

Altova Authentic 2024 Browser Edition



Manual del usuario y referencia

Altova Authentic 2024 Browser Edition Manual del usuario y referencia

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

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

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

Fecha de publicación: 2024

© 2018-2024 Altova GmbH

Contenido

1	Introducción	6
1.1	Ventajas de Authentic Browser.....	7
1.2	Funcionamiento.....	8
1.3	Versiones de Authentic Browser.....	10
1.4	Información sobre este manual.....	12
2	Configuración del servidor	13
2.1	IIS: Configurar el servicio de explorador.....	14
2.2	Archivos XSD, XML y SPS/PXF.....	18
2.3	Página HTML para el complemento Authentic.....	20
2.3.1	Licencia para la edición Enterprise.....	21
2.3.2	Página HTML para Internet Explorer.....	22
2.3.3	Página HTML para otros exploradores.....	31
2.4	Paquetes de extensión para instalación por encargo.....	36
3	Configuración del cliente	37
3.1	Requisitos del explorador.....	38
3.2	Complemento para exploradores Authentic Browser.....	39
3.2.1	Instalación por encargo.....	39
3.2.2	Instalación manual con MSI.....	39
3.2.3	Instalación de inserción con MSI.....	40
3.2.4	Actualizaciones automáticas.....	41
3.2.5	Desinstalación y deshabilitación.....	41
3.3	Configuración de seguridad en IE9.....	42
3.4	Configuración de seguridad en IE10.....	44
4	Referencia del usuario	45

4.1	Mecanismos.....	46
4.1.1	Eventos: Punto de conexión para IE.....	46
4.1.2	Eventos: Botones de la barra de herramientas.....	47
4.1.3	Eventos: Referencia.....	48
4.1.4	Acceder al contenido del documento y modificarlo.....	49
4.1.5	Operaciones de edición.....	49
4.1.6	Buscar y reemplazar.....	49
4.1.7	Operaciones en filas de tablas.....	50
4.1.8	Teclas de acceso rápido.....	50
4.1.9	Botones de estado del texto.....	51
4.1.10	Ayudantes de entrada.....	51
4.1.11	Paquetes.....	52
4.1.12	Interfaz XMLData.....	57
4.1.13	DOM y XMLData.....	60
4.1.14	Scripting de Authentic.....	64
4.2	Objetos.....	66
4.2.1	Authentic.....	66
4.2.2	AuthenticCommand.....	92
4.2.3	AuthenticCommands.....	93
4.2.4	AuthenticContextMenu.....	94
4.2.5	AuthenticDataTransfer.....	95
4.2.6	AuthenticEvent.....	97
4.2.7	AuthenticEventContext.....	101
4.2.8	AuthenticLoadObject.....	104
4.2.9	AuthenticRange.....	104
4.2.10	AuthenticSelection.....	133
4.2.11	AuthenticToolbarButton.....	134
4.2.12	AuthenticToolbarButtons.....	134
4.2.13	AuthenticToolbarRow.....	137
4.2.14	AuthenticToolbarRows.....	138
4.2.15	AuthenticView.....	139
4.2.16	AuthenticXMLTableCommands.....	160
4.2.17	XMLData.....	168
4.3	Enumeraciones.....	178

4.3.1	SPYAuthenticActions.....	178
4.3.2	SPYAuthenticCommand.....	178
4.3.3	SPYAuthenticCommandGroup.....	180
4.3.4	SPYAuthenticDocumentPosition.....	180
4.3.5	SPYAuthenticElementActions.....	180
4.3.6	SPYAuthenticElementKind.....	181
4.3.7	SPYAuthenticEntryHelperWindows.....	181
4.3.8	SPYAuthenticMarkupVisibility.....	181
4.3.9	SPYAuthenticToolBarAlignment.....	182
4.3.10	SPYAuthenticToolBarButtonState.....	182
4.3.11	SPYXMLDataKind.....	182

5 Información sobre licencias 184

5.1	Altova Authentic: contrato de licencia para el usuario final.....	185
-----	-------------------------------------------------------------------	-----

Índice 186

1 Introducción

Altova® Authentic® 2024 Browser Edition permite crear y editar contenido XML y de BD desde una interfaz de usuario muy parecida a la de un procesador de texto. La edición Browser Edition se puede incrustar en cualquier página web para que el usuario edite el contenido desde el explorador web. La edición Browser Edition está disponible en forma de complemento para el explorador **Microsoft Internet Explorer**.



Authentic Browser está disponible en dos versiones distintas: [una versión segura \(trusted\)](#) y [una versión no segura \(untrusted\)](#)¹⁰. Para ambas versiones es necesario adquirir una licencia.

1.1 Ventajas de Authentic Browser

Estas son las principales ventajas del complemento Authentic Browser para edición XML en exploradores web:

- Permite a varios acceder a documentos XML y editarlos desde el explorador web. De momento, es compatible con **Microsoft Internet Explorer 5.5 o versiones superiores**. Las versiones de 64 bits de Internet Explorer 10 y 11 no son compatibles.
- Facilita la implementación en toda la organización y su mantenimiento mientras reduce el coste total de propiedad.
- Está basado en estándares abiertos como XML Schema y XSLT.
- Es totalmente compatible con Unicode.
- Utiliza la vista Authentic de Altova, que se basa en el explorador Internet Explorer. La vista Authentic sirve para editar archivos XML en una interfaz WYSIWYG (es decir, el usuario no ve el código XML subyacente).
- No requiere la implementación de otro software porque Authentic Browser es un complemento para exploradores web.
- Authentic Browser es un control ActiveX en el que el objeto [Authentic](#)⁶⁶ define la interfaz COM. El modelo de objetos completo se describe en la sección [Referencia del usuario: Objetos](#)⁶⁶ de este manual.

1.2 Funcionamiento

Para implementar un proyecto de Authentic Browser necesitará un equipo servidor y, como mínimo, un equipo cliente conectado al servidor.

Servidor de Authentic Browser

El servidor de Authentic Browser lleva a cabo estas funciones:

- El servidor almacena los archivos relacionados con el documento XML que se va a editar. Se trata de estos archivos:
 1. El esquema XML (XSD) en el que se basa el documento XML editable.
 2. El documento XML que se va a editar.
 3. El archivo SPS o PXF que controla el diseño y los mecanismos de entrada del documento XML en la vista Authentic.
- El servidor también almacena la [página HTML para el complemento Authentic](#)²⁰. Esta página HTML es el punto de acceso para la edición en la vista Authentic. Contiene instrucciones para acceder al documento XML y sirve de contenedor de la ventana de la vista Authentic en la que se carga y edita el documento XML. Para acceder a esta página se debe escribir su URL en los exploradores clientes.
- Si implementa la edición Enterprise de Authentic Browser, la licencia de la edición Enterprise se debe almacenar en el servidor. Altova puede emitir licencias de la edición Enterprise Edition para uno o varios servidores.
- Si tiene pensado instalar el complemento Authentic Browser por encargo, el servidor almacena los paquetes de extensión del complemento (archivos CAB) para la descarga e instalación del complemento en equipos cliente. De lo contrario, el complemento se instala directamente en el explorador web cliente.

Las instrucciones para preparar el servidor se describen en la sección [Configuración del servidor](#)¹³.

Clientes de Authentic Browser

En la vista Authentic puede ver y editar documentos XML que tengan asignados un archivo SPS o PXF. La edición en la vista Authentic se lleva a cabo en equipos cliente que deben configurarse de la siguiente manera:

- Cada cliente debe tener instalado **Internet Explorer 5.5 o superior (32 o 64 bits)**.
- Si no tiene pensada una instalación por encargo, instale el complemento Authentic Browser en los exploradores cliente directamente.

Las instrucciones para preparar los clientes se describen en la sección [Configuración del cliente](#)³⁷.

Mecanismo de Authentic Browser

Cuando el servidor y los clientes ya están preparados, el usuario introduce la URL de la página HTML del complemento Authentic en el explorador cliente. Si el complemento no está instalado todavía en el explorador cliente, la página HTML puede incluir instrucciones para realizar una instalación por encargo.

Cuando el complemento ya está instalado en el cliente, el código de la página HTML abre una ventana de edición de la vista Authentic dentro de la ventana del explorador. El documento XML que desea editar se carga

en la ventana de la vista Authentic desde el servidor y el usuario puede empezar a editarlo y a guardar cambios en el documento XML directamente.

1.3 Versiones de Authentic Browser

Altova ofrece estas versiones del complemento Authentic Browser para exploradores web:

- *Idioma:* Altova ofrece versiones en español (ES), inglés (EN), alemán (DE), francés (FR) y japonés (JA).
- *Seguridad:* Altova ofrece versiones seguras (Trusted) y no seguras (Untrusted).
- *Explorador:* Altova ofrece complementos para Microsoft Internet Explorer (32 y 64 bits). Consulte también el apartado [Requisitos del explorador](#)³⁸.

Hay versiones en diferentes idiomas (español, inglés, alemán, francés japonés) y cada una de ellas tiene versiones seguras y no seguras para los exploradores Internet Explorer de 32 bits y de 64 bits. En cada equipo cliente puede instalar una, varias o todas las versiones del complemento Authentic Browser. Cada versión aparece por separado en el administrador de complementos del explorador web.

A continuación puede ver las versiones en inglés (EN), alemán (DE), español (ES), francés (FR) y japonés (JA) con sus IDs de clase (para archivos CAB). El ID de clase correspondiente debe especificarse en la [página HTML para el complemento Authentic](#)²⁰.

- **Archivos CAB y sus IDs de clase (en Internet Explorer 32/64 bits se usa el mismo ID de clase)**

EN	Trusted	B4628728-E3F0-44a2-BEC8-F838555AE780
EN	Untrusted	A5985EA9-3332-4ddf-AD7F-F6E98BF9AF94
DE	Trusted	91DDF44A-DFD1-4F47-8EE3-4CBE874584F7
DE	Untrusted	28A640E8-EAEE-4B5D-BEBE-BFA956081E66
ES	Trusted	23B503E7-269B-45CE-BAB2-22AA97BED8E2
ES	Untrusted	8AD3EF86-AC1E-4574-8C13-DE5B6CBECEBE
FR	Trusted	1B768F46-A9E8-4a88-91B0-2917FE47612A
FR	Untrusted	070F4B50-26F7-48cc-9AC1-52D562C1E749
JA	Trusted	5B15DB5A-1720-4264-BB65-70C3F7A860DA
JA	Untrusted	4B9512D2-A3D3-46e3-82C1-34248BBDCE58

Todas estas versiones se pueden descargar del [sitio web de Altova](#).

Debe tener en cuenta algunos aspectos de las versiones del complemento Authentic Browser:

- Todas las versiones son Unicode y son totalmente compatibles con múltiples juegos de caracteres del documento XML. Las versiones Unicode son compatibles con estas plataformas: Windows 10, Windows 11.
- También hay un archivo CAB distinto para cada versión Internet Explorer de 32 y 64 bits. Los IDs de clase de los dos archivos CAB (para exploradores IE de 32 y 64 bits) son idénticos, es decir, los que aparecen en la tabla anterior para las versiones ES/EN/FR/DE/JA seguras y no seguras.

- La **versión segura Trusted** no permite acceder a archivos locales y, por tanto, se considera segura para secuencias de comandos. Se puede usar en casos de uso basados en exploradores web y se puede invocar desde cualquier página web sin causar alertas de seguridad en el lado cliente.
- La **versión no segura Untrusted** está diseñada para la implementación en intranets o para usar el complemento Authentic Browser como control ActiveX en su aplicación. Ofrece acceso a archivos locales y, por tanto, no se considera segura para secuencias de comandos. Si intenta usar esta versión desde una ventana del explorador, aparecerá un aviso. Puede elegir si los controles ActiveX se deben habilitar o deshabilitar o si el explorador debe pedir permiso para habilitarlos (consulte el apartado sobre [Internet Explorer 9](#)⁴² para obtener más información).

Nota: Para instalar y configurar el servicio de explorador (p.ej. Internet Information Services de Microsoft) consulte la documentación del proveedor.

1.4 Información sobre este manual

Este es el manual del usuario del complemento Authentic Browser. El manual se divide en varias secciones:

- La [introducción](#)⁶ explica (i) las [ventajas de usar Authentic Browser](#)⁷, (ii) su [funcionamiento](#)⁸ y (iii) las diferencias entre las [versiones de Authentic Browser](#)¹⁰.
- La sección [Configuración del servidor](#)¹³ describe los pasos necesarios para configurar un servidor para un proyecto de Authentic Browser e incluye una descripción detallada de la [página HTML para el complemento Authentic](#)²⁰ y una sección que explica cómo usar los paquetes de extensión de Authentic Browser para [instalaciones por encargo](#)³⁶.
- La sección [Configuración del cliente](#)³⁷ explica cómo [instalar el complemento Authentic Browser en los exploradores cliente](#)³⁹.
- La referencia del usuario se divide en tres grandes secciones ([Mecanismos](#)⁴⁶, [Objetos](#)⁶⁶ y [Enumeraciones](#)¹⁷⁸) que describen los mecanismos, los objetos y las enumeraciones que se utilizan para crear y personalizar la vista Authentic en Authentic Browser.

Documentación relacionada

También puede leer la documentación de Altova sobre estos productos y componentes relacionados:

- La documentación para crear archivos StyleVision Power Stylesheet (SPS) en Altova StyleVision disponible en el [sitio web de Altova](#). Los archivos SPS controlan la vista Authentic de un documento XML. Consulte esta documentación para desarrollar la interfaz de la vista Authentic para editar datos XML.
- La documentación para usar vista Authentic. Para las personas que usan vista Authentic podría ser oportuno ver los tutoriales y las descripciones de uso de vista Authentic en el manual de usuario de Authentic Desktop que está disponible en el [sitio web de Altova](#).

2 Configuración del servidor

Varios pasos son necesarios para configurar el servidor para el complemento Authentic Browser.

Configurar el servicio de explorador

Instale y [configure el servicio de explorador de su servidor](#)¹⁴. Si usa Internet Information Services (IIS) de Microsoft, tenga en cuenta que durante la instalación se crea el directorio predeterminado `Inetpub`, que contiene varias subcarpetas. El directorio raíz del servidor sería entonces: `//Inetpub/wwwroot`. Este es el directorio al que se llega con la dirección IP del servidor, a no ser que indique otro directorio como directorio raíz (en la configuración del servicio de explorador). Para más información sobre cómo instalar y configurar el servicio de explorador, consulte la documentación del proveedor.

Configurar los archivos XSD, XML y SPS/PXF

Los archivos SPS permiten representar documentos XML en un formato editable en la vista Authentic. Están basados en esquemas XML (archivos XSD) y se diseñan con [Altova StyleVision](#). El archivo SPS, junto con el archivo XSD y el documento XML que se debe editar, debe almacenarse en una ubicación de la red a la que tengan acceso todos los equipos clientes (por lo general, se almacenan en el servidor de Authentic Browser). El apartado [Archivos XSD, XML y SPS/PXF](#)¹⁸ describe cómo preparar el servidor para esto.

Crear la página HTML para el complemento Authentic Browser

La [página HTML para el complemento Authentic](#)²⁰ es el punto de acceso para editar en la vista Authentic. Contiene instrucciones para acceder al documento XML y sirve de contenedor de la ventana de la vista Authentic en la que se carga y edita el documento XML. Para acceder a esta página se debe escribir su URL en los exploradores clientes. Esta página HTML debe crearse correctamente y debe almacenarse en el servidor. Para más información sobre cómo crear la página HTML consulte el apartado [Página HTML para el complemento Authentic](#)²⁰.

Almacenar paquetes de extensión para instalación por encargo

Descargue el complemento Authentic Browser (un archivo comprimido CAB) del sitio web de Altova y guárdelo en cualquier ubicación del servidor. Si quiere implementar la edición Enterprise de Authentic Browser debe almacenar el paquete de instalación en el servidor para el que se registró el producto. **No desempaquete este archivo.** Por cada complemento de Authentic Browser hay una versión segura y una versión no segura en todos los idiomas disponibles (español, inglés, francés, alemán y japonés) en cuatro formatos diferentes (CAB de 32 y 64 bits). Para más información sobre los formatos de archivo y las diferentes [versiones](#)¹⁰, consulte los apartados de esta sección.

Nota: Antes de instalar el complemento Authentic Browser, compruebe que no hay ninguna versión antigua de Authentic Browser en ejecución. De lo contrario, la versión más reciente no se registrará correctamente y la instalación será defectuosa. Si esto ocurre, registre el complemento ejecutando:
`regsvr32 C:\Windows\Downloaded Program Files\AuthenticPlugin.dll.` (Recuerde que se necesitan privilegios de administrador para ejecutar el programa `regsvr32.exe`.)

2.1 IIS: Configurar el servicio de explorador

Internet Information Services (IIS) 6 de Microsoft solamente sirve los tipos de archivo que se definen en los tipos MIME del sitio (sitio web o carpeta). Por tanto, los tipos de archivo necesarios deben añadirse a la lista de tipos MIME del sitio.

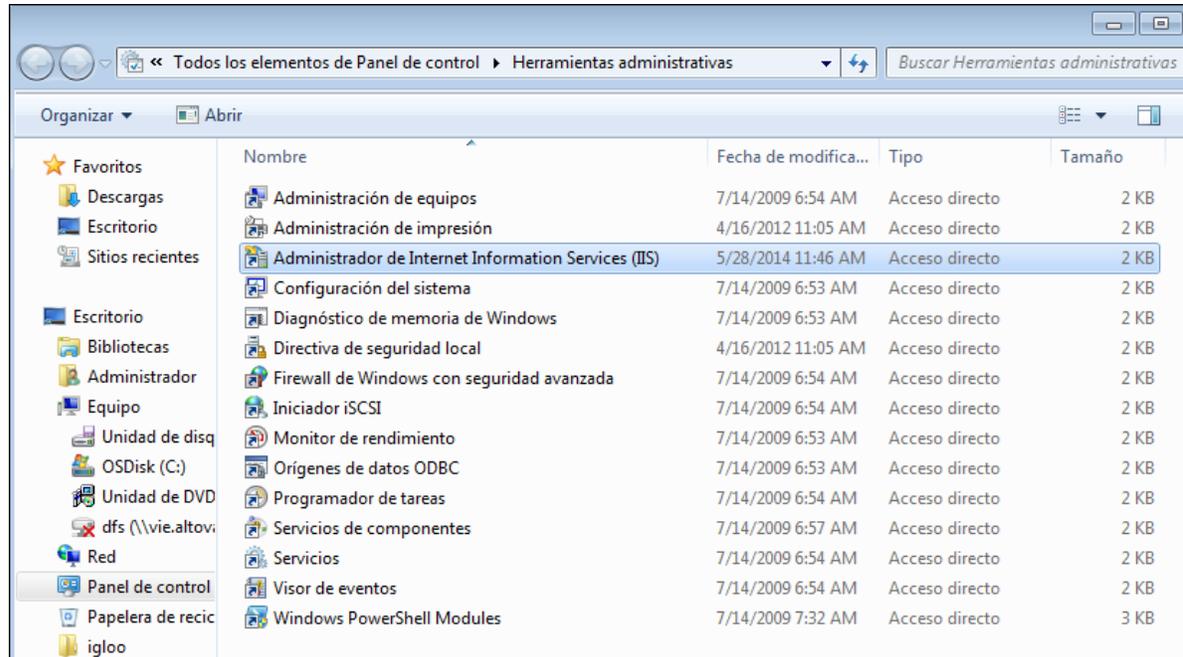
Para trabajar con el complemento Authentic Browser son necesarios estos tipos de archivo:

Extensión	Tipo MIME	Comentario
xsd	text/plain	
sps	text/plain	
pxf	application/x-zip-compressed	

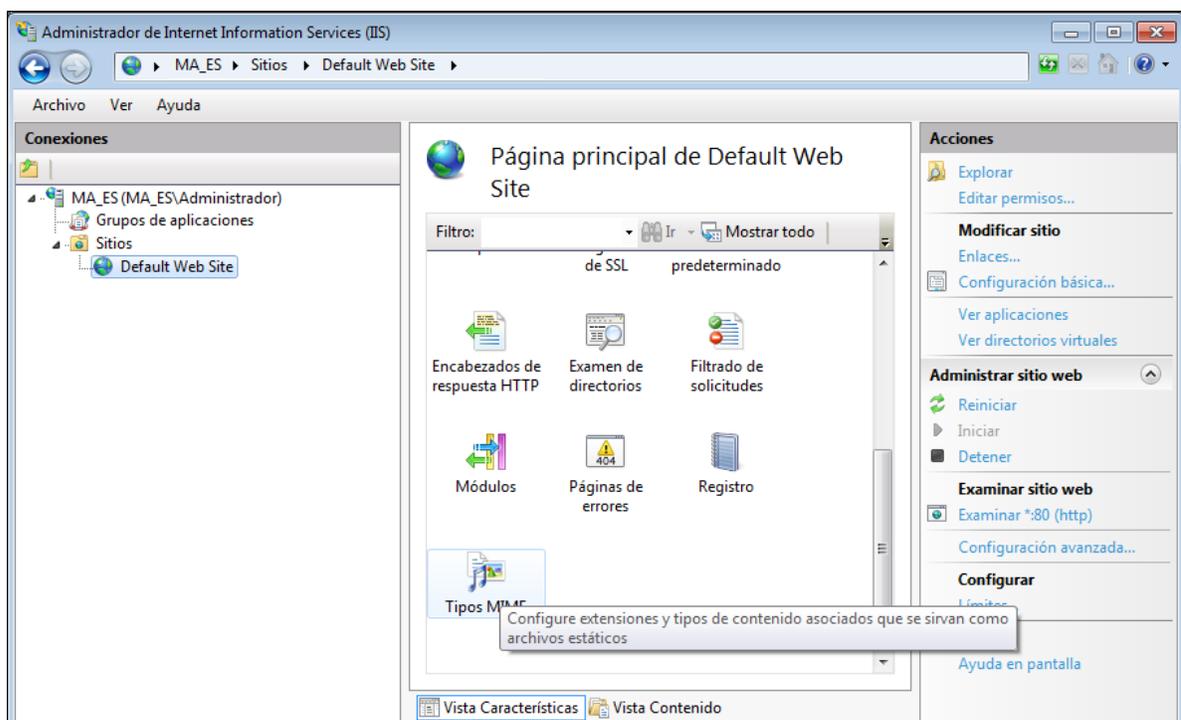
Agregar tipos MIME para un sitio en Internet Information Services

A continuación explicamos cómo agregar un tipo MIME a la lista de tipos MIME de un sitio concreto en un equipo Windows XP. El proceso es similar en los demás sistemas compatibles (Windows 10, Windows 11).

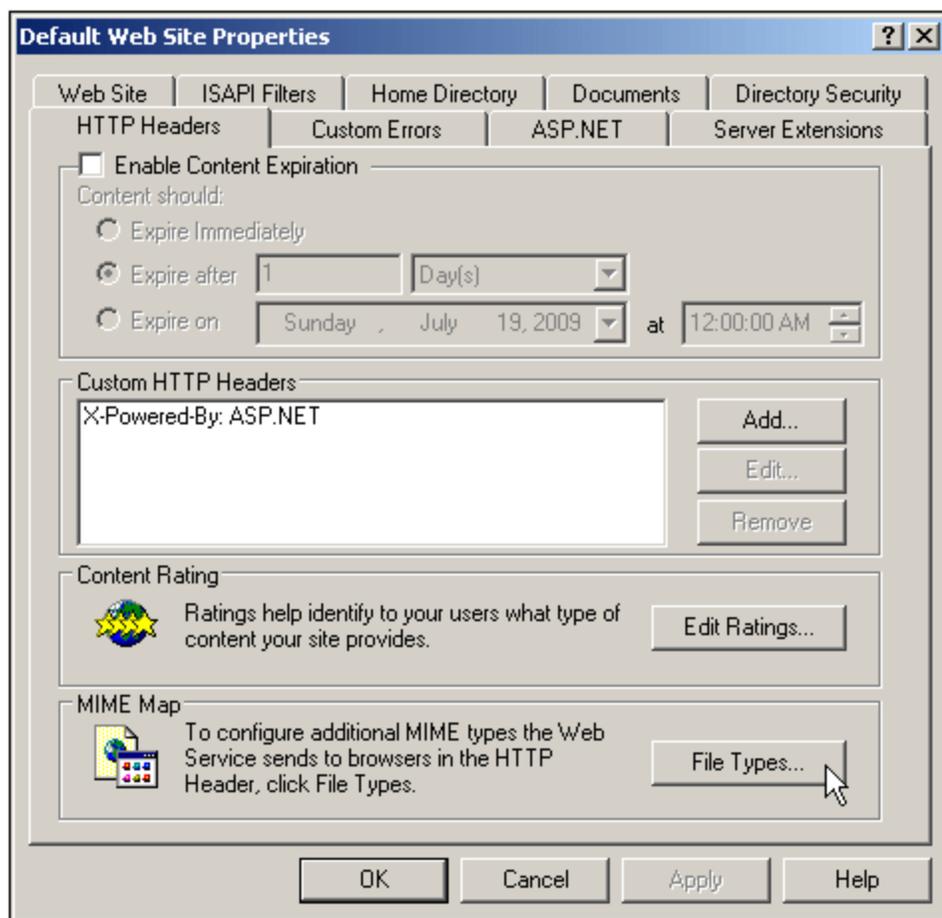
1. Abra el panel de control y haga doble clic en Herramientas administrativas.
2. En el cuadro de diálogo que aparece (*imagen siguiente*) haga doble clic en Administrador de Internet Information Services (IIS).



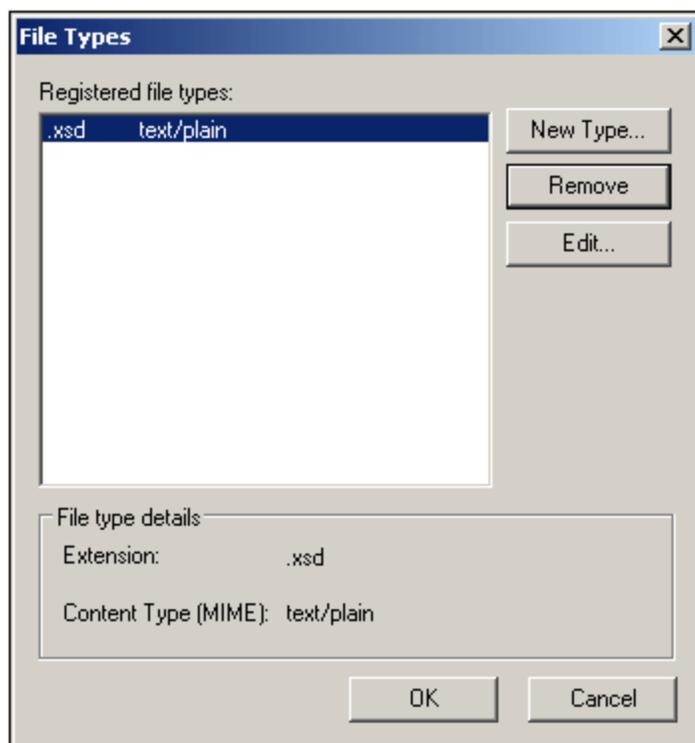
3. Aparece el cuadro de diálogo "Internet Information Services" (*imagen siguiente*). Seleccione el sitio pertinente (sitio web o carpeta) en el panel izquierdo y haga clic en el icono **Propiedades** (debajo del cursor en la imagen siguiente). También puede hacer clic en el comando **Propiedades** del menú contextual (que se accede haciendo clic con el botón derecho).



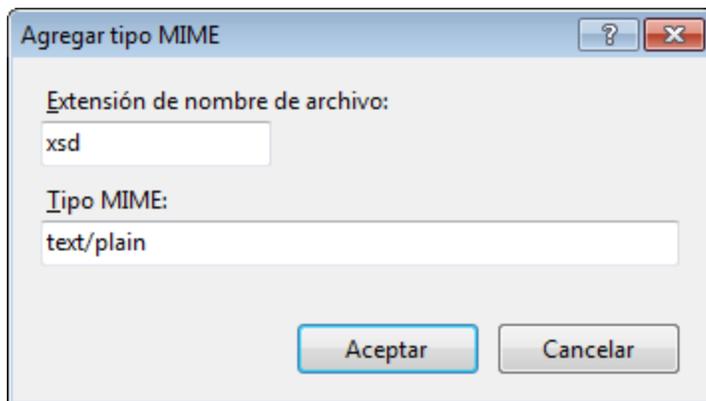
4. En la pestaña Encabezados de respuesta HTTP del cuadro de diálogo Propiedades, haga clic en el botón **Tipos MIME** (ver imagen siguiente).



5. Esto abre el cuadro de diálogo **Agregar tipo MIME** (imagen siguiente).



6. Escriba la extensión de archivo correspondiente y su tipo MIME. Consulte la tabla que aparece al principio del apartado para encontrar el tipo MIME y la extensión correcta.



7. Haga clic en **Aceptar** para confirmar.

Para más información sobre cómo configurar el acceso remoto a una carpeta en un servidor interno a través de WebDAV, consulte [este artículo](#).

2.2 Archivos XSD, XML y SPS/PXF

El principal objetivo de implementar un proyecto de Authentic Browser es ver y editar documentos XML en la vista Authentic. Este formato oculta el marcado XML del documento XML y habilita la edición del documento en una interfaz WYSIWYG con un diseño personalizado. El diseño y los mecanismos de entrada de datos de un documento XML en la vista Authentic se definen en un archivo SPS. Por tanto, el archivo SPS, el archivo XML y el esquema XML (archivo XSD) en el que se basan el archivo SPS y los documentos XML deben almacenarse en la red para que los clientes de Authentic Browser puedan acceder a ellos.

Archivos SPS

Los archivos SPS se crean con [Altova StyleVision](#). Se basa en el mismo esquema XML que el archivo XML. En StyleVision el diseñador del archivo SPS puede usar funciones de diseño gráfico para diseñar el SPS, que determina el diseño y los mecanismos de entrada de datos del documento XML cuando se abre en la vista Authentic.

En la [página HTML para el complemento Authentic Browser](#)²⁰ se especifica la ubicación del archivo XML, del archivo SPS y del archivo XSD. El archivo XML se carga en la ventana de la vista Authentic (en la página HTML) y se presenta siguiendo el diseño especificado en el archivo SPS. Tanto el archivo XML como el archivo SPS están basados en el mismo esquema.

Nota: para garantizar el correcto funcionamiento de Authentic Browser, **el esquema debe estar asignado como esquema principal** en el archivo SPS. Para obtener más información sobre cómo crear un archivo SPS (StyleVision Power Stylesheet), consulte el [manual del usuario de Altova StyleVision](#). También puede obtener más información sobre Altova StyleVision en el [sitio web de Altova](#).

Archivos XML

El usuario de la vista Authentic edita el documento XML en un equipo cliente a través de la [página HTML para el complemento de Authentic](#)²⁰. Esta página contiene una ventana de la vista Authentic donde se carga el archivo XML. El usuario de la vista Authentic edita el documento en el explorador cliente y guarda los cambios en el documento XML.

