

67     node.add( newNode ) ;
68     fillTree( newNode, newElem ) ;
69   }
70 }
71 catch (Exception e)
72 {
73     e.printStackTrace();
74 }
75 }
76
77 }

```

## 15.4.6 Command Reference

This section lists the names and identifiers of all menu commands that are available within Authentic Desktop. Every sub-section lists the commands from the corresponding top-level menu of Authentic Desktop. The command tables are organized as follows:

- The "Menu Item" column shows the command's menu text as it appears in Authentic Desktop, 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 **ActiveXImages** folder of the Authentic Desktop installation directory.
- The "ID" column shows the numeric identifier of the column that must be supplied as argument to methods which execute or query this command.

To execute a command, use the [AuthenticDesktopControl.Exec](#)<sup>653</sup> or the [AuthenticDesktopControlDocument.Exec](#)<sup>659</sup> methods. To query the status of a command, use the [AuthenticDesktopControl.QueryStatus](#)<sup>654</sup> or [AuthenticDesktopControlDocument.QueryStatus](#)<sup>660</sup> methods.

Depending on the edition of Authentic Desktop you have installed, some of these commands might not be supported.

### 15.4.6.1 "File" Menu

The "File" menu has the following commands:

Menu item	Command name	ID
New...	ID_FILE_NEW	57600
Open...	ID_FILE_OPEN	57601
Reload	IDC_FILE_RELOAD	34065
Encoding...	IDC_ENCODING	34061
Close	ID_FILE_CLOSE	57602
Close All	IDC_CLOSE_ALL	34050
Close All But Active	IDC_CLOSE_OTHERS	34271

Menu item	Command name	ID
Save	ID_FILE_SAVE	57603
Save As...	ID_FILE_SAVE_AS	57604
Save All	ID_FILE_SAVE_ALL	34208
Send by Mail...	ID_FILE_SEND_MAIL	57612
Print...	ID_FILE_PRINT	57607
Print Preview	IDC_PRINT_PREVIEW	34104
Print Setup...	ID_FILE_PRINT_SETUP	57606
Recent File	ID_FILE_MRU_FILE1	57616
Exit	ID_APP_EXIT	57665

### 15.4.6.2 "Edit" Menu

The "Edit" menu has the following commands:

Menu item	Command name	ID
Undo	ID_EDIT_UNDO	57643
Redo	ID_EDIT_REDO	57644
Cut	ID_EDIT_CUT	57635
Copy	ID_EDIT_COPY	57634
Paste	ID_EDIT_PASTE	57637
Delete	ID_EDIT_CLEAR	57632
Select All	ID_EDIT_SELECT_ALL	57642
Find...	ID_EDIT_FIND	57636
Find Next	ID_EDIT_REPEAT	57640
Replace...	ID_EDIT_REPLACE	57641

### 15.4.6.3 "Project" Menu

The "Project" menu has the following commands:

Menu item	Command name	ID
New Project	IDC_ICPROJECTGUI_NEW	37200
Open Project...	IDC_ICPROJECTGUI_OPEN	37201
Reload Project	IDC_ICPROJECTGUI_RELOAD	37202
Close Project	IDC_ICPROJECTGUI_CLOSE	37203
Save Project	IDC_ICPROJECTGUI_SAVE	37204
Save Project As...	IDC_ICPROJECTGUI_SAVE_AS	37207
Enable Source Control	ID_SCC_ENABLE	38602
Add Files to Project...	IDC_ICPROJECTGUI_ADD_FILES_TO_PROJECT	37205
Add Global Resource to Project...	IDC_ICPROJECTGUI_ADD_GLOBAL_RESOURCE_TO_PROJECT	37239
Add URL to Project...	IDC_ICPROJECTGUI_ADD_URL_TO_PROJECT	37206
Add Active File to Project	IDC_ICPROJECTGUI_ADD_ACTIVE_FILE_TO_PROJECT	37208
Add Active and Related Files to Project	IDC_ICPROJECTGUI_ADD_ACTIVE_AND_RELATED_FILES_TO_PROJECT	37209
Add Project Folder to Project...	IDC_ICPROJECTGUI_ADD_FOLDER_TO_PROJECT	37210
Add External Folder to Project...	IDC_ICPROJECTGUI_ADD_EXT_FOLDER_TO_PROJECT	37211
Add External Web Folder to Project...	IDC_ICPROJECTGUI_ADD_EXT_URL_FOLDER_TO_PROJECT	37212
Script settings...	IDC_PROJECT_SCRIPT_SETTINGS	34136
Properties...	IDC_ICPROJECTGUI_PROJECT_PROPERTIES	37223
Recent Project	IDC_ICPROJECTGUI_RECENT	37224

#### 15.4.6.4 "XML" Menu

The "XML" menu has the following commands:

Menu item	Command name	ID
Check Well-Formedness	IDC_CHECK_WELL_FORM	34049
Validate XML	IDC_VALIDATE	32954

### 15.4.6.5 "XSL/XQuery" Menu

The "XSL/XQuery" menu has the following commands:

Menu item	Command name	ID
XSL Transformation	IDC_TRANSFORM_XSL	33006
XSL-FO Transformation	IDC_TRANSFORM_XSLFO	33007
XSL Parameters / XQuery Variables...	IDC_TRANSFORM_XSL_PARAMS	33008

### 15.4.6.6 "Authentic" Menu

The "Authentic" menu has the following commands:

Menu item	Command name	ID
New Document...	IDC_AUTHENTIC_NEW_FILE	34036
Edit Database Data...	IDC_AUTHENTIC_EDIT_DB	34035
Edit StyleVision Stylesheet	IDC_EDIT_SPS	34060
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

Menu item	Command name	ID
Toggle Underline	IDC_AUTHENTICGUI_RICHEDIT_TOGGLEUNDERLINE	32815
Toggle Strikethrough	IDC_AUTHENTICGUI_RICHEDIT_TOGGLESTRIKETHROUGH	32816
Foreground Color	IDC_AUTHENTICGUI_RICHEDIT_COLOR_FOREGROUND	32824
Background Color	IDC_AUTHENTICGUI_RICHEDIT_COLOR_BACKGROUND	32830
Align Left	IDC_AUTHENTICGUI_RICHEDIT_ALIGN_LEFT	32818
Center	IDC_AUTHENTICGUI_RICHEDIT_ALIGN_CENTER	32819
Align Right	IDC_AUTHENTICGUI_RICHEDIT_ALIGN_RIGHT	32820
Append Row	IDC_ROW_APPEND	32806
Insert Row	IDC_ROW_INSERT	32809
Duplicate Row	IDC_ROW_DUPLICATE	32808
Move Row Up	IDC_ROW_MOVE_UP	32811
Move Row Down	IDC_ROW_MOVE_DOWN	32810
Delete Row	IDC_ROW_DELETE	32807
Generate an HTML document	IDC_PXF_GENERATE_HTML	34283
Generate an RTF document	IDC_PXF_GENERATE_RTF	34284
Generate a PDF document	IDC_PXF_GENERATE_PDF	34285
Generate a Word 2007+ document	IDC_PXF_GENERATE_DOCX	34286
Trusted Locations...	IDC_TRUSTED_LOCATIONS	34288

### 15.4.6.7 "View" Menu

The "View" menu has the following commands:

Menu item	Command name	ID
Authentic View	IDC_VIEW_CONTENT	34177
Browser View	IDC_VIEW_BROWSER	34176

Menu item	Command name	ID
Text View Settings	IDC_TEXTVIEW_SETTINGS	34119

### 15.4.6.8 "Browser" Menu

The "Browser" menu has the following commands:

Menu item	Command name	ID
Back	IDC_STEP_BACK	32958
Forward	IDC_STEP_FORWARD	32957
Stop	IDC_BROWSER_STOP	34047
Refresh	IDC_BROWSER_REFRESH	34046
Largest	IDC_BROWSER_FONT_LARGEST	34041
Larger	IDC_BROWSER_FONT_LARGE	34040
Medium	IDC_BROWSER_FONT_MEDIUM	34042
Smaller	IDC_BROWSER_FONT_SMALL	34043
Smallest	IDC_BROWSER_FONT_SMALLEST	34044

### 15.4.6.9 "Tools" Menu

The "Tools" menu has the following commands:

Menu item	Command name	ID
Spelling...	IDC_SPELL_CHECK	34154
Spelling Options...	IDC_SPELL_OPTIONS	34155
Scripting Editor...	ID_SCRIPTFORMEDITOR_EDIT_PROJECT	39666
none	ID_SCRIPTFORMEDITOR_EXECUTE_MACRO_MENU_UPPDATE	39600
	IDC_TOOLS_ENTRY	34292
Global Resources	IDC_GLOBALRESOURCES	37401
	IDC_GLOBALRESOURCES_SUBMENUENTR Y1	37408
Customize...	IDC_APP_TOOLS_CUSTOMIZE	32959

Menu item	Command name	ID
Options...	IDC_SETTINGS	34133
	ID_SCRIPTING_MACROITEMS	34249

### 15.4.6.10 "Window" Menu

The "Window" menu has the following commands:

Menu item	Command name	ID
Cascade	ID_WINDOW_CASCADE	57650
Tile horizontally	ID_WINDOW_TILE_HORZ	57651
Tile vertically	ID_WINDOW_TILE_VERT	57652
Project window	IDC_PROJECT_WINDOW	34128
Info window	IDC_INFO_WINDOW	34085
Entry Helpers	IDC_ENTRY_HELPERS	34062
Output windows	IDC_OUTPUT_DIALOGBARS	34004
Project and Entry Helpers	IDC_PROJECT_ENTRYHELPERS	34006
All on/off	IDC_ALL_BARS	34031

### 15.4.6.11 "Help" Menu

The "Help" menu has the following commands:

Menu item	Command name	ID
Table of Contents...	IDC_HELP_CONTENTS	32966
Index...	IDC_HELP_INDEX	32967
Search...	IDC_HELP_SEARCH	32969
Keyboard Map...	IDC_HELP_KEYMAPDLG	32968
Software Activation...	IDC_ACTIVATION	32970
Order Form...	IDC_OPEN_ORDER_PAGE	32971
Registration...	IDC_REGISTRATION	32972
Check for Updates...	IDC_CHECK_FOR_UPDATES	32973

Menu item	Command name	ID
XMLSpy Product Comparison...	IDC_PRODUCT_COMPARISON	32955
Support Center...	IDC_OPEN_SUPPORT_PAGE	32961
FAQ on the Web...	IDC_OPEN_FAQ_PAGE	32962
Download Components and Free Tools...	IDC_OPEN_COMPONENTS_PAGE	32963
Authentic on the Internet..	IDC_OPEN_HOME_PAGE	32964
Authentic Training...	IDC_OPEN_TRAINING_PAGE	32965
About Authentic...	ID_APP_ABOUT	57664

## 15.4.7 Object Reference

### Objects:

[Authentic DesktopCommand](#) <sup>646</sup>  
[Authentic DesktopCommands](#) <sup>648</sup>  
[AuthenticDesktopControl](#) <sup>649</sup>  
[AuthenticDesktopControlDocument](#) <sup>657</sup>  
[AuthenticDesktopControlPlaceHolder](#) <sup>663</sup>

To give access to standard Authentic Desktop functionality, objects of the **Authentic Desktop automation interface** can be accessed as well. See [AuthenticDesktopControl.Application](#) <sup>650</sup>, [AuthenticDesktopControlDocument.Document](#) <sup>658</sup> and [AuthenticDesktopControlPlaceHolder.Project](#) <sup>664</sup> for more information.