Archivos XSD

Los archivos XSD tienen dos funciones. Sirven para validar el documento XML y aporta la estructura genérica de documento en el que se basa el SPS. Tanto el archivo XML como el archivo SPS hacen referencia al archivo XSD. Por tanto, el archivo XSD también se debe almacenar en la ubicación correcta, tal y como se indica en los archivos XML y SPS.

Ubicación de los archivos XSD, XML y SPS/PXF

Los archivos XSD, XML y SPS (o PXF) deben guardarse en ubicaciones de la red a las que puedan acceder todos los equipos clientes. La mejor ubicación es el servidor de Authentic Browser. Lo mejor es guardar estos archivos en una sola carpeta del servidor y especificar en ellos referencias a los demás en forma de rutas de acceso relativas. Por ejemplo, el archivo XML se puede crear con la referencia al XSD en forma de ruta de acceso relativa.

Nota sobre archivos SPS basados en BD

Si quiere usar el complemento Authentic Browser para ver o editar bases de datos (BD) mediante un archivo SPS basado en una BD, deberá configurar el cliente para que pueda conectarse a la BD correctamente.

Información sobre la conexión en el archivo SPS: Toda la información necesaria para conectarse a la BD se almacena en una cadena de conexión en el archivo SPS. Esta cadena se crea con StyleVision cuando se crea el archivo SPS. El mecanismo utilizado para acceder a bases de datos MS Access es distinto al de las demás bases de datos, que utilizan conexiones ADO. Esta es la configuración que debe utilizar para cada tipo de mecanismo de conexión:

- **Para bases de datos MS Access:** En la cadena de conexión debe usar una ruta de acceso UNC para que los clientes se puedan conectar correctamente con a BD. Esta ruta de acceso UNC se especifica cuando la SPS se crea con StyleVision. Además, compruebe que la carpeta que contiene la BD (o alguna carpeta antecesora) está compartida. En este equipo también debe habilitar el uso compartido avanzado. En los clientes no es necesario cambiar la configuración.

Nota: El formato de la ruta de acceso UNC que se utiliza en la cadena de conexión es `\nombreServidor\carpetaCompartida\ruta\archivo.mdb`.

- **Para conexiones ADO:** La cadena de conexión ADO del archivo SPS se especifica al crear el SPS con StyleVision. Contiene toda la información necesaria para la conexión, incluida la información de seguridad. Compruebe que el controlador utilizado para probar la cadena de conexión en StyleVision también está instalado en todos los equipos cliente que hospedarán el complemento Authentic Browser. Esto permite al archivo SPS conectarse correctamente a la BD desde los clientes. Recuerde también que las conexiones ADO solamente funcionan con rutas de acceso locales y no con direcciones URL de tipo `http://`.

Nota sobre archivos PXF

Un diseño SPS que utiliza XSLT 2.0 se puede guardar en el formato de archivo PXF (Power XML Form). Este formato de archivo fue desarrollado por Altova para empaquetar el diseño SPS con los archivos relacionados (como el archivo de esquema, el archivo XML de origen, los archivos de imagen utilizados en el diseño y los archivos XSLT necesarios para transformar el XML de origen en el formato de salida). La ventaja de usar este formato de archivo es que todos los archivos necesarios para la edición en la vista Authentic y para generar los resultados de la vista Authentic se pueden distribuir en un solo archivo con total comodidad.

Nota: Si el archivo PXF está ubicado en un servidor web y se usará con el complemento Authentic Browser, compruebe que el servidor no bloquea el archivo. Para ello debe añadir este tipo MIME para las extensiones de archivo PXF : `application/x-zip-compressed`.

2.3 Página HTML para el complemento Authentic

La página HTML para el complemento Authentic desempeña estas funciones importantes:

1. La primera vez que se abre la página HTML para el complemento de Authentic en un cliente, el código de la página HTML descarga el complemento Authentic Browser del servidor y lo instala en el cliente. Este código debe estar escrito correctamente para poder identificar el archivo CAB correcto en el servidor.
2. El código de la página HTML configura la interfaz de vista Authentic dentro de la ventana del explorador, incluidas las dimensiones de la interfaz y la ubicación de los archivos XML, XSD y SPS.
3. Especifica qué archivo XML se debe editar y en qué esquema o archivo SPS se basa.
4. Contiene, dentro de elementos HTML `SCRIPT`, definiciones para las subrutinas y el control de eventos. Por ejemplo, puede indicar qué acción se lleva a cabo cuando se hace clic en un botón de la página HTML. Para ver algunos ejemplos consulte el apartado [Internet Explorer. Ejemplo nº1: ejemplo básico](#) ²⁷.

Nota: si implementa la **edición Enterprise de Authentic Browser**, la página HTML debe instalarse en el servidor para el que se registró la licencia de la edición Enterprise Edition

Incrustar el complemento Authentic Browser

Para usar el complemento Authentic Browser es necesario incrustar un objeto que identifique el complemento en la página HTML que descarga el complemento. Esto se hace mediante el elemento HTML `OBJECT`.

```
<OBJECT clsid="clsid:<CLSID>" />
```

Para más información sobre qué valores usar para CLSID consulte el apartado [Versiones de Authentic Browser](#) ¹⁰.

Temas de esta sección

Los apartados de esta sección describen cómo implementar las funciones anteriores en la página HTML. Esto se debe a que cada explorador descarga el DLL del complemento Authentic Browser de una manera diferente:

- [Licencia para la edición Enterprise](#) ²¹: este apartado describe el mecanismo para asignar licencias a la edición Enterprise de Authentic Browser.
- [Internet Explorer](#) ²²: este apartado describe cómo descargar un archivo CAB (con una extensión `.cab`) mediante un elemento HTML `OBJECT`.
- [Otros exploradores web](#) ³¹: describe cómo detectar qué tipo de explorador está realizando la solicitud y cómo descargar la versión correcta del complemento para ese explorador.

En estos apartados encontrará descripciones y ejemplos de uso de los elementos HTML `OBJECT` ²² y `SCRIPT` ²⁶ y ejemplos de páginas HTML que invocan el complemento Authentic Browser. Si necesita información sobre algún objeto en particular, consulte la descripción correspondiente en la sección [Objetos](#) ⁶⁶ de la [Referencia del usuario](#) ⁴⁵.

2.3.1 Licencia para la edición Enterprise

La licencia para la edición Enterprise de Authentic Browser se puede comprar en el [sitio web de Altova](#).

Resumen: cómo configurar la licencia para la edición Enterprise

A continuación resumimos los pasos para configurar correctamente la información sobre licencia para la edición Enterprise de Authentic Browser.

- La edición Enterprise de Authentic Browser requiere esta información sobre la licencia: (i) **nombre del servidor** para el que se registró la licencia, (ii) **nombre de la compañía** para la que se registró la licencia y (iii) la **clave de licencia**. Esta información está en el correo electrónico que recibirá después de comprar la licencia para la edición Enterprise de Authentic Browser.
- La información sobre la licencia se encuentra en la [página HTML para el complemento Authentic](#)²⁰. A continuación explicamos cómo hacer esto. (Tenga en cuenta que Authentic Browser no hará referencia a ningún archivo específico de clave de licencia.)
- Si la información sobre la licencia indicada en la [página HTML para el complemento Authentic](#)²⁰ es válida, las funciones de la edición Enterprise de Authentic Browser se desbloquearán.
- La [página HTML para el complemento Authentic](#)²⁰ debe almacenarse en el servidor para el que se registró la licencia.

Cómo introducir la información sobre la licencia

Los tres parámetros de la licencia (nombre del servidor, nombre de la compañía y clave de licencia) deben insertarse en la [página HTML para el complemento Authentic](#)²⁰. Esto puede hacerse de varias maneras:

- Como valores de parámetro del elemento OBJECT. Esto se explica en el subapartado correspondiente del apartado [Internet Explorer](#)²².
- Si se trata de una página HTML compatible con cualquier tipo de explorador, los parámetros de la licencia se pueden registrar como en el ejemplo que aparece en el apartado [Página HTML para otros exploradores](#)³¹.
- Los parámetros de la licencia también se pueden insertar en el objeto directamente, tal y como muestra el ejemplo siguiente (que es una adaptación del ejemplo del apartado [Página HTML para otros exploradores](#)³¹).

```
<SCRIPT LANGUAGE="javascript" FOR=objPlugIn EVENT="ControlInitialized">
// suscripción a evento si se ejecuta en Internet Explorer
if ( isIEOnWindows() )
{
  InitAuthenticPluginPage();
}
</SCRIPT>
```

```
<SCRIPT type="text/javascript" LANGUAGE="javascript" >
function InitAuthenticPluginPage( )
{
var serverstr='DevAuthBrowTest';
var basedir='Authentic/';
objPlugIn.LicServer = 'DevAuthBrowTest';
objPlugIn.LicCompany = 'Altova';
objPlugIn.LicKey = 'XXXXXXXXXXXX';
objPlugIn.SchemaLoadObject.URL = 'http://' + serverstr + basedir + 'OrgChart.xsd';
```

```
objPlugIn.XMLDataLoadObject.URL = 'http://' + serverstr + basedir + 'OrgChart.xml' ;
objPlugIn.DesignDataLoadObject.URL = 'http://' + serverstr + basedir +
'OrgChart.sps' ;
objPlugIn.StartEditing();
}
</SCRIPT>
```

2.3.2 Página HTML para Internet Explorer

Si la página HTML para el complemento de Authentic se abre en Internet Explorer (32 o 64 bits), debe incluir estos elementos:

- Un elemento HTML [OBJECT](#)²² que (i) provoca la descarga del DLL para el complemento de Authentic (32 o 64 bits) desde el servidor al cliente y (ii) especifica las dimensiones de la ventana de la vista Authentic en el explorador del cliente. El elemento [OBJECT](#)²² contiene la ubicación del complemento Authentic Browser. Recuerde que el complemento de Authentic para Internet Explorer puede ser de 32 o de 64 bits (excepto IE 10 y 11 de 64 bits), así que debe descargar la versión correcta (archivo CAB) en el cliente. En el apartado sobre [otros exploradores](#)³¹ puede ver un ejemplo de código que comprueba automáticamente la versión de Internet Explorer y descarga el complemento de Authentic correspondiente.
- Uno o varios elementos HTML [SCRIPT](#)²⁶ para definir las subrutinas y el control de eventos. Un elemento [SCRIPT](#) se puede usar para especificar qué documento XML se debe editar y el esquema XML y el archivo SPS en el que se basa.

Temas de esta sección

Esta sección tiene varios apartados:

- [El elemento OBJECT](#)²²: describe cómo se utiliza el elemento HTML [OBJECT](#)²² en una página HTML para el complemento Authentic.
- [El elemento SCRIPT](#)²⁶: describe cómo se deberían utilizar los elementos HTML [SCRIPT](#)²⁶ en una página HTML para el complemento Authentic.
- Ejemplos de páginas HTML completas: [IE. Ejemplo nº1: Ejemplo básico](#)²⁷ y [IE. Ejemplo nº2: Ordenar una tabla](#)²⁸.

Si necesita información sobre algún objeto en particular, consulte la descripción correspondiente en la sección [Objetos](#)⁶⁶ de la [Referencia del usuario](#)⁴⁵.

Nota: El complemento Authentic Browser es compatible con **Internet Explorer versión 5.5 o superior**. Sin embargo, las versiones de 64 bits de Internet Explorer 10 y 11 no son compatibles.

2.3.2.1 El elemento OBJECT

El elemento [OBJECT](#) cumple estas funciones:

- Pone un nombre al complemento Authentic Browser (a través del atributo `id`).
- Selecciona qué [versión del complemento Authentic Browser](#)¹⁰ se debe utilizar (con el valor del atributo `classid`).

- Especifica un archivo CAB y un número de versión (atributo `codebase`). El archivo CAB contiene un DLL que registra un objeto de COM (el complemento Authentic Browser) con un ID que es el valor del atributo `classid`. Por lo general, tanto la versión de 32 bits como la de 64 bits del complemento se almacenan en el servidor. Cada versión se identifica con un nombre de archivo distinto. El atributo `codebase` especifica el nombre del archivo CAB necesario (*ver ejemplo de código más abajo*). Para los archivos CAB de 32 y 64 bits se usa el mismo ID de clase (consulte la [tabla del apartado Versiones \(inglés/alemán y seguras/no seguras\)](#)¹⁰ de Authentic Browser).

Nombre de archivo para la versión de 32 bits: `AuthenticBrowserEdition.CAB`

Nombre de archivo para la versión de 64 bits: `AuthenticBrowserEdition_x64.CAB`

- Especifica la dimensión de la ventana de la vista Authentic en el explorador del cliente (mediante el atributo `style`).
- Puede especificar otros parámetros.

Ejemplo del elemento OBJECT

A continuación puede ver un ejemplo de elemento HTML `OBJECT`. Este elemento selecciona la versión Trusted Unicode (con el valor dado en el atributo `classid`) y establece unas dimensiones de 600 x 500 píxeles para la ventana de la vista Authentic en el explorador del cliente. Más abajo se describen también todos los atributos y parámetros que se pueden usar en el elemento `OBJECT`.

Nota: El número de versión dado más abajo no es necesariamente el número de la versión actual. Para más información consulte la descripción del atributo `codebase`²⁴.

Para el complemento Authentic Browser de 32 bits:

```
<OBJECT id="objPlugIn" style="WIDTH:600px; HEIGHT:500px"
  codeBase="http://suservidor/cabfiles/AuthenticBrowserEdition.CAB#Version=12,3,0,0"
  classid="clsid:B4628728-E3F0-44a2-BEC8-F838555AE780">
  <PARAM NAME="XMLDataURL" VALUE="http://suservidor/OrgChart.xml">
  <PARAM NAME="SPSDataURL" VALUE="http://suservidor/OrgChart.sps">
  <PARAM NAME="SchemaDataURL" VALUE="http://suservidor/OrgChart.xsd">
</OBJECT>
```

Para el complemento Authentic Browser de 64 bits:

```
<OBJECT id="objPlugIn" style="WIDTH:600px; HEIGHT:500px"
  codeBase="http://suservidor/cabfiles/AuthenticBrowserEdition_x64.CAB#Version=12,3,0,0"
  classid="clsid:B4628728-E3F0-44a2-BEC8-F838555AE780">
  <PARAM NAME="XMLDataURL" VALUE="http://suservidor/OrgChart.xml">
  <PARAM NAME="SPSDataURL" VALUE="http://suservidor/OrgChart.sps">
  <PARAM NAME="SchemaDataURL" VALUE="http://suservidor/OrgChart.xsd">
</OBJECT>
```

id

El valor del atributo `id` se utiliza como nombre de los objetos del complemento Authentic Browser cuando se utilizan en scripts. Por ejemplo, `objPlugIn.SchemaLoadObject.URL` es una llamada al objeto que carga el archivo de esquema. Para más información consulte el apartado [El elemento SCRIPT](#)²⁶.

style

Este es el atributo HTML `style` normal y corriente y sirve para especificar las dimensiones de la ventana de la vista Authentic en el explorador del cliente.

codebase

El atributo `codebase` indica la ubicación del archivo `CAB`. Recuerde que hay un archivo `CAB` para el complemento de 32 bits y otro para el complemento Authentic Browser de 64 bits:

`AuthenticBrowserEdition.CAB` y `AuthenticBrowserEdition_x64.CAB`.

El valor de la extensión opcional `#Version` aporta el número de versión del componente que está disponible en el servidor. Si el cliente tiene una versión anterior y en el atributo `codebase` se indica una versión más reciente, se instalará la versión más reciente del servidor. Si no se especifica la extensión `#Version`, no se llevará a cabo ninguna actualización hasta que el componente se elimine a mano del cliente. El número de versión actual del componente se enumera junto con las propiedades del archivo `DLL` del archivo `CAB` del componente (haga clic con el botón derecho en el archivo y seleccione Propiedades en el menú contextual).

classid

Los IDs de clase para los archivos `CAB` de 32 y 64 bits son idénticos y son los que aparecen en la tabla del apartado [Versiones de Authentic Browser](#)¹⁰.

A partir de la versión 5.0 del complemento, el valor de `classid` para las versiones Unicode es distinto. Por tanto, si actualiza el archivo `CAB` Unicode del servidor cuya versión sea anterior a la versión 5.0, asegúrese de cambiar los valores de `classid` en los archivos HTML. También debe tener en cuenta que si un archivo `CAB` nuevo del servidor tiene el mismo ID de clase que un archivo `CAB` instalado previamente en el cliente, el archivo `CAB` nuevo no reemplazará automáticamente el antiguo en el cliente. Deberá quitar a mano el archivo `CAB` instalado antes de descargar el archivo `CAB` nuevo. Los valores de `classid` difieren dependiendo del idioma del complemento.

Parámetros

El elemento `OBJECT` puede usar estos parámetros tantas veces como se quiera:

LicServer

El nombre del servidor para el que se registró la licencia de la edición Enterprise de Authentic Browser.

LicKey

La licencia para validar el uso de la edición Enterprise de Authentic Browser.

LicCompany

El nombre de la compañía con la que se registró la licencia de la edición Enterprise de Authentic Browser.

XMLDataURL

Una URL absoluta que aporta la ubicación del archivo XML que se debe editar. En la versión no segura (Untrusted) también puede usar una ruta de acceso local completa.

XMLDataSaveURL

Una URL absoluta que aporta la ubicación donde se debe guardar el archivo XML. En la versión no segura (Untrusted) también puede usar una ruta de acceso local completa.

SPSDataURL

Una URL absoluta que aporta la ubicación del archivo StyleVision Power Stylesheet. En la versión no segura (Untrusted) también puede usar una ruta de acceso local completa.

SchemaDataURL

Una URL absoluta que aporta la ubicación del archivo de esquema asociado. En la versión no segura (Untrusted) también puede usar una ruta de acceso local completa.

TextStateBmpURL

La carpeta donde se deben almacenar las imágenes para los iconos de estado del texto.

TextStateToolbarLine

La línea de la barra de herramientas donde se deben colocar los iconos de estado del texto. El valor predeterminado es 1.

AutoHideUnusedCommandGroups

Indica si se deben ocultar o no los grupos de comandos no utilizados de las barras de herramientas. El valor predeterminado es true.

ToolbarsEnabled

Indica si se habilitan o no las barras de herramientas. El valor predeterminado es true.

ToolbarTooltipsEnabled

Indica si se habilita o no la función de información rápida.

HideSaveButton

Si su valor es True, quita el botón Guardar de la barra de herramientas Authentic, que aparece por defecto.

BaseURL

Aporta la URL base que se debe usar con rutas de acceso relativas.

SaveButtonUsePOST

Si su valor es True, el comando POST HTTP se usa en lugar de PUT a la hora de guardar documentos.

EntryHelpersEnabled

Si su valor es True, se muestran los ayudantes de entrada de Authentic.

EntryHelperSize

El ancho de la ventana del ayudante de entrada medido en píxeles.

EntryHelperAlignment

Especifica la ubicación de los ayudantes de entrada con respecto a la ventana de documento.

0 = Alinear la barra de herramientas encima del documento.

1 = Alinear la barra de herramientas a la izquierda del documento.

2 = Alinear la barra de herramientas debajo del documento.

3 = Alinear la barra de herramientas a la derecha del documento.

EntryHelperWindows

Selecciona qué ventanas secundarias del ayudante de entrada se pueden ver.

1 = Elementos

2 = Atributos

4 = Entidades

Se permiten todas las combinaciones posibles

SaveButtonAutoEnable

Consulte el apartado [Authentic.SaveButtonAutoEnable](#) ⁸⁶ para más información.

LoaderSettingsFileURL

Aporta la URL de LoaderSettingsFile para la gestión de paquetes.

2.3.2.2 El elemento SCRIPT

Los elementos `SCRIPT` definen los controladores de eventos y las subrutinas a los que se puede llamar desde el archivo HTML.

Aquí puede ver un script de muestra para controlar un evento:

```
<SCRIPT LANGUAGE="javascript" FOR=objPlugIn EVENT="ControlInitialized">
  objPlugIn.SchemaLoadObject.URL = "http://suservidor/OrgChart.xsd"
  objPlugIn.XMLDataLoadObject.URL = "http://suservidor/OrgChart.xml"
  objPlugIn.DesignDataLoadObject.URL = "http://suservidor/OrgChart.sps"
  objPlugIn.StartEditing
</SCRIPT>
```

Y aquí puede ver otro script que contiene subrutinas:

```
<SCRIPT ID=clientEventHandlers LANGUAGE=vbscript>
  Sub BtnOnClick
    objPlugIn.SchemaLoadObject.URL = "http://suservidor/OrgChart.xsd"
    objPlugIn.XMLDataLoadObject.URL = "http://suservidor/OrgChart.xml"
    objPlugIn.DesignDataLoadObject.URL = "http://suservidor/OrgChart.sps"
    objPlugIn.StartEditing
  End Sub
  Sub OnClickFind
    objPlugIn.FindDialog
  End Sub
  Sub BtnOnTestProp
    If objPlugIn.IsRowInsertEnabled Then
      msgbox "true"
    Else
      msgbox "false"
    End If
  End Sub
</SCRIPT>
```

Lenguajes de programación

El complemento Authentic Browser se ha probado con JavaScript y VBScript.

Controlar eventos

El valor del atributo `ID` del elemento `OBJECT` del cuerpo HTML se especifica como valor del atributo `FOR`. Los objetos del complemento Authentic Browser a los que se llama deben tener un nombre que es este valor. Para ver una lista con todos los eventos consulte el apartado [Eventos: Referencia](#) ⁴⁸.

Subrutinas

Se pueden crear subrutinas para cualquier evento que se desee definir en el archivo HTML. El nombre del objeto del complemento Authentic Browser debe ser el mismo que el valor del atributo `ID` del elemento `OBJECT` del cuerpo HTML. En el ejemplo anterior, el prefijo es `objPlugIn`, que debe ser el valor del atributo `ID` del elemento `OBJECT`. Los métodos, propiedades y subobjetos disponibles en el complemento Authentic Browser se describen en la sección de referencia de esta documentación.

2.3.2.3 Ejemplo nº1: Ejemplo básico

El código HTML que aparece más abajo genera una página con estas características:

- Instala la versión Trusted Unicode de Authentic Browser en el cliente si no está instalado todavía.
- La página contiene una ventana de 600 x 500 píxeles en la que se carga Authentic Browser.
- Debajo de la ventana de Authentic Browser hay una fila de cuatro botones.
- Se carga la vista Authentic del archivo `OrgChart.xml`.
- Los botones **Buscar** y **Reemplazar** abren los cuadros de diálogo "Buscar" y "Reemplazar" respectivamente.
- El botón **Guardar** guarda los cambios en un archivo llamado `SaveFile.xml` ubicado en el directorio raíz del servidor.
- El botón **Probar propiedad** prueba una propiedad simple.

Cuando se abre esta página HTML en el cliente, el usuario puede empezar a editar el archivo XML `OrgChart.xml` y guardar el archivo editado como `SaveFile.xml`.

Si quiere, puede usar esta sencilla página HTML para probar si Authentic Browser funciona correctamente. Antes debe asegurarse de utilizar las URL correctas para ubicar el archivo `CAB`, los archivos `XSD`, `XML` y `SPS` y cualquier otro recurso en el servidor. Recuerde que algunos servidores distinguen entre mayúsculas y minúsculas, así que si algún archivo no se encuentra, revise el uso de mayúsculas y minúsculas en los nombres de archivo y en los comandos del código. Este ejemplo se puede modificar o ampliar para desarrollar soluciones más complejas con la ayuda de Authentic Browser. Consulte el apartado [El elemento OBJECT](#) ²² para obtener más información.

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
  <title>Página XMLSpyDocEditPlugIn básica</title>

  <!-- Script para controlar el evento ControlInitialized -->
  <SCRIPT LANGUAGE="javascript" FOR=objPlugIn EVENT="ControlInitialized">
    objPlugIn.SchemaLoadObject.URL = "http://suservidor/OrgChart.xsd"
    objPlugIn.XMLDataLoadObject.URL = "http://suservidor/OrgChart.xml"
    objPlugIn.DesignDataLoadObject.URL = "http://suservidor/OrgChart.sps"
    objPlugIn.StartEditing()
  </SCRIPT>

  <!-- Script con subrutinas -->
```

```

<SCRIPT ID=clientEventHandlers LANGUAGE=vbscript>
  Sub OnClickFind
    objPlugIn.FindDialog
  End Sub

  Sub OnClickReplace
    objPlugIn.ReplaceDialog
  End Sub

  Sub BtnOnSave
    objPlugIn.XMLDataSaveUrl = "http://suservidor/SaveFile.xml"
    objPlugIn.Save
  End Sub

  Sub BtnOnTestProp
    If objPlugIn.IsRowInsertEnabled Then
      msgbox "true"
    Else
      msgbox "false"
    End If
  End Sub
</SCRIPT>
</head>

<body>
  <!-- El elemento Object tiene un atributo id con el valor que se debe usar -->
  <!-- como nombre de los objetos de Authentic Browser -->
  <!-- Classid selecciona la versión Trusted Unicode -->
  <OBJECT id="objPlugIn"
  <!-- CodeBase selecciona el archivo CAB de 32 bits (AuthenticBrowserEdition.CAB) -->
  <!-- o el archivo CAB de 64 bits (AuthenticBrowserEdition_x64.CAB) -->
  CodeBase="http://suservidor/AuthenticBrowserEdition.CAB#Version=12,3,0,0"
  <!-- El atributo Classid para el archivo CAB de 32 bits y para el de 64 bits es el mismo
-->
  Classid="clsid:B4628728-E3F0-44a2-BEC8-F838555AE780" width="600" height="500">
</OBJECT>
<p>
  <input type="button" value="Find" name="B4" onclick="OnClickFind()">
  <input type="button" value="Replace" name="B5" onclick="OnClickReplace()">
  <input type="button" value="Save" name="B6" onclick="BtnOnSave()">
  <input type="button" value="Test property" name="B7" onclick="BtnOnTestProp">
</p>
</body>
</html>

```

2.3.2.4 Ejemplo nº2: Ordenar una tabla

A continuación aparece un ejemplo de página HTML que tiene un JavaScript incrustado. Este ejemplo requiere tener instalado Authentic Browser (archivo CAB) en el equipo. Recuerde que algunos servidores distinguen entre mayúsculas y minúsculas, así que si algún archivo no se encuentra, revise el uso de mayúsculas y minúsculas en los nombres de archivo y en los comandos del código.

El código del ejemplo:

- Muestra cómo acceder al complemento. Modifique el código para remitir al archivo CAB y al identificador de clase (CLSID) de su versión del complemento (Trusted o Untrusted).
- Muestra cómo cargar un archivo en el complemento. Modifique el código para remitir a su documento de muestra.
- Implementa botones para el posicionamiento del cursor.
- Implementa comandos más complejos como comandos para ordenar tablas.
- Muestra cómo usar el evento `SelectionChanged`.

Consulte el apartado [El elemento OBJECT](#)²² para obtener más información.

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
    <title>Página de prueba para el complemento Authentic Browser</title>

    <SCRIPT LANGUAGE="javascript" for=objPlugIn EVENT="ControlInitialized">
      var strSampleRoot = "http://miRaíz/miRuta/miNombreDocumento";
      objPlugIn.SchemaLoadObject.URL = strSampleRoot + ".xsd";
      objPlugIn.XMLDataLoadObject.URL = strSampleRoot + ".xml";
      objPlugIn.DesignDataLoadObject.URL = strSampleRoot + ".sps";
      objPlugIn.StartEditing();
    </SCRIPT>

    <SCRIPT ID="clientEventHandlers" LANGUAGE="javascript">
      var objCurrentRange = Null;

      Function BtnDocumentBegin() { objPlugIn.AuthenticView.DocumentBegin.Select(); }
      Function BtnDocumentEnd() { objPlugIn.AuthenticView.DocumentEnd.Select(); }
      Function BtnWholeDocument() { objPlugIn.AuthenticView.WholeDocument.Select(); }
      Function BtnSelectNextWord()
    { objPlugIn.AuthenticView.Selection.SelectNext(1).Select(); }
      Function BtnSortDepartmentOnClick()
      {
        var objCursor = Null;
        var objTableStart = Null;
        var objBubble = Null;
        var strField1 = "";
        var strField1 = "";
        var nColIndex = 0;
        var nRows = 0;

        objCursor = objPlugIn.AuthenticView.Selection;
        If (objCursor.IsInDynamicTable())
        {
          // calcular índice de columnas actual
          nColIndex = 0;
          While (True)
          {
            try { objCursor.GotoPrevious(1); }
            catch (err) { break; }
            nColIndex++;
          }

          // Ir al principio de la tabla
```



```

        <TR>
            <TD><Input Type="button" value="Ordenar tabla en función de esta columna"
id="B6" onclick="BtnSortDepartmentOnClick()" </TD>
        </TR>
    </TABLE>
    <TABLE id=SelTable border=1>
        <TR><TD id=SelTable_FirstTextPosition></TD><TD
id=SelTable_LastTextPosition></TD></TR>
        <TR><TD id=SelTable_FirstXMLData></TD><TD id=SelTable_FirstXMLDataOffset></TD></TR>
        <TR><TD id=SelTable_LastXMLData></TD><TD id=SelTable_LastXMLDataOffset></TD></TR>
        <TR><TD id=SelTable_Text></TD></TR>
    </TABLE>
</body>

<SCRIPT LANGUAGE=javascript For=objPlugIn EVENT=selectionchanged>
    var CurrentSelection = Null;
    CurrentSelection = objPlugIn.AuthenticView.Selection;
    SelTable_FirstTextPosition.innerHTML = CurrentSelection.FirstTextPosition;
    SelTable_LastTextPosition.innerHTML = CurrentSelection.LastTextPosition;
    SelTable_FirstXMLData.innerHTML = CurrentSelection.FirstXMLData.Parent.Name;
    SelTable_FirstXMLDataOffset.innerHTML = CurrentSelection.FirstXMLDataOffset;
    SelTable_LastXMLData.innerHTML = CurrentSelection.LastXMLData.Parent.Name;
    SelTable_LastXMLDataOffset.innerHTML = CurrentSelection.LastXMLDataOffset;
</SCRIPT>

</html>

```

2.3.3 Página HTML para otros exploradores

En algunos casos no se conoce qué explorador se usará en el cliente. En estos casos se pueden almacenar [diferentes versiones de Authentic Browser para Internet Explorer](#)¹⁰ en el servidor. En la página HTML puede insertar un script para determinar qué explorador se utilizó para abrir la página HTML y provocar la carga del complemento [Authentic Browser](#)¹⁰ correspondiente.

Además, si el explorador utilizado en el cliente es IE, debe seleccionarse y descargarse el archivo **.CAB** correcto (para IE de 32 o de 64 bits) desde el servidor. Puede usar un script para determinar la versión de IE y seleccionar el archivo **.CAB** correspondiente para descargarlo desde el servidor.

A continuación puede ver un ejemplo que determina el tipo de explorador, carga la versión correcta del complemento Authentic Browser y lleva a cabo algunas funciones. Si necesita información sobre algún objeto en particular, consulte la descripción correspondiente en la sección [Objetos](#)⁶⁶ de la [Referencia del usuario](#)⁴⁵.

Archivo de muestra

El código HTML que aparece más abajo genera una página con estas características:

- Comprueba qué explorador web está instalado en el cliente.
- Si el explorador detectado es IE, comprueba si el sistema es de 32 o 64 bits y selecciona el [archivo CAB correspondiente](#)²².
- La ventana de Authentic Browser dentro de la página tiene un ancho que es 100% de la ventana del explorador y 60% de su alto.

- Debajo de la ventana de Authentic Browser hay una fila de cinco botones.
- El botón **Iniciar edición** carga la vista Authentic del archivo `OrgChart.xml` que está en el directorio raíz del servidor.
- Los botones **Buscar** y **Reemplazar** abren los cuadros de diálogo "Buscar" y "Reemplazar" respectivamente.
- El botón **Guardar** guarda los cambios en un archivo llamado `SaveFile_OrgChart.xml` que está en el directorio raíz del servidor.
- El botón **Probar propiedad** prueba una propiedad simple.

Cuando se abre esta página HTML en el cliente, el usuario puede empezar a editar el archivo XML `OrgChart.xml` y guardar el archivo editado como `SaveFile_OrgChart.xml`.

Si quiere, puede usar esta sencilla página HTML para probar si Authentic Browser funciona correctamente. Antes debe asegurarse de utilizar la dirección IP y la ruta de acceso correctas de los archivos correspondientes en las URL que buscan el archivo `XPI`, los archivos `XSD`, `XML` y `SPS` y cualquier otro recurso en el servidor. Recuerde que algunos servidores distinguen entre mayúsculas y minúsculas, así que si algún archivo no se encuentra, revise el uso de mayúsculas y minúsculas en los nombres de archivo y en los comandos del código. Este ejemplo se puede modificar o ampliar para desarrollar soluciones más complejas con la ayuda de Authentic Browser.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Página de prueba para Orgchart.sps compatible con cualquier explorador</title>
    <script type="text/javascript">
      <!--
        function BtnOnSave() { objPlugIn.Save();}

        function InitAuthenticPluginPage( )
        {

          var schema= document.getElementById('xsd');
          var instance=document.getElementById('xml');
          var design=document.getElementById('sps');
          objPlugIn.XMLDataLoadObject.URL =instance.innerHTML;
          objPlugIn.DesignDataLoadObject.URL = design.innerHTML;
          objPlugIn.SchemaLoadObject.URL= schema.innerHTML;
          // alert(schema.innerHTML+" "+instance.innerHTML+" "+design.innerHTML);

          /*
            var serverstr='su-servidor/';
            var basedir='Authentic/';
            objPlugIn.SchemaLoadObject.URL = 'http://' + serverstr + basedir + 'OrgChart.xsd';
            objPlugIn.XMLDataLoadObject.URL = 'http://' + serverstr + basedir + 'OrgChart.xml'
          ;
            objPlugIn.DesignDataLoadObject.URL = 'http://' + serverstr + basedir +
'OrgChart.sps';
          */

          objPlugIn.StartEditing();
        }
      </script>
    </head>
  </html>
```



```

        else {
            return('CodeBase="http://su-
servidor/AuthenticBrowserEdition.CAB#Version=12,2,0,0" ');
        }
    }

    // Crear la instancia de objeto del complemento en función del explorador que
cargue la página
    // -IE usa la etiqueta <OBJECT> para incrustar complementos y permite usar el
atributo CODEBASE
    // para señalar el archivo .cab necesario para la instalación
    // si el complemento no está instalado todavía

function createObject( codebase, clsid)
{if ( isIEOnWindows() )
{
    document.write ( '<OBJECT ' +
        'id="objPlugIn" ' +
        getCodeBase() +
        'Classid="clsid:B4628728-E3F0-44a2-BEC8-F838555AE780" ' +
        'width="100%" ' +
        'height="60%" ' +
        '>' +
        '<PARAM NAME="XMLDataSaveUrl" VALUE="http://su-
servidor/Authentic/SaveFile_OrgChart.xml"> ' +
        '<PARAM NAME="EntryHelpersEnabled" VALUE="TRUE"> ' +
        '<PARAM NAME="SaveButtonAutoEnable" VALUE="TRUE"> ' +
        '<PARAM NAME="LicServer" VALUE="your-server"> ' +
        '<PARAM NAME="LicCompany" VALUE="Altova"> ' +
        '<PARAM NAME="LicKey" VALUE="XXXXXXXXXX"> ' +
        '</OBJECT>');
}
}

createObject();
// tras ejecutar createObject el objeto del complemento existe. Inicializar la
variable de javascript que se debe usar en la variables de los scripts
var objPlugIn = document.getElementById('objPlugIn');
</script>

<br><br>
<button onclick="objPlugIn.StartEditing()"><span>Iniciar edición</span></button>
<button onclick="objPlugIn.FindDialog()"><span>Buscar</span></button>
<button onclick="objPlugIn.ReplaceDialog();"><span>Reemplazar</span></button>
<button onclick="BtnOnSave()"><span>Guardar</span></button>
<button onclick="alert
( objPlugIn.IsRowInsertEnabled );"><span>Probar</span><br></button>
</center>

<script language="javascript" type="text/javascript">
    // suscripción a evento si se ejecuta en Firefox
    if ( isFirefoxOnWindows() )
    {

```

```
        objPlugIn.addEventListener("ControlInitialized", InitAuthenticPluginPage,
false);
    }
</script>

<script event="ControlInitialized" for="objPlugIn" language="javascript"
type="text/javascript">
    // suscripción a evento si se ejecuta en Internet Explorer
    if ( isIEOnWindows() )
    {
        InitAuthenticPluginPage();
        //if ( isIEEx64OnWindows() ) alert("IE x64");
    }
</script>

</body>
</html>
```

Nota: El script anterior contiene la información sobre la licencia necesaria para activar la edición Enterprise de Authentic Browser.

2.4 Paquetes de extensión para instalación por encargo

Si tiene pensado que la instalación se realice por encargo, el paquete o los paquetes de extensión comprimido(s) de Authentic Browser (archivo(s) CAB) deben estar almacenados en el servidor. Cuando se acceda por primera vez a la [página HTML para el complemento Authentic](#)²⁰ desde el explorador cliente, las instrucciones de la página HTML provocan la descarga del paquete de instalación correspondiente desde el servidor al cliente, lo descomprime y lo instala en el cliente.

Los paquetes de extensión de Authentic Browser necesarios para la instalación por encargo se pueden descargar del sitio web de Altova. Después debe almacenarlos en el servidor como archivos CAB comprimidos.

Internet Explorer 5.5 o superior (32 bits)	Archivo CAB (para IE de 32 bits)
Internet Explorer (64 bits)	Archivo CAB (para IE de 64 bits)

Los archivos CAB están disponibles para exploradores IE de 32 bits y de 64 bits. En el servidor puede almacenar el archivo CAB para IE de 32 bits y el archivo CAB para IE de 64 bits. En la [página HTML para el complemento Authentic](#)²⁰ puede incluir un script que detecte si el explorador es de 32 o de 64 bits y que descargue el archivo CAB correspondiente. En el apartado [Página HTML para el complemento Authentic | Otros exploradores](#)³¹ se explica cómo crear ese script.

Descargar y almacenar el archivo CAB

El archivo CAB se descarga del [sitio web de Altova](#) y se puede guardar en cualquier ubicación del servidor. Si quiere implementar la edición Enterprise de Authentic Browser **debe almacenar el paquete de instalación en el servidor para el que se registró el producto.**

El archivo de instalación (CAB) es un formato de archivo comprimido. **No descomprima estos archivos.** La extracción e instalación en el cliente se lleva a cabo automáticamente cuando el cliente abre la [página HTML para el complemento Authentic](#)²⁰ por primera vez. La ubicación del archivo CAB en el servidor también se especifica en la [página HTML para el complemento Authentic](#)²⁰.

3 Configuración del cliente

El equipo cliente es el equipo en el que se abre la [página HTML para el complemento de Authentic](#)²⁰ y en el que se edita el documento XML. Para poder habilitar estas funciones el complemento Authentic Browser tiene que estar instalado en el explorador cliente que se usará para abrir la [página HTML para el complemento de Authentic](#)²⁰.

El complemento Authentic Browser se puede instalar en el explorador cliente [automáticamente desde el servidor](#)³⁹ cuando la [página HTML para el complemento de Authentic](#)²⁰ se abre por primera vez en el explorador cliente. También puede instalarse [a mano](#)³⁹ o con el [archivo de instalación MSI](#)⁴⁰.

En esta sección explicamos cómo configurar el equipo cliente para trabajar con el complemento Authentic Browser, en concreto:

- Los [requisitos del explorador web](#)³⁸ para poder habilitar las funciones de Authentic Browser.
- Los [métodos de instalación del complemento Authentic Browser](#)³⁹ en los exploradores compatibles en el equipo cliente. También describe cómo [actualizar](#)⁴¹ y [desinstalar](#)⁴¹ Authentic Browser.
- [Configuración de seguridad de IE9](#)⁴² para garantizar el correcto funcionamiento del complemento.
- [Configuración de seguridad de IE10](#)⁴⁴ para garantizar el correcto funcionamiento del complemento. Recuerde que Authentic Browser funcione en Internet Explorer 10, debe habilitar el modo compatible de IE 10.

3.1 Requisitos del explorador

Para poder ver una página XML de Altova Authentic en un equipo cliente, deberá tener un explorador que admita complementos. Varios proveedores de exploradores web han dejado de ofrecer compatibilidad con la [arquitectura NPAPI](#), por lo cual Authentic Browser Edition ya solo es compatible con **Microsoft Internet Explorer (32 bits) versión 9 o superior**. Si prefiere usar otro explorador web, deberá usar una versión antigua que admita complementos.

A continuación puede ver una lista de exploradores web populares y su compatibilidad con Authentic Browser:

- *Microsoft Internet Explorer (32 bits) versión 9 o superior*: compatible con Authentic Browser.
- *Microsoft Internet Explorer (64 bits) versión 9*: es la última versión de 64 bits compatible con Authentic Browser. Las versiones más recientes son Microsoft Edge.
- *Microsoft Edge* no es compatible con Authentic Browser porque no admite controles ActiveX como complementos.

Nota: Es **obligatorio** tener instalado Internet Explorer porque la interfaz de la vista Authentic (que aparece en la ventana del explorador web) se genera con la ayuda de Internet Explorer.

3.2 Complemento para exploradores Authentic Browser

Los siguientes apartados describen cómo instalar Authentic Browser de varias maneras y cómo actualizar y desinstalar el complemento.

- [Instalación por encargo](#) ³⁹
- [Instalación manual con MSI](#) ³⁹
- [Instalación de inserción con MSI](#) ⁴⁰
- [Actualizaciones automáticas](#) ⁴¹
- [Desinstalación y deshabilitación](#) ⁴¹

Instalación de varias versiones

Hay varias versiones de Authentic Browser. Hay versiones en diferentes idiomas (español, inglés, alemán, francés japonés) y cada una de ellas tiene versiones seguras y no seguras para Microsoft Internet Explorer (32 y 64 bits).

En cada equipo cliente puede instalar una, varias o todas las versiones del complemento Authentic Browser. Cada versión aparece por separado en el administrador de complementos del explorador web.

Para más información consulte el apartado [Versiones de Authentic Browser](#) ¹⁰.

3.2.1 Instalación por encargo

Lo mejor es instalar el complemento Authentic Browser automáticamente cuando lo necesite: Así es como funciona este mecanismo:

1. Los paquetes de extensión CAB se almacenan en el servidor.
2. Cuando se abre la [página HTML para el complemento Authentic](#) ²⁰ en el explorador del cliente, las instrucciones de la página HTML descargan el paquete de extensión correspondiente e instalan Authentic Browser en el explorador del cliente.

Use el atributo `CODEBASE` del elemento `OBJECT` para especificar la dirección URL del archivo CAB. Para más información consulte el [ejemplo para Internet Explorer](#) ²².

3.2.2 Instalación manual con MSI

En Internet Explorer no se pueden instalar complementos guardados en archivos CAB. Sin embargo, puede instalar Authentic Browser manualmente con el archivo de instalación MSI.

El archivo de instalación MSI (Microsoft Windows Installer) para Authentic Browser se puede descargar del sitio web de Altova. Se trata de un archivo de instalación ejecutable (con una extensión `.exe`) que instala el complemento Authentic Browser en los exploradores web compatibles que estén instalados en el equipo cliente (Microsoft Internet Explorer de 32 y 64 bits).

Instalación

Descargue el archivo de instalación MSI en el equipo cliente y haga doble clic en él para iniciar la instalación. El complemento se instalará (por defecto en la carpeta `C:\Archivos de programa (x86)\Altova\AuthenticBrowserPlugin\...`) y se integrará en los exploradores web compatibles que estén instalados. Si no quiere instalar el complemento en todos los exploradores, puede configurar la instalación en la pantalla *Instalación personalizada*.

Las versiones instaladas de Authentic Browser aparecen en la lista de programas *Agregar/Quitar programas* del Panel de control del equipo cliente.

Versiones del archivo MSI

Para cada una de las distintas versiones de Authentic Browser (en inglés, alemán, español, francés o japonés; versión trusted o untrusted; exploradores de 32 o 64 bits) hay un archivo de instalación MSI distinto. Por ejemplo, el archivo MSI para el complemento en inglés versión Trusted para exploradores de 32 bits es diferente del archivo MSI en inglés versión Untrusted para exploradores de 32 bits. Este también será diferente para exploradores de 64 bits, etc.

Para más información consulte el apartado [Versiones de Authentic Browser](#)¹⁰.

3.2.3 Instalación de inserción con MSI

Si la instalación de software nuevo la gestiona el equipo de soporte técnico y los archivos de instalación se distribuyen dentro de la red de administración, es conveniente usar un archivo de instalación MSI nativo. Sobre todo en el caso de Internet Explorer porque para instalar complementos en este explorador hacen falta derechos de administrador.

Desde el sitio web de Altova puede descargar el archivo de instalación MSI para el complemento Authentic Browser. Se trata de un archivo de instalación ejecuta (con la extensión `.exe`) que instala el complemento Authentic Browser en todos los exploradores compatibles que estén instalados en el equipo cliente (Microsoft Internet Explorer de 32 y 64 bits).

Instalación

Descargue el archivo de instalación MSI en el servidor de administración e inclúyalo en los procesos administrativos para instalar programas nuevos en equipos cliente. El complemento se instalará (en la carpeta `C:\Archivos de programa (x86)\Altova\AuthenticBrowserPlugin\...`) y se integrará en los exploradores web compatibles que estén instalados. Si no quiere instalar el complemento en todos los exploradores, puede configurar la instalación en la pantalla *Instalación personalizada*.

Para instalarlo en silencio, use el parámetro de entrada de la línea de comandos `/q`.

Las versiones instaladas de Authentic Browser aparecen en la lista de programas *Agregar/Quitar programas* del Panel de control del equipo cliente.

Versiones del archivo MSI

Para cada una de las distintas versiones de Authentic Browser (en inglés, alemán, español, francés o japonés; versión trusted o untrusted; exploradores de 32 o 64 bits) hay un archivo de instalación MSI distinto. Por ejemplo, el archivo MSI para el complemento en inglés versión Trusted para exploradores de 32 bits es diferente del archivo MSI en inglés versión Untrusted para exploradores de 32 bits. Este también será diferente para exploradores de 64 bits, etc.

Para más información consulte el apartado [Versiones de Authentic Browser](#)¹⁰.

3.2.4 Actualizaciones automáticas

La actualización del componente Authentic Browser y su instalación en Internet Explorer se lleva a cabo de la siguiente manera:

- Configure el valor del atributo `codebase` del elemento `object` de la [página HTML para el complemento Authentic](#)²⁰ con el sufijo `#Version=...` para indicar el número de versión del complemento que está disponible en el servidor.
- Si la versión del complemento que está instalado en el equipo cliente es inferior a la versión del atributo, se le preguntará al usuario si quiere o no actualizar el complemento.

3.2.5 Desinstalación y deshabilitación

Si usó un archivo de instalación MSI para la instalación, en la lista *Agregar o quitar programas* del Panel de control debe aparecer el complemento Authentic Browser. Si lo desinstala desde esta lista, el complemento se quita del explorador y de la lista de programas.

Los complementos instalados en el explorador web aparecen como *habilitado* en el administrador de complementos del explorador. Si quita el complemento desde este administrador, el complemento aparecerá como *deshabilitado* pero seguirá en el disco del equipo cliente.

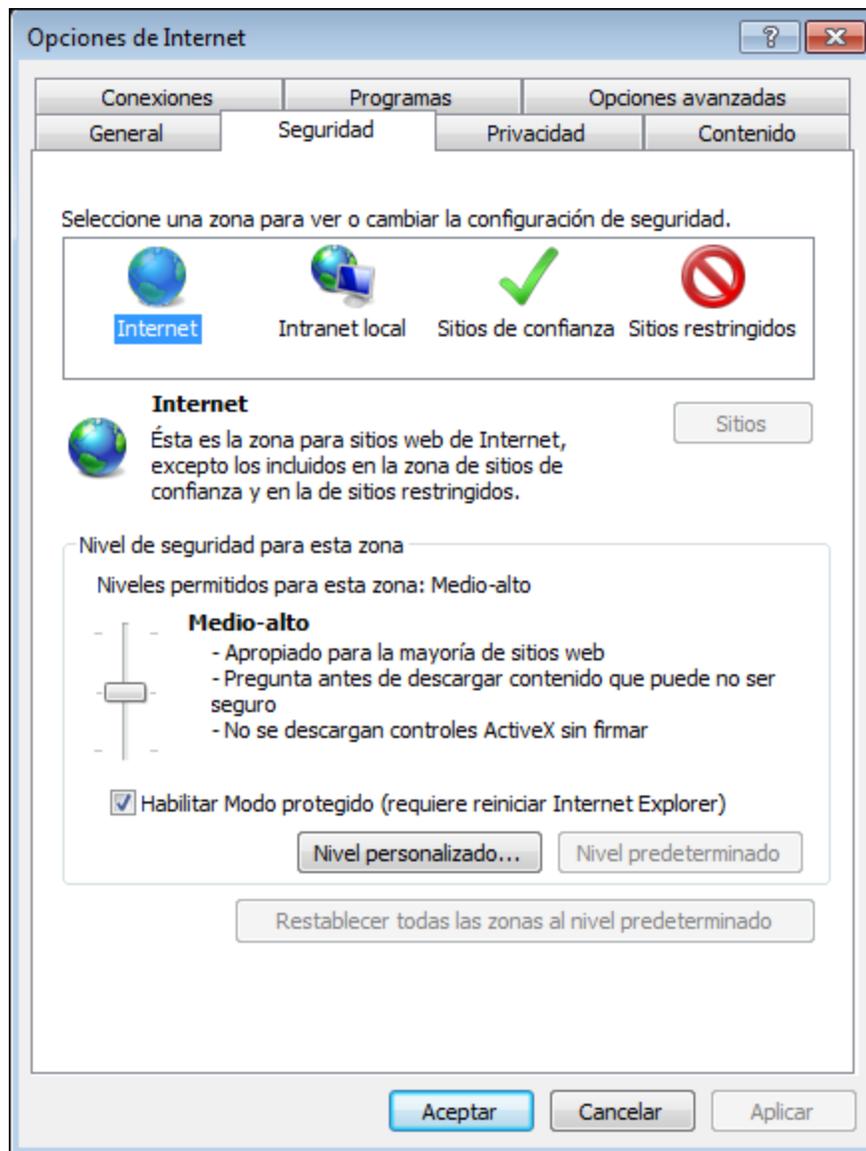
3.3 Configuración de seguridad en IE9

Si usa la [versión no segura de Authentic Browser](#)¹⁰ en Internet Explorer 9, puede que reciba este mensaje de error:

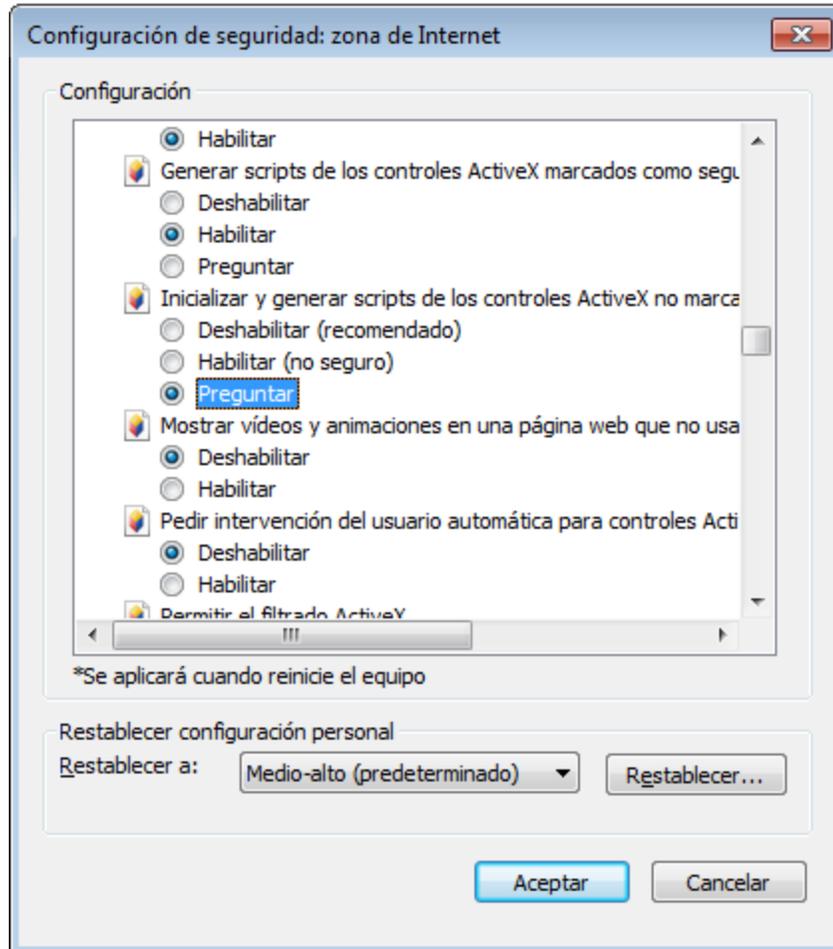
Internet Explorer bloqueó un control ActiveX. Es posible que esta página no se muestre correctamente.

Para habilitar los controles ActiveX, debe configurar la opción de seguridad correspondiente en Internet Explorer, siguiendo las instrucciones que aparecen a continuación:

1. Abra el cuadro de diálogo "Opciones de Internet" (*imagen siguiente*) de Internet Explorer con el comando de menú **Herramientas | Opciones de Internet**.



2. Seleccione la pestaña *Seguridad (imagen anterior)* y la zona para la que quiere cambiar la configuración (Internet o intranet local).
3. Ahora haga clic en el botón **Nivel personalizado...** para abrir el cuadro de diálogo "Configuración de seguridad: zona de Internet" (*imagen siguiente*).



4. Desplácese hasta ver la sección *Controles y complementos ActiveX* y hasta la opción Iniciar y generar scripts de los controles ActiveX no marcados como seguros para scripts. Seleccione el botón de opción *Preguntar (imagen anterior)*. Para confirmar haga clic en **Aceptar**.
5. Reinicie Internet Explorer para aplicar la nueva configuración.

A partir de ahora cada vez que una [página HTML para el complemento de Authentic](#)²⁰ intente cargar un archivo ActiveX, Internet Explorer le preguntará si quiere cargar o no el control.

3.4 Configuración de seguridad en IE10

Internet Explorer 10 (IE10) tiene dos modos:

- El modo estilo Metro, que ejecuta el modo de seguridad *Enhanced Protection Mode* y
- Y el modo escritorio, que ejecuta el modo de seguridad *Modo compatible*.

Para poder ejecutar el complemento Authentic Browser en IE10 debe usar el Modo compatible (**Herramientas | Opciones | Seguridad**).

4 Referencia del usuario

En la referencia del usuario se enumeran y describen todos los mecanismos, objetos y enumeraciones de Authentic Browser.

- [Mecanismos](#) ⁴⁶
- [Objetos](#) ⁶⁶
- [Enumeraciones](#) ¹⁷⁸

4.1 Mecanismos

Esta sección describe los mecanismos de Authentic Browser más frecuentes.

4.1.1 Eventos: Punto de conexión para IE

Authentic Browser ofrece varios eventos de punto de conexión (consulte [Eventos: Referencia](#)⁴⁸), para los que puede suministrar controladores de eventos en los bloques `SCRIPT` de la página HTML.

Nota: La descripción de esta sección **corresponde solamente a Internet Explorer**.

Los siguientes ejemplos incluyen bloques `SCRIPT` para los eventos `ControlInitialized` y `SelectionChanged`:

ControlInitialized

Este evento de punto de conexión se desencadena inmediatamente después de crearse e inicializarse el control. Fuera del controlador del evento `ControlInitialized` se pueden controlar más scripts de inicio para el control.

```
<SCRIPT LANGUAGE=javascript FOR=objPlugIn EVENT=controlinitialized>
  // añada aquí su código
</SCRIPT>
```

SelectionChanged

El evento `SelectionChanged` se desencadena cada vez que cambia la selección actual en la vista. Puede usar un bloque `SCRIPT` para ejecutar el código personal del evento.

```
<SCRIPT LANGUAGE=javascript FOR=objPlugIn EVENT=selectionchanged>
  // añada aquí su código
</SCRIPT>
```

Recuerde que el objeto [Authentic.event](#)⁷⁵ no se rellena cuando ocurre este evento. Si en su script hay controladores de eventos registrados, las propiedades del objeto [Authentic.event](#)⁷⁵ contienen valores del último evento que ocurrió. El objeto [Authentic.CurrentSelection](#)⁷¹ ahora contiene información válida.

Cargar archivos SPS, XSD y XML sin la intervención del usuario

Si quiere cargar archivos `.sps`, `.xsd` y `.xml` sin la intervención del usuario cuando se cargue la página HTML, lo mejor es escribir un controlador de eventos que controle el evento `ControlInitialized`. Otra opción es usar un contenedor de propiedades, como en el segundo ejemplo que aparece a continuación:

Método recomendado:

```
<SCRIPT LANGUAGE="javascript" FOR=objPlugIn EVENT="ControlInitialized">
objPlugIn.SchemaLoadObject.URL = "http://suservidor/OrgChart.xsd"
objPlugIn.XMLDataLoadObject.URL = "http://suservidor/OrgChart.xml"
objPlugIn.DesignDataLoadObject.URL = "http://suservidor/OrgChart.sps"
```

```
objPlugIn.StartEditing()
</SCRIPT>
```

O

```
<OBJECT id=objPlugIn style="WIDTH:600px; HEIGHT:500px"
codeBase="http://suservidor/cabfiles/AuthenticBrowserEdition.CAB#Version=12,3,0,0"
classid=clsid:B4628728-E3F0-44a2-BEC8-F838555AE780>
<PARAM NAME="XMLDataURL" VALUE="http://suservidor/OrgChart.xml">
<PARAM NAME="SPSDataURL" VALUE="http://suservidor/OrgChart.sps">
<PARAM NAME="SchemaDataURL" VALUE="http://suservidor/OrgChart.xsd">
</OBJECT>
```

No recomendamos cargar estos archivos en un controlador de eventos que controle los elementos "body" del evento "onload" porque puede que el control de Authentic Browser se inicialice después de que se desencadene el evento "onload". Si eso ocurre, los métodos y las propiedades del complemento no estarán disponibles y los archivos no se cargarán. Consulte el apartado [El elemento OBJECT](#)²² para obtener más información.

Método no recomendado:

```
<SCRIPT LANGUAGE="javascript">
function load () {
objPlugIn.SchemaLoadObject.URL = "http://suservidor/OrgChart.xsd"
objPlugIn.XMLDataLoadObject.URL = "http://suservidor/OrgChart.xml"
objPlugIn.DesignDataLoadObject.URL = "http://suservidor/OrgChart.sps"
objPlugIn.StartEditing()
}
</SCRIPT>

<body onload = "load files">
```

4.1.2 Eventos: Botones de la barra de herramientas

Cada botón de las barras de herramientas tiene un comportamiento predeterminado que puede cambiarse. El evento `AuthenticCommand` sirve para añadir tareas adicionales o para redefinir por completo el comportamiento predeterminado de los botones de las barras de herramientas. Los scripts pueden usar el evento `AuthenticCommand` para recibir una notificación cada vez que el usuario haga clic en un icono de una barra de herramientas.

Recuerde que no todos los comandos (de la colección [Authentic.UICommands](#)⁹⁰) tiene su propio evento asociado. Para averiguar en qué icono hizo clic el usuario, el script debe revisar la propiedad [AuthenticEvent.srcElement](#)¹⁰⁰, que contiene una referencia a un objeto [AuthenticCommand](#)⁹².

Ejemplo

```
// controlador de eventos para OnDocEditCommand
<SCRIPT LANGUAGE=javascript FOR=objPlugIn EVENT=doceditcommand>
// nos interesa el botón k_CommandSave
if(objPlugIn.event.srcElement.CommandID == 1)
```

```

{
    // en lugar de HTTP PUT
    // queremos usar HTTP POST
    objPlugIn.SavePOST();