### 15.4.7.1 Authentic DesktopCommand

#### Properties:

[ID](#) <sup>647</sup>  
[Label](#) <sup>647</sup>  
[Name](#) <sup>648</sup>  
[IsSeparator](#) <sup>647</sup>  
[ToolTip](#) <sup>648</sup>  
[StatusText](#) <sup>648</sup>  
[Accelerator](#) <sup>647</sup>  
[SubCommands](#) <sup>648</sup>

#### 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	<ul style="list-style-type: none"> <li>• <code>ID</code> is greater than zero</li> </ul>

The Command object is...	When...
	<ul style="list-style-type: none"> <li>• <code>IsSeparator</code> is false</li> <li>• <code>SubCommands</code> is empty</li> </ul>
A command container	<ul style="list-style-type: none"> <li>• ID is zero</li> <li>• <code>IsSeparator</code> is false</li> <li>• <code>SubCommands</code> contains a collection of Command objects.</li> </ul>
Separator	<ul style="list-style-type: none"> <li>• ID is zero</li> <li>• <code>IsSeparator</code> is true</li> </ul>

#### 15.4.7.1.1 Accelerator

**Property:** `Accelerator` as [string](#)

**Description:**

Returns the accelerator key defined for the command. If the command has no accelerator key assigned, this property returns the empty string. The string representation of the accelerator key has the following format:

[ALT+] [CTRL+] [SHIFT+] key

Where `key` is converted using the Windows Platform SDK function `GetKeyNameText`.

#### 15.4.7.1.2 ID

**Property:** `ID` as [long](#)

**Description:**

This property gets the unique identifier of the command. A command's ID is required to execute the command (using [Exec](#)<sup>653</sup>) or query its status (using [QueryStatus](#)<sup>654</sup>). If the command is a container for other commands (for example, a top-level menu), or a separator, the ID is 0.

#### 15.4.7.1.3 IsSeparator

**Property:** `IsSeparator` as [boolean](#)

**Description:**

The property returns `true` if the command object is a menu separator; `false` otherwise. See also [Command](#)<sup>646</sup>.

#### 15.4.7.1.4 Label

**Property:** `Label` as [string](#)

**Description:**

This property gets the text of the command as it is displayed in the graphical user interface of Authentic Desktop. 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.

### 15.4.7.1.5 Name

**Property:** Name as [string](#)

**Description:**

This property gets the unique name of the command. This value can be used to get the icon file of the command, where it is available. The available icon files can be found in the folder `<ApplicationFolder>\Examples\ActiveX\Images` of your Authentic Desktop installation.

### 15.4.7.1.6 StatusText

**Property:** Label as [string](#)

**Description:**

The status text is the text shown in the status bar of Authentic Desktop 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.

### 15.4.7.1.7 SubCommands

**Property:** SubCommands as [Commands](#) <sup>648</sup>

**Description:**

The `SubCommands` property gets the collection of [Command](#) <sup>646</sup> 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`.

### 15.4.7.1.8 ToolTip

**Property:** ToolTip as [string](#)

**Description:**

This property gets the text that is shown as a tool-tip for each command. If the command does not have a tooltip text, the property returns an empty string.

## 15.4.7.2 Authentic DesktopCommands

**Properties:**

[Count](#) <sup>649</sup>

## [Item](#) <sup>649</sup>

### **Description:**

Collection of [Command](#) <sup>646</sup> objects to get access to command labels and IDs of the AuthenticDesktopControl. Those commands can be executed with the [Exec](#) <sup>653</sup> method and their status can be queried with [QueryStatus](#) <sup>654</sup>.

### 15.4.7.2.1 Count

**Property:** Count as [long](#)

### **Description:**

Number of [Command](#) <sup>646</sup> objects on this level of the collection.

### 15.4.7.2.2 Item

**Property:** Item (n as [long](#)) as [Command](#) <sup>646</sup>

### **Description:**

Gets the command with the index *n* in this collection. Index is 1-based.

## 15.4.7.3 AuthenticDesktopControl

### **Properties:**

[IntegrationLevel](#) <sup>651</sup>

[Appearance](#) <sup>650</sup>

[Application](#) <sup>650</sup>

[BorderStyle](#) <sup>650</sup>

[CommandsList](#) <sup>651</sup>

[EnableUserPrompts](#) <sup>651</sup>

[MainMenu](#) <sup>652</sup>

[Toolbars](#) <sup>652</sup>

### **Methods:**

[Open](#) <sup>654</sup>

[Exec](#) <sup>653</sup>

[QueryStatus](#) <sup>654</sup>

### **Events:**

[OnUpdateCmdUI](#) <sup>656</sup>

[OnOpenedOrFocused](#) <sup>656</sup>

[OnCloseEditingWindow](#) <sup>655</sup>

[OnFileChangedAlert](#) <sup>655</sup>

[OnContextChanged](#) <sup>655</sup>

[OnDocumentOpened](#) <sup>655</sup>

[OnValidationWindowUpdated](#) <sup>657</sup>

This object is a complete ActiveX control and should only be visible if the Authentic Desktop library is used in the Application Level mode.

### 15.4.7.3.1 Properties

The following properties are defined:

[IntegrationLevel](#) <sup>651</sup>

[EnableUserPrompts](#) <sup>651</sup>

[Appearance](#) <sup>650</sup>

[BorderStyle](#) <sup>650</sup>

Command related properties:

[CommandsList](#) <sup>651</sup>

[MainMenu](#) <sup>652</sup>

[Toolbars](#) <sup>652</sup>

Access to AuthenticDesktopAPI:

[Application](#) <sup>650</sup>

#### 15.4.7.3.1.1 Appearance

**Property:** Appearance as [short](#)

**Dispatch Id:** -520

**Description:**

A value not equal to 0 displays a client edge around the control. Default value is 0.

#### 15.4.7.3.1.2 Application

**Property:** Application as [Application](#)

**Dispatch Id:** 1

**Description:**

The `Application` property gives access to the `Application` object of the complete Authentic Desktop automation server API. The property is read-only.

#### 15.4.7.3.1.3 BorderStyle

**Property:** BorderStyle as [short](#)

**Dispatch Id:** -504

**Description:**

A value of 1 displays the control with a thin border. Default value is 0.

#### 15.4.7.3.1.4 *CommandsList*

**Property:** `CommandList` as [Commands](#)<sup>648</sup> (read-only)

**Dispatch Id:** 1004

**Description:**

This property returns a flat list of all commands defined available with `AuthenticDesktopControl`. To get commands organized according to their menu structure, use [MainMenu](#)<sup>652</sup>. To get toolbar commands, use [Toolbars](#)<sup>652</sup>.

```
public void GetAllAuthenticCommands()
{
    // Get all commands from the Authentic ActiveX control assigned to the current form
    AuthenticControlLib.XMLSpyCommands commands =
    this.axAuthenticDesktopControl1.CommandList;
    // Loop through all commands
    for (int i = 0; i < commands.Count; i++)
    {
        // Get each command by index and output it to the console
        AuthenticControlLib.XMLSpyCommand cmd = axAuthenticDesktopControl1.CommandList[i];
        Console.WriteLine("{0} {1} {2}", cmd.ID, cmd.Name, cmd.Label.Replace("&", ""));
    }
}
```

*C# example*

#### 15.4.7.3.1.5 *EnableUserPrompts*

**Property:** `EnableUserPrompts` as `boolean`

**Dispatch Id:** 1006

**Description:**

Setting this property to *false*, disables user prompts in the control. The default value is *true*.