    // no continúa una ejecución estándar
    objPlugIn.event.cancelBubble = true;
}
</SCRIPT>

```

Referencia

Consulte el apartado [AuthenticToolBarButton.CommandID](#)¹³⁴ para ver los comandos disponibles.

4.1.3 Eventos: Referencia

Lista de eventos de punto de conexión

Nombre	desde TypeLib	Descripción
SelectionChanged	1.0	Cambió la selección en el editor. El objeto Authentic.CurrentSelection ⁷¹ ahora contiene información válida. Las propiedades del objeto Authentic.event ⁷⁵ no están definidas.
Controllinitialized	1.2	El control está totalmente cargado e inicializado. Las propiedades del objeto Authentic.event ⁷⁵ no están definidas.
dragover	1.8	Está produciéndose una operación de arrastrar.
drop	1.8	Está produciéndose una operación de colocar.
keydown	1.8	Se pulsó una tecla pero no se soltó.
keyup	1.8	Se dejó de pulsar una tecla del teclado. Es el evento que se suele usar para reaccionar a pulsaciones de teclas.
keypressed	1.8	Con cualquier pulsación de teclado.
mousemove	1.8	Se movió el puntero del ratón.
buttonup	1.8	Se dejó de pulsar uno de los botones del ratón.
buttondown	1.8	Se pulsó uno de los botones del ratón.
contextmenu	1.8	Se envía después de recibir WM_CONTEXTMENU.
editpaste	1.8	Se le llama antes de que tenga lugar una operación de pegar.
editcut	1.8	Se le llama antes de que tenga lugar una operación de cortar.
editcopy	1.8	Se le llama antes de que tenga lugar una operación de copiar.
editclear	1.8	Se le llama antes de que tenga lugar una operación de borrar.
doceditcommand	1.8	Cuando se ejecuta un comando de Authentic y para implementar un controlador de comandos personal. Consulte Eventos: Botones de la barra de herramientas ⁴⁷ . Las propiedades del objeto Authentic.event ⁷⁵ no están definidas.
buttondoubleclick	1.8	Se pulsó dos veces seguidas uno de los botones del ratón.

Si no se menciona ninguna excepción en la descripción, las propiedades del objeto [Authentic.event](#)⁷⁵ se definen al llamar al controlador de eventos.

Nota: Estos controladores de eventos son sincrónicos, es decir, se les llama en cuanto ocurre el evento.

4.1.4 Acceder al contenido del documento y modificarlo

Para acceder al contenido del documento y modificarlo puede usar los objetos `AuthenticRange`, `AuthenticView` y `Authentic` (así como sus propiedades y métodos).

Cuando haya una coincidencia entre las funciones de las interfaces `AuthenticRange` y `AuthenticView`, por una parte, y la interfaz que aporta el objeto `Authentic`, por otra, la interfaz preferida es `AuthenticRange` y `AuthenticView`. Las funciones coincidentes del objeto `Authentic` se están eliminando de forma gradual y serán obsoletas en futuras versiones.

4.1.5 Operaciones de edición

Cuando los datos XML se presentan en la vista `Authentic`, los elementos se pueden manipular uno por uno mediante operaciones de edición estándar (cortar, copiar, pegar, etc.). Sin embargo, no todos los elementos de datos XML se pueden editar. Por tanto, primero hace falta probar si está permitido editarlos. El mecanismo utilizado es este: primero compruebe si la operación de edición está habilitada. Si lo está, llame al método que debe realizar la operación de edición. El único método que no tiene un test de prueba es el método `EditSelectAll`, que selecciona automáticamente todos los elementos que aparecen en el documento.

A continuación aparecen las propiedades y los métodos que realizan operaciones de edición. Cada propiedad devuelve un valor booleano. Los métodos no tienen parámetros.

IsEditUndoEnabled	Authentic.EditUndo ⁷³	Deshacer una operación de edición.
IsEditRedoEnabled	Authentic.EditRedo ⁷³	Rehacer una operación de edición.
IsEditCopyEnabled	Authentic.EditCopy ⁷²	Copiar el texto seleccionado en el portapapeles.
IsEditCutEnabled	Authentic.EditCut ⁷³	Cortar el texto seleccionado y copiarlo en el portapapeles.
IsEditPasteEnabled	Authentic.EditPaste ⁷³	Pegar el texto del portapapeles en la posición actual del cursor.
IsEditClearEnabled	Authentic.EditClear ⁷²	Borrar el texto seleccionado del documento XML.

4.1.6 Buscar y reemplazar

El método [Authentic.FindDialog](#)⁷⁵ abre el cuadro de diálogo "Buscar", donde el usuario puede insertar el término de búsqueda.

El método [Authentic.FindNext](#)⁷⁵ encuentra la siguiente instancia del término de búsqueda. El método `FindNext` se puede probar con la propiedad booleana `IsFindNextEnabled`.

Una variación del método `FindDialog` es el método [Authentic.ReplaceDialog](#)⁸⁴. Esta operación busca un término concreto y lo reemplaza con un valor indicado por el usuario en el cuadro de diálogo "Reemplazar". Si después se llama al método `FindNext`, se busca y reemplaza la siguiente instancia del término de búsqueda.

4.1.7 Operaciones en filas de tablas

En la vista `Authentic` se pueden crear tablas dinámicas a partir de estructuras de elementos repetidos. En la tabla dinámica cada fila representa una instancia del elemento repetido. Una vez creada la tabla dinámica, el usuario de la vista `Authentic` puede manipular las filas y sus datos. Estas operaciones en las filas se pueden llevar a cabo con ayuda de un script.

Si quiere usar un script externo para realizar operaciones en filas, debe seguir estos dos pasos:

1. El primer paso consiste en comprobar si la fila en la que está el cursor usa una propiedad. Se usa una propiedad como `IsRowInsertEnabled`, por ejemplo, que devuelve el valor `True` o `False`.
2. Si el valor devuelto es `True`, se puede llamar al método de fila necesario.

A continuación aparecen las propiedades y los métodos que realizan operaciones en filas. Cada propiedad devuelve un valor booleano. Los métodos no tienen parámetros.

<code>IsRowInsertEnabled</code>	Authentic.RowInsert ⁸⁵	Operación de inserción de filas.
<code>IsRowAppendEnabled</code>	Authentic.RowAppend ⁸⁴	Operación de anexión de filas.
<code>IsRowDeleteEnabled</code>	Authentic.RowDelete ⁸⁴	Operación de eliminación de filas.
<code>IsRowMoveUpEnabled</code>	Authentic.RowMoveUp ⁸⁵	Subir los datos XML una fila.
<code>IsRowMoveDownEnabled</code>	Authentic.RowMoveDown ⁸⁵	Bajar los datos XML una fila.
<code>IsRowDuplicateEnabled</code>	Authentic.RowDuplicate ⁸⁵	Hacer una copia de la fila actual.

4.1.8 Teclas de acceso rápido

Estas son las teclas de acceso rápido que puede usar dentro de la ventana de `Authentic Browser`:

CTRL + P	Imprimir documento
CTRL + Z	Deshacer
CTRL + Y	Rehacer
CTRL + X	Cortar
CTRL + C	Copiar
CTRL + V	Pegar
CTRL + A	Seleccionar todo
CTRL + F	Abrir el cuadro de diálogo "Buscar"
CTRL + H	Abrir el cuadro de diálogo "Buscar y reemplazar"

4.1.9 Botones de estado del texto

La barra de herramientas de la vista Authentic del complemento Authentic Browser ofrece iconos de estado del texto. Estos iconos insertan elementos que tienen propiedades de formato de texto predefinidas. Los requisitos para poder usar esta función son:

- El elemento que se debe crear como icono de estado del texto debe estar declarado como plantilla global en el esquema, y
- El archivo SPS debe contener las definiciones necesarias (consulte la documentación de Altova StyleVision).

El complemento necesita tener la ruta de acceso del archivo `.png` que se usará como icono de estado del texto o como botón en la barra de herramientas. Esta URL del archivo de imagen del icono se suministra como valor de [Authentic.TextStateBmpURL](#)⁸⁹.

4.1.10 Ayudantes de entrada

El complemento Authentic Browser permite usar ventanas de ayudantes de entrada en la vista Authentic igual que en XMLSpy. De este modo, el usuario de vista Authentic tiene acceso a los elementos, atributos y entidades que están permitidos en cada posición del documento. Los ayudantes de entrada están deshabilitados por defecto y no aparecen a no ser que se activen explícitamente. Para habilitar los ayudantes de entrada puede utilizar estos parámetros `PROPERTYBAG` en la etiqueta `OBJECT` de [Internet Explorer](#)²²:

<code>EntryHelpersEnabled</code>	TRUE / FALSE
<code>EntryHelperSize</code>	Tamaño en píxeles
<code>EntryHelperAlignment</code>	Toma los valores de SPYAuthenticToolbarAlignment ¹⁸²
<code>EntryHelperWindows</code>	Toma los valores de SPYAuthenticEntryHelperWindows ¹⁸¹ . Se permiten todas las combinaciones posibles

En lugar de usar los parámetros en la etiqueta `OBJECT` también puede usar las propiedades y métodos en la API:

Propiedades

- [Authentic.EntryHelpersEnabled](#)⁷⁴
- [Authentic.EntryHelperAlignment](#)⁷⁴
- [Authentic.EntryHelperSize](#)⁷⁴
- [Authentic.EntryHelperWindows](#)⁷⁴

Método

- [Authentic.RedrawEntryHelpers](#)⁸³

4.1.11 Paquetes

El complemento Authentic Browser admite el uso de paquetes para expandir las funciones del módulo del programa.

Los paquetes se almacenan en el servidor y se instalan en el cliente por encargo desde la interfaz gráfica de Authentic Browser directamente. Una vez instalado, el paquete se activa cada vez que se inicia sesión y hasta que el usuario lo quite.

Actualmente Authentic Browser es compatible con paquetes de corrector ortográfico.

Esta sección tiene dos apartados:

- [Trabajar con paquetes](#)⁵², que describe el funcionamiento de los paquetes.
- [Paquetes del corrector ortográfico](#)⁵⁴, que explica cómo se almacenan en el servidor los paquetes del corrector ortográfico, cómo se instalan y cómo se utilizan.

4.1.11.1 Trabajar con paquetes

Los paquetes se almacenan en el servidor y se instalan localmente (en el cliente) por encargo desde la interfaz gráfica de Authentic Browser directamente. Una vez instalado, el paquete se activa cada vez que se inicia sesión y hasta que el usuario lo quite.

Gestión de paquetes

Authentic Browser ofrece una función que permite al usuario controlar los paquetes disponibles. Esta función está disponible en el cuadro de diálogo "Gestión de paquetes".

Para habilitar la función de gestión de paquetes debe usar el parámetro `LoaderSettingsFileURL` en la [página HTML para el complemento Authentic](#)²⁰. Este parámetro especifica la URL del archivo `LoaderSettings` para la gestión de paquetes (que puede encontrarse en cualquier ubicación a la que tenga acceso):

En una [página HTML para Internet Explorer](#)²² el parámetro `LoaderSettingsFileURL` se usa como elemento `PARAM` secundario del elemento `OBJECT`:

```
<OBJECT>
...
<PARAM NAME="LoaderSettingsFileURL"
VALUE="http://www.server.com/AuthenticFiles/XMLSpyPlugInLoaderSettings.xml" />
</OBJECT>
```

El archivo `LoaderSettings` contiene las definiciones de los paquetes y sus URL. Más abajo puede ver un fragmento de un archivo `LoaderSettings`.

Ejemplo de un archivo XML `LoaderSettings`

El elemento de documento del archivo XML `LoaderSettings` es `loadersettings` (ver *fragmento más abajo*). Este elemento puede contener varios elementos secundarios `package` y varios elementos `zippackage`.

- El elemento `package` hace referencia a los paquetes de ortografía `.pck` antiguos y se usa en las versiones de Authentic Browser anteriores a la versión 2011r3. Estas versiones de Authentic Browser ignoran el elemento `zippackage`.
- El elemento `zippackage` hace referencia a los paquetes de ortografía `.zip` y se usa en las versiones de Authentic Browser posteriores a la versión 2011r3. Estas versiones de Authentic Browser ignoran el elemento `package`.

Para obtener más información consulte el apartado [Paquetes del corrector ortográfico](#) ⁵⁴.

```
<?xml version="1.0" encoding="UTF-8"?>
<loadersettings>

  <zippackage mode="user_demand" category="spelling">
    <packageurl>Portuguese (Brazilian).zip</packageurl>
    <description>Portuguese (BR) language pack.</description>
  </zippackage>

  <zippackage mode="user_demand" category="spelling">
    <packageurl>Portuguese (Brazilian).zip</packageurl>
    <description>Portuguese (BR) language pack.</description>
  </zippackage>

  <package mode="user_demand" id="SentrySpellChecker_EAM_only"
    category="spelling" version="1">
    <packageurl>PlugIn/SentrySpellChecker_EAM_only.pck</packageurl>
    <description>Sentry Spellchecker (EN-US)</description>
  </package>

  <package mode="user_demand" id="SentrySpellChecker_EALL"
    category="spelling" version="1">
    <packageurl>PlugIn/SentrySpellChecker_EALL.pck</packageurl>
    <description>Sentry SpellChecker EN with Legal and Medical.</description>
  </package>

</loadersettings>
```

Es necesario tener en cuenta que:

- La ubicación del paquete se debe especificar como contenido del elemento secundario `packageurl` del elemento `package` o `zippackage`. La ruta de acceso puede ser absoluta o relativa al archivo HTML que llama al archivo `LoaderSettings`. El paquete puede almacenarse en cualquier ubicación del servidor.
- Si se elimina un paquete del archivo XML (eliminando el elemento `package`) el paquete no se podrá instalar en el cliente.
- El atributo `mode` del elemento `package` especifica el nivel de influencia que puede tener el usuario sobre la decisión de instalar el paquete. Se pueden usar estos valores:
 1. `user`: Se le pregunta al usuario cada vez que se inicia Authentic Browser si desea instalar el paquete.
 2. `user_demand`: Instala el paquete cuando lo solicita el usuario. El usuario lo solicita haciendo clic en el botón de la barra de herramientas para revisar la ortografía o para abrir el cuadro de diálogo "Gestión de paquetes".

3. force: Provoca la instalación del paquete sin notificar al usuario.
- El contenido del elemento secundario `description` del elemento `package` se puede editar y se usará como texto descriptivo en el cuadro de diálogo "Gestión de paquetes".

Una sola instalación de los paquetes

Si actualiza su versión de Authentic Browser e instala una versión más reciente, no hace falta volver a instalar los paquetes que ya están instalados en el cliente. Los paquetes que ya están instalados se utilizarán en la nueva versión. Authentic Browser lee el archivo `LoaderSettings` del servidor, recupera la información del paquete de este archivo y comprueba si el paquete está instalado localmente.

Nota: Los paquetes del corrector ortográfico compatibles con la versión 2011r3 y posterior (paquetes .zip) usan un corrector ortográfico distinto al de las versiones anteriores (paquetes .pck). Las versiones 2011r3 y superiores de Authentic Browser usan paquetes de corrector ortográfico .zip. Las versiones anteriores de Authentic Browser usan paquetes .pck.

4.1.11.2 Paquetes del corrector ortográfico

A partir de la versión 2011r3, Authentic Browser utiliza diccionarios Hunspell.

Ubicación de la instalación

En el equipo cliente los diccionarios están en la carpeta `Lexicons` situada en esta ubicación:

En Windows 7/8: `C:\ProgramData\Altova\SpellChecker\Lexicons`

En Windows XP: `C:\Documents and Settings\All Users\Application Data\Altova\SharedBetweenVersions\SpellChecker\Lexicons`

Si hay algún producto de Altova se instaló en el equipo cliente, entonces se instalaron también diccionarios integrados en la carpeta `Lexicons` de la ubicación correspondiente. Todos los diccionarios instalados se comparten con todos los usuarios del equipo y con los demás productos de Altova que estén instalados.

Puede descargar más diccionarios desde el [sitio web de Altova](http://wiki.services.openoffice.org/wiki/Dictionaryes) o desde otros sitios web, como por ejemplo <http://wiki.services.openoffice.org/wiki/Dictionaryes> o <http://extensions.services.openoffice.org/en/dictionaries>. Como los diccionarios Hunspell están basados en los diccionarios Myspell, estos también se pueden instalar en la carpeta `Lexicons`.

Nota: los diccionarios integrados en el software de Altova no refleja ningún tipo de preferencia por parte de Altova en cuanto al uso de ciertos idiomas. Simplemente se basa en la disponibilidad de diccionarios que pueden distribuirse con software comercial, como las licencias [MPL](#), [LGPL](#) o [BSD](#). Existen muchos otros diccionarios de código abierto pero se distribuyen con licencias más restrictivas, como la licencia [GPL](#). Muchos de estos diccionarios se pueden descargar desde el [sitio web de Altova](http://wiki.services.openoffice.org/wiki/Dictionaryes). 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.

Archivos de los diccionarios y estructura de la carpeta de diccionarios en el equipo cliente

Cada diccionario del equipo cliente está compuesto por dos archivos de diccionario Hunspell: el archivo `.aff` y el archivo `.dic`.

Todos los diccionarios están instalados en la carpeta `Lexicons` de la ubicación correspondiente. Dentro de la carpeta `Lexicons` cada idioma se almacena en una carpeta distinta: `<idioma>\<archivos del diccionario>`. El nombre de la carpeta del idioma equivale al nombre que tiene el diccionario en Authentic Browser. A continuación puede ver la estructura de la carpeta `Lexicons` y el nombre correspondiente de los diccionarios tal y como aparecerían en Authentic Browser.

```
Lexicons
|
|-- English (British)           Nombre del diccionario: English (British)
|   |
|   |-- archivo .aff
|   |   archivo .dic
|
|-- English (US)             Nombre del diccionario: English (US)
|   |
|   |-- archivo .aff
|   |   archivo .dic
|
|-- Spanish                   Nombre del diccionario: Spanish
|   |
|   |-- archivo .aff
|   |   archivo .dic
```

Archivos de los diccionarios en el servidor

En el servidor los diccionarios se deben guardar en archivos ZIP. Cuando el archivo ZIP se descarga e instala en el cliente, se crea una carpeta del idioma en la carpeta `Lexicons` (ver más arriba) y los archivos `.aff` y `.dic` descomprimidos se colocan en esta carpeta. Por ejemplo, el archivo `English (British).zip` crea la carpeta: `<ruta de acceso>\Lexicons\English (British)` e instala los archivos `.aff` y `.dic` en esta carpeta.

También se pueden comprimir varios archivos ZIP en un solo archivo ZIP, que se puede guardar después en el servidor. Cuando este archivo ZIP se descarga e instala en el cliente, el nombre de los archivos ZIP del directorio más interior se usarán para el nombre de las carpetas de la carpeta `Lexicons`. Por ejemplo, los archivos `English (British).zip` y `English (US).zip` se pueden comprimir en un archivo ZIP llamado `English.zip`, que se coloca en el servidor. Una vez completada su instalación, se crearán dos carpetas en la carpeta `Lexicons`: `<ruta de acceso>\Lexicons\English (British)` y `<ruta de acceso>\Lexicons\English (US)`.

Cómo se instalan los diccionarios

Cuando se abre una página HTML para el complemento Authentic Browser en el explorador, los paquetes de instalación se pueden instalar en el cliente bien automáticamente, bien cuando lo solicite el usuario. Así funciona el mecanismo de instalación:

1. La página HTML que se abre en Authentic Browser remite al [archivo LoaderSettings](#)⁵².
2. El [archivo LoaderSettings](#)⁵² remite al paquete del corrector ortográfico que está en el servidor. El [archivo LoaderSettings](#)⁵² también especifica, por cada paquete, si el paquete se debe instalar

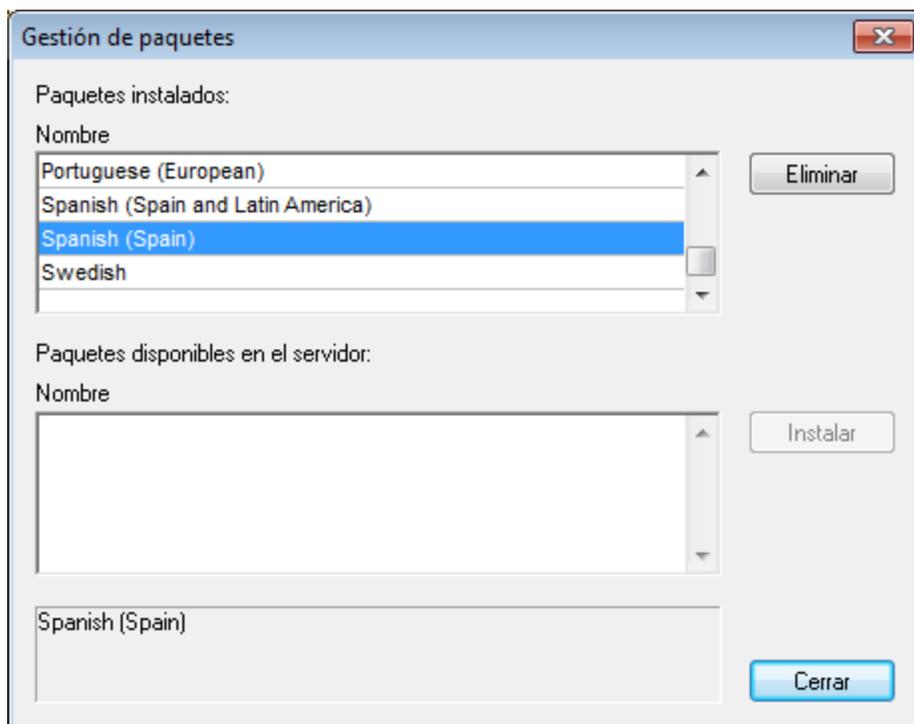
automáticamente justo después de que se cargue la página HTML o si se debe preguntar al usuario por la instalación.

3. El paquete del diccionario se instala en el cliente desde el archivo comprimido del servidor.

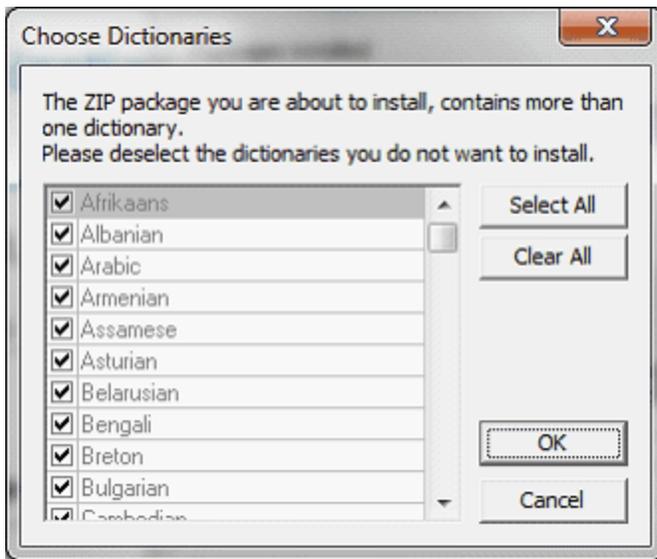
Instalación de los diccionarios

Los paquetes del corrector ortográfico se almacenan en el servidor como archivos comprimidos ZIP y se pueden instalar localmente desde la interfaz gráfica de Authentic Browser directamente. Para instalar un paquete haga clic en el icono **Gestión de paquetes** de la barra de herramientas de Authentic Browser. Esto abre el cuadro de diálogo "Gestión de paquetes" (*imagen siguiente*). Este cuadro de diálogo también aparece si no hay ningún diccionario instalado y se hace clic en el icono **Corrección ortográfica**.

En el cuadro de diálogo "Gestión de paquetes" se enumeran todos los paquetes que están disponibles en el servidor. Los nombres que aparecen en el cuadro de diálogo equivalen a los nombres de los archivos ZIP. Seleccione qué archivos desea instalar y después haga clic en **Instalar**. Para eliminar paquetes instalados, seleccione un paquete del panel superior y haga clic en **Eliminar**.



Si el archivo ZIP seleccionado contiene varios paquetes ZIP, aparece el cuadro de diálogo "Elegir diccionarios" (imagen siguiente). En este cuadro de diálogo puede elegir qué paquetes desea instalar o desactivar la casilla de los paquetes que no desea instalar.



Si la instalación de un paquete sobrescribirá un paquete ya existente, aparece un mensaje de advertencia. El usuario de Authentic Browser puede continuar y sobrescribir el paquete actual o cancelar la instalación.

4.1.12 Interfaz XMLData

XMLData ofrece acceso a los elementos del archivo XML que está visible. Permite realizar todas las modificaciones necesarias a los elementos de la estructura XML. La principal función de XMLData es:

1. Acceder a los nombres y los valores de todo tipo de elementos (p. ej. elementos, atributos...).
2. Crear elementos nuevos de todo tipo.
3. Insertar y anexar elementos nuevos.
4. Borrar elementos secundarios ya existentes.

Si ya conoce la interfaz XMLData porque la usó en la API de XMLSpy, debe tener en cuenta algunos aspectos cuando se creen e inserten elementos nuevos en el archivo XML o cuando se cambie el nombre de los elementos ya existentes. Para más información siga leyendo este apartado.

Estructura de XMLData

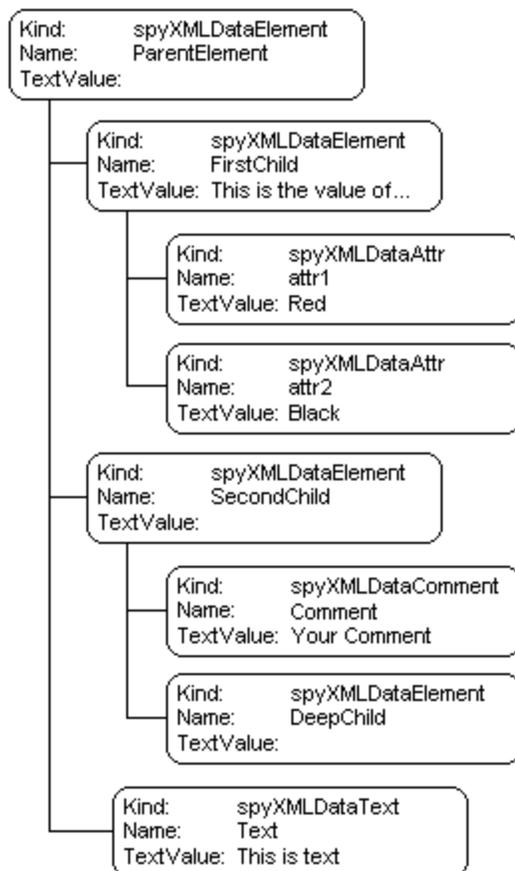
Para poder usar la interfaz XMLData, antes es necesario conocer cómo se crean asignaciones de datos entre el archivo XML y la estructura XMLData. También es muy importante saber que XMLData no tiene una rama separada de objetos para los atributos. Los atributos de un elemento también son sus secundarios. Un elemento puede tener otros tipos de secundarios (como elementos secundarios contenido de texto). La propiedad [XMLData.Kind](#)¹⁷⁶ le ofrece la oportunidad de diferenciar entre tipos de secundarios distintos del mismo elemento. Puede ver un ejemplo a continuación:

Ejemplo

Este código XML,

```
<ElementoPrimario>
  <PrimerSecundario atr1="Rojo" atr2="Negro">
    Este es el valor de PrimerSecundario
  </PrimerSecundario>
  <SegundoSecundario>
    <!--Comentario-->
    </SecundarioProfundo>
  </SegundoSecundario>
  Este es el texto
</ElementoPrimario>
```

se asigna a esta estructura de objetos de XMLData:



El primario de todos los elementos XML que hay dentro de un archivo es la propiedad [Authentic.XMLRoot](#)⁹². Use este objeto XMLData para obtener referencias a los demás elementos XML de la estructura.

Nombre y valor de los elementos

Para conocer y modificar el nombre y el valor de todo tipo de elementos XML use las propiedades [XMLData.Name](#)¹⁷⁶ y [XMLData.TextValue](#)¹⁷⁷. Es posible que varios tipos de objetos de XMLData y elementos vacíos no tengan asociado ningún valor de texto. No se recomienda cambiar el nombre de un elemento XML ya existente en la vista Authentic. El nombre tiene una gran relevancia en la forma en que la vista Authentic

presenta el contenido del elemento. Consulte la documentación de Altova StyleVision para obtener más información.

Creación e inserción de objetos XMLData nuevos

Cree el objeto XMLData nuevo (como un elemento, atributo o nodo de texto):

1. Use el método [Authentic.CreateChild](#)⁷¹ para crear el objeto XMLData nuevo. Defina el nombre y el valor del objeto nuevo antes de insertarlo (ver punto 3 más abajo).
2. Inserte el objeto XMLData nuevo en la ubicación correcta haciendo referencia a su primario: (i) si el secundario nuevo debe ser el último secundario del primario, utilice el método [XMLData.AppendChild](#)¹⁶⁹ para insertar el objeto XMLData, (ii) para insertar el secundario nuevo en otra posición de la secuencia de objetos secundarios, entonces use el método [XMLData.GetFirstChild](#)¹⁷³ o el método [XMLData.GetNextChild](#)¹⁷³ para localizar al secundario antes del cual debe insertarse el secundario nuevo; ahora inserte el secundario nuevo con [XMLData.InsertChild](#)¹⁷⁵. El secundario nuevo se inserta inmediatamente antes del secundario actual.

Nota: en vista Authentic se puede crear un elemento nuevo junto con su subestructura porque la estructura del elemento se define en el esquema XML. Sin embargo, es en la configuración del nodo del SPS donde se define si se crea o no la subestructura. Consulte la documentación de StyleVision para más información.

En el ejemplo siguiente se añade un tercer secundario entre el elemento <PrimerSecundario> y el <SegundoSecundario>:

```
Dim objParent
Dim objChild
Dim objNewChild

Set objNewChild = objPlugIn.CreateChild(spyXMLDataElement)
objNewChild.Name = "OneAndAHalf"

'objParent is set to <ElementoPrimario>
'GetFirstChild(-1) gets all children of the parent element
'and move to <SegundoSecundario>
Set objChild = objParent.GetFirstChild(-1)
Set objChild = objParent.GetNextChild

objParent.InsertChild objNewChild
Set objNewChild = Nothing
```

Los nodos secundarios de un elemento se deben insertar en determinado orden. Inserte primero los atributos del elemento y después los elementos secundarios del mismo. El contenido de texto de un elemento se debe añadir como nodo secundario de tipo texto; para crearlo puede usar [Authentic.CreateChild](#)⁷¹ con el valor de parámetro 6. Para establecer el contenido de texto de un elemento debe introducirlo como el valor de texto del nodo secundario.

Copiar objetos XMLData ya existentes

Si quiere mover objetos XMLData a otra posición del mismo archivo, no puede usar los métodos XMLData.InsertChild y XMLData.AppendChild. Estos métodos solo sirven para objetos XMLData nuevos.

En cambio, debe copiar la jerarquía de objetos a mano. Por ejemplo, podría usar esta función escrita en JavaScript para copiar XMLData recursivamente:

```
// esta función devuelve una copia completa del objeto XMLData
function GetCopy(objXMLData)
{
    var objNew;
    objNew = objPlugIn.CreateChild(objXMLData.Kind);

    objNew.Name = objXMLData.Name;
    objNew.TextValue = objXMLData.TextValue;

    if(objXMLData.HasChildren) {
        var objChild;
        objChild = objXMLData.GetFirstChild(-1);

        while(objChild) {
            try {
                objNew.AppendChild(GetCopy(objChild));
                objChild = objXMLData.GetNextChild();
            }
            catch(e) {
                objChild = null;
            }
        }
    }

    return objNew;
}
```

Eliminar objetos XMLData

XMLData ofrece dos métodos para la eliminación de objetos secundarios: [XMLData.EraseAllChildren](#)¹⁷⁰ y [XMLData.EraseCurrentChild](#)¹⁷⁰.

Para eliminar objetos XMLData primero debe tener acceso al primario de los elementos que desea eliminar. Para obtener una referencia al objeto XMLData primario utilice [XMLData.GetFirstChild](#)¹⁷³ y [XMLData.GetNextChild](#)¹⁷³.

Para ver un ejemplo de cómo eliminar elementos XML, consulte la descripción de los métodos [XMLData.EraseAllChildren](#)¹⁷⁰ y [XMLData.EraseCurrentChild](#)¹⁷⁰.

4.1.13 DOM y XMLData

La interfaz XMLData ofrece total acceso a la estructura XML del documento actual con menos métodos que DOM y de manera mucho más sencilla. La interfaz XMLData ofrece una manera minimalista de leer y modificar datos XML nuevos o ya existentes. Sin embargo, puede usar la estructura DOM si quiere tener acceso desde una fuente de datos externa o si prefiere la implementación MSXML DOM.

Las funciones **ProcessDOMNode()** y **ProcessXMLDataNode()** que aparecen más abajo convierten los segmentos de una estructura XML de XMLData a DOM.

Para usar la función **ProcessDOMNode()**:

- pase el elemento raíz del segmento DOM que quiere convertir en **objNode** y
- pase el objeto del complemento con el método CreateChild() en **objCreator**

Para usar la función **ProcessXMLDataNode()**:

- pase el elemento raíz del segmento XMLData en **objXMLData** y
- pase el objeto DOMDocument creado con MSXML en **xmlDoc**

Conversión de DOM en XMLData

El siguiente código muestra cómo convertir segmentos DOM en XMLData.

```

////////////////////////////////////
// conversión de segmentos DOM en XMLData

function ProcessDOMNode(objNode,objCreator)
{
    var objRoot;
    objRoot = CreateXMLDataFromDOMNode(objNode,objCreator);

    if(objRoot) {
        if((objNode.nodeValue != null) && (objNode.nodeValue.length > 0))
            objRoot.TextValue = objNode.nodeValue;

        // agregar atributos
        if(objNode.attributes) {
            var Attribute;
            var oNodeList = objNode.attributes;

            for(var i = 0;i < oNodeList.length; i++) {
                Attribute = oNodeList.item(i);

                var newNode;
                newNode = ProcessDOMNode(Attribute,objCreator);

                objRoot.AppendChild(newNode);
            }
        }

        if(objNode.hasChildNodes) {
            try {
                // agregar secundarios
                var Item;
                oNodeList = objNode.childNodes;

                for(var i = 0;i < oNodeList.length; i++) {
                    Item = oNodeList.item(i);

                    var newNode;

```

```

        newNode = ProcessDOMNode(Item,objCreator);

        objRoot.appendChild(newNode);
    }
}
catch(err) {
}
}
}

return objRoot;
}

```

```

function CreateXMLDataFromDOMNode(objNode,objCreator)
{
    var bSetName = true;
    var bSetValue = true;

    var nKind = 4;

    switch(objNode.nodeType) {
        case 2:nKind = 5;break;
        case 3:nKind = 6;bSetName = false;break;
        case 4:nKind = 7;bSetName = false;break;
        case 8:nKind = 8;bSetName = false;break;
        case 7:nKind = 9;break;
    }

    var objNew = null;
    objNew = objCreator.CreateChild(nKind);

    if(bSetName)
        objNew.Name = objNode.nodeName;

    if(bSetValue && (objNode.nodeValue != null))
        objNew.TextValue = objNode.nodeValue;

    return objNew;
}

```

Conversión de XMLData en DOM

El siguiente código muestra cómo convertir segmentos XMLData en DOM.

```

////////////////////////////////////
// conversión de segmentos XMLData en DOM

function ProcessXMLDataNode(objXMLData,xmlDoc)
{
    var objRoot;

```

```

objRoot = CreateDOMNodeFromXMLData(objXMLData,xmlDoc);

if(objRoot) {
    if(IsTextNodeEnabled(objRoot) && (objXMLData.TextValue.length > 0))
        objRoot.appendChild(xmlDoc.createTextNode(objXMLData.TextValue));

    if(objXMLData.HasChildren) {
        try {
            var objChild;
            objChild = objXMLData.GetFirstChild(-1);

            while(true) {
                if(objChild) {
                    var newNode;
                    newNode = ProcessXMLDataNode(objChild,xmlDoc);

                    if(newNode.nodeType == 2) {
                        // el nodo secundario es un atributo
                        objRoot.attributes.setNamedItem(newNode);
                    }
                    else
                        objRoot.appendChild(newNode);
                }

                objChild = objXMLData.GetNextChild();
            }
        }
        catch(err) {
        }
    }
}

return objRoot;
}

function CreateDOMNodeFromXMLData(objXMLData,xmlDoc)
{
    switch(objXMLData.Kind) {
        case 4: return xmlDoc.createElement(objXMLData.Name);
        case 5: return xmlDoc.createAttribute(objXMLData.Name);
        case 6: return xmlDoc.createTextNode(objXMLData.TextValue);
        case 7: return xmlDoc.createCDATASection(objXMLData.TextValue);
        case 8: return xmlDoc.createComment(objXMLData.TextValue);
        case 9: return
xmlDoc.createProcessingInstruction(objXMLData.Name,objXMLData.TextValue);
    }

    return xmlDoc.createElement(objXMLData.Name);
}

function IsTextNodeEnabled(objNode)
{

```

```

switch(objNode.nodeType) {
    case 1:
    case 2:
    case 5:
    case 6:
    case 11: return true;
}

return false;
}

```

4.1.14 Scripting de Authentic

La función **Scripting de Authentic** permite crear diseños SPS más flexibles e interactivos. Estos diseños se pueden crear o editar en StyleVision Enterprise y StyleVision Professional y se pueden ver en la vista Authentic de los productos de Altova.

A continuación puede ver con qué productos de Altova es compatible esta función. No obstante, recuerde que en la versión segura del complemento Authentic Browser, los scripts internos se desactivan por motivos de seguridad.

Producto de Altova	¿Creación de scripts de Authentic?	¿Scripts de Authentic habilitados?
StyleVision Enterprise	Sí	Sí
StyleVision Professional	Sí	Sí
StyleVision Basic *	No	No
XMLSpy Enterprise	No	Sí
XMLSpy Professional	No	Sí
AuthenticDesktop Enterprise	No	Sí
Authentic Browser Ent Trusted **	No	Sí
Authentic Browser Ent Untrusted	No	Sí

* No tiene la vista Authentic

** Muestra los diseños que incluyen scripts. Incompatible con la ejecución de macros y con el control de eventos. Los eventos externos se desencadenan.

Los scripts de Authentic se comportan de la misma manera en todos los productos de Altova, por lo que no es necesaria una configuración ni un código especial para cada producto.

Cómo funcionan los scripts de Authentic

El autor del diseño SPS puede usar la función Scripting de Authentic de dos maneras:

- Puede asignar scripts a acciones definidas por el usuario (macros) para diseñar elementos, botones de barras de herramientas y comandos de menús contextuales.
- Puede añadir controladores de eventos al diseño que reaccionen a eventos de la vista Authentic.

Todos los scripts necesarios para convertir los documentos de Authentic en documentos interactivos se crean en StyleVision. Los formularios, las macros y los controladores de eventos se crean dentro del editor de scripts de StyleVision y los scripts se guardan con el archivo SPS. Después, en la vista Diseño de StyleVision, los scripts guardados se asignan a elementos de diseño, a botones de barras de herramientas o a menús contextuales. Cuando en la vista Authentic de un producto de Altova compatible con scripts de Authentic se abre un documento XML basado en este diseño SPS, el documento tendrá las opciones interactivas creadas en el editor de scripts.

Documentación para scripts de Authentic

La documentación necesaria para aprender a trabajar con scripts de Authentic está incluida en la documentación de Altova StyleVision y se puede consultar en el [sitio web de Altova](#).

4.2 Objetos

Esta sección enumera y describe los objetos de Authentic Browser.

4.2.1 Authentic

Métodos

- [StartEditing](#) ⁸⁸
- [LoadXML](#) ⁸²
- [Reset](#) ⁸⁴
- [Save](#) ⁸⁵
- [SavePOST](#) ⁸⁶
- [SaveXML](#) ⁸⁷
- [ValidateDocument](#) ⁹⁰
- [EditClear](#) ⁷²
- [EditCopy](#) ⁷²
- [EditCut](#) ⁷³
- [EditPaste](#) ⁷³
- [EditRedo](#) ⁷³
- [EditSelectAll](#) ⁷³
- [EditUndo](#) ⁷³
- [RowAppend](#) ⁸⁴
- [RowDelete](#) ⁸⁴
- [RowDuplicate](#) ⁸⁵
- [RowInsert](#) ⁸⁵
- [RowMoveDown](#) ⁸⁵
- [RowMoveUp](#) ⁸⁵
- [FindDialog](#) ⁷⁵
- [FindNext](#) ⁷⁵
- [ReplaceDialog](#) ⁸⁴
- [ApplyTextState](#) ⁶⁸
- [IsTextStateApplied](#) ⁸²
- [IsTextStateEnabled](#) ⁸²
- [MarkUpView](#) ⁸²
- [Print](#) ⁸³
- [PrintPreview](#) ⁸³
- [CreateChild](#) ⁷¹
- [GetAllowedElements](#) ⁷⁷
- [GetAllAttributes](#) ⁷⁵
- [GetNextVisible](#) ⁷⁹
- [GetPreviousVisible](#) ⁷⁹
- [SelectionMoveTabOrder](#) ⁸⁸
- [SelectionSet](#) ⁸⁸
- [ClearSelection](#) ⁷⁰
- [attachCallBack](#) ⁶⁸
- [ReloadToolbars](#) ⁸³
- [StartSpellChecking](#) ⁸⁹
- [GetFileVersion](#) ⁷⁸
- [RedrawEntryHelpers](#) ⁸³

[SetUnmodified](#) ⁸⁸
[ClearUndoRedo](#) ⁷¹

Propiedades

[AuthenticView](#) ⁶⁹
[IsEditClearEnabled](#) ⁷⁹
[IsEditCopyEnabled](#) ⁷⁹
[IsEditCutEnabled](#) ⁷⁹
[IsEditPasteEnabled](#) ⁸⁰
[IsEditRedoEnabled](#) ⁸⁰
[IsEditUndoEnabled](#) ⁸⁰
[IsFindNextEnabled](#) ⁸⁰
[IsRowAppendEnabled](#) ⁸⁰
[IsRowDeleteEnabled](#) ⁸¹
[IsRowDuplicateEnabled](#) ⁸¹
[IsRowInsertEnabled](#) ⁸¹
[IsRowMoveDownEnabled](#) ⁸¹
[IsRowMoveUpEnabled](#) ⁸²
[SchemaLoadObject](#) ⁸⁷
[XMLDataLoadObject](#) ⁹¹
[DesignDataLoadObject](#) ⁷²
[XMLDataSaveUrl](#) ⁹¹
[XMLRoot](#) ⁹²
[CurrentSelection](#) ⁷¹
[event](#) ⁷⁵
[validationBadData](#) ⁹¹
[validationMessage](#) ⁹¹
[ToolbarsEnabled](#) ⁹⁰
[ToolbarTooltipsEnabled](#) ⁹⁰
[AutoHideUnusedCommandGroups](#) ⁷⁰
[TextStateToolbarLine](#) ⁸⁹
[TextStateBmpURL](#) ⁸⁹
[ToolbarRows](#) ⁸⁹
[UICommands](#) ⁹⁰
[XMLTable](#) ⁹²
[BaseUrl](#) ⁷⁰
[EntryHelpersEnabled](#) ⁷⁴
[EntryHelperSize](#) ⁷⁴
[EntryHelperAlignment](#) ⁷⁴
[EntryHelperWindows](#) ⁷⁴
[EnableModifications](#) ⁷³
[Modified](#) ⁸³
[SaveButtonAutoEnable](#) ⁸⁶

Eventos

[SelectionChanged](#) ⁸⁷
[ControlInitialized](#) ⁷¹

Descripción

Clase de Authentic

4.2.1.1 Authentic.ApplyTextState

Declaración `ApplyTextState(elementName as String)`

Descripción

Aplica o quita el estado de texto definido por el parámetro `elementName`. Por ejemplo, el parámetro `elementName` podría ser `strong` (negrita) e `italic` (cursiva).

En un documento XML hay segmentos de datos que pueden contener subelementos. Por ejemplo:

```
<b>fragment </ b>
```

La etiqueta HTML `` hace que la palabra aparezca en negrita. Sin embargo, esto ocurre porque el analizador HTML sabe que la etiqueta `` corresponde al estilo negrita. Con XML puede tener mucha más flexibilidad: es posible definir cualquier etiqueta XML para lo que se quiera. Es decir, puede aplicar estado de texto usando XML, pero el estado de texto que se aplica debe formar parte del esquema.

Por ejemplo, en el ejemplo `OrgChart.xml`, `OrgChart.sps` y `OrgChart.xsd` la etiqueta `` es como la etiqueta para el estilo negrita. Para aplicar la negrita se llama al método **ApplyTextState()**. Pero al igual que las operaciones de filas y de edición, es necesario probar si se puede aplicar el estado de texto.

Consulte también [IsTextStateEnabled](#)⁸² y [IsTextStateApplied](#)⁸².

4.2.1.2 Authentic.attachCallback

Obsoleto

En su lugar utilice los eventos de punto de conexión tal y como se describe [aquí](#)⁴⁶.

Declaración `attachCallback(bstrName as String, varCallback as Variant)`

Descripción

La vista `Authentic` ofrece eventos que se pueden controlar por medio de funciones personales de devolución de llamada. Ningún controlador de eventos toma parámetros y se ignorará el valor devuelto. Para recuperar información cuando se genera un evento concreto debe leer las propiedades correspondientes del objeto del [evento](#)⁷⁵.

Lista de eventos disponibles:

```
ondragover
ondrop
onkeydown
onkeyup
```

onkeypressed
onmousemove
onbuttonup
onbuttondown
oneditpaste
oneditcut
oneditcopy

Desde la versión 3.0.0.0:

ondoceditcommand

Desde la versión 5.3.0.0:

onbuttondoubleclick

Ejemplo de JavaScript:

```
// en alguna parte del script:
objPlugIn.attachCallBack("ondragover",OnDragOver);
objPlugIn.attachCallBack("ondrop",OnDrop);

// controladores de eventos
function OnDragOver()
{
    if(    !objPlugIn.event.dataTransfer.ownDrag &&
        objPlugIn.event.dataTransfer.type == "TEXT"))
    {
        objPlugIn.event.dataTransfer.dropEffect = 1;
        objPlugIn.event.cancelBubble = true;
    }
}

// OnDrop() reemplaza el valor de texto completo del elemento XML
// con la selección de la operación de arrastrar
function OnDrop()
{
    var objTransfer = objPlugIn.event.dataTransfer;

    if(    !objTransfer.ownDrag &&
        (objTransfer.type == "TEXT"))
        objPlugIn.event.srcElement.TextValue = objTransfer.getData();
}

```

4.2.1.3 AuthenticView

Declaración [AuthenticView](#) as [AuthenticView](#)¹³⁹ (read-only)

Descripción

Devuelve un objeto que ofrece acceso a propiedades y métodos propios de la vista Authentic.

[AuthenticView](#)¹³⁹ se solapa con las funciones concretas de la vista existente. Las versiones futuras de AuthenticView incluirá todas las funciones propias de la vista. El objeto AuthenticView es la interfaz recomendada para todas las futuras implementaciones.

Ejemplos

Consulte [Ordenar tablas](#)²⁸ para obtener más información sobre cómo usar el objeto AuthenticView.

4.2.1.4 Authentic.AutoHideUnusedCommandGroups

Declaración `AutoHideUnusedCommandGroups` as `Boolean`

Descripción

True si es una barra de herramientas sin usar. Los grupos de comandos se ocultan automáticamente (p. ej. XML, los comandos de tabla, si el archivo SPS actual no admite el uso de tablas XML).

Valor predeterminado: true

4.2.1.5 Authentic.BaseURL

Declaración `BaseURL` as `String`

Descripción

Esta propiedad establece la URL base para resolver rutas de acceso relativas. Si no se establece la URL, se utiliza la ubicación del archivo XML actual.

4.2.1.6 Authentic.ClearSelection

Declaración `ClearSelection()`

Descripción

El método borra la selección actual. A partir de ese momento no se puede obtener la selección por medio de la propiedad CurrentSelection hasta que el usuario seleccione un nodo o se procese una llamada a [SelectionSet\(\)](#)⁸⁸ correctamente.

Para poder borrar un nodo que tiene la selección actual, la selección debe establecerse en otro nodo o borrarse.

4.2.1.7 Authentic.ClearUndoRedo

Declaración [ClearUndoRedo\(\)](#)

Descripción

El método borra el búfer de las operaciones Deshacer/Rehacer.

4.2.1.8 Authentic.Controllnitialized

Declaración [Controllnitialized](#)

Descripción

Este evento se genera cuando se crea e inicializa el control.

Consulte también el apartado [Eventos de punto de conexión](#)⁴⁶.

4.2.1.9 Authentic.CreateChild

Declaración [CreateChild](#)(*nKind* as [SPYXMLDataKind](#)¹⁸²) as [XMLData](#)¹⁶⁸

Valor devuelto

XML nodo nuevo

Descripción

El método CreateChild sirve para crear nodos nuevos que se pueden insertar en la estructura XML del documento actual usando la interfaz [XMLData](#)¹⁶⁸.

Consulte también [XMLData.AppendChild](#)¹⁶⁹ y [XMLData.InsertChild](#)¹⁷⁵.

4.2.1.10 Authentic.CurrentSelection

Declaración [CurrentSelection](#) as [AuthenticSelection](#)¹³³

Descripción

La propiedad ofrece acceso a la selección actual en la vista Authentic.

El código del ejemplo siguiente recupera el texto completo de la selección actual:

JavaScript:

```
// en alguna parte del script:  
GetSelection(objPlugIn.CurrentSelection);
```

```
// GetSelection() recopila la selección total del texto
function GetSelection(objSel)
{
    var strText = "";

    var objCurrent = objSel.Start;

    while(!objSel.End.IsSameNode(objCurrent))
    {
        objCurrent = objPlugIn.GetNextVisible(objCurrent);
        strText += objCurrent.TextValue;
    }

    strText += objSel.End.TextValue.substring(0,objSel.EndTextPosition);
    return objSel.Start.TextValue.substr(objSel.StartTextPosition) + strText;
}
```

4.2.1.11 Authentic.DesignDataLoadObject

See also

Declaración [DesignDataLoadObject](#) as [AuthenticLoadObject](#)¹⁰⁴

Descripción

DesignDataLoadObject contiene una referencia al documento SPS. El documento SPS sirve para generar el entorno de edición WYSIWYG y se genera por lo general con StyleVision.

Consulte [SchemaLoadObject](#)⁸⁷ para ver un ejemplo.

4.2.1.12 Authentic.EditClear

Declaración [EditClear](#)

Descripción

Borra la selección actual.

4.2.1.13 Authentic.EditCopy

Declaración [EditCopy](#)

Descripción

Copia la selección actual en el portapapeles.

4.2.1.14 Authentic.EditCut

Declaración [EditCut](#)

Descripción

Corta la selección actual del documento y la copia en el portapapeles.

4.2.1.15 Authentic.EditPaste

Declaración [EditPaste](#)

Descripción

Pega el contenido del portapapeles en el documento.

4.2.1.16 Authentic.EditRedo

Declaración [EditRedo](#)

Descripción

Rehace el último paso deshacer.

4.2.1.17 Authentic.EditSelectAll

Declaración [EditSelectAll](#)

Descripción

El método selecciona el documento completo.

4.2.1.18 Authentic.EditUndo

Declaración [EditUndo](#)

Descripción

Deshace la última acción.

4.2.1.19 Authentic.EnableModifications

Declaración [EnableModifications](#) as [Boolean](#)

Descripción

True si está habilitada la modificación del contenido XML en Authentic Browser. Consulte también la propiedad [Modified](#)⁸³.

Valor predeterminado: TRUE

4.2.1.20 Authentic.EntryHelperAlignment

Declaración [EntryHelperAlignment](#) as [SPYAuthenticToolbarAlignment](#)¹⁸²

Descripción

Esta propiedad se puede usar para establecer la posición de los ayudantes de entrada. El valor predeterminado es 3, que coloca los ayudantes de entrada en el lado derecho de la interfaz.

4.2.1.21 Authentic.EntryHelpersEnabled

Declaración [EntryHelpersEnabled](#) as [Boolean](#)

Descripción

True si están habilitados los ayudantes de entrada de Authentic Browser. Esta propiedad se puede usar para habilitar o deshabilitar los ayudantes de entrada.

Valor predeterminado: FALSE

4.2.1.22 Authentic.EntryHelperSize

Declaración [EntryHelperSize](#) as [Integer](#)

Descripción

Esta propiedad se puede usar para establecer el tamaño inicial del área de los ayudantes de entrada en píxels. Si quiere que se ignore el valor, asigne el valor -1 a la propiedad.

Valor predeterminado: -1

4.2.1.23 Authentic.EntryHelperWindows

Declaración [EntryHelperWindows](#) as [SPYAuthenticEntryHelperWindows](#)¹⁸¹

Descripción

Esta propiedad se puede usar para definir qué ayudantes de entrada se muestran. Los valores se pueden combinar para mostrar más de un ayudante de entrada. El valor predeterminado para mostrar los tres ayudantes de entrada es 7.

4.2.1.24 Authentic.event

See also

Declaración `event` as [AuthenticEvent](#)⁹⁷

Descripción

Esta propiedad almacena un objeto de datos de evento que contiene información sobre el evento actual.

4.2.1.25 Authentic.FindDialog

See also

Declaración `FindDialog`

Descripción

Muestra el cuadro de diálogo "Buscar".

Consulte también el apartado [Buscar y reemplazar](#)⁴⁹.

4.2.1.26 Authentic.FindNext

Declaración `FindNext`

Descripción

El método realiza la operación Buscar siguiente.

Consulte también el apartado [Buscar y reemplazar](#)⁴⁹.

4.2.1.27 Authentic.GetAllAttributes

Declaración `GetAllAttributes`(*pForElement* as [XMLData](#)¹⁶⁸, *pElements* as [Variant](#))

Descripción

`GetAllAttributes()` devuelve los atributos permitidos para el elemento indicado en forma de matriz de cadenas.

Ejemplo de JavaScript:

```
function GetAllAttributes()  
{  
    var arrElements = new Array(1);
```

```

var objStart = objPlugIn.CurrentSelection.Start;

var strText;
strText = "Valid attributes at current selection:\n\n";

for(var i = 1;i <= 4;i++)
{
    objPlugIn.GetAllAttributes(objStart, arrElements);
    strText = strText + ListArray(arrElements) + "-----\n";
}

return strText;
}

function ListArray(arrIn)
{
    var strText = "";

    if(typeof(arrIn) == "object")
    {
        for(var i = 0;i <= (arrIn.length - 1);i++)
            strText = strText + arrIn[i] + "\n";
    }

    return strText;
}

```

Ejemplo de VBScript:

```

Sub DisplayAllowedAttributes
    dim arrElements()

    dim objStart
    dim objEnd
    set objStart = objPlugIn.CurrentSelection.Start
    set objEnd = objPlugIn.CurrentSelection.End

    dim strText
    strText = "Valid attributes at current selection:" & chr(13) & chr(13)

    dim i

    For i = 1 To 4
        objView.GetAllAttributes objStart, arrElements
        strText = strText & ListArray(arrElements) & "-----" &
chr(13)
    Next

    msgbox strText
End Sub

```

```

Function ListArray(arrIn)
    dim strText

    If IsArray(arrIn) Then
        dim i

        For i = 0 To UBound(arrIn)
            strText = strText & arrIn(i) & chr(13)
        Next
    End If

    ListArray = strText
End Function

```

4.2.1.28 Authentic.GetAllowedElements

Declaración `GetAllowedElements(nAction as SPYAuthenticElementActions180, pStartElement as XMLData168, pEndElement as XMLData168, pElements as Variant)`

Descripción

`GetAllowedElements()` devuelve los elementos permitidos para las acciones especificadas por `nAction`.

Ejemplo de JavaScript:

```

function GetAllowed()
{
    var arrElements = new Array(1);

    var objStart = objPlugIn.CurrentSelection.Start;
    var objEnd = objPlugIn.CurrentSelection.End;

    var strText;
    strText = "valid elements at current selection:\n\n";

    for(var i = 0; i <= 4; i++) {
        objPlugIn.GetAllowedElements(i, objStart, objEnd, arrElements);
        strText = strText + ListArray(arrElements) + "-----\n";
    }

    return strText;
}

function ListArray(arrIn)
{
    var strText = "";

    if(typeof(arrIn) == "object") {
        for(var i = 0; i <= (arrIn.length - 1); i++)
            strText = strText + arrIn[i] + "\n";
    }
}

```

```

    }
    return strText;
}

```

Ejemplo de VBScript:

```

Sub DisplayAllowed
    dim arrElements()

    dim objStart
    dim objEnd
    set objStart = objPlugIn.CurrentSelection.Start
    set objEnd = objPlugIn.CurrentSelection.End

    dim strText
    strText = "Valid elements at current selection:" & chr(13) & chr(13)

    dim i

    For i = 1 To 4
        objView.GetAllowedElements i,objStart,objEnd,arrElements
        strText = strText & ListArray(arrElements) & "-----" &
chr(13)
    Next

    msgbox strText
End Sub

Function ListArray(arrIn)
    dim strText

    If IsArray(arrIn) Then
        dim i

        For i = 0 To UBound(arrIn)
            strText = strText & arrIn(i) & chr(13)
        Next
    End If

    ListArray = strText
End Function

```

4.2.1.29 Authentic.GetFileVersion

Declaración `GetFileVersion(strVersion as String)`

Descripción

El método simplemente devuelve la versión del componente en forma de cadena, con el formato 5.0.0.0.

4.2.1.30 Authentic.GetNextVisible

Declaración `GetNextVisible(pElement as XMLData168) as XMLData168`

Descripción

El método obtiene el siguiente elemento XML visible del documento.

4.2.1.31 Authentic.GetPreviousVisible

Declaración `GetPreviousVisible(pElement as XMLData168) as XMLData168`

Descripción

El método obtiene el elemento XML visible anterior del documento.

4.2.1.32 Authentic.IsEditClearEnabled

Declaración `IsEditClearEnabled` as `Boolean`

Descripción

True si [EditClear](#)⁷² es posible.

Consulte también el apartado [Operaciones de edición](#)⁴⁹.

4.2.1.33 Authentic.IsEditCopyEnabled

Declaración `IsEditCopyEnabled` as `Boolean`

Descripción

True si se puede copiar en el portapapeles.

Consulte también [EditCopy](#)⁷² y [Operaciones de edición](#)⁴⁹.

4.2.1.34 Authentic.IsEditCutEnabled

Declaración `IsEditCutEnabled` as `Boolean`

Descripción

True si [EditCut](#)⁷³ es posible en ese momento.

Consulte también el apartado [Operaciones de edición](#)⁴⁹.

4.2.1.35 Authentic.IsEditPasteEnabled

Declaración `IsEditPasteEnabled` as `Boolean`

Descripción

True si [EditPaste](#)⁷³ es posible.

Consulte también el apartado [Operaciones de edición](#)⁴⁹.

4.2.1.36 Authentic.IsEditRedoEnabled

Declaración `IsEditRedoEnabled` as `Boolean`

Descripción

True si [EditRedo](#)⁷³ es posible en ese momento.

Consulte también el apartado [Operaciones de edición](#)⁴⁹.

4.2.1.37 Authentic.IsEditUndoEnabled

Declaración `IsEditUndoEnabled` as `Boolean`

Descripción

True si [EditUndo](#)⁷³ es posible.

Consulte también el apartado [Operaciones de edición](#)⁴⁹.

4.2.1.38 Authentic.IsFindNextEnabled

Declaración `IsFindNextEnabled` as `Boolean`

Descripción

True si FindNext es posible en ese momento. False si no quedan más instancias.

Consulte también el apartado [Buscar y reemplazar](#)⁴⁹ y [FindDialog](#)⁷⁵.

4.2.1.39 Authentic.IsRowAppendEnabled

Declaración `IsRowAppendEnabled` as `Boolean`

Descripción

True si [RowAppend](#)⁸⁴ es posible.

Consulte también [Operaciones en filas](#)⁵⁰.

4.2.1.40 Authentic.IsRowDeleteEnabled

Declaración [IsRowDeleteEnabled](#) as **Boolean**

Descripción

True si [RowDelete](#)⁸⁴ es posible.

Consulte también [Operaciones en filas](#)⁵⁰.

4.2.1.41 Authentic.IsRowDuplicateEnabled

Declaración [IsRowDuplicateEnabled](#) as **Boolean**

Descripción

True si [RowDuplicate](#)⁸⁵ es posible en ese momento.

Consulte también [Operaciones en filas](#)⁵⁰.

4.2.1.42 Authentic.IsRowInsertEnabled

Declaración [IsRowInsertEnabled](#) as **Boolean**

Descripción

True si [RowInsert](#)⁸⁵ es posible.

Consulte también [Operaciones en filas](#)⁵⁰.

4.2.1.43 Authentic.IsRowMoveDownEnabled

Declaración [IsRowMoveDownEnabled](#) as **Boolean**

Descripción

True si [RowMoveDown](#)⁸⁵ es posible en ese momento.

Consulte también [Operaciones en filas](#)⁵⁰.

4.2.1.44 Authentic.IsRowMoveUpEnabled

Declaración `IsRowMoveUpEnabled` as `Boolean`

Descripción

True si [RowMoveUp](#)⁸⁵ es posible.

Consulte también [Operaciones en filas](#)⁵⁰.

4.2.1.45 Authentic.IsTextStateApplied

Declaración `IsTextStateApplied`(*elementName* as `String`) as `Boolean`

Descripción

Comprueba si el estado del texto ya se ha aplicado. Ejemplos del parámetro `elementName` serían `strong` e `italic`.

4.2.1.46 Authentic.IsTextStateEnabled

Declaración `IsTextStateEnabled`(*elementName* as `String`) as `Boolean`

Descripción

Comprueba si se puede aplicar un estado del texto. Ejemplos del parámetro `elementName` serían `strong` e `italic`.

4.2.1.47 Authentic.LoadXML

Declaración `LoadXML`(*xmlString* as `String`)

Descripción

Carga el documento XML actual con la cadena XML aplicada. El contenido nuevo aparece inmediatamente.

4.2.1.48 Authentic.MarkUpView

Declaración `MarkUpView`(*kind* as [SPYAuthenticMarkupVisibility](#)¹⁸¹)

Descripción

El documento se presenta por defecto usando técnicas HTML. Pero a veces es necesario ver las etiquetas de edición. Con este método puede mostrar las diferentes etiquetas de marcado.

4.2.1.49 Authentic.Modified

Declaración [Modified](#) as [Boolean](#)

Descripción

True si se modificó el contenido XML.

Esta propiedad es de solo lectura.

4.2.1.50 Authentic.Print

Declaración [Print](#)

Descripción

Imprimir el documento que se está editando.

4.2.1.51 Authentic.PrintPreview

Declaración [PrintPreview](#)

Descripción

Vista previa de impresión del documento que se está editando.

4.2.1.52 Authentic.RedrawEntryHelpers

Declaración [RedrawEntryHelpers\(\)](#)

Descripción

[RedrawEntryHelpers](#) toma los valores de las propiedades [EntryHelpersEnabled](#)⁷⁴, [EntryHelperAlignment](#)⁷⁴, [EntryHelperSize](#)⁷⁴ y [EntryHelperWindows](#)⁷⁴ y vuelve a dibujar las ventanas de los ayudantes de entrada.

4.2.1.53 Authentic.ReloadToolbars

Declaración [ReloadToolbars\(\)](#)

Descripción

[ReloadToolbars](#) lee la colección [ToolbarRows](#)⁸⁹ y vuelve a dibujar las barras de herramientas y la vista.

4.2.1.54 Authentic.ReplaceDialog

Declaración [ReplaceDialog](#)

Descripción

Muestra el cuadro de diálogo "Reemplazar".

Consulte también [Buscar y reemplazar](#)⁴⁹.

4.2.1.55 Authentic.Reset

Obsoleto

Use [Authentic.StartEditing](#)⁸⁸ en su lugar.

Declaración [Reset](#)

Descripción

Restaurar los datos que se están editando. Se le suele llamar antes de editar un conjunto nuevo de documentos XML, XSL y SPS.

El método no cambia la vista y todavía se puede seguir trabajando con el documento que está activo en la vista.

4.2.1.56 Authentic.RowAppend

Declaración [RowAppend](#)

Descripción

Anexa una fila en la posición actual.

Consulte también [Operaciones en filas](#)⁵⁰.

4.2.1.57 Authentic.RowDelete

Declaración [RowDelete](#)

Descripción

Elimina las filas seleccionadas.

Consulte también [Operaciones en filas](#)⁵⁰.

4.2.1.58 Authentic.RowDuplicate

Declaración [RowDuplicate](#)

Descripción

El método duplica las filas seleccionadas.

Consulte también [Operaciones en filas](#) ⁵⁰.

4.2.1.59 Authentic.RowInsert

Declaración [RowInsert](#)

Descripción

Inserta una fila nueva justo encima de la selección.

Consulte también [Operaciones en filas](#) ⁵⁰.

4.2.1.60 Authentic.RowMoveDown

Declaración [RowMoveDown](#)

Descripción

Baja la fila actual una posición.

Consulte también [Operaciones en filas](#) ⁵⁰.

4.2.1.61 Authentic.RowMoveUp

Declaración [RowMoveUp](#)

Descripción

Sube la fila actual una posición.

Consulte también [Operaciones en filas](#) ⁵⁰.

4.2.1.62 Authentic.Save

Declaración [Save](#)

Descripción

Guarda el documento en la URL indicada por la propiedad [XMLDataSaveUrl](#)⁹¹. En la versión no segura (*Untrusted*) también puede usar una ruta de acceso local completa.

El complemento envía una solicitud HTTP PUT al servidor para guardar el archivo XML activo en la vista.

4.2.1.63 Authentic.SaveButtonAutoEnable

Declaración [SaveButtonAutoEnable](#) as **Boolean**

Descripción

Si esta propiedad tiene el valor TRUE, el estado habilitado/deshabilitado del botón **Guardar** de la barra de herramientas del control se establece en función de la marca [Modified](#)⁸³ del documento.

Valor predeterminado: FALSE.

4.2.1.64 Authentic.SavePOST

Declaración [SavePOST](#)

Descripción

Guarda el documento en la URL indicada por la propiedad [XMLDataSaveUrl](#)⁹¹. En la versión no segura (*Untrusted*) también puede usar una ruta de acceso local completa. El complemento envía una solicitud HTTP POST al servidor para guardar el archivo XML activo en la vista.

Comprobar si el archivo se guardó o no

Si el complemento Authentic Browser recibe una respuesta HTTP de ≥ 300 , entenderá que el archivo **no** se guardó y mostrará por defecto un mensaje con la respuesta de error HTTP y después otro mensaje informando al usuario de que el archivo no se guardó. El desarrollador de la aplicación debe asegurarse de que se devuelva la respuesta HTTP correcta. Por ejemplo puede usar este código PHP:

```
<?php
// suprimir mensajes de error para evitar
// que se generen resultados para poder enviar encabezados
error_reporting (0);
$error = false;
$handle = fopen ( "result.xml", "w+" );
if (! $handle)
    $error = true;
else
{
    if (! fwrite($handle, $HTTP_RAW_POST_DATA))
        $error = true;
    else
        fclose($handle);
}
if ($error)
    header( "HTTP/1.1 500 Server Error" );
```

?>

4.2.1.65 Authentic.SaveXML

Declaración [SaveXML](#) as [String](#)

Valor devuelto

Estructura XML como cadena

Descripción

Guarda los datos XML actuales en una cadena de texto que se devuelve al autor de la llamada.

4.2.1.66 Authentic.SchemaLoadObject

Declaración [SchemaLoadObject](#) as [AuthenticLoadObject](#) ¹⁰⁴

Descripción

SchemaLoadObject contiene una referencia al documento de esquema XML para el archivo XML actual. El documento de esquema se suele generar con Altova XMLSpy.

Ejemplo

```
objPlugIn.SchemaLoadObject.URL = "http://www.suservidor.com/OrgChart.xsd"
objPlugIn.XMLDataLoadObject.URL = "http://www.suservidor.com/OrgChart.xml"
objPlugIn.DesignDataLoadObject.URL = "http://www.suservidor.com/OrgChart.sps"
objPlugIn.StartEditing
```

El código del ejemplo anterior establece todas las propiedades URL de los objetos y llama [StartEditing](#) ⁸⁸ para cargar y mostrar los archivos. En la versión no segura (*Untrusted*) también puede usar una ruta de acceso local completa. El contenido actual y el estado del complemento se borrará.

4.2.1.67 Authentic.SelectionChanged

Declaración [SelectionChanged](#) as [VT_0019](#)

Descripción

Este evento se genera cada vez que el usuario cambia la selección actual.

Consulte también [Eventos del punto de conexión](#) ⁴⁶.

4.2.1.68 Authentic.SelectionMoveTabOrder

Declaración `SelectionMoveTabOrder(bForward as Boolean, bTag as Boolean)`

Descripción

`SelectionMoveTabOrder()` mueve hacia delante o hacia detrás la selección actual.

Si `bTag` es `false` y la selección actual está en la última celda de una tabla, entonces se añadirá una línea nueva.

4.2.1.69 Authentic.SelectionSet

Declaración `SelectionSet(pStartElement as XMLData168, nStartPos as long, pEndElement as XMLData168, nEndPos as long) as Boolean`

Descripción

Use `SelectionSet()` para establecer una selección nueva en la vista `Authentic`. Es posible dar a `pEndElement` el valor `null` (nada) si la selección debe estar justo encima de un elemento XML (`StartElement`).

4.2.1.70 Authentic.SetUnmodified

Declaración `SetUnmodified()`

Descripción

Tras llamar a este método la condición actual del búfer de las operaciones `Deshacer/Rehacer` se toma como estado limpio del documento XML subyacente y la marca `Modified`⁸³ toma el valor `FALSE`.

4.2.1.71 Authentic.StartEditing

Declaración `StartEditing as Boolean`

Valor devuelto

`True` si todos los archivos se cargaron y presentaron en la vista correctamente.

Descripción

Empezar a editar el documento actual. Es importante configurar primero las propiedades de los objetos `SchemaLoadObject`⁸⁷, `DesignDataLoadObject`⁷² y `XMLDataLoadObject`⁹¹.

4.2.1.72 Authentic.StartSpellChecking

Declaración [StartSpellChecking\(\)](#)

Descripción

Este comando abre el cuadro de diálogo de corrección ortográfica si hay un paquete con un motor de corrección ortográfica y con los diccionarios necesarios y si ese paquete está activado.

4.2.1.73 Authentic.TextStateBmpURL

Declaración [TextStateBmpURL](#) as [String](#)

Descripción

La URL de la que se deben recuperar los mapas de bits para los iconos de estados del texto. Si no se indica ninguna URL, no habrá botones de estado del texto.

Ejemplos

```
objPlugin.TextStateBmpURL = "<http://plugin.xmlspy.com/textstates/>"
```

```
<PARAM NAME="TextStateBmpURL" VALUE="http://plugin.xmlspy.com/textstates/">
```

4.2.1.74 Authentic.TextStateToolBarLine

Declaración [TextStateToolBarLine](#) as [long](#)

Descripción

La barra de herramientas (número de línea) en la que se deben colocar los iconos de los estados del texto. Estos iconos se pueden anexar a las barras de herramientas actuales o en barras de herramientas nuevas.

Valor predeterminado: 1

4.2.1.75 Authentic.ToolbarRows

Declaración [ToolbarRows](#) as [AuthenticToolbarRows](#)¹³⁸

Descripción

Obtiene una colección de todas las barras de herramientas que se deben mostrar. Consulte la descripción de [AuthenticToolbarRows](#)¹³⁸, que explica cómo eliminar, añadir o modificar las barras de herramientas.

4.2.1.76 Authentic.ToolbarsEnabled

Declaración `ToolbarsEnabled` as `Boolean`

Descripción

True si las barras de herramientas de Authentic Browser están habilitadas.

Esta propiedad se puede usar para habilitar o deshabilitar todas las barras de herramientas.

Valor predeterminado: true

4.2.1.77 Authentic.ToolbarTooltipsEnabled

Declaración `ToolbarTooltipsEnabled` as `Boolean`

Descripción

True si está habilitada la información rápida de las barras de herramientas de Authentic Browser.

Valor predeterminado: true

4.2.1.78 Authentic.UICommands

Declaración `UICommands` as `AuthenticCommands` ⁹³

Descripción

Obtiene una colección de todos los comandos de las barras de herramientas disponibles (y su descripción). En las barras de herramientas se pueden colocar comandos para todos estos botones. Solo lectura.

Ejemplo

Obtener todos los comandos, grupos de comandos y descripciones disponibles y presentarlos en un cuadro de mensaje.

```
dim str
for each UICommand in objPlugin.UICommands
str = str & UICommand.CommandID & " | " & UICommand.Group & " | " &
UICommand.ShortDescripción & chr(13)
next
msgbox str
```

4.2.1.79 Authentic.ValidateDocument

Declaración `ValidateDocument(showResults` as `Boolean`) as `Boolean`

Valor devuelto

Resultado de la validación

Descripción

Valida los datos XML actuales con los datos del esquema XML. Si el parámetro `showResults` es `FALSE`, entonces se suprimirán los errores de validación. De lo contrario se presentan los errores de validación.

4.2.1.80 `Authentic.validationBadData`

Declaración `validationBadData` as [XMLData](#)¹⁶⁸

Descripción

Esta propiedad puede aportar información adicional sobre el último error de validación. Se configura después de la llamada a `ValidateDocument()` y su valor puede ser `null` o una referencia al elemento XML que provocó el error.

4.2.1.81 `Authentic.validationMessage`

Declaración `validationMessage` as `String`

Descripción

Si falla la validación (después de la llamada a `ValidateDocument`), esta propiedad almacena una cadena de texto con el mensaje de error.

4.2.1.82 `Authentic.XMLDataLoadObject`

Declaración `XMLDataLoadObject` as [AuthenticLoadObject](#)¹⁰⁴

Descripción

`XMLDataLoadObject` contiene una referencia al documento XML que se está editando. El documento XML se suele definir con `Altova XMLSpy`, pero se suele generar con ayuda de una base de datos o cualquier otro proceso de negocio.

Consulte también [SchemaLoadObject](#)⁸⁷ para ver un ejemplo.

4.2.1.83 `Authentic.XMLDataSaveUrl`

Declaración `XMLDataSaveUrl` as `String`

Descripción

Cuando los datos XML han sido modificados, se pueden volver a guardar en el servidor con una URL. Cuando se guardan a través de una solicitud `HTTP PUT` con ayuda del método `Authentic.Save`, esta propiedad define la ubicación donde se guardarán los datos XML. Cuando se envían a través de una solicitud `HTTP POST` con

ayuda del método `Authentic.SavePOST`, esta propiedad defina la ubicación de una aplicación o de un script del lado servidor que procesará los datos de `POST`. En la versión no segura (*Untrusted*) también puede usar una ruta de acceso local completa.

Consulte también los métodos [Authentic.Save](#)⁸⁵ y [Authentic.SavePOST](#)⁸⁶.

4.2.1.84 Authentic.XMLRoot

Declaración `XMLRoot` as [XMLData](#)¹⁶⁸

Descripción

`XMLRoot` es el elemento primario de la estructura XML activa en la vista. La interfaz [XMLData](#)¹⁶⁸ otorga acceso a todo el contenido del archivo.

Consulte también [Using XMLData](#)⁵⁷.

4.2.1.85 Authentic.XMLTable

Declaración `XMLTable` as [AuthenticXMLTableCommands](#)¹⁶⁰

Descripción

Obtiene el conjunto formado por todos los comandos de tabla XML. Solo lectura.

4.2.2 AuthenticCommand

Métodos

Propiedades

[CommandID](#)⁹²

[Group](#)⁹³

[ShortDescripción](#)⁹³

[Name](#)⁹³

4.2.2.1 AuthenticCommand.CommandID

Declaración `CommandID` as [SPYAuthenticCommand](#)¹⁷⁸

Descripción

Obtiene el `CommandId` del comando.

Valores posibles: véase `AuthenticToolBarButton`

Solo lectura

Ejemplo

Consulte el ejemplo de [Authentic.UICommands](#) ⁹⁰

4.2.2.2 AuthenticCommand.Group

Declaración `Group` as [SPYAuthenticCommandGroup](#) ¹⁸⁰

Descripción

El grupo de comandos al que pertenece el comando.
Solo lectura

Ejemplo

Consulte el ejemplo de [Authentic.UICommands](#) ⁹⁰

4.2.2.3 AuthenticCommand.ShortDescription

Declaración `ShortDescripción` as `String`

Descripción

Breve descripción del comando (p. ej. el texto de información rápida)
Solo lectura

Ejemplo

Consulte el ejemplo de [Authentic.UICommands](#) ⁹⁰

4.2.2.4 AuthenticCommand.Name

Declaración `Name` as `String`

Descripción

Para su uso posterior

4.2.3 AuthenticCommands

Métodos

[Item](#) ⁹⁴

Propiedades

[Count](#) ⁹⁴

4.2.3.1 AuthenticCommands.Count

Declaración `Count` as `long`

Descripción

Número de comandos disponibles en la interfaz del usuario
Solo lectura

4.2.3.2 AuthenticCommands.Item

Declaración `Item` (`nPosition` as `long`) as `AuthenticCommand` ⁹²

Descripción

Obtiene el comando de la posición `nPosition`. `nPosition` empieza por el número 1.

4.2.4 AuthenticContextMenu

La interfaz `ContextMenu` permite al usuario personalizar los menús contextuales que aparecen en `Authentic`. La interfaz tiene los métodos que se enumeran en esta sección.

4.2.4.1 CountItems

Método: `CountItems`()

Valor devuelto

Devuelve el número de elementos del menú.

Errores

2501 Objeto no válido.

4.2.4.2 DeleteItem

Método: `DeleteItem`(`position` as `integer`)

Valor devuelto

Elimina un elemento de menú.

Errores

2501 Objeto no válido
2502 Índice no válido

4.2.4.3 GetItemText

Método: `GetItemText`(`position as integer`) nombre del elemento de menú como cadena de texto

Valor devuelto

Obtiene el nombre del elemento del menú.

Errores

- 2501 Objeto no válido
- 2502 Índice no válido

4.2.4.4 InsertItem

Método: `InsertItem`(`position as integer`, `menu item name as string`, `macro name as string`)

Valor devuelto

Inserta un elemento de menú definido por el usuario. El elemento de menú iniciará una macro, por lo que es necesario suministrar un nombre de macro válido.

Errores

- 2501 Objeto no válido
- 2502 Índice no válido
- 2503 La macro no existe
- 2504 Error interno

4.2.4.5 SetItemText

Método: `SetItemText`(`position as integer`, `menu item name as string`)

Valor devuelto

Establece el nombre del elemento de menú.

Errores

- 2501 Objeto no válido
- 2502 Índice no válido

4.2.5 AuthenticDataTransfer

Métodos

[getData](#)⁹⁶

Propiedades

[dropEffect](#)⁹⁶

[ownDrag](#)⁹⁶

[type](#) ⁹⁶

Descripción

Interfaz AuthenticDataTransfer.

4.2.5.1 AuthenticDataTransfer.dropEffect

Declaración `dropEffect` as `long`

Descripción

La propiedad almacena el efecto de la operación colocar del controlador de evento predeterminado. Este efecto de la operación de colocar se puede configurar cambiando este valor y asignando el valor TRUE a

[AuthenticEvent.cancelBubble](#) ⁹⁸.

4.2.5.2 AuthenticDataTransfer.getData

Declaración `getData` as `Variant`

Descripción

`getData` obtiene los datos reales asociados con este objeto `dataTransfer`. Consulte también

[AuthenticDataTransfer.type](#) ⁹⁶.

4.2.5.3 AuthenticDataTransfer.ownDrag

Declaración `ownDrag` as `Boolean`

Descripción

La propiedad tiene el valor TRUE si el origen de la operación de arrastrar actual viene de la vista Authentic.

4.2.5.4 AuthenticDataTransfer.type

Declaración `type` as `String`

Descripción

Almacena el tipo de datos que se obtienen con el método [AuthenticDataTransfer.getData](#) ⁹⁶.

Los tipos de datos compatibles actualmente son:

OWN	datos del propio complemento
TEXT	texto sin formato
UNICODETEXT	texto sin formato como UNICODE
IUNKNOWN	referencia IUnknown a la instancia IDataObject

4.2.6 AuthenticEvent

Propiedades

[altKey](#)⁹⁷

[altLeft](#)⁹⁷

[ctrlKey](#)⁹⁸

[ctrlLeft](#)⁹⁹

[shiftKey](#)¹⁰⁰

[shiftLeft](#)¹⁰⁰

[keyCode](#)⁹⁹

[repeat](#)¹⁰⁰

[button](#)⁹⁷

[clientX](#)⁹⁸

[clientY](#)⁹⁸

[dataTransfer](#)⁹⁹

[srcElement](#)¹⁰⁰

[fromElement](#)⁹⁹

[propertyName](#)⁹⁹

[cancelBubble](#)⁹⁸

[returnValue](#)¹⁰⁰

[type](#)¹⁰¹

Descripción

Interfaz AuthenticEvent.

4.2.6.1 AuthenticEvent.altKey

Declaración [altKey](#) as **Boolean**

Descripción

True si se pulsa la tecla **Alt** derecha.

4.2.6.2 AuthenticEvent.altLeft

Declaración [altLeft](#) as **Boolean**

Descripción

True si se pulsa la tecla **Alt** izquierda.

4.2.6.3 AuthenticEvent.button

Declaración [button](#) as **long**

Descripción

Especifica qué botón del ratón se pulsó:

- 0 Ninguno
- 1 El botón izquierdo
- 2 El botón derecho
- 3 Se pulsó el botón izquierdo y el botón derecho a la vez
- 4 El botón del medio
- 5 Se pulsó el botón izquierdo y el botón del medio a la vez
- 6 Se pulsó el botón derecho y el botón del medio a la vez
- 7 Todos los botones a la vez

Los eventos `onbuttondown` y `onbuttonup` configuran el valor del botón. El evento `onbuttonup` configura el valor del botón que se soltó y genera el evento `up` independientemente de qué otros botones estén pulsados en ese momento.

4.2.6.4 `AuthenticEvent.cancelBubble`

Declaración `cancelBubble` as `Boolean`

Descripción

Define el valor `TRUE` para `cancelBubble` si no se debe llamar al controlador de eventos predeterminado.

4.2.6.5 `AuthenticEvent.clientX`

Declaración `clientX` as `long`

Descripción

Valor X de la posición actual del ratón en las coordenadas del cliente.

4.2.6.6 `AuthenticEvent.clientY`

Declaración `clientY` as `long`

Descripción

El valor Y de la posición actual del ratón en las coordenadas del cliente.

4.2.6.7 `AuthenticEvent.ctrlKey`

Declaración `ctrlKey` as `Boolean`

Descripción

True si se pulsa la tecla **Ctrl** derecha.

4.2.6.8 AuthenticEvent.ctrlLeft

Declaración ctrlLeft as Boolean

Descripción

True si se pulsa la tecla **Ctrl** izquierda.

4.2.6.9 AuthenticEvent.dataTransfer

Declaración dataTransfer as Variant

Descripción

propiedad dataTransfer

4.2.6.10 AuthenticEvent.fromElement

Declaración fromElement as Variant

Descripción

Ningún evento establece esta propiedad

4.2.6.11 AuthenticEvent.keyCode

Declaración keyCode as long

Descripción

Código clave de la tecla que está pulsada.

La propiedad es de lectura-escritura.

4.2.6.12 AuthenticEvent.propertyName

Declaración propertyName as String

Descripción

Ahora mismo no hay ningún evento que establezca esta propiedad.

4.2.6.13 AuthenticEvent.repeat

Declaración `repeat` as `Boolean`

Descripción

True si se repite el evento onkeydown.

4.2.6.14 AuthenticEvent.returnValue

Declaración `returnValue` as `Variant`

Descripción

Use `returnValue` para definir un valor devuelto para su controlador de eventos.

4.2.6.15 AuthenticEvent.shiftKey

Declaración `shiftKey` as `Boolean`

Descripción

True si se pulsa la tecla **Mayús** derecha.

4.2.6.16 AuthenticEvent.shiftLeft

Declaración `shiftLeft` as `Boolean`

Descripción

True si se pulsa la tecla **Mayús** izquierda.

4.2.6.17 AuthenticEvent.srcElement

Declaración `srcElement` as `Variant`

Descripción

Elemento que desencadena el evento actual.

Suele ser un objeto [XMLData](#)¹⁶⁸.

Desde la versión 3.0.0.0: la propiedad también puede almacenar una referencia a un objeto [AuthenticCommand](#)⁹² si se estableció desde un evento `ondoceditcommand`.

4.2.6.18 AuthenticEvent.type

Declaración `type` as `String`

Descripción

Ahora mismo no hay ningún evento que establezca esta propiedad.

4.2.7 AuthenticEventContext

La interfaz `EventContext` ofrece acceso a muchas propiedades del contexto donde se ejecuta una macro.

4.2.7.1 EvaluateXPath

Método: `EvaluateXPath` (`string` expression) as `string`

Valor devuelto

El método evalúa la expresión XPath en el contexto del nodo en el que se desencadenó el evento y devuelve una cadena de texto.

Descripción

`EvaluateXPath()` ejecuta una expresión Xpath con el contexto del evento dado. El resultado se devuelve como cadena de texto y, si se trata de una secuencia, entonces es una cadena de texto separada por espacios.

Errores

- 2201 Objeto no válido.
- 2202 No hay contexto.
- 2209 Parámetro no válido.
- 2210 Error interno.
- 2211 Error de XPath.

4.2.7.2 GetEventContextType

Método: `GetEventContextType` () as `EventContextType` enumeration

Valor devuelto

Devuelve el tipo de nodo de contexto.

Descripción

`GetEventContextType` permite al usuario determinar si la macro está en un nodo XML o en un contexto de un elemento atómico XPath. La enumeración `AuthenticEventContextType` se define de la siguiente manera:

```
authenticEventContextXML,  
authenticEventContextAtomicItem,  
authenticEventContextOther
```

Si el contexto es un nodo XML normal, la función `GetXMLNode()` da acceso (de lo contrario devuelve `NULL`).

Errores

- 2201 Objeto no válido.
- 2202 No hay contexto.
- 2209 Parámetro no válido.

4.2.7.3 GetNormalizedTextValue

Método: `GetNormalizedTextValue()` value as string

Valor devuelto

Devuelve el valor del nodo actual en forma de cadena.

Errores

- 2201 Objeto no válido.
- 2202 No hay contexto.
- 2203 Contexto no válido
- 2209 Parámetro no válido.

4.2.7.4 GetVariableValue

Método: `GetVariableValue`(nombre como cadena, valor como cadena)

Valor devuelto

Obtiene el valor de la variable nombrada.

Descripción

`GetVariableValue` obtiene el valor de la variable en el ámbito del contexto.

```
nZoom = parseInt( AuthenticView.EventContext.GetVariableValue( 'Zoom' ) );
if ( nZoom > 1 )
{
    AuthenticView.EventContext.SetVariableValue( 'Zoom', nZoom - 1 );
}
```

Errores

- 2201 Objeto no válido.
- 2202 No hay contexto.
- 2204 Variable fuera de intervalo
- 2205 La variable no se puede evaluar
- 2206 La variable devuelve una secuencia
- 2209 Parámetro no válido

4.2.7.5 GetXMLNode

Método: `GetXMLNode()` [XMLData](#)¹⁶⁸ object

Valor devuelto

Devuelve el nodo XML de contexto o NULL

Errores

- 2201 Objeto no válido.
- 2202 No hay contexto.
- 2203 Contexto no válido
- 2209 Parámetro no válido.

4.2.7.6 IsAvailable

Método: `IsAvailable()`

Valor devuelto

Devuelve true si se estableció `EventContext`.

Errores

- 2201 Objeto no válido.

4.2.7.7 SetVariableValue

Método: `SetVariableValue`(nombre como cadena, valor como cadena)

Valor devuelto

Establece el valor de la variable nombrada.

Descripción

`SetVariableValue` establece el valor de la variable en el ámbito del contexto.

```
nZoom = parseInt( AuthenticView.EventContext.GetVariableValue( 'Zoom' ) );
if ( nZoom > 1 )
{
    AuthenticView.EventContext.SetVariableValue( 'Zoom', nZoom - 1 );
}
```

Errores

- 2201 Objeto no válido.
- 2202 No hay contexto.
- 2204 Variable fuera de intervalo
- 2205 La variable no se puede evaluar
- 2206 La variable devuelve una secuencia
- 2207 Variable de solo lectura

2208 No se permiten cambios

4.2.8 AuthenticLoadObject

Propiedades

[String](#)¹⁰⁴

[URL](#)¹⁰⁴

Descripción

El objeto XMLSpyXMLLoadSave sirve para establecer el origen de los archivos que se deben cargar. Puede establecer el contenido a través de la propiedad String directamente o como ubicación externa a través de la propiedad URL.

Para más información consulte [Authentic.SchemaLoadObject](#)⁸⁷, [Authentic.DesignDataLoadObject](#)⁷² y [Authentic.XMLDataLoadObject](#)⁹¹.

4.2.8.1 AuthenticLoadObject.String

Declaración `String` as `String`

Descripción

Puede usar esta propiedad para establecer la estructura XML a partir de una cadena. La propiedad URL del objeto debe estar vacía si quiere usar esta propiedad.

4.2.8.2 AuthenticLoadObject.URL

Declaración `URL` as `String`

Descripción

La propiedad debería contener una URL válida para la operación Cargar o Guardar. Los protocolos HTTP compatibles son http, https, ftp y gopher.

4.2.9 AuthenticRange

La primera tabla que aparece a continuación enumera las propiedades y los métodos de AuthenticRange que se pueden usar para navegar por el documento y seleccionar partes concretas.

Propiedades	Métodos	
Application ¹⁰⁷	Clone ¹⁰⁸	MoveBegin ¹²⁶
FirstTextPosition ¹¹¹	CollapsToBegin ¹⁰⁸	MoveEnd ¹²⁶
FirstXMLData ¹¹²	CollapsToEnd ¹⁰⁸	NextCursorPosition ¹¹⁷

FirstXMLDataOffset ¹¹³	ExpandTo ¹¹¹	PreviousCursorPosition ¹¹⁹
LastTextPosition ¹²³	Goto ¹¹⁶	Select ¹²⁹
LastXMLData ¹²⁴	GotoNext ¹¹⁷	SelectNext ¹²⁹
LastXMLDataOffset ¹²⁵	GotoPrevious ¹¹⁸	SelectPrevious ¹³⁰
Parent ¹²⁷	IsEmpty ¹²¹	SetFromRange ¹³²
	IsEqual ¹²¹	

Esta tabla enumera los métodos de modificación de contenido. La mayoría puede encontrarse en el menú que aparece al hacer clic con el botón derecho del ratón.

Propiedades	Operaciones de edición	Operaciones de tablas dinámicas
Text ¹³³	Copy ¹⁰⁹	AppendRow ¹⁰⁶
	Cut ¹⁰⁹	DeleteRow ¹¹⁰
	Delete ¹⁰⁹	DuplicateRow ¹¹⁰
	Paste ¹²⁸	InsertRow ¹²⁰
		IsInDynamicTable ¹²²
		MoveRowDown ¹²⁷
		MoveRowUp ¹²⁷

Estos métodos ofrecen las funciones de los ayudantes de entrada de Authentic para los objetos de intervalo.

Operaciones de los ayudantes de entrada		
Elementos	Atributos	Entidades
CanPerformActionWith ¹⁰⁷	GetElementAttributeValue ¹¹⁵	GetEntityNames ¹¹⁶
CanPerformAction ¹⁰⁷	GetElementAttributeNames ¹¹⁴	InsertEntity ¹¹⁹
PerformAction ¹²⁸	GetElementHierarchy ¹¹⁵	
	HasElementAttribute ¹¹⁹	
	SetElementAttributeValue ¹³¹	

Descripción

Los objetos de `AuthenticRange` son las selecciones realizadas con el cursor de la interfaz de automatización. Puede usarlos para apuntar a cualquier selección realizada con el cursor en la vista Authentic o para seleccionar parte del documento. Después las operaciones disponibles para los objetos de `AuthenticRange` trabajan en la selección de la misma manera que las operaciones correspondientes de la interfaz del usuario en la selección actual de la interfaz del usuario. La principal diferencia es que puede usar un número aleatorio de objetos de `AuthenticRange` al mismo tiempo, mientras que en la interfaz del usuario hay una sola selección realizada con el cursor.

Para obtener un objeto de intervalo inicial utilice `AuthenticView.Selection` ¹⁵⁷, con lo que obtendrá el intervalo que se corresponde con la selección realizada con el cursor en la interfaz del usuario. Otra opción es acceder a intervalos de segundo orden con las propiedades de solo lectura `AuthenticView.DocumentBegin` ¹⁵³,

[AuthenticView.DocumentEnd](#)¹⁵³ y [AuthenticView.WholeDocument](#)¹⁵⁹. El método más flexible es [AuthenticView.Goto](#)¹⁵⁵, que permite navegar hasta una parte concreta del documento dentro de una llamada. Para realizar selecciones más complejas, puede combinar las propiedades con los métodos de navegación en los objetos de intervalo que aparecen en la primera tabla de este apartado.

Otro método para seleccionar una porción del documento consiste en usar las propiedades de posición del objeto de intervalo. Existen dos sistemas de posicionamiento y se pueden combinar de forma aleatoria:

- **Posiciones absolutas del cursor de texto:** empezando por la posición 0 al principio del documento. Se puede establecer y recuperar para el principio y el final del intervalo. Para más información consulte [FirstTextPosition](#)¹¹¹ y [LastTextPosition](#)¹²³. Este método requiere cálculos internos complejos y debería utilizarse con cuidado.
- El elemento **XMLData** y una posición de texto dentro de este elemento: se puede establecer y recuperar para el principio y el final del intervalo. Para más información consulte [FirstXMLData](#)¹¹², [FirstXMLDataOffset](#)¹¹³, [LastXMLData](#)¹²⁴ y [LastXMLDataOffset](#)¹²⁵. Este método es muy eficaz pero requiere conocimientos previos sobre la estructura subyacente del documento. Se puede utilizar para ubicar objetos XMLData y realizar operaciones en ellos que no están disponibles desde la interfaz del usuario.

Se pueden realizar modificaciones en el contenido del documento de varias maneras:

- La propiedad [Text](#)¹³³ permite recuperar el texto del documento seleccionado por el objeto de intervalo. Si se le asigna un valor, el texto seleccionado del documento se reemplaza con el texto nuevo.
- Las funciones estándar de edición [Cortar](#)¹⁰⁹, [Copiar](#)¹⁰⁹, [Pegar](#)¹²⁶ y [Eliminar](#)¹⁰⁹.
- Operaciones de tabla para tabla que pueden crecer de forma dinámica.
- Métodos que crean asignaciones entre las funciones de los ayudantes de entrada de Authentic.
- Acceso a los objetos [XMLData](#) del documento subyacente para modificarlos directamente.

4.2.9.1 AuthenticRange.AppendRow

Método: [AppendRow](#) () as [Boolean](#)

Descripción

Si el principio del intervalo está dentro de una tabla dinámica, este método inserta una fila nueva al final de la tabla seleccionada. La selección del intervalo se modifica para apuntar al principio de la nueva fila. La función devuelve *true* si la operación de anexión finaliza correctamente. De lo contrario devuelve *false*.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

Ejemplos

```

'-----
'                               VBScript
' Anexar fila al final de la tabla dinámica actual
'-----

Dim objRange
Set objRange = objPlugin.ActiveDocument.AuthenticView.Selection

' comprobar si se puede insertar algo

```

```
If objRange.IsInDynamicTable Then
    objRange.AppendRow
    ' objRange apunta al principio de la nueva fila
    objRange.Select
End If
```

4.2.9.2 AuthenticRange.Application

Propiedad: `Application` as [Authentic](#)⁶⁶ (read-only)

Descripción

Acceder al objeto de aplicación de .

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.3 AuthenticRange.CanPerformAction

Método: `CanPerformAction` (`eAction` as `SPYAuthenticActions`, `strElementName` as `String`) as `Boolean`

Descripción

`CanPerformAction` y sus métodos permiten el acceso a las funciones de los ayudantes de entrada de `Authentic`. Esta función permite modificar el contenido del documento de forma sencilla y coherente sin necesidad de saber dónde tienen lugar los cambios exactamente. El principio del objeto de intervalo se usa para encontrar la siguiente ubicación válida donde se puede llevar a cabo la acción indicada. Si se encuentra la ubicación, el método devuelve `True`. De lo contrario devuelve `False`.

CONSEJO: utilice [CanPerformActionWith](#)¹⁰⁷ para encontrar todos los nombres de elemento válidos para una acción determinada.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.
- 2007 La acción indicada no es válida.

Ejemplos

Consulte también [PerformAction](#)¹²⁸.

4.2.9.4 AuthenticRange.CanPerformActionWith

Método: `CanPerformActionWith` (`eAction` as `SPYAuthenticActions`, `out_arrElementNames` as `Variant`)

Descripción

CanPerformActionWith y sus métodos permiten acceder a las funciones de los ayudantes de entrada de Authentic. Esta función permite modificar el contenido del documento de forma sencilla y coherente sin necesidad de saber dónde tienen lugar los cambios exactamente.

Este método devuelve una serie de nombres de elementos con los que se puede llevar a cabo la acción especificada.

CONSEJO: utilice [PerformAction](#)¹²⁸ para aplicar la acción.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.
- 2007 La acción indicada no es válida.

Ejemplos

Consulte también [PerformAction](#)¹²⁸.

4.2.9.5 AuthenticRange.Clone

Método: [Clone](#) () as [AuthenticRange](#)¹⁰⁴

Descripción

Devuelve una copia del objeto de intervalo.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.6 AuthenticRange.CollapsToBegin

Método: [CollapsToBegin](#) () as [AuthenticRange](#)¹⁰⁴

Descripción

Establece el final del objeto de intervalo en su inicio. El método devuelve el objeto de intervalo modificado.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.7 AuthenticRange.CollapsToEnd

Método: [CollapsToEnd](#) () as [AuthenticRange](#)¹⁰⁴

Descripción

Establece el principio del objeto de intervalo en su final. El método devuelve el objeto de intervalo modificado.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.8 AuthenticRange.Copy

Método: `Copy ()` as `Boolean`

Descripción

Devuelve `False` si el intervalo no contiene porciones del documento que no se puedan copiar. Devuelve `True` si se copió texto (y si hay elementos XML seleccionados, elementos también) en el búfer de copia/pega.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.9 AuthenticRange.Cut

Método: `Cut ()` as `Boolean`

Descripción

Devuelve `False` si el intervalo no contiene porciones del documento que no se pueden eliminar. Devuelve `True` después de que se elimine texto (y si hay elementos XML seleccionados, elementos también) y se guarde en el búfer de copia/pega.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.10 AuthenticRange.Delete

Método: `Delete ()` as `Boolean`

Descripción

Devuelve `False` si el intervalo no contiene porciones del documento que no se pueden eliminar. Devuelve `True` después de que se elimine texto (y si hay elementos XML seleccionados, elementos también) del documento.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.11 AuthenticRange.DeleteRow

Método: `DeleteRow ()` as **Boolean**

Descripción

Si el principio del intervalo está dentro de una tabla dinámica, este método elimina la fila seleccionada. La selección del intervalo se modifica para apuntar al siguiente elemento situado después de la fila eliminada. La función devuelve *true* si la operación de eliminar se lleva a cabo correctamente. De lo contrario devuelve *false*.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

Ejemplos

```

'-----
'                               VBScript
' Eliminar fila seleccionada de la tabla dinámica
'-----

Dim objRange
Set objRange = objPlugin.ActiveDocument.AuthenticView.Selection

' comprobar si estamos en una tabla
If objRange.IsInDynamicTable Then
    objRange.DeleteRow
End If

```

4.2.9.12 AuthenticRange.DuplicateRow

Método: `DuplicateRow ()` as **Boolean**

Descripción

Si el principio del intervalo está dentro de una tabla dinámica, este método inserta una copia de la fila actual después de la fila seleccionada. La selección del intervalo se modifica para apuntar al principio de la nueva fila. La función devuelve *true* si la operación se completa correctamente. De lo contrario devuelve *false*.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

Ejemplos

```

'-----
'                               VBScript
' crear copia de la fila en la tabla dinámica actual
'-----

Dim objRange
Set objRange = objPlugin.ActiveDocument.AuthenticView.Selection

' comprobar si se puede insertar algo

```

```
If objRange.IsInDynamicTable Then
    objRange.DuplicateRow
    ' objRange apunta al principio de la nueva fila
    objRange.Select
End If
```

4.2.9.13 AuthenticRange.EvaluateXPath

Método: [EvaluateXPath](#) (expresión de cadena) como cadena de texto

Valor devuelto

Este método devuelve una cadena de texto

Descripción

`EvaluateXPath()` ejecuta una expresión XPath con el nodo de contexto que está al principio de la selección del intervalo. El resultado se devuelve como cadena de texto y, si se trata de una secuencia, como cadena separada por espacios. Si el nodo de contexto XML es irrelevante, el usuario puede indicar cualquier otro nodo, como `AuthenticView.XMLDataRoot` por ejemplo.

Errores

2001	Objeto no válido
2005	Parámetro no válido
2008	Error interno
2202	Falta el nodo de contexto
2211	Error de XPath

4.2.9.14 AuthenticRange.ExpandTo

Método: [ExpandTo](#) (*eKind* as [SPYAuthenticElementKind](#)) as [AuthenticRange](#) ¹⁰⁴

Descripción

Selecciona todo el elemento de tipo *eKind*, que empieza en o contiene la primera posición del cursor del intervalo. El método devuelve el objeto de intervalo modificado.

Errores

2001	El objeto de intervalo o su objeto relacionado ya no es válido.
2003	La expansión del intervalo sobrepasaría el final del documento.
2005	La dirección indicada para el parámetro de devolución no es válida.

4.2.9.15 AuthenticRange.FirstTextPosition

Propiedad: [FirstTextPosition](#) as [Long](#)

Descripción

Establecer u obtener el índice de posición del texto situado más a la izquierda del objeto de intervalo. Este índice siempre es menor o igual que [LastTextPosition](#)¹²³. La indización empieza por 0 al principio del documento e incrementa con cada nueva posición que pueda ocupar el cursor de texto. Incrementar la posición de prueba 1 punto tiene el mismo efecto que la tecla a la derecha del cursor. Reducir la posición de prueba 1 punto tiene el mismo efecto que la tecla a la izquierda del cursor.

Si para *FirstTextPosition* define un valor mayor que el actual [LastTextPosition](#)¹²³, [LastTextPosition](#)¹²³ se establece en el nuevo *FirstTextPosition*.

CONSEJO: tenga cuidado al utilizar las posiciones del cursor de texto porque es una operación costosa comparada con el posicionamiento del cursor basado en XMLData.

Errores

- 2001 El objeto de intervalo o su objeto relacionado no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.
- 2006 Se indicó una posición del texto situada fuera del documento.

Ejemplos

```
' -----
'                               VBScript
' -----

Dim objAuthenticView
' imaginamos que el documento activo está abierto en la vista Authentic
Set objAuthenticView = objPlugin.AuthenticView

nDocStartPosition = objAuthenticView.DocumentBegin.FirstTextPosition
nDocEndPosition = objAuthenticView.DocumentEnd.FirstTextPosition

' creamos un intervalo que selecciona todo el documento
' de manera eficaz
Dim objRange
' primero tenemos que obtener un objeto de intervalo (cualquiera)
Set objRange = objAuthenticView.DocumentBegin
objRange.FirstTextPosition = nDocStartPosition
objRange.LastTextPosition = nDocEndPosition

' comprobamos si lo hemos hecho correctamente
If objRange.isEqual(objAuthenticView.WholeDocument) Then
    MsgBox "Funcionó la prueba usando posicionamiento directo del cursor de texto"
Else
    MsgBox "Ooops!"
End If
```

4.2.9.16 AuthenticRange.FirstXMLData

Propiedad: [FirstXMLData](#) as [XMLData](#)

Descripción

Establecer u obtener el primer elemento XMLData del documento subyacente que esté parcial o completamente seleccionado por el intervalo. El principio exacto de la selección viene definido por el atributo [FirstXMLDataOffset](#)¹¹³.

Cuando establezca *FirstXMLData* en un objeto de datos nuevo, [FirstXMLDataOffset](#)¹¹³ se establece en la primera posición del cursor dentro de este elemento. Solamente se pueden utilizar objetos XMLData que tengan una posición de cursor. Si establece *FirstXMLData* / [FirstXMLDataOffset](#)¹¹³, se selecciona una posición mayor que el actual [LastXMLData](#)¹²⁴ / [LastXMLDataOffset](#)¹²⁵ (este último se mueve a la nueva posición de inicio).

CONSEJO: puede usar las propiedades [FirstXMLData](#)¹¹² y [LastXMLData](#)¹²⁴ para acceder directamente al documento XML subyacente y manipularlo en los casos en que los métodos disponibles con el objeto [AuthenticRange](#)¹⁰⁴ no sean suficientes.

Errores

- 2001 El objeto de intervalo o su objeto relacionado no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.
- 2008 Error interno
- 2009 No se puede acceder al objeto XMLData.

Ejemplos

```

' -----
'           VBScript
' mostrar nombre del elemento XMLData que está seleccionado
' -----

Dim objAuthenticView
Set objAuthenticView = objPlugin.AuthenticView

Dim objXMLData
Set objXMLData = objAuthenticView.Selection.FirstXMLData
' la vista Authentic añade un elemento secundario "text" a los elementos
' del documento que tienen contenido. Así que tenemos que subir
' un elemento.
Set objXMLData = objXMLData.Parent
MsgBox "La selección actual selecciona el elemento " & objXMLData.Name

```

4.2.9.17 AuthenticRange.FirstXMLDataOffset

Propiedad: [FirstXMLDataOffset](#) as Long

Descripción

Establecer u obtener el desplazamiento de la posición del cursor dentro del elemento [FirstXMLData](#)¹¹² para el principio del intervalo. Las posiciones de desplazamiento se basan en los caracteres que devuelve la propiedad [Text](#)¹³³ y empiezan en la posición 0. Cuando establezca un desplazamiento nuevo, utilice -1 para establecer el desplazamiento en la última posición posible del documento. En estos casos es recomendable prestar particular atención:

- La forma textual de las entradas en cuadros combinados, casillas de verificación y controles similares pueden diferir de lo que aparece en pantalla. Aunque el desplazamiento de los datos se basa en este texto, solamente hay dos posiciones de desplazamiento válidas: una al principio y otra al final de la

entrada. Si intenta establecer el desplazamiento en cualquier otra posición en el medio de la entrada, el desplazamiento se establece al final.

- La forma textual de las entidades XML pueden diferir en longitud de su representación en pantalla. El desplazamiento está basado en esta forma textual.

Si [FirstXMLData](#) / [FirstXMLDataOffset](#)¹¹³ selecciona una posición situada después del actual [LastXMLData](#)¹²⁴ / [LastXMLDataOffset](#)¹²⁵, este último se mueve hasta la posición de inicio nueva.

Errores

- 2001 El objeto de intervalo o su objeto relacionado no es válido.
 2005 El desplazamiento indicado no es válido.
 La dirección indicada para el parámetro de devolución no es válida.

Ejemplos

```
' -----
'                               VBScript
' Seleccionar todo el texto de un elemento XMLData
' usando selección basada en XMLData y ExpandTo
' -----
Dim objAuthenticView
Set objAuthenticView = objPlugin.AuthenticView

' primero usamos las propiedades de intervalo basadas en XMLData
' para seleccionar todo el texto del primer elemento XMLData
' de la selección actual
Dim objRange
Set objRange = objAuthenticView.Selection
objRange.FirstXMLDataOffset = 0 ' empezar al principio del elemento text
objRange.LastXMLData = objRange.FirstXMLData ' seleccionar un elemento solamente
objRange.LastXMLDataOffset = -1 ' seleccionar hasta el final

' esto mismo se puede hacer con el método ExpandTo
Dim objRange2
Set objRange2 = objAuthenticView.Selection.ExpandTo(spyAuthenticTag)

' ¿lo hemos conseguido?
If objRange.IsEqual(objRange2) Then
    objRange.Select()
Else
    MsgBox "Ooops"
End If
```

4.2.9.18 AuthenticRange.GetElementAttributeNames

Método: [GetElementAttributeNames](#) (*strElementName* as String, *out_arrAttributeNames* as Variant)

Descripción

Recupera el nombre de todos los atributos para el elemento envolvente que tiene el nombre indicado. Use pares de elementos/atributos para establecer u obtener el valor de atributo con los métodos [GetElementAttributeValue](#)¹¹⁵ y [SetElementAttributeValue](#)¹³¹.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 El nombre de elemento indicado no es válido.
La dirección indicada para el parámetro de devolución no es válida.

Ejemplos

Consulte también [SetElementAttributeValue](#)¹³¹.

4.2.9.19 AuthenticRange.GetElementAttributeValue

Método: `GetElementAttributeValue` (*strElementName* as `String`, *strAttributeName* as `String`) as `String`

Descripción

Recuperar el valor del atributo especificado en *strAttributeName*, para el elemento identificado con *strElementName*. Si el atributo está permitido pero no tiene asignado ningún valor, se devuelve la cadena vacía. Para buscar el nombre de los atributos que admite un elemento utilice [GetElementAttributeNames](#)¹¹⁴ o [HasElementAttribute](#)¹¹⁹.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 El nombre de elemento indicado no es válido.
El nombre de atributo indicado no es válido.
La dirección indicada para el parámetro de devolución no es válida.

Ejemplos

Consulte también [SetElementAttributeValue](#)¹³¹.

4.2.9.20 AuthenticRange.GetElementHierarchy

Método: `GetElementHierarchy` (*out_arrElementNames* as `Variant`)

Descripción

Recuperar el nombre de todos los elementos XML que son elementos primarios de la selección actual. Los elementos internos se enumeran antes de los elementos envolventes. Si la selección actual no está dentro de un elemento XMLData, se devuelve una lista vacía.

Los nombres de la jerarquía de elementos, junto con el objeto de intervalo, identifican de forma inequívoca los elementos XMLData del documento. A los atributos de estos elementos se puede acceder directamente con [GetElementAttributeNames](#)¹¹⁴ y sus métodos.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

Ejemplos

Consulte también [SetElementAttributeValue](#)¹³¹.

4.2.9.21 AuthenticRange.GetEntityNames

Método: `GetEntityNames` (*out_arrEntityNames* as *Variant*)

Descripción

Recuperar los nombres de todas las entidades definidas. La lista de las entidades recuperadas es independiente de la selección o ubicación actual. Use uno de estos nombres con la función [InsertEntity](#)¹¹⁹.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

Ejemplos

Consulte también [InsertEntity](#)¹¹⁹.

4.2.9.22 AuthenticRange.GetVariableValue

Método: `GetVariableValue` (nombre como cadena, valor como cadena)

Valor devuelto

Obtiene el valor de la variable nombrada.

Errores

- 2001 Objeto no válido.
- 2005 Parámetro no válido
- 2202 No hay contexto.
- 2204 Variable fuera de intervalo
- 2205 La variable no se puede evaluar
- 2206 La variable devuelve una secuencia

4.2.9.23 AuthenticRange.Goto

Método: `Goto` (*eKind* as *SPYAuthenticElementKind*, *nCount* as *Long*, *eFrom* as *SPYAuthenticDocumentPosition*) as [AuthenticRange](#)¹⁰⁴

Descripción

Configura el intervalo para apuntar al principio del elemento nº *nCount* de tipo *eKind*. La posición inicial viene definida por el parámetro *eFrom*.

Utilice valores positivos para *nCount* para navegar hasta el final del documento. Para navegar hasta el principio del documento utilice valores negativos. El método devuelve el objeto de intervalo modificado.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.

- 2003 El destino está después del final del documento.
- 2004 El destino está antes del principio del documento.
- 2005 El tipo de elemento indicado no es válido.
La posición inicial indicada no es válida.
La dirección indicada para el parámetro de devolución no es válida.

4.2.9.24 AuthenticRange.GotoNext

Método: `GotoNext (eKind as SPYAuthenticElementKind)` as [AuthenticRange](#)¹⁰⁴

Descripción

Establece el intervalo en el principio del siguiente elemento de tipo *eKind*. El método devuelve el objeto de intervalo modificado.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2003 El destino está después del final del documento.
- 2005 El tipo de elemento indicado no es válido.
La dirección indicada para el parámetro de devolución no es válida.

Ejemplos

```

'-----
'           VBScript
' Explorar todo el documento palabra por palabra
'-----

Dim objAuthenticView
Set objAuthenticView = objPlugin.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 ' error de reenvío
    End If
Wend

```

4.2.9.25 AuthenticRange.GotoNextCursorPosition

Método: `GotoNextCursorPosition ()` as [AuthenticRange](#)¹⁰⁴

Descripción

Establece el intervalo en la siguiente posición del cursor después de su posición final actual. Devuelve el objeto modificado.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2003 El destino está después del final del documento.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.26 AuthenticRange.GotoPrevious

Método: `GotoPrevious` (*eKind* as `SPYAuthenticElementKind`) as [AuthenticRange](#)¹⁰⁴

Descripción

Establece el intervalo en el principio del elemento de tipo *eKind* que está antes del principio del intervalo actual. El método devuelve el objeto de intervalo modificado.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2004 El destino está antes del principio del documento.
- 2005 El tipo de elemento indicado no es válido.
La dirección indicada para el parámetro de devolución no es válida.

Ejemplos

```

-----
'                               VBScript
' Explorar todo el documento etiqueta por etiqueta
'-----
Dim objAuthenticView
Set objAuthenticView = objPlugin.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentEnd
Dim bEndOfDocument
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 ' error de reenvío
    End If
Wend

```

4.2.9.27 AuthenticRange.GotoPreviousCursorPosition

Método: `GotoPreviousCursorPosition ()` as [AuthenticRange](#)¹⁰⁴

Descripción

Establece el intervalo en la posición del cursor situada inmediatamente antes de la posición actual. Devuelve el objeto modificado.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2004 El destino está antes del principio del documento.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.28 AuthenticRange.HasElementAttribute

Método: `HasElementAttribute (strElementName as String, strAttributeName as String)` as Boolean

Descripción

Comprueba si el elemento envolvente llamado `strElementName` admite el atributo indicado por `strAttributeName`.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 El nombre de elemento indicado no es válido.
La dirección indicada para el parámetro de devolución no es válida.

4.2.9.29 AuthenticRange.InsertEntity

Método: `InsertEntity (strEntityName as String)`

Descripción

Reemplaza la selección del intervalo con la entidad indicada, que debe ser una de las entidades devueltas por [GetEntityNames](#)¹¹⁶.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 Nombre de entrada desconocido.

Ejemplos

```
-----  
'                                     VBScript  
' Insertar la primera entidad de la lista de entidades disponibles  
'-----  
  
Dim objRange  
Set objRange = objPlugin.AuthenticView.Selection  
  
' primero obtenemos el nombre de todas las entidades
```

```

' tal y como aparecen en el ayudante de entrada de XMLSpy
Dim arrEntities
objRange.GetEntityNames arrEntities

' insertamos la primera de la lista
If UBound(arrEntities) >= 0 Then
    objRange.InsertEntity arrEntities(0)
    objRange.Select()
Else
    MsgBox "Para este documento no hay entidades"
End If

```

4.2.9.30 AuthenticRange.InsertRow

Método: `InsertRow ()` as **Boolean**

Descripción

Si el principio del intervalo está dentro de una tabla dinámica, este método inserta una fila nueva antes de la fila actual. La selección del intervalo se modifica para apuntar al principio de la fila recién insertada. La función devuelve *true* si la operación de inserción se completa correctamente. De lo contrario devuelve *false*.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

Ejemplos

```

' -----
'                               VBScript
' Insertar fila al principio de la fila dinámica actual
' -----

Dim objRange
Set objRange = objPlugin.AuthenticView.Selection

' comprobar si se puede insertar algo
If objRange.IsInDynamicTable Then
    objRange.InsertRow
    ' objRange apunta al principio de la nueva fila
    objRange.Select
End If

```

4.2.9.31 AuthenticRange.IsCopyEnabled

Propiedad: `IsCopyEnabled` as **Boolean** (solo lectura)

Descripción

Comprueba si la operación Copiar está permitida para este intervalo.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.

2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.32 AuthenticRange.IsCutEnabled

Propiedad: `IsCutEnabled` as Boolean (solo lectura)

Descripción

Comprueba si la operación Cortar está permitida para este intervalo.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.33 AuthenticRange.IsDeleteEnabled

Propiedad: `IsDeleteEnabled` as Boolean (solo lectura)

Descripción

Comprueba si la operación Eliminar está permitida para este intervalo.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.34 AuthenticRange.IsEmpty

Método: `IsEmpty ()` as Boolean

Descripción

Comprueba si la primera y la última posición del intervalo son iguales.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.35 AuthenticRange.IsEqual

Método: `IsEqual (objCmpRange as AuthenticRange104)` as Boolean

Descripción

Comprueba si el principio y el final de ambos intervalos son el mismo.

Errores

- 2001 Uno de los dos objetos de intervalo comparados no es válido.
- 2005 Dirección no válida para un parámetro de devolución.

4.2.9.36 AuthenticRange.IsFirstRow

Propiedad: `IsFirstRow()` as Boolean (solo lectura)

Descripción

Comprueba si el intervalo está en la primera fila de una tabla. La tabla que se consulta depende de la extensión del intervalo. Si la selección sobrepasa una sola fila de una tabla, se comprueba si esta tabla es el primer elemento de una tabla de incrustación. Consulte la documentación de los ayudantes de entrada para obtener más información.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.37 AuthenticRange.IsInDynamicTable

Método: `IsInDynamicTable()` as Boolean

Descripción

Comprobar si el principio del intervalo está dentro de una tabla que admite las diferentes operaciones en filas.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.38 AuthenticRange.IsLastRow

Propiedad: `IsLastRow()` as Boolean (solo lectura)

Descripción

Comprobar si el intervalo está en la última fila de una tabla. La tabla que se consulta depende de la extensión del intervalo. Si la selección sobrepasa una sola fila de una tabla, se comprueba si esta tabla es el último elemento de una tabla de incrustación. Consulte la documentación de los ayudantes de entrada para obtener más información.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.39 AuthenticRange.IsPasteEnabled

Propiedad: `IsPasteEnabled` as Boolean (solo lectura)

Descripción

Comprueba si la operación Pegar está permitida para este intervalo.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.40 AuthenticRange.IsSelected

Propiedad: `IsSelected` as Boolean

Descripción

Devuelve true() si está presente la selección. El intervalo de selección puede estar vacío (p. ej. si solamente está el cursor).

4.2.9.41 AuthenticRange.IsTextStateApplied

Método: `IsTextStateApplied` (`i_strElementName` as String) as Boolean

Descripción

Comprueba si todo el texto seleccionado está incrustado en un elemento XML llamado `i_strElementName`. Algunos ejemplos típicos para el parámetro `i_strElementName` son "strong", "bold" y "italic".

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.42 AuthenticRange.LastTextPosition

Propiedad: `LastTextPosition` as Long

Descripción

Establecer u obtener el índice de posición del texto situado más a la derecha del objeto de intervalo. Este índice siempre es mayor o igual que [FirstTextPosition](#)¹¹¹. La indización empieza por 0 al principio del documento e incrementa con cada nueva posición que pueda ocupar el cursor de texto. Incrementar la posición de prueba 1 punto tiene el mismo efecto que la tecla a la derecha del cursor. Reducir la posición de prueba 1 punto tiene el mismo efecto que la tecla a la izquierda del cursor.

Si para `LastTextPosition` define un valor menor que el actual [FirstTextPosition](#)¹¹¹, [FirstTextPosition](#)¹¹¹ se establece en el nuevo `LastTextPosition`.

CONSEJO: tenga cuidado al utilizar las posiciones del cursor de texto porque es una operación costosa comparada con el posicionamiento del cursor basado en XMLData.

Errores

- 2001 El objeto de intervalo o su objeto relacionado no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.
- 2006 Se indicó una posición del texto situada fuera del documento.

Ejemplos

```

'-----
'                               VBScript
'-----

Dim objAuthenticView
Set objAuthenticView = objPlugin.AuthenticView

nDocStartPosition = objAuthenticView.DocumentBegin.FirstTextPosition
nDocEndPosition = objAuthenticView.DocumentEnd.FirstTextPosition

' creamos un intervalo que selecciona todo el documento
' de manera eficaz
Dim objRange
' primero tenemos que obtener un objeto de intervalo (cualquiera)
Set objRange = objAuthenticView.DocumentBegin
objRange.FirstTextPosition = nDocStartPosition
objRange.LastTextPosition = nDocEndPosition

' comprobamos si lo hemos hecho correctamente
If objRange.IsEqual(objAuthenticView.WholeDocument) Then
    MsgBox "Funcionó la prueba usando posicionamiento directo del cursor de texto"
Else
    MsgBox "Oops!"
End If

```

4.2.9.43 AuthenticRange.LastXMLData

Consulte también

Propiedad: [LastXMLData](#) as [XMLData](#)

Descripción

Establecer u obtener el último elemento XMLData del documento subyacente que esté parcial o completamente seleccionado por el intervalo. El final exacto de la selección viene definido por el atributo [LastXMLDataOffset](#)¹²⁵.

Cuando establezca *LastXMLData* en un objeto de datos nuevo, [LastXMLDataOffset](#)¹²⁵ se establece en la última posición del cursor dentro de este elemento. Solamente se pueden utilizar objetos XMLData que tengan una posición de cursor. Si establece *LastXMLData* / [LastXMLDataOffset](#)¹²⁵, se selecciona una posición menor que el actual [FirstXMLData](#)¹¹² / [FirstXMLDataOffset](#)¹¹³ (este último se mueve a la nueva posición final).

CONSEJO: puede usar las propiedades [FirstXMLData](#)¹¹² y [LastXMLData](#)¹²⁴ para acceder directamente al documento XML subyacente y manipularlo en los casos en que los métodos disponibles con el objeto [AuthenticRange](#)¹⁰⁴ no sean suficientes.

Errores

- 2001 El objeto de intervalo o su objeto relacionado no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.
- 2008 Error interno
- 2009 No se puede acceder al objeto XMLData.

4.2.9.44 AuthenticRange.LastXMLDataOffset

Propiedad: [LastXMLDataOffset](#) as Long

Descripción

Establecer u obtener la posición del cursor dentro del elemento [LastXMLData](#)¹²⁴ para el final del intervalo.

Las posiciones de desplazamiento se basan en los caracteres que devuelve la propiedad [Text](#)¹³³ y empiezan en la posición 0. Cuando establezca un desplazamiento nuevo, utilice -1 para establecer el desplazamiento en la última posición posible del documento. En estos casos es recomendable prestar particular atención:

- La forma textual de las entradas en cuadros combinados, casillas de verificación y controles similares pueden diferir de lo que aparece en pantalla. Aunque el desplazamiento de los datos se basa en este texto, solamente hay dos posiciones de desplazamiento válidas: una al principio y otra al final de la entrada. Si intenta establecer el desplazamiento en cualquier otra posición en el medio de la entrada, el desplazamiento se establece al final.
- La forma textual de las entidades XML pueden diferir en longitud de su representación en pantalla. El desplazamiento está basado en esta forma textual.

Si [LastXMLData](#)¹²⁴ / [LastXMLDataOffset](#)¹²⁵ selecciona una posición situada antes del actual [FirstXMLData](#)¹¹² / [FirstXMLDataOffset](#)¹¹³, este último se mueve hasta la posición final nueva.

Errores

- 2001 El objeto de intervalo o su objeto relacionado no es válido.
- 2005 El desplazamiento indicado no es válido.
La dirección indicada para el parámetro de devolución no es válida.

Ejemplos

```
'-----
'                               VBScript
' Seleccionar todo el texto de un elemento XMLData
' usando selección basada en XMLData y ExpandTo
'-----
Dim objAuthenticView
Set objAuthenticView = objPlugin.AuthenticView

' primero usamos las propiedades de intervalo basadas en XMLData
```

```

' para seleccionar todo el texto del primer elemento XMLData
' de la selección actual
Dim objRange
Set objRange = objAuthenticView.Selection
objRange.FirstXMLDataOffset = 0 ' empezar al principio del elemento text
objRange.LastXMLData = objRange.FirstXMLData ' seleccionar un elemento solamente
objRange.LastXMLDataOffset = -1 ' seleccionar hasta el final

' esto mismo se puede hacer con el método ExpandTo
Dim objRange2
Set objRange2 = objAuthenticView.Selection.ExpandTo(spyAuthenticTag)