#### 15.4.7.3.1.6 *IntegrationLevel*

**Property:** `IntegrationLevel` as [ICActiveXIntegrationLevel](#)<sup>686</sup>

**Dispatch Id:** 1000

**Description:**

The `IntegrationLevel` property determines the operation mode of the control. See also [Integration at Application Level](#)<sup>621</sup> and [Integration at Document Level](#)<sup>623</sup> for more information.

**Note:** It is important to set this property immediately after the creation of the `AuthenticDesktopControl` object.

#### 15.4.7.3.1.7 MainMenu

**Property:** `MainMenu` as [Command](#)<sup>646</sup> (read-only)

**Dispatch Id:** 1003

##### Description:

This property provides information about the structure and commands available in the `AuthenticDesktopControl` main menu, as a `Command` object. The `Command` object contains all available submenus of Authentic Desktop (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 GetAuthenticMenus ()
{
    // Get the main menu from the Authentic ActiveX control assigned to the current form
    AuthenticControlLib.XMLSpyCommand mainMenu =
    this.axAuthenticDesktopControl1.MainMenu;

    // Loop through entries of the main menu (e.g. File, Edit, etc.)
    for (int i = 0; i < mainMenu.SubCommands.Count; i++)
    {
        AuthenticControlLib.XMLSpyCommand menu = mainMenu.SubCommands[i];
        Console.WriteLine("{0} menu has {1} children items (including separators)",
        menu.Label.Replace("&", ""), menu.SubCommands.Count);
    }
}
```

*C# example*

#### 15.4.7.3.1.8 Toolbars

**Property:** `Toolbars` as [Commands](#)<sup>648</sup> (read-only)

**Dispatch Id:** 1005

##### Description:

This property provides information about the structure of `AuthenticDesktopControl` toolbars, as a `Command` object. The `Command` object contains all available toolbars of Authentic Desktop. 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 GetAuthenticToolbars()
{
    // Get the application toolbars from the Authentic ActiveX control assigned to the
    // current form
    AuthenticControlLib.XMLSpyCommands toolbars =
    this.axAuthenticDesktopControl1.Toolbars;

    // Iterate through all toolbars
    for (int i = 0; i < toolbars.Count; i++)
    {
        AuthenticControlLib.XMLSpyCommand toolbar = toolbars[i];
        Console.WriteLine();
        Console.WriteLine("The toolbar \"{0}\" has the following commands:",
        toolbar.Label);

        // Iterate through all commands of this toolbar
        for (int j = 0; j < toolbar.SubCommands.Count; j++)
        {
            AuthenticControlLib.XMLSpyCommand cmd = toolbar.SubCommands[j];
            // Output only command objects that are not separators
            if ( ! cmd.IsSeparator)
            {
                Console.WriteLine("{0}, {1}, {2}", cmd.ID, cmd.Name, cmd.Label.Replace("&",
                ""));
            }
        }
    }
}
```

C# example

### 15.4.7.3.2 Methods

The following methods are defined:

[Open](#) <sup>654</sup>

[Exec](#) <sup>653</sup>

[QueryStatus](#) <sup>654</sup>

#### 15.4.7.3.2.1 Exec

**Method:** Exec (nCmdID as long) as boolean

**Dispatch Id:** 6

**Description:**

This method calls the Authentic Desktop 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](#) <sup>651</sup>. To retrieve the status of any command, use [QueryStatus](#) <sup>654</sup>.

### 15.4.7.3.2.2 Open

**Method:** Open (strFilePath as string) as boolean

**Dispatch Id:** 5

**Description:**

The result of the method depends on the extension passed in the argument `strFilePath`. If the file extension is `.sps`, a new document is opened. If the file extension is `.svp`, the corresponding project is opened. If a different file extension is passed into the method, the control tries to load the file as a new component into the active document.

Do not use this method to load documents or projects when using the control in document-level integration mode. Instead, use [AuthenticDesktopControlDocument.Open](#)<sup>660</sup> and [AuthenticDesktopControlPlaceholder.OpenProject](#)<sup>665</sup>.

### 15.4.7.3.2.3 QueryStatus

**Method:** QueryStatus (nCmdID as long) as long

**Dispatch Id:** 7

**Description:**

`QueryStatus` returns the enabled/disabled and checked/unchecked status of the command specified by `nCmdID`. The status is returned as a bit mask.

Bit	Value	Name	Meaning
0	1	Supported	Set if the command is supported.
1	2	Enabled	Set if the command is enabled (can be executed).
2	4	Checked	Set if the command is checked.

This means that if `QueryStatus` returns 0 the command ID is not recognized as a valid Authentic Desktop command. If `QueryStatus` returns a value of 1 or 5, the command is disabled.

### 15.4.7.3.3 Events

The AuthenticDesktopControl ActiveX control provides the following connection point events:

[OnUpdateCmdUI](#)<sup>656</sup>

[OnOpenedOrFocused](#)<sup>656</sup>

[OnCloseEditingWindow](#)<sup>655</sup>

[OnFileChangedAlert](#)<sup>655</sup>

[OnContextChanged](#)<sup>655</sup>

[OnDocumentOpened](#)<sup>655</sup>

[OnValidationWindowUpdated](#)<sup>657</sup>

#### 15.4.7.3.3.1 OnCloseEditingWindow

**Event:** OnCloseEditingWindow (i\_strFilePath as String) as boolean

**Dispatch Id:** 1002

**Description:**

This event is triggered when Authentic Desktop 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 AuthenticDesktopControl should try to close the editor and destroy the associated document control.

#### 15.4.7.3.3.2 OnContextChanged

**Event:** OnContextChanged (i\_strContextName as String, i\_bActive as bool) as bool

**Dispatch Id:** 1004

**Description:**

This event is not used in Authentic Desktop

#### 15.4.7.3.3.3 OnDocumentOpened

**Event:** OnDocumentOpened (objDocument as Document)

**Dispatch Id:** 1

**Description:**

This event is triggered whenever a document is opened. The argument *objDocument* is a *Document* object from the Authentic Desktop 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 [AuthenticDesktopControlDocument.OnDocumentOpened](#)<sup>662</sup> instead.

#### 15.4.7.3.3.4 OnFileChangedAlert

**Event:** OnFileChangedAlert (i\_strFilePath as String) as bool

**Dispatch Id:** 1001

**Description:**

This event is triggered when a file loaded with AuthenticDesktopControl is changed on the hard disk by another application. Clients should return *true*, if they handled the event, or *false*, if Authentic Desktop should handle it in its customary way, i.e. prompting the user for reload.

#### 15.4.7.3.3.5 *OnLicenseProblem*

**Event:** OnLicenseProblem (i\_strLicenseProblemText as String)

**Dispatch Id:** 1005

**Description:**

This event is triggered when AuthenticDesktopControl 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).

#### 15.4.7.3.3.6 *OnOpenedOrFocused*

**Event:** OnOpenedOrFocused (i\_strFilePath as String, i\_bOpenWithThisControl as bool)

**Dispatch Id:** 1000

**Description:**

When integrating at application level, this event informs clients that a document has been opened, or made active by Authentic Desktop.

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 AuthenticDesktopControl, since internal access is required. Otherwise, the file can be opened with different editors.

#### 15.4.7.3.3.7 *OnToolWindowUpdated*

**Event:** OnToolWindowUpdated (pToolWnd as long )

**Dispatch Id:** 1006

**Description:**

This event is triggered when the tool window is updated.

#### 15.4.7.3.3.8 *OnUpdateCmdUI*

**Event:** OnUpdateCmdUI ( )

**Dispatch Id:** 1003

**Description:**

Called frequently to give integrators a good opportunity to check status of Authentic Desktop commands using [AuthenticDesktopControl.QueryStatus](#)<sup>654</sup>. Do not perform long operations in this callback.

### 15.4.7.3.3.9 OnValidationWindowUpdated

**Event:** OnValidationWindowUpdated ()

**Dispatch Id:** 3

**Description:**

This event is triggered whenever the validation output window is updated with new information.

## 15.4.7.4 AuthenticDesktopControlDocument

**Properties:**

[Appearance](#) <sup>658</sup>  
[BorderStyle](#) <sup>658</sup>  
[Document](#) <sup>658</sup>  
[IsModified](#) <sup>658</sup>  
[Path](#) <sup>659</sup>  
[ReadOnly](#) <sup>659</sup>

**Methods:**

[Exec](#) <sup>659</sup>  
[New](#) <sup>660</sup>  
[Open](#) <sup>660</sup>  
[QueryStatus](#) <sup>660</sup>  
[Reload](#) <sup>661</sup>  
[Save](#) <sup>661</sup>  
[SaveAs](#) <sup>661</sup>

**Events:**

[OnDocumentOpened](#) <sup>662</sup>  
[OnDocumentClosed](#) <sup>662</sup>  
[OnModifiedFlagChanged](#) <sup>663</sup>  
[OnContextChanged](#) <sup>662</sup>  
[OnFileChangedAlert](#) <sup>662</sup>  
[OnActivate](#) <sup>661</sup>

If the AuthenticDesktopControl is integrated in the Document Level mode each document is displayed in an own object of type AuthenticDesktopControlDocument. The AuthenticDesktopControlDocument 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.

### 15.4.7.4.1 Properties

The following properties are defined:

[ReadOnly](#) <sup>659</sup>  
[IsModified](#) <sup>658</sup>  
[Path](#) <sup>659</sup>

[Appearance](#) <sup>658</sup>  
[BorderStyle](#) <sup>658</sup>

Access to AuthenticDesktopAPI:  
[Document](#) <sup>658</sup>

#### 15.4.7.4.1.1 Appearance

**Property:** Appearance as `short`

**Dispatch Id:** -520

**Description:**

A value not equal to 0 displays a client edge around the document control. Default value is 0.

#### 15.4.7.4.1.2 BorderStyle

**Property:** BorderStyle as `short`

**Dispatch Id:** -504

**Description:**

A value of 1 displays the control with a thin border. Default value is 0.

#### 15.4.7.4.1.3 Document

**Property:** Document as Document

**Dispatch Id:** 1

**Description:**

The Document property gives access to the Document object of the Authentic Desktop automation server API. This interface provides additional functionality which can be used with the document loaded in the control. The property is read-only.

#### 15.4.7.4.1.4 IsModified

**Property:** IsModified as `boolean` (read-only)

**Dispatch Id:** 1006

**Description:**

IsModified is `true` if the document content has changed since the last open, reload or save operation. It is `false`, otherwise.

#### 15.4.7.4.1.5 Path

**Property:** Path as `string`

**Dispatch Id:** 1005

**Description:**

Sets or gets the full path name of the document loaded into the control.

#### 15.4.7.4.1.6 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.

### 15.4.7.4.2 Methods

The following methods are defined:

Document handling:

[New](#) <sup>660</sup>

[Open](#) <sup>660</sup>

[Reload](#) <sup>661</sup>

[Save](#) <sup>661</sup>

[SaveAs](#) <sup>661</sup>

Command Handling:

[Exec](#) <sup>659</sup>

[QueryStatus](#) <sup>660</sup>

#### 15.4.7.4.2.1 Exec

**Method:** Exec (`nCmdID` as `long`) as `boolean`

**Dispatch Id:** 8

**Description:**

`Exec` calls the Authentic Desktop 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](#)<sup>652</sup> property of AuthenticDesktopControl. To get toolbar commands, use the [Toolbars](#)<sup>652</sup> property of the AuthenticDesktopControl.

#### 15.4.7.4.2.2 *New*

**Method:** `New ()` as `boolean`

**Dispatch Id:** 1000

**Description:**

This method initializes a new document inside the control.

#### 15.4.7.4.2.3 *Open*

**Method:** `Open (strFileName as string)` as `boolean`

**Dispatch Id:** 1001

**Description:**

`Open` loads the file `strFileName` as the new document into the control.

#### 15.4.7.4.2.4 *QueryStatus*

**Method:** `QueryStatus (nCmdID as long)` as `long`

**Dispatch Id:** 9

**Description:**

`QueryStatus` returns the enabled/disabled and checked/unchecked status of the command specified by `nCmdID`. The status is returned as a bit mask.

Bit	Value	Name	Meaning
0	1	Supported	Set if the command is supported.
1	2	Enabled	Set if the command is enabled (can be executed).
2	4	Checked	Set if the command is checked.

This means that if `QueryStatus` returns 0 the command ID is not recognized as a valid Authentic Desktop 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.

#### 15.4.7.4.2.5 Reload

**Method:** Reload () as [boolean](#)

**Dispatch Id:** 1002

**Description:**

Reload updates the document content from the file system.

#### 15.4.7.4.2.6 Save

**Method:** Save () as [boolean](#)

**Dispatch Id:** 1003

**Description:**

Save saves the current document at the location [Path](#)<sup>659</sup>.

#### 15.4.7.4.2.7 SaveAs

**Method:** SaveAs (strFileName as [string](#)) as [boolean](#)

**Dispatch Id:** 1004

**Description:**

SaveAs sets [Path](#)<sup>659</sup> to strFileName and then saves the document to this location.

### 15.4.7.4.3 Events

The AuthenticDesktopControlDocument ActiveX control provides following connection point events:

[OnDocumentOpened](#)<sup>662</sup>

[OnDocumentClosed](#)<sup>662</sup>

[OnModifiedFlagChanged](#)<sup>663</sup>

[OnContextChanged](#)<sup>662</sup>

[OnFileChangedAlert](#)<sup>662</sup>

[OnActivate](#)<sup>661</sup>

[OnSetEditorTitle](#)<sup>663</sup>

#### 15.4.7.4.3.1 OnActivate

**Event:** OnActivate ()

**Dispatch Id:** 1005

**Description:**

This event is triggered when the document control is activated, has the focus, and is ready for user input.

#### 15.4.7.4.3.2 *OnContextChanged*

**Event:** OnContextChanged (i\_strContextName as String, i\_bActive as bool) as bool

**Dispatch Id:** 1004

**Description:** None

#### 15.4.7.4.3.3 *OnDocumentClosed*

**Event:** OnDocumentClosed (objDocument as Document)

**Dispatch Id:** 1001

**Description:**

This event is triggered whenever the document loaded into this control is closed. The argument `objDocument` is a `Document` object from the Authentic Desktop automation interface and should be used with care.

#### 15.4.7.4.3.4 *OnDocumentOpened*

**Event:** OnDocumentOpened (objDocument as Document)

**Dispatch Id:** 1000

**Description:**

This event is triggered whenever a document is opened in this control. The argument `objDocument` is a `Document` object from the Authentic Desktop automation interface, and can be used to query for more details about the document, or perform additional operations.

#### 15.4.7.4.3.5 *OnDocumentSaveAs*

**Event:** OnContextDocumentSaveAs (i\_strFileName as String)

**Dispatch Id:** 1007

**Description:**

This event is triggered when this document gets internally saved under a new name.

#### 15.4.7.4.3.6 *OnFileChangedAlert*

**Event:** OnFileChangedAlert () as bool

**Dispatch Id:** 1003

**Description:**

This event is triggered when the file loaded into this document control is changed on the hard disk by another application. Clients should return true, if they handled the event, or false, if Authentic Desktop should handle it in its customary way, i.e. prompting the user for reload.

#### 15.4.7.4.3.7 OnModifiedFlagChanged

**Event:** OnModifiedFlagChanged (i\_bIsModified as [boolean](#))

**Dispatch Id:** 1002

**Description:**

This event gets triggered whenever the document changes between modified and unmodified state. The parameter *i\_bIsModified* is *true* if the document contents differs from the original content, and *false*, otherwise.

#### 15.4.7.4.3.8 OnSetEditorTitle

**Event:** OnSetEditorTitle ()

**Dispatch Id:** 1006

**Description:**

This event is being raised when the contained document is being internally renamed.

### 15.4.7.5 AuthenticDesktopControlPlaceholder

**Properties available for all kinds of placeholder windows:**

[PlaceholderWindowID](#)<sup>664</sup>

**Properties for project placeholder window:**

[Project](#)<sup>664</sup>

**Methods for project placeholder window:**

[OpenProject](#)<sup>665</sup>

[CloseProject](#)<sup>665</sup>

The `AuthenticDesktopControlPlaceholder` control is used to show the additional Authentic Desktop windows like Overview, Library or Project window. It is used like any other ActiveX control and can be placed anywhere in the client application.

#### 15.4.7.5.1 Properties

The following properties are defined:

[PlaceholderWindowID](#)<sup>664</sup>

Access to AuthenticDesktopAPI:

[Project](#)<sup>664</sup>

#### 15.4.7.5.1.1 Label

**Property:** Label as `String` (read-only)

**Dispatch Id:** 1001

**Description:**

This property gives access to the title of the placeholder. The property is read-only.

#### 15.4.7.5.1.2 PlaceholderWindowID

**Property:** PlaceholderWindowID as [AuthenticDesktopControlPlaceholderWindow](#)<sup>666</sup>

**Dispatch Id:** 1

**Description:**

This property specifies which Authentic Desktop 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 [AuthenticDesktopControlPlaceholderWindow](#)<sup>666</sup> enumeration. The control changes its state immediately and shows the new Authentic Desktop window.

#### 15.4.7.5.1.3 Project

**Property:** Project as `Project` (read-only)

**Dispatch Id:** 2

**Description:**

The `Project` property gives access to the `Project` object of the Authentic Desktop 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](#)<sup>664</sup> with a value of `Authentic DesktopXProjectWindow (=3)`. The property is read-only.

### 15.4.7.5.2 Methods

The following method is defined:

[OpenProject](#)<sup>665</sup>

[CloseProject](#)<sup>665</sup>

### 15.4.7.5.2.1 *OpenProject*

**Method:** `OpenProject (strFileName as string) as boolean`

Dispatch Id: 3

**Description:**

`OpenProject` loads the file `strFileName` as the new project into the control. The method will fail if the placeholder window has a [PlaceholderWindowID](#)<sup>664</sup> different to `Authentic DesktopXProjectWindow (=3)`.

### 15.4.7.5.2.2 *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](#)<sup>664</sup> different to `Authentic DesktopXProjectWindow (=3)`.

### 15.4.7.5.3 *Events*

The `AuthenticDesktopControlPlaceholder` ActiveX control provides following connection point events:

[OnModifiedFlagChanged](#)<sup>665</sup>

#### 15.4.7.5.3.1 *OnModifiedFlagChanged*

**Event:** `OnModifiedFlagChanged (i_bIsModified as boolean)`

**Dispatch Id:** 1

**Description:**

This event gets triggered only for placeholder controls with a [PlaceholderWindowID](#)<sup>664</sup> of `Authentic DesktopXProjectWindow (=3)`. The event is fired whenever the project content changes between modified and unmodified state. The parameter `i_bIsModified` is `true` if the project contents differs from the original content, and `false`, otherwise.