' ¿lo hemos conseguido?
If objRange.IsEqual(objRange2) Then
    objRange.Select()
Else
    MsgBox "Ooops"
End If

```

4.2.9.45 AuthenticRange.MoveBegin

Método: `MoveBegin` (*eKind* as `SPYAuthenticElementKind`, *nCount* as `Long`) as [AuthenticRange](#) ¹⁰⁴

Descripción

Mover el inicio del intervalo al principio del elemento nº *nCount* de tipo *eKind*. El recuento comienza por el inicio actual del objeto de intervalo.

Utilice números positivos para *nCount* para desplazarse hacia el final del documento y negativos para desplazarse hacia el principio. El final del intervalo no se mueve, a no ser que el nuevo inicio sea mayor. En este caso el final se mueve al nuevo inicio. El método devuelve el objeto de intervalo modificado.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2003 El destino está después del final del documento.
- 2004 El destino está antes del principio del documento.
- 2005 El tipo de elemento indicado no es válido.
La dirección indicada para el parámetro de devolución no es válida.

4.2.9.46 AuthenticRange.MoveEnd

Método: `MoveEnd` (*eKind* as `SPYAuthenticElementKind`, *nCount* as `Long`) as [AuthenticRange](#) ¹⁰⁴

Descripción

Mover el final del intervalo al principio del elemento nº *nCount* de tipo *eKind*. El recuento comienza por el final actual del objeto de intervalo.

Utilice números positivos para *nCount* para desplazarse hacia el final del documento y negativos para desplazarse hacia el principio. El inicio del intervalo no se mueve, a no ser que el nuevo final sea menor. En este caso el inicio se mueve al nuevo final. El método devuelve el objeto de intervalo modificado.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2003 El destino está después del final del documento.
- 2004 El destino está antes del principio del documento.
- 2005 El tipo de elemento indicado no es válido.
La dirección indicada para el parámetro de devolución no es válida.

4.2.9.47 AuthenticRange.MoveRowDown

Método: [MoveRowDown](#) () as [Boolean](#)

Descripción

Si el principio del intervalo está dentro de una tabla dinámica y selecciona una fila que no es la última fila de esta tabla, este método intercambia esta fila con la fila situada justo debajo. La selección del intervalo se mueve con la fila, pero sigue en el mismo lugar. La función devuelve *true* si la operación finaliza correctamente. De lo contrario devuelve *false*.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.48 AuthenticRange.MoveRowUp

Método: [MoveRowUp](#) () as [Boolean](#)

Descripción

Si el principio del intervalo está dentro de una tabla dinámica y selecciona una fila que no es la primera fila de esta tabla, este método intercambia esta fila con la fila situada encima. La selección del intervalo se mueve con la fila, pero sigue en el mismo lugar. La función devuelve *true* si la operación finaliza correctamente. De lo contrario devuelve *false*.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

Ejemplos

Consulte también [Ejemplo: ordenar una tabla](#)²⁸.

4.2.9.49 AuthenticRange.Parent

Consulte también

Propiedad: [Parent](#) as [AuthenticView](#)¹³⁹ (solo lectura)

Descripción

Acceder a la vista propietaria de este objeto de intervalo.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.50 AuthenticRange.Paste

Método: `Paste ()` as `Boolean`

Descripción

Devuelve *False* si el búfer de copia/pega está vacío o si su contenido no puede reemplazar la selección actual.

De lo contrario, elimina la selección actual, inserta el contenido del búfer de copia/pega y devuelve *True*.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.

4.2.9.51 AuthenticRange.PerformAction

Método: `PerformAction (eAction as SPYAuthenticActions, strElementName as String)` as `Boolean`

Descripción

`PerformAction` y sus métodos ofrecen acceso a las funciones de los ayudantes de entrada de `Authentic`. Esta función permite modificar el contenido del documento de forma sencilla y coherente sin necesidad de saber dónde tienen lugar los cambios exactamente. El principio del objeto de intervalo se usa para encontrar la siguiente ubicación válida donde se puede llevar a cabo la acción indicada. Si dicha ubicación no se encuentra, el método devuelve *False*. De lo contrario, el documento se modifica y el intervalo apunta al principio de la modificación.

CONSEJO: utilice [CanPerformActionWith](#)¹⁰⁷ para averiguar el nombre de los elementos que se pueden pasar como segundo parámetro.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 La dirección indicada para el parámetro de devolución no es válida.
- 2007 La acción indicada no es válida.

Ejemplos

```
'-----
'           VBScript
' Insertar el elemento más interior
'-----
Dim objRange
Set objRange = objPlugin.AuthenticView.Selection
```

```

' determinamos qué elementos se pueden insertar en la posición actual
Dim arrElements()
objRange.CanPerformActionWith spyAuthenticInsertBefore, arrElements

' insertamos el primer elemento (es decir, el más interior)
If UBound(arrElements) >= 0 Then
    objRange.PerformAction spyAuthenticInsertBefore, arrElements(0)
    ' objRange ahora apunta al principio del elemento insertado
    ' establecemos un valor y una posición predeterminados en su final
    objRange.Text = "Hello"
    objRange.ExpandTo(spyAuthenticTag).CollapsToEnd().Select
Else
    MsgBox "En la posición actual no se puede insertar ningún elemento"
End If

```

4.2.9.52 AuthenticRange.Select

Método: [Select](#) ()

Descripción

Convierte el intervalo en la selección actual de la interfaz del usuario. Con '*objRange.Parent.Selection = objRange*' puede conseguir el mismo resultado.

Errores

2001 El objeto de intervalo o su objeto relacionado ya no es válido.

Ejemplos

```

' -----
'                               VBScript
' -----

' establecer selección actual en final del documento
objPlugin.objAuthenticView.DocumentEnd.Select()

```

4.2.9.53 AuthenticRange.SelectNext

Método: [SelectNext](#) (*eKind* as [SPYAuthenticElementKind](#)) as [AuthenticRange](#) ¹⁰⁴

Descripción

Selecciona el elemento de tipo *eKind* situado tras el final actual del intervalo. El método devuelve el objeto de intervalo modificado.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2003 El destino está después del final del documento.
- 2005 El tipo de elemento indicado no es válido.
La dirección indicada para el parámetro de devolución no es válida.

Ejemplos

```

' -----
'                               VBScript
' Explorar todo el documento palabra por palabra
' -----

Dim objAuthenticView
Set objAuthenticView = objPlugin.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 ' error de reenvío
    End If
Wend

```

4.2.9.54 AuthenticRange.SelectPrevious

Método: [GotoPrevious](#) (*eKind* as *SPYAuthenticElementKind*) as [AuthenticRange](#) ¹⁰⁴

Descripción

Selecciona el elemento de tipo *eKind* situado antes del inicio actual del intervalo. El método devuelve el objeto de intervalo modificado.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2004 El destino está antes del principio del documento.
- 2005 El tipo de elemento indicado no es válido.
La dirección indicada para el parámetro de devolución no es válida.

Ejemplos

```

' -----
'                               VBScript
' Explorar todo el documento etiqueta por etiqueta
' -----

Dim objAuthenticView
Set objAuthenticView = objPlugin.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentEnd
Dim bEndOfDocument
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 ' error de reenvío
    End If
Wend

```

4.2.9.55 AuthenticRange.SetElementAttributeValue

Método: `SetElementAttributeValue` (*strElementName* as String, *strAttributeName* as String, *strAttributeValue* as String)

Descripción

Recuperar el valor del atributo especificado en *strAttributeName* para el elemento identificado por *strElementName*. Si el atributo es compatible pero no tiene asignado ningún valor, se devuelve una cadena vacía. Para averiguar el nombre de los atributos compatibles con un elemento utilice [GetElementAttributeNames](#)¹¹⁴ o [HasElementAttribute](#)¹¹⁹.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 El nombre de elemento indicado no es válido.
El nombre de atributo indicado no es válido.
Se indicó un valor de atributo no válido.

Ejemplos

```

'-----
'           VBScript
' Obtener y establecer atributos del elemento
'-----

Dim objRange
Set objRange = objPlugin.AuthenticView.Selection

' primero averiguamos todos los elementos situados al principio del intervalo
Dim arrElements
objRange.GetElementHierarchy arrElements

If IsArray(arrElements) Then
    If UBound(arrElements) >= 0 Then
        ' usamos el elemento de nivel superior y averiguamos sus atributos válidos
        Dim arrAttrs()
        objRange.GetElementAttributeNames arrElements(0), arrAttrs

        If UBound(arrAttrs) >= 0 Then
            ' recuperamos el valor actual del primer atributo válido
            Dim strAttrVal
            strAttrVal = objRange.GetElementAttributeValue (arrElements(0),
arrAttrs(0))

```

```

        MsgBox "el valor actual de " & arrElements(0) & "/" & arrAttrs(0) & "
es: " & strAttrVal

        ' cambiamos este valo y lo volvemos a leer
        strAttrVal = "Hello"
        objRange.SetElementAttributeValue arrElements(0), arrAttrs(0),
strAttrVal
        strAttrVal = objRange.GetElementAttributeValue (arrElements(0),
arrAttrs(0))
        MsgBox "el nuevo valor de " & arrElements(0) & "/" & arrAttrs(0) & "
es: " & strAttrVal
    End If
End If
End If

```

4.2.9.56 AuthenticRange.SetFromRange

Método: [SetFromRange](#) (*objSrcRange* as [AuthenticRange](#)¹⁰⁴)

Descripción

Establece el objeto de intervalo en las mismas posiciones de inicio y fin que *objSrcRange*.

Errores

- 2001 Uno de los dos objetos de intervalo no es válido.
- 2005 Se indicó un objeto null como objeto de origen.

4.2.9.57 AuthenticRange.SetVariableValue

Método: [SetVariableValue](#)(nombre como cadena, valor como cadena)

Valor devuelto

Establece el valor de la variable nombrada.

Errores

- 2001 Objeto no válido.
- 2002 Falta el nodo de contexto.
- 2204 Variable fuera de intervalo
- 2205 La variable no se puede evaluar
- 2206 La variable devuelve una secuencia
- 2207 Variable de solo lectura
- 2208 No se permiten cambios

4.2.9.58 AuthenticRange.Text

Propiedad: [Text](#) as [String](#)

Descripción

Establecer u obtener el contenido textual seleccionado por el objeto de intervalo.

El número de caracteres recuperados no es necesariamente idéntico porque hay posiciones del cursor de texto entre el principio y el final del intervalo seleccionado. La mayoría de los elementos del documento admiten una posición final del cursor diferente a la posición inicial del cursor del elemento siguiente. Las listas desplegables conservan una sola posición del cursor, pero pueden seleccionar cadenas de cualquier longitud. En el caso de los botones de opción y de las casillas de verificación, el valor de la propiedad Text almacena la cadena del elemento XML correspondiente.

Si el intervalo selecciona más de un elemento, el texto es la concatenación de los textos. Las entidades XML se expanden para que '&' se espere como '&'.

Al asignarse el texto a la cadena vacía, no se elimina ningún elemento XML. Utilice [Cut](#)¹⁰⁹, [Delete](#)¹⁰⁹ o [PerformAction](#)¹²⁸ en su lugar.

Errores

- 2001 El objeto de intervalo o su objeto relacionado ya no es válido.
- 2005 Dirección no válida para un parámetro de devolución.

4.2.10 AuthenticSelection

Propiedades

[Start](#)¹³⁴

[StartTextPosition](#)¹³⁴

[End](#)¹³³

[EndTextPosition](#)¹³³

4.2.10.1 AuthenticSelection.End

Declaración [End](#) as [XMLData](#)¹⁶⁸

Descripción

Elemento XML donde finaliza la selección actual.

4.2.10.2 AuthenticSelection.EndTextPosition

Declaración [EndTextPosition](#) as [long](#)

Descripción

Posición en [Authentic.End](#)¹³³ donde finaliza la selección.

4.2.10.3 AuthenticSelection.Start

Declaración [Start](#) as [XMLData](#)¹⁶⁸

Descripción

Elemento XML donde comienza la selección actual.

4.2.10.4 AuthenticSelection.StartTextPosition

Declaración [StartTextPosition](#) as [long](#)

Descripción

Posición en [Authentic.Start](#)¹³⁴ donde comienza la selección.

4.2.11 AuthenticToolBarButton

Métodos**Propiedades**

[CommandID](#)¹³⁴

4.2.11.1 AuthenticToolBarButton.CommandID

Declaración [CommandID](#) as [SPYAuthenticCommand](#)¹⁷⁸

Descripción

El CommandID para el botón de la barra de herramientas.

4.2.12 AuthenticToolBarButtons

Métodos

[Item](#)¹³⁵

[NewButton](#)¹³⁵

[NewCustomButton](#)¹³⁶

[NewSeparator](#)¹³⁷

[Remove](#)¹³⁷

Propiedades[Count](#) ¹³⁵

4.2.12.1 AuthenticToolBarButtons.Count

Declaración `Count` as `long`**Descripción**

Obtener el número de botones que hay en la barra de herramientas.
Solo lectura.

4.2.12.2 AuthenticToolBarButtons.Item

Declaración `Item`(`n` as `long`) as [AuthenticToolBarButton](#) ¹³⁴**Descripción**

Obtener el botón nº `n` de la barra herramientas actual. `n` empieza por 1.

Ejemplo

Consulte el ejemplo disponible en [AuthenticToolBarButtons.NewButton](#) ¹³⁵

4.2.12.3 AuthenticToolBarButtons.NewButton

Declaración `NewButton`(`nPosition` as `long`, `nCommandId` as [SPYAuthenticCommand](#) ¹⁷⁸)**Descripción**

Inserta un botón nuevo para el comando `nCommandId` en la posición `nPosition` de la barra de herramientas. `nPosition` empieza por 1.

Ejemplo

Añadir una barra de herramientas nueva, alinearla en la parte inferior y añadir botones nuevos

```
objPlugIn.ToolbarRows.NewRow(3)           // añadir una barra de
                                           herramientas nueva (fila 3)

set ToolbarRow = objPlugIn.ToolbarRows.Item(3)
set Buttons = ToolbarRow.Buttons
ToolbarRow.Alignment = 2                  // alinear la barra de
                                           herramientas en la parte
                                           inferior

Buttons.NewButton 1, 2                    // agregar el botón Imprimir
Buttons.NewButton 1, 3                    // agregar el botón Vista
                                           previa de impresión
Buttons.NewSeparator 2                    // agregar separador
Buttons.NewButton 1, 4                    // agregar el botón Validar
```

Cuando se llama a `StartEditing` o `ReloadToolbars`, se usa la configuración modificada de la barra de herramientas.

4.2.12.4 AuthenticToolBarButtons.NewCustomButton

Declaración `NewCustomButton(nPosition as long, strName as String, strTooltip as String, strBitmapURL as String)`

Descripción

Inserta un botón personalizado nuevo llamado `strName` en la posición `nPosition` de la barra de herramientas. `nPosition` empieza por 1. `strTooltip` se usa como texto de información rápida.

`strBitmapURL` es la ubicación del mapa de bits que debe utilizarse para el nuevo botón. La ruta de acceso es relativa a la ruta definida con la propiedad `TextStateBmpURL`⁸⁹.

Ejemplo

Imagine que insertó un botón personalizado llamado `MiFunción` en las barras de herramientas. El siguiente controlador de eventos para `doceditcommand`⁴⁸ demuestra la reacción que se produce cuando el usuario hace clic en el botón y cómo se puede configurar su estado habilitado / deshabilitado.

```
<SCRIPT LANGUAGE=javascript FOR=objPlugIn EVENT=doceditcommand>
  // event.type recibe el valor "command" si el usuario hizo clic en el botón
  if(objPlugIn.event.type == "command")
  {
    if(objPlugIn.event.srcElement.Name == "MiFunción")
      window.alert("Acaba de pulsar el botón personalizado.");
  }

  // event.type recibe el valor "update" si se debe establecer el estado del botón
  if(objPlugIn.event.type == "update")
  {
    if(objPlugIn.event.srcElement.Name == "MiFunción")
    {
      // habilitamos el botón si hay un solo elemento seleccionado

if(objPlugIn.CurrentSelection.Start.IsSameNode(objPlugIn.CurrentSelection.End))
      objPlugIn.event.returnValue = 1;
    else
      objPlugIn.event.returnValue = 0;

    objPlugIn.event.cancelBubble = true;
  }
  }
</SCRIPT>
```

4.2.12.5 AuthenticToolBarButtons.NewSeparator

Declaración `NewSeparator(nPosition as long)`

Descripción

Inserta un separador en la posición `nPosition` de la barra de herramientas. `nPosition` empieza por 1.

Ejemplo

Consulte el ejemplo disponible en [AuthenticToolBarButtons.NewButton](#) ¹³⁵

4.2.12.6 AuthenticToolBarButtons.Remove

Declaración `Remove(nPosition as long)`

Descripción

Quita el botón o el separador de la posición `nPosition` de la barra de herramientas. `nPosition` empieza por 1.

4.2.13 AuthenticToolBarRow

Métodos

[Alignment](#) ¹³⁷

Propiedades

[Buttons](#) ¹³⁸

4.2.13.1 AuthenticToolBarRowAlignment

Declaración `Alignment(nAlign as SPYAuthenticToolBarAlignment)` ¹⁸²

Descripción

Obtiene o establece la alineación de la barra de herramientas dentro del complemento.

Ejemplo

Alinear todas las barras de herramientas en la parte inferior:

```
for each ToolbarRow in objPlugin.ToolbarRows
    ToolbarRow.Alignment = 2
next
```

4.2.13.2 AuthenticToolBarRowButtons

Declaración `Buttons` as [AuthenticToolBarButtons](#) ¹³⁴

Descripción

Obtener todos los botones de la barra de herramientas.

Ejemplo

Consulte el ejemplo disponible en [AuthenticToolBarButtons.NewButton](#) ¹³⁵

4.2.14 AuthenticToolBarRows

Métodos

[Item](#) ¹³⁸

[RemoveRow](#) ¹³⁹

[NewRow](#) ¹³⁹

Propiedades

[Count](#) ¹³⁸

4.2.14.1 AuthenticToolBarRows.Count

Declaración: `Count` as `long`

Descripción

Obtener el número de barras de herramientas definidas.
Solo lectura.

4.2.14.2 AuthenticToolBarRows.Item

Declaración `Item(nPosition as long)` as [AuthenticToolBarRow](#) ¹³⁷

Descripción

Obtener la barra de herramientas de la posición `nPosition`. `nPosition` empieza por 1
Solo lectura.

Ejemplo

Consulte el ejemplo disponible en [AuthenticToolBarButtons.NewButton](#) ¹³⁵

4.2.14.3 AuthenticToolbarRows.RemoveRow

Declaración `RemoveRow(nPosition as long)`

Descripción

Quitar la barra de herramientas de la posición nPosition de la interfaz del usuario. nPosition empieza por 1.

4.2.14.4 AuthenticToolbarRows.NewRow

Declaración `NewRow(nPosition as long)`

Descripción

Crea e inserta una fila nueva de barra de herramientas. nPosition empieza por 1.

Ejemplo

Consulte el ejemplo disponible en [AuthenticToolbarButtons.NewButton](#)¹³⁵

4.2.15 AuthenticView

Propiedades	Métodos	Eventos
Application ¹⁵¹	Goto ¹⁵⁵	
DocumentBegin ¹⁵³	Print ¹⁵⁷	
DocumentEnd ¹⁵³	Redo ¹⁵⁷	
MarkupVisibility ¹⁵⁶	Undo ¹⁵⁸	
Parent ¹⁵⁶	UpdateXMLInstanceEntities ¹⁵⁸	
Selection ¹⁵⁷		
WholeDocument ¹⁵⁹		

Descripción

AuthenticView y su objeto secundario [AuthenticRange](#)¹⁰⁴ ofrecen una interfaz para la vista Authentic que permite modificar de forma coherente y sencilla el contenido de los documentos. Si hay métodos y propiedades repetidos del objeto Authentic, las funciones de modificación de contenido del objeto Authentic se convierten en obsoletas. De hecho, se recomienda usar la nueva interfaz AuthenticView.

AuthenticView ofrece acceso a funciones concretas como la función de impresión, el búfer de las operaciones Deshacer y la selección o posición actual del cursor.

AuthenticView utiliza objetos de tipo [AuthenticRange](#)¹⁰⁴ para facilitar la navegación dentro del documento y para permitir una selección flexible de los elementos de texto lógicos. Use las propiedades [DocumentBegin](#)¹⁵³, [DocumentEnd](#)¹⁵³ y [WholeDocument](#)¹⁵⁹ para selecciones sencillas y el método [Goto](#)¹⁵⁵ para selecciones

más complejas. Para navegar por un intervalo concreto del documento, consulte las propiedades y los métodos del objeto [AuthenticRange](#)¹⁰⁴.

4.2.15.1 Eventos

4.2.15.1.1 OnBeforeCopy

Evento: `OnBeforeCopy()` as Boolean

Entorno de scripting de XMLSpy: VBScript:

```
Function On_AuthenticBeforeCopy()  
    ' On_AuthenticBeforeCopy = False ' para deshabilitar la operación  
End Function
```

Entorno de scripting de XMLSpy: JScript:

```
function On_AuthenticBeforeCopy()  
{  
    // devolver false; /* para deshabilitar la operación */  
}
```

Complemento de XMLSpy para entornos IDE:

```
IXMLSpyPlugIn.OnEvent (21, ...) // nEventId = 21
```

Descripción

Este evento se desencadena antes de que se lleve a cabo una operación Copiar en el documento. Devolver *True* (o nada) para permitir la operación Copiar. Devolver *False* para deshabilitar la copia.

4.2.15.1.2 OnBeforeCut

Evento: `OnBeforeCut()` as Boolean

Entorno de scripting de XMLSpy: VBScript:

```
Function On_AuthenticBeforeCut()  
    ' On_AuthenticBeforeCut = False ' para deshabilitar la operación  
End Function
```

Entorno de scripting de XMLSpy: JScript:

```
function On_AuthenticBeforeCut()  
{  
    // devolver false; /* para deshabilitar la operación */  
}
```

Complemento de XMLSpy para entornos IDE:

```
IXMLSpyPlugIn.OnEvent (20, ...) // nEventId = 20
```

Descripción

Este evento se desencadena antes de que se lleve a cabo una operación Cortar en el documento. Devolver *True* (o nada) para permitir la operación Cortar. Devolver *False* para deshabilitar la operación.

4.2.15.1.3 OnBeforeDelete

Evento: `OnBeforeDelete()` as Boolean

Entorno de scripting de XMLSpy: VBScript:

```
Function On_AuthenticBeforeDelete()
    ' On_AuthenticBeforeDelete = False ' para deshabilitar la operación
End Function
```

Entorno de scripting de XMLSpy: JScript:

```
function On_AuthenticBeforeDelete()
{
    // devolver false; /* para deshabilitar la operación */
}
```

Complemento de XMLSpy para entornos IDE:

```
IXMLSpyPlugIn.OnEvent (22, ...) // nEventId = 22
```

Descripción

Este evento se desencadena antes de que se lleve a cabo una operación Eliminar en el documento. Devolver *True* (o nada) para permitir la operación Eliminar. Devolver *False* para deshabilitar la operación.

4.2.15.1.4 OnBeforeDrop

Evento: `OnBeforeDrop` (*i_nXPos* as Long, *i_nYPos* as Long, *i_ipRange* as AuthenticRange, *i_ipData* as cancelBoolean)

Entorno de scripting de XMLSpy: VBScript:

```
Function On_AuthenticBeforeDrop(nXPos, nYPos, objRange, objData)
    ' On_AuthenticBeforeDrop = False ' para deshabilitar la operación
End Function
```

Entorno de scripting de XMLSpy: JScript:

```
function On_AuthenticBeforeDrop(nXPos, nYPos, objRange, objData)
{
    // devolver false; /* para deshabilitar la operación */
}
```

Complemento de XMLSpy para entornos IDE:

```
IXMLSpyPlugIn.OnEvent (11, ...) // nEventId = 11
```

Descripción

Este evento se desencadena cada vez que arrastra y coloca un objeto dentro de la ventana de la aplicación. Toda la información relacionada con el evento se pasa en forma de parámetros.

Los dos primeros parámetros especifican la posición del ratón cuando ocurre el evento. El parámetro *objRange* pasa un objeto de intervalo que selecciona el elemento XML situado en la posición del ratón. El valor de este parámetro puede ser NULL. Compruébelo antes de acceder al objeto de intervalo. El parámetro *objData* permite acceder a la información disponible sobre el objeto que se está arrastrando.

Devolver *False* para cancelar la operación Colocar. Devolver *True* (o nada) para continuar con la operación.

Ejemplos

```
' -----
' Fragmento de código VB: conectarse a eventos a nivel de objeto
' -----
' acceder a XMLSpy (sin comprobar si hay errores)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' esta es la rutina de devolución de llamada del evento
' conectada al evento OnBeforeDrop del objeto objView
Private Function objView_OnBeforeDrop(ByVal i_nXPos As Long, ByVal i_nYPos As Long,
                                      ByVal i_ipRange As IAuthenticRange,
                                      ByVal i_ipData As IAuthenticDataTransfer) As
Boolean

    If (Not i_ipRange Is Nothing) Then
        MsgBox ("Dropping on content is prohibited");
        Return False;
    Else
        Return True;
    End If
End Function

' use la palabra clave de VBA WithEvents para conectarse al evento a nivel de objeto
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continuar con más código ...
' y servir el bucle de mensajes
```

4.2.15.1.5 OnBeforePaste

Evento: *OnBeforePaste* (*objData* as Variant, *strType* as String) as Boolean

Entorno de scripting de XMLSpy: VBScript:

```
Function On_AuthenticBeforePaste(objData, strType)
    ' On_AuthenticBeforePaste = False ' para deshabilitar la operación
End Function
```

Entorno de scripting de XMLSpy: JScript:

```
function On_AuthenticBeforePaste(objData, strType)
{
    // devolver false; /* para deshabilitar la operación */
}
```

Complemento de XMLSpy para entornos IDE:

```
IXMLSpyPlugIn.OnEvent(19, ...) // nEventId = 19
```

Descripción

Este evento se desencadena antes de que se lleve a cabo la operación Pegar en el documento. El parámetro *strType* es "TEXT", "UNICODETEXT" o "IUNKNOWN". En los dos primeros casos *objData* contiene una representación de cadena del objeto que se pegará. En el último, *objData* contiene un puntero que apunta a una interfaz IUnknown de COM.

Devolver *True* (o nada) para permitir la operación Pegar. Devolver *False* para deshabilitar la operación.

4.2.15.1.6 OnBeforeSave

Evento: *OnBeforeSave* (SaveAs flag) as Boolean

Descripción: *OnBeforeSave* da la oportunidad de, por ejemplo, advertir al usuario antes de sobrescribir el documento XML o de convertir el documento en un documento de solo lectura cuando no se cumplan ciertos requisitos. El evento se desencadenará antes de que aparezca el cuadro de diálogo del archivo. **Nota:** el evento se desencadena cuando se guarda el documento XML y no cuando se guarda el diseño SPS en StyleVision.

4.2.15.1.7 OnDragOver

Evento: *OnDragOver* (*nXPos* as Long, *nYPos* as Long, *eMouseEvent* as SPYMouseEvent, *objRange* as AuthenticRange, *objData* as AuthenticDataTransfer) as Boolean

Entorno de scripting de XMLSpy: VBScript:

```
Function On_AuthenticDragOver(nXPos, nYPos, eMouseEvent, objRange, objData)
    ' On_AuthenticDragOver = False ' para deshabilitar la operación
End Function
```

Entorno de scripting de XMLSpy: JScript:

```
function On_AuthenticDragOver(nXPos, nYPos, eMouseEvent, objRange, objData)
{
    // devolver false; /* para deshabilitar la operación */
}
```

Complemento de XMLSpy para entornos IDE:

```
IXMLSpyPlugIn.OnEvent(10, ...) // nEventId = 10
```

Descripción

Este evento se desencadena cada vez que se arrastra un objeto desde fuera de la vista Authentic por encima de la ventana de la aplicación. Toda la información relacionada con el evento se pasa en forma de parámetros.

Los tres primeros parámetros especifican la posición del ratón, el estado del botón del ratón y el estado de las claves virtuales en el momento en que tiene lugar el evento. El parámetro *objRange* pasa un objeto de intervalo que selecciona el elemento XML situado bajo la posición del ratón. El valor de este parámetro puede ser NULL. El parámetro *objData* permite acceder a la información disponible sobre el objeto que se está arrastrando.

Devolver *False* para cancelar la operación de arrastrar. Devolver *True* (o nada) para continuar con la operación.

Ejemplos

```
' -----
' fragmento de código VB: conectarse a eventos a nivel de objeto
' -----
' acceder a XMLSpy (sin buscar errores)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' esta es la rutina de devolución de llamada
' conectada al evento OnDragOver del objeto objView
Private Function objView_OnDragOver(ByVal i_nXPos As Long, ByVal i_nYPos As Long,
    ByVal i_eMouseEvent As SPYMouseEvent,
    ByVal i_ipRange As IAuthenticRange,
    ByVal i_ipData As IAuthenticDataTransfer) As Boolean

    If (((i_eMouseEvent And spyShiftKeyDownMask) <> 0) And
        (Not i_ipRange Is Nothing)) Then
        MsgBox ("Floating over element " & i_ipRange.FirstXMLData.Parent.Name);
    End If

    Return True;
End Function

' use la palabra clave de VBA WithEvents para conectarse al evento a nivel de objeto
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continuar con más código ...
' y servir el bucle de mensajes
```

4.2.15.1.8 OnKeyboardEvent

Evento: *OnKeyboardEvent* (*eKeyEvent* as SPYKeyEvent, *nKeyCode* as Long, *nVirtualKeyStatus* as Long) as Boolean

Entorno de scripting de XMLSpy: VBScript:

```
Function On_AuthenticKeyboardEvent(eKeyEvent, nKeyCode, nVirtualKeyStatus)
    ' On_AuthenticKeyboardEvent = True ' to cancel bubbling of event
End Function
```

Entorno de scripting de XMLSpy: JScript:

```
function On_AuthenticKeyboardEvent(eKeyEvent, nKeyCode, nVirtualKeyStatus)
{
    // devolver false; /* to cancel bubbling of event */
}
```

Complemento de XMLSpy para entornos IDE:

```
IXMLSpyPlugIn.OnEvent (30, ...) // nEventId = 30
```

Descripción

Este evento se desencadena para los mensajes de Windows *WM_KEYDOWN*, *WM_KEYUP* y *WM_CHAR*.

El tipo de mensaje está disponible en el parámetro *eKeyEvent*. El estado de las teclas virtuales está en el parámetro *nVirtualKeyStatus*. Utilice las máscaras de bits definidas en el tipo de datos de enumeración *SPYVirtualKeyMask* para probar las distintas claves y sus combinaciones.

NOTA: estos eventos del entorno de scripting y del complemento de Altova XMLSpy para entornos IDE son compatibles pero se vuelven obsoletos con este evento:

```
On_AuthenticKeyUp()           IXMLSpyPlugIn.OnEvent (13, ...) // nEventId = 13
On_AuthenticKeyDown()        IXMLSpyPlugIn.OnEvent (12, ...) // nEventId = 12
On_AuthenticKeyPressed()     IXMLSpyPlugIn.OnEvent (14, ...) // nEventId = 14
```

Ejemplos

```
' -----
' fragmento de código VB: conectarse a eventos a nivel de objeto
' -----
' acceder a XMLSpy (sin buscar errores)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' esta es la rutina de devolución de llamada
' conectada al evento OnKeyboard del objeto objView
Private Function objView_OnKeyboardEvent(ByVal i_keyEvent As Long, ByVal io_pnKeyCode As Long,
ByVal i_nVirtualKeyStatus As Long) As Boolean
    If ((i_keyEvent = XMLSpyLib.spyKeyUp) And ((i_nVirtualKeyStatus And
XMLSpyLib.spyCtrlKeyMask) <> 0)) Then
        MsgBox ("Ctrl " & io_pnKeyCode & " pressed")
        objView_OnKeyboardEvent = True
    Else
        objView_OnKeyboardEvent = False
    End If
End Function

' use la palabra clave de VBA WithEvents para conectarse al evento a nivel de objeto
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continuar con más código ...
' y servir el bucle de mensajes
```

4.2.15.1.9 OnLoad

Evento: OnLoad ()

Descripción: OnLoad se puede usar para, por ejemplo, restringir alguna función de la vista Authentic.

Ejemplo

```
function On_AuthenticLoad( )
{
    // Deshabilitamos todos