#### 15.4.7.5.3.2 *OnSetLabel*

**Event:** `OnSetLabel (i_strNewLabel as string)`

**Dispatch Id:** 1000

**Description:**

Raised when the title of the placeholder window is changed.

## 15.4.7.6 Enumerations

The following enumerations are defined:

[ICActiveXIntegrationLevel](#) <sup>666</sup>

AuthenticDesktopControlPlaceholderWindow

### 15.4.7.6.1 IActiveXIntegrationLevel

Possible values for the [IntegrationLevel](#) <sup>651</sup> property of the AuthenticDesktopControl.

ICActiveXIntegrationOnApplicationLevel = 0

ICActiveXIntegrationOnDocumentLevel = 1

### 15.4.7.6.2 AuthenticDesktopControlPlaceholderWindow

This enumeration contains the list of the supported additional Authentic Desktop windows.

AuthenticDesktopControlNoToolWnd = -1

AuthenticDesktopControlEntryHelperTopToolWnd = 0

AuthenticDesktopControlEntryHelperMiddleToolWnd = 1

AuthenticDesktopControlEntryHelperBottomToolWnd = 2

AuthenticDesktopControlValidatorOutputToolWnd = 3

AuthenticDesktopControlProjectWindowToolWnd = 4

AuthenticDesktopControlInfoToolWnd = 18

## 16 Anexos

Estos anexos contienen información técnica sobre Authentic Desktop e información importante sobre las licencias. Cada anexo contiene varios apartados.

### [Datos técnicos](#) <sup>668</sup>

- Requisitos de sistema operativo y de memoria
- Analizador XML de Altova
- Motores XSLT y XQuery de Altova
- Compatibilidad con Unicode
- Uso de Internet
- 

### [Información sobre licencias](#) <sup>671</sup>

- Distribución electrónica de software
- Derechos de autor
- Contrato de licencia para el usuario final

## 16.1 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](#) <sup>668</sup>
- [Motores de Altova](#) <sup>668</sup>
- [Compatibilidad con Unicode](#) <sup>669</sup>
- [Uso de Internet](#) <sup>669</sup>

### 16.1.1 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.

### 16.1.2 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.

## 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.

### 16.1.3 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.

### 16.1.4 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 Authentic Desktop, 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.

## 16.2 Información sobre licencias

En esta sección encontrará información sobre:

- la distribución de este producto de software
- 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](#).

### 16.2.1 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](http://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.

## 16.2.2 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 ayude 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](http://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.

### Nota sobre los certificados

Su aplicación de Altova contacta con el servidor de licencias de Altova ([link.altova.com](https://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.

## 16.2.3 Altova Authentic: contrato de licencia para el usuario final

- Encontrará el Contrato de licencia de Altova para el usuario final de Authentic en: <https://www.altova.com/legal/authentic-eula>
- Encontrará la Política de privacidad de Altova en: <https://www.altova.com/privacy>

# Índice

■

## .NET,

- diferencias con la versión independiente de Authentic Desktop, 168
- integración de Authentic Desktop con, 166

## A

### Acceso rápido, 265

- asignar/eliminar, 265

### ActiveX,

- integration at application level, 621
- integration at document level, 623
- integration prerequisites, 618

### ActiveX controls,

- adding to the Visual Studio Toolbox, 619

### Administrador del control de código fuente, 214

### Alias,

- ver Recursos globales, 105

### Altova products, 23

### Altova support, 23

### Analizador,

- XSLT, 286

### Ancho de celda máximo, 286

### Ancho de tabla óptimo, 286

### Anexos, 667

### API,

- documentation, 351
- overview, 352

### API COM,

- en el editor de scripting, 316

### API de XMLSpy,

- documentation, 351
- overview, 352

### Aplicaciones externas,

- abrir archivos en, 264

### Aplicar, 278

### Application,

- ActiveDocument, 385
- AddMacroMenuItem, 385

AddXSLT\_XQParameter, 386

Application, 386

ClearMacroMenu, 386

CurrentProject, 387

Dialogs, 387

Documents, 388

GetDatabaseImportElementList, 388

GetDatabaseSettings, 389

GetDatabaseTables, 389

GetExportSettings, 390

GetTextImportElementList, 390

GetTextImportExportSettings, 391

GetXSLT\_XQParameterCount, 391

GetXSLT\_XQParameterName, 392

GetXSLT\_XQParameterXPath, 392

ImportFromDatabase, 392

ImportFromSchema, 393

ImportFromText, 394

ImportFromWord, 395

NewProject, 396

OnBeforeOpenDocument, 383

OnBeforeOpenProject, 384

OnDocumentOpened, 384

OnProjectOpened, 385

OpenProject, 396

Parent, 396

Quit, 397

ReloadSettings, 397

RemoveXSLT\_XQParameter, 397

RunMacro, 398

ScriptingEnvironment, 398

ShowApplication, 398

ShowForm, 399

URLDelete, 400

URLMakeDirectory, 400

WarningNumber, 401

WarningText, 401

### Archivo, 279

cerrar, 184

codificación predeterminada, 283

codificar, 184

crear nuevo, 177

guardar, 185

guardar por correo electrónico, 190

opciones de impresión, 191

opciones para abrir, 279

sección, 279

### Archivo nuevo,

- Archivo nuevo,**
  - crear, 177
- Archivo PXF,**
  - generar documentos de salida desde la Authentic View, 247
- Archivo XML de plantilla,**
  - en la Authentic View, 37
- Archivo XML de recursos globales, 106**
- Archivos,**
  - agregarlos al control de código fuente, 207
  - usados recientemente, 192
- Archivos de plantilla,**
  - para documentos nuevos, 177
- Archivos no XML, 281**
- Archivos usados recientemente,**
  - lista de, 192
- Asignar,**
  - tecla de acceso rápido a un comando, 265
- ATL,**
  - archivos de muestra para complementos, 338
- Atributos (ayudante de entrada),**
  - en la Authentic View, 60
- Authentic Desktop,**
  - features, 23
  - help, 23
  - integration, 618
  - manual del usuario, 11
- Authentic DesktopCommand,**
  - in AuthenticDesktopControl, 646
- Authentic DesktopCommands,**
  - in AuthenticDesktopControl, 648
- AuthenticDataTransfer,**
  - dropEffect, 403
  - getData, 404
  - ownDrag, 404
  - type, 404
- AuthenticDesktopControl, 649**
  - documentation of, 618
  - examples of integration at document level, 626
  - integration using C#, 626
  - object reference, 646
- AuthenticDesktopControlDocument, 657**
- AuthenticDesktopControlIPlaceHolder, 663**
- AuthenticRange,**
  - AppendRow, 409
  - Application, 410
  - CanPerformAction, 410
  - CanPerformActionWith, 411
  - Close, 411
  - CollapsToBegin, 411
  - CollapsToEnd, 412
  - Copy, 412
  - Cut, 412
  - Delete, 412
  - DeleteRow, 413
  - DuplicateRow, 413
  - ExpandTo, 414
  - FirstTextPosition, 414
  - FirstXMLData, 415
  - FirstXMLDataOffset, 416
  - GetElementAttributeNames, 417
  - GetElementAttributeValue, 418
  - GetElementHierarchy, 418
  - GetEntityNames, 418
  - Goto, 419
  - GotoNext, 420
  - GotoNextCursorPosition, 420
  - GotoPrevious, 421
  - GotoPreviousCursorPosition, 421
  - HasElementAttribute, 422
  - InsertEntity, 422
  - InsertRow, 423
  - IsCopyEnabled, 423
  - IsCutEnabled, 423
  - IsDeleteEnabled, 424
  - IsEmpty, 424
  - IsEqual, 424
  - IsFirstRow, 424
  - IsInDynamicTable, 425
  - IsLastRow, 425
  - IsPasteEnabled, 425
  - IsTextStateApplied, 426
  - LastTextPosition, 426
  - LastXMLData, 427
  - LastXMLDataOffset, 427
  - MoveBegin, 428
  - MoveEnd, 429
  - MoveRowDown, 429
  - MoveRowUp, 429
  - Parent, 430
  - Paste, 430
  - PerformAction, 430
  - Select, 431
  - SelectNext, 432
  - SelectPrevious, 432
  - SetElementAttributeValue, 433
  - SetFromRange, 434

**AuthenticRange,**

Text, 435

**AuthenticView, 451**

Application, 443  
 AsXMLString, 443  
 DocumentBegin, 445  
 DocumentEnd, 445  
 Event, 446  
 Goto, 447  
 IsRedoEnabled, 448  
 IsUndoEnabled, 448  
 Markup Visibility, 449  
 OnBeforeCopy, 436  
 OnBeforeCut, 436  
 OnBeforeDelete, 437  
 OnBeforeDrop, 437  
 OnBeforePaste, 438  
 OnDragOver, 439  
 OnKeyboardEvent, 439  
 OnMouseEvent, 440  
 OnSelectionChanged, 441  
 Parent, 449  
 Print, 449  
 Redo, 449  
 Selection, 450  
 Undo, 451  
 WholeDocument, 452  
 XMLDataRoot, 452

**Ayuda,**

mapa de teclado, 298

**Ayudantes de entrada, 19, 29**

ver, 295

**B****Barra de estado, 20, 30****Barra de herramientas, 20, 30, 262**

activar/desactivar, 262  
 agregar comando a, 260  
 agregar macro a, 270  
 crear nueva, 262  
 mostrar iconos grandes, 274  
 restablecer barra de herramientas y comandos de menú, 262

**Barra de menú, 20, 30****Bases de datos,**

edición en la Authentic View, 240

ver también BD, 84

**BD, 84, 85**

consultas en la Authentic View, 84  
 crear consultas, 85  
 editar en la Authentic View, 84, 89  
 filtrar opciones de visualización en la Authentic View, 85  
 navegar por tablas en la Authentic View, 84  
 parámetros en consultas de BD, 85

**BD XML,**

cargar filas de datos XML nuevas, 84  
 cargar nueva fila de datos en la Authentic View, 241

**Big-endian, 283****Browser View,**

font size, 251  
 moving back and forward, 251  
 refresh content of, 251  
 separate windows, 251  
 stop loading page, 251

**Buscar, 195**

texto en el documento, 194  
 y reemplazar texto en el documento, 195

**Buscar cambios, 279****C****C#,**

integration of Authentic Desktop, 626

**Cambiar de vista,**

a la Authentic View, 54

**Catálogo,**

Oasis XML, 230

**CodeGeneratorDlg,**

Application, 453  
 CPPSettings\_DOMType, 453  
 CPPSettings\_LibraryType, 454  
 CPPSettings\_UseMFC, 455  
 CSharpSettings\_ProjectType, 455  
 OutputPath, 455  
 OutputPathDialogAction, 456  
 OutputResultDialogAction, 456  
 Parent, 456  
 ProgrammingLanguage, 456  
 PropertySheetDialogAction, 457  
 TemplateFileName, 457

**Codificación,**

de archivos, 184

- Codificación,**
    - predeterminada, 283
  - Color de sintaxis, 281, 286**
  - Comando, 268**
    - agregar a menú/barra de herramientas, 260
    - eliminar de menú, 268
    - menú contextual, 268
    - restablecer menú, 268
  - Comandos,**
    - listado del mapa de teclado, 298
  - COM-API,**
    - documentation, 351
  - Command reference, 639**
  - Compatibilidad con Unicode,**
    - de los productos de Altova, 669
  - Complemento,**
    - archivos de muestra ATL, 338
    - configuración de la interfaz del usuario, 335
    - registro, 334
    - XMLSPY, 334
  - Complemento de Authentic para Eclipse,**
    - instalar, 170
  - Complemento de Authentic para VS .NET,**
    - instalar, 167
  - Comprobación de formato XML, 230**
  - Configuración, 22, 32, 278**
    - script, 290
  - Configuración activa,**
    - para recursos globales, 259
  - Configuración de Java, 290**
  - Configuración del programa, 278**
  - Configuraciones,**
    - de un recurso global, 106
  - Configuraciones en recursos globales, 120**
  - Configurar,**
    - interfaz de usuario de XMLSpy, 335
  - Configurar impresión, 191**
  - Conformidad con XML, 281**
  - Consultas,**
    - para visualizar BD en la Authentic View, 85
  - Contrato de licencia para el usuario final, 671**
  - Control de código fuente, 121, 291**
    - abrir proyecto, 201
    - actualizar estado, 214
    - agregar al control de código fuente, 207
    - anular desprotección, 207
    - cambiar de proveedor, 214
    - compartir desde, 208
    - desproteger, 204
    - habilitar, deshabilitar, 202
    - mostrar diferencias, 211
    - mostrar historial, 210
    - obtener archivos, 202
    - obtener la versión más reciente, 202
    - propiedades, 213
    - proveedores compatibles, 200
    - quitar del, 208
  - Controles ActiveX,**
    - compatibilidad, 335
  - Copiar (comando), 193**
  - Correo electrónico,**
    - enviar archivos por, 190
  - Cortar (comando), 193**
  - CPU,**
    - carga - acelerar actualización de estado en segundo plano, 121
  - CR & LF, 279**
  - CustomCatalog, 230**
  - CVS, 121**
- ## D
- DatabaseConnection,**
    - ADOConnection, 458
    - AsAttributes, 459
    - CreateMissingTables, 459
    - CreateNew, 459
    - DatabaseKind, 460
    - ExcludeKeys, 460
    - File, 460
    - IncludeEmptyElements, 461
    - NumberDateTimeFormat, 462
    - ODBCConnection, 462
    - SQLSelect, 463
    - TextFieldLen, 463
  - Delete,**
    - Application.URLDelete, 400
  - Descripción de la IGU, 15, 25**
  - Descripción de la interfaz gráfica, 15, 25**
  - Deshabilitar validación automática, 281**
  - Deshacer (comando), 193**
  - Dialogs,**
    - Application, 464
    - CodeGeneratorDlg, 464

**Dialogs,**

- DTDSchemaGeneratorDlg, 466
- FileSelectionDlg, 465
- GenerateSampleXMLDlg, 465
- Parent, 465
- SchemaDocumentationDlg, 465

**Diccionario, 252**

- añadir diccionario personalizado, 252
- modificar, 252
- revisión ortográfica, 252

**Diccionario personalizado, 252****directories,**

- creating with Application.URLMakeDirectory, 400

**Distribución,**

- de productos de software de Altova, 671

**Document, 479**

- Application, 472
- AssignDTD, 472
- AssignSchema, 472
- AssignXSL, 472
- AssignXSLFO, 473
- AuthenticView, 473
- Close, 474
- ConvertDTDOOrSchema, 474
- CreateChild, 476
- CreateSchemaDiagram, 477
- CurrentViewMode, 477
- DataRoot, 477
- DocEditView, 477
- Encoding, 478
- EndChanges, 478
- ExecuteXQuery, 479
- ExportToDatabase, 479
- ExportToText, 480
- FullName, 481
- GenerateDTDOOrSchema, 482
- GenerateProgramCode, 483
- GenerateSampleXML, 483
- GenerateSchemaDocumentation, 483
- GetExportElementList, 485
- GetPathName, 486
- GridView, 486
- IsModified, 486
- IsValid, 487
- IsWellFormed, 489
- Name, 489
- OnBeforeCloseDocument, 470
- OnBeforeSaveDocument, 469

- OnBeforeValidate, 470
- OnCloseDocument, 471
- OnViewActivation, 471
- Path, 490
- RootElement, 490
- Save, 490
- SaveAs, 491
- Saved, 491
- SaveInString, 491
- SaveToURL, 492
- SetActiveDocument, 492
- SetEncoding, 492
- SetExternalsValid, 493
- SetPathName, 493
- StartChanges, 494
- SwitchViewMode, 494
- Title, 495
- TransformXSL, 495
- TransformXSLFO, 495
- UpdateViews, 496
- UpdateXMLData, 496
- XQuery, 479

**Document-level,**

- examples of integration of <%SPY-GEN%>, 626

**Documento,**

- revisión ortográfica, 252

**Documento de salida HTML,**

- generar en la Authentic View a partir del archivo PXF, 247

**Documento de salida PDF,**

- generar en la Authentic View a partir del archivo PXF, 247

**Documento de salida RTF,**

- generar en la Authentic View a partir del archivo PXF, 247

**Documento de salida Word 2007+,**

- generar en la Authentic View a partir del archivo PXF, 247

**Documento XML,**

- abrir en la Authentic View, 37

**Documentos en la ventana principal, 16, 26****Documents,**

- Count, 498
- Item, 498
- NewAuthenticFile, 498
- NewFile, 498
- NewFileFromText, 499
- OpenAuthenticFile, 499
- OpenFile, 500
- OpenURL, 500
- OpenURLDialog, 501

**DTD, 279, 281**

**DTDSchemaGeneratorDlg,**

Application, 502  
AttributeTypeDefinition, 502  
DTDSchemaFormat, 502  
FrequentElements, 502  
GlobalAttributes, 503  
MaxEnumLength, 503  
MergeAllEqualNamed, 503  
OnlyStringEnums, 504  
OutputPath, 504  
OutputPathDialogAction, 504  
Parent, 504  
ResolveEntities, 505  
TypeDetection, 505  
ValueList, 505

**E****Eclipse platform,**

Authentic Desktop entry points in, 175

**Editado con XMLSPY, 279****Editar,**

botón de macro, 275

**Editor de script,**

iniciar, 258

**Editor de scripting,**

vista general, 308

**Editor predeterminado, 281****ElementList,**

Count, 506  
Item, 506  
RemoveElement, 506

**ElementListItem,**

ElementKind, 507  
FieldCount, 507  
Name, 507  
RecordCount, 507

**Elementos (ayudante de entrada),**

en la Authentic View, 60

**Elementos vacíos, 281****Eliminar, 260**

acceso rápido, 265  
barra de herramientas, 262  
comando de la barra de herramientas, 260  
comando del menú contextual, 268  
icono de la barra de herramientas, 260

**Eliminar (comando), 193****En paralelo, 286****Entidades,**

definir en la Authentic View, 71, 94  
insertar en la Authentic View, 49, 71

**Entidades (ayudante de entrada),**

en la Authentic View, 60

**Entidades externas analizadas, 281****Entorno de scripting, 306****Enumerations,**

in AuthenticDesktopControl, 666  
SPYAttributeTypeDefinition, 604  
SPYAuthenticActions, 605  
SPYAuthenticDocumentPosition, 605  
SpyAuthenticElementActions, 605  
SPYAuthenticElementKind, 605  
SPYAuthenticMarkupVisibility, 606  
SPYDatabaseKind, 606  
SPYDialogAction, 607  
SPYDOMType, 607  
SPYDTDSchemaFormat, 607  
SPYEncodingByteOrder, 607  
SPYExportNamespace, 608  
SPYFrequentElements, 608  
SPYKeyEvent, 609  
SPYLibType, 609  
SPYLoading, 609  
SPYMouseEvent, 610  
SPYNumberDateTimeFormat, 610  
SPYProgrammingLanguage, 611  
SPYProjectItemTypes, 611  
SPYProjectType, 611  
SPYSampleXMLGenerationOptimization, 612  
SPYSampleXMLGenerationSchemaOrDTDAssignment, 612  
SPYSchemaDefKind, 613  
SPYSchemaDocumentationFormat, 613  
SPYTextDelimiters, 614  
SPYTextEnclosing, 614  
SPYTypeDetection, 615  
SPYURLTypes, 615  
SPYViewModes, 616  
SPYVirtualKeyMask, 616  
SPYXMLDataKind, 617

**Equipo virtual Java,**

configurar ruta de acceso, 290

**Esquema,**

configuración, 279

**Event**, 383, 384, 385, 436, 437, 438, 439, 440, 441, 469, 470, 471, 529, 530, 531

**Events**, 358

**Explorador**, 281, 286

Vista, 103

**ExportSettings**,

CreateKeys, 508

ElementList, 508

EntitiesToText, 508

ExportAllElements, 508

FromAttributes, 509

FromSingleSubElements, 509

FromTextValues, 509

IndependentPrimaryKey, 509

Namespace, 509

SubLevelLimit, 510

**Expresiones regulares**,

en la cadena de búsqueda, 194

**Extensiones de archivo**,

personalizar, 230

## F

**Fechas**,

cambiar manualmente, 92

**Fila**,

subir/bajar, 247

**FileSelectionDlg**,

Application, 510

DialogAction, 511

FullName, 511

Parent, 511

**Firmas XML**, 96

**Formato de salida**, 279

**Formatos de imagen**,

en la Authentic View, 98

**Formatos gráficos**,

en la Authentic View, 98

## G

**Generate Sample XML**, 604, 612

**GenerateSampleXMLDlg**,

Application, 524

FillWithSampleData, 525

NonMandatoryAttributes, 526

NonMandatoryElements, 526

Parent, 527

RepeatCount, 527

TakeFirstChoice, 528

**Gestor de esquemas**,

aplicar un parche a un esquema, 156

desinstalar un esquema, 158

estado de los esquemas en el, 154

funcionamiento del, 151

help (comando ILC), 159

info (comando ILC), 160

initialize (comando ILC), 160

instalar un esquema, 156

install (comando ILC), 161

introducción, 147

introducción a la ILC, 159

list (comando ILC), 161

lista de esquemas por estado, 154

reset (comando ILC), 162

restaurar, 158

uninstall (comando ILC), 163

update (comando ILC), 164

upgrade (comando ILC), 164

**Global**,

configuración, 278

**Gramática**, 281

**GridView**,

CurrentFocus, 531

Deselect, 531

IsVisible, 532

OnBeforeDrag, 529

OnBeforeDrop, 529

OnBeforeStartEditing, 530

OnEditingFinished, 530

OnFocusChanged, 531

Select, 532

SetFocus, 532

**Grupo de inicio**,

agregar (menú contextual), 275

**Guardar archivos**,

codificación de, 184

## H

**Herramientas**,

**Herramientas,**

ver también Aplicaciones externas, 264

**Hipervínculos,**

navegar por la Authentic View, 71

**Hojas de estilos XSLT para FO,**

asignar al documento XML, 233

**I****Icono,**

agregar a menú/barra de herramientas, 260

mostrar en tamaño grande, 274

**Imprimir,**

desde la Authentic View, 50

**Información general, 668****Información legal, 671****Información rápida,**

mostrar, 274

mostrar teclas de acceso rápido en, 274

**Información sobre derechos de autor, 671****Información técnica, 668****Instrucción de procesamiento stylesheet, 233****Integrating,**

Authentic Desktop in applications, 618

**Interfaz del usuario,**

configurar usando complementos, 335

**J****Java, 630****JRE,**

para el complemento de Authentic para Eclipse, 170

**Juego de caracteres,**

codificación, 283

**L****Lenguaje de scripting, 290****Licencia,**

información sobre, 671

**Little-endian, 283****loading, 500****M****Macro,**

agregar a menú/barra de herramientas, 270

botón, 275

**Macros,**

desarrollar, 312

ejecutar, 332

ejecutar macros de aplicación, 258

habilitar, 319, 331

**MainCatalog, 230****Manual del usuario, 11**

Authentic Desktop, 12

**Mapa de teclado, 298****Marcado,**

en la Authentic View, 54, 57

**Marcado (en la Authentic View),**

mostrar en tamaño pequeño/grande/mixto, 246

ocultar, 246

**Marcado en tamaño grande (en la Authentic View), 246****Marcado en tamaño pequeño (en la Authentic View), 246****Marcado mixto (en la Authentic View), 246****Medición de licencias,**

en los productos de Altova, 672

**Mensajes de validación, 20, 30****Menú, 268**

agregar macro a, 270

agregar/eliminar comando, 260

Authentic, 239

Edición, 193

eliminar comandos de, 268

Herramientas, 252

personalizar, 268

predeterminado/XMLSPY, 268

Proyecto, 196

XML, 230

XSL/XQuery, 232

**Menú Archivo, 177****Menú Authentic, 239**

edición de tablas dinámicas, 54

presentación del marcado, 54

**Menú contextual,**

comandos, 268

para personalización, 275

**Menú Edición, 193**  
**Menú Herramientas, 252**  
**Menú Proyecto, 196**  
**Menú Ventanas, 295**  
**Menú XML, 230**  
**Menú XSL/XQuery, 232**  
**Menús contextuales,**  
   en la Authentic View, 65  
**Microsoft® SharePoint® Server, 220**  
**MIME, 281**  
**Mostrar marcado en tamaño grande, 54, 57**  
**Mostrar marcado en tamaño pequeño, 54, 57**  
**Mostrar marcado mixto, 54, 57**  
**Motores,**  
   de los productos de Altova, 668  
**MS Visual Source Safe, 121**  
**MSXML, 286**  
**Multiusuario, 279**

## O

**OASIS,**  
   Catálogo XML, 230  
**Ocultar marcado, 54, 57**  
**Ocultar marcado (en la Authentic View), 246**  
**Ocultar ventanas automáticamente, 15, 25**  
**Opciones de impresión, 191**  
**Opciones de la copia de seguridad automática, 279**  
**Opciones de ortografía, 255**  
**Opciones de soporte, 33**  
**Opciones de validación, 279**  
**Opciones para abrir,**  
   archivos, 279  
**Overview,**  
   of API de XMLSpy, 352

## P

**Paquete de integración de Authentic, 167, 170**  
**Parámetros,**  
   en las consultas de BD, 85  
   pasarlos a la hoja de estilos a través de la interfaz, 234  
**Parámetros XSLT,**  
   pasarlos a la hoja de estilos a través de la interfaz, 234  
**Parent, 490**  
**Pegar,**  
   como texto, 71  
   como XML, 71  
**Pegar (comando), 193**  
**Pegar como,**  
   Texto, 65  
   XML, 65  
**Período de evaluación,**  
   de los productos de software de Altova, 671  
**Personalización, 22, 32**  
**Personalizar, 268**  
   comandos de menú/barra de herramientas, 260  
   macros, 270  
   menú, 268  
   menú contextual, 268  
   menú contextual Personalizar, 275  
**Perspectiva Authentic en Eclipse, 172**  
**Plantillas,**  
   de documentos XML en la Authentic View, 240  
**Plantillas en la Authentic View, 37**  
**Plataforma Eclipse,**  
   perspectiva Authentic, 172  
   y Authentic Desktop, 169  
   y el paquete de integración de Authentic, 170  
**Plataformas,**  
   para productos de Altova, 668  
**Predeterminada,**  
   codificación, 283  
**Predeterminado,**  
   menú, 268  
**Presentación, 286**  
**Procesador XQuery,**  
   de los productos de Altova, 668  
**Procesador XSL externo, 286**  
**Procesadores XSLT,**  
   de los productos de Altova, 668  
**Productos de Altova, 33**  
**Proveedor,**  
   de control de código fuente, 121  
**Proyecto,**  
   abrir, 199  
   agregar archivo activo al, 216  
   agregar archivos al, 215  
   agregar carpetas al, 216  
   agregar carpetas externas al, 217  
   agregar carpetas web externas al, 220  
   agregar recursos globales al, 215

**Proyecto,**

- agregar URL al, 215
- agregarlo al control de código fuente, 207
- cerrar, 199
- crear nuevo, 199
- guardar, 200
- propiedades, 226
- volver a cargar, 199

**Proyectos,**

- agregar archivos relacionados al, 216
- introducción, 196
- usados recientemente, 229

**PUBLIC,**

- identificador - catálogo, 230

**PVCS Version Manager, 121**

## R

**Recargar, 279****Recursos,**

- acelerar actualización de estado en segundo plano, 121

**Recursos globales, 105**

- activación de barras de herramientas, 262
- cambiar de configuración, 120
- configuración activa para, 259
- definir, 106, 258
- definir el tipo de archivo, 108
- definir el tipo de base de datos, 114
- definir el tipo de carpeta, 113
- usar, 117, 120
- usar tipo de archivo y tipo de carpeta, 117

**Recursos globales de Altova,**

- ver Recursos globales, 105

**Reemplazar,**

- texto, 194
- texto en el documento, 195

**Referencia para programadores, 304****Registrar,**

- complemento, 334

**Registro,**

- configuración, 278

**Rehacer (comando), 193****Repositorios, 121****Requisitos de memoria, 668****Restablecer,**

- acceso rápido, 265

- barras de herramientas y comandos de menú, 262
- comandos de menú, 268

**Retorno de carro,**

- ver Tecla Entrar, 99

**Revisar,**

- ortografía, 252

**Revisión ortográfica, 252**

- diccionario personalizado, 252

**RichEdit, 246**

## S

**Saltos de línea, 279****save, 492****schema, 393****SchemaDocumentationDlg,**

- AllDetails, 533
- Application, 534
- IncludeAll, 535
- IncludeAttributeGroups, 536
- IncludeComplexTypes, 536
- IncludeGlobalElements, 536
- IncludeGroups, 537
- IncludeIndex, 537
- IncludeLocalElements, 538
- IncludeRedefines, 538
- IncludeSimpleTypes, 538
- OptionsDialogAction, 539
- OutputFile, 539
- OutputFileDialogAction, 540
- OutputFormat, 540
- Parent, 540
- ShowAnnotations, 540
- ShowAttributes, 541
- ShowChildren, 541
- ShowConstraints, 542
- ShowDiagram, 541
- ShowEnumerations, 542
- ShowNamespace, 542
- ShowPatterns, 543
- ShowProgressBar, 543
- ShowProperties, 543
- ShowResult, 543
- ShowSingleFacets, 544
- ShowSourceCode, 544
- ShowType, 544

**SchemaDocumentationDlg,**

ShowUsedBy, 545

**Script, 290****Scripts de Authentic,**

configuración de seguridad, 248

ubicaciones de confianza, 248

**Secciones CDATA,**

insertar en la Authentic View, 71

**Segundo plano,**

actualización de estado - aumentar el intervalo, 121

**Seleccionar todo (comando), 194****Selector de fecha,**

usar en la Authentic View, 91

**SharePoint® Server, 220****SO,**

para productos de Altova, 668

**Soporte técnico de Altova, 33****SPS,**

asignar a un archivo XML nuevo, 177

**SpyProject,**

CloseProject, 546

ProjectFile, 546

RootItems, 546

SaveProject, 546

SaveProjectAs, 547

**SpyProjectItem,**

ChildItems, 547

FileExtensions, 547

ItemType, 548

Name, 548

Open, 548

ParentItem, 548

Path, 548

ValidateWith, 548

XMLForXSLTtransformation, 549

XSLForXMLTtransformation, 549

XSLTransformationFileExtension, 549

XSLTransformationFolder, 549

**SpyProjectItems,**

AddFile, 550

AddFolder, 550

AddURL, 550

Count, 551

Item, 551

RemoveItem, 551

**StarTeam, 121****Subir/bajar,**

fila (Authentic View), 247

**Support options, 23****T****Tabla,**

generar automáticamente, 281

**Tablas,**

editar tablas dinámicas (SPS), 54

en la Authentic View, 42

**Tablas (SPS) dinámicas en la Authentic View,**

uso de, 76

**Tablas (SPS) estáticas en la Authentic View,**

uso de, 76

**Tablas dinámicas,**

edición, 54

**Tablas en la Authentic View,**

iconos para editar tablas XML, 81

usar tablas SPS (estáticas y dinámicas), 76

usar tablas XML, 77

uso de, 76

**Tablas SPS,**

editar tablas dinámicas, 54

**Tablas SPS en la Authentic View,**

uso de, 76

**Tablas XML en la Authentic View,**

iconos para editar, 81

uso de, 77

**Tecla de acceso rápido, 265****Tecla Entrar,**

efectos de usar, 99

**Teclas de acceso rápido, 265**

mostrar en información rápida, 274

**terminate, 397****TextImportExportSettings,**

DestinationFolder, 552

EnclosingCharacter, 552

Encoding, 552

EncodingByteOrder, 552

FieldDelimiter, 552

FileExtension, 553

HeaderRow, 553

ImportFile, 553

**Texto,**

buscar en el documento, 194

buscar y reemplazar, 195

editar en la Authentic View, 71

**Texto,**

formato en la Authentic View, 71

**Tipos de archivo, 281****Título de la pantalla, 286****Transformación XSLT, 232****U****Ubicaciones de confianza para scripts de Authentic, 248****Ubicaciones del archivo SPP, 196****UCS-2, 283****URL, 400, 492, 500, 501**

enviar por correo electrónico, 190

**User Manual of Authentic Desktop, 13****Uso de Internet,**

en los productos de Altova, 669

**UTF-16, 283****V****Validación, 22, 32, 230****Validación automática, 281****Validar al modificar el comando, 231****Valores de atributo,**

introducir en la Authentic View, 48

**Velocidad,**

acelerar actualización de estado en segundo plano, 121

**Ventana Información, 19, 29**

ver, 295

**Ventana Mensajes, 20, 30**

ver, 295

**Ventana principal, 16, 26****Ventana Proyecto, 17, 27**

ver, 295

**Ventana Resultados,**

ver, 295

**Ventanas,**

activar/desactivar, 295

administrar la visualización de, 15, 25

colocar, 295

en cascada, 295

en mosaico, 295

flotantes, acopladas, en pestañas, 15, 25

ocultar automáticamente, 15, 25

**Ventanas acopladas, 15, 25****Ventanas flotantes, 15, 25****Vista,**

Explorador, 103

**Vista Authentic,**

abrir archivo XML nuevo en, 240

abrir un documento XML en, 37

agregar nodos, 42

aplicar elementos, 42

ayudantes de entrada, 39

ayudantes de entrada en, 60

borrar elementos, 42

caracteres especiales en la, 45

edición de datos de BD en, 240

editar datos en una BD XML, 241

entidades en la, 45

formato de texto en, 54

generar documentos de salida a partir del archivo PXF, 247

iconos de la barra de herramientas, 54

iconos para tablas XML, 81

imprimir un documento XML desde la, 50

insertar entidades en la, 49

insertar nodos, 42

interfaz, 51

interfaz gráfica del usuario (IGU), 52

introducir datos en la, 45

introducir valores de atributo, 48

menú contextual, 39

menús contextuales, 65

mostrar etiquetas de marcado, 39

pegar como XML o como texto, 65

presentación de marcado en, 57

presentación del marcado en, 54

quitar nodos, 42

secciones CDATA en la, 45

tablas (SPS y XML), 76

tablas en la, 42

Tablas SPS, 76

tablas XML, 77

Tutorial, 35

uso de funciones importantes, 68

uso de tablas XML, 77

ventana principal de, 57

visualización de documentos, 57

**Vista Explorador, 103****Vista predeterminada,**

configuración de la ventana principal, 281

**Vista previa de los atributos, 286**

**Visual Studio,**

adding the Authentic Desktop ActiveX Controls to the toolbox, 619

**Visual Studio .Net,**

diferencias con la versión independiente de Authentic Desktop, 168  
y Authentic Desktop, 166

**Volcar a cargar,**

archivos cambiados, 183

**VS .NET,**

y el paquete de integración de Authentic, 167

**W****Windows,**

compatibilidad con productos de Altova, 668

**X****XML,**

Catálogo Oasis, 230  
revisión ortográfica, 252

**XMLData,**

AppendChild, 594  
EraseAllChildren, 595  
EraseCurrentChild, 596  
GetChild, 596  
GetChildKind, 597  
GetCurrentChild, 598  
GetFirstChild, 598  
GetNextChild, 599  
HasChildren, 600  
HasChildrenKind, 601  
InsertChild, 601  
IsSameNode, 602  
Kind, 602  
MayHaveChildren, 602  
Name, 603  
Parent, 603  
TextValue, 604

**XMLSPY,**

ayuda, 33  
complemento, 334  
funciones, 33  
registro del complemento, 334

**XMLSpyLib, 351, 352**

Application, 382  
AuthenticDataTransfer, 403  
AuthenticRange, 407  
AuthenticView, 435  
CodeGeneratorDlg, 452  
DatabaseConnection, 457  
Dialogs, 464  
Document, 467  
Documents, 497  
DTDSchemaGeneratorDlg, 501  
ElementList, 505  
ElementListItem, 506  
ExportSettings, 507  
FileSelectionDlg, 510  
GenerateSampleXMLDlg, 523  
GridView, 529  
ProjectItem, 547  
SchemaDocumentationDlg, 532  
SpyProject, 545  
SpyProjectItems, 549  
TextImportExportSettings, 551  
XMLData, 593

**XPath al nodo seleccionado, 52****XQuery,**

pasar variables al documento XQuery, 234

**XSLT,**

procesador, 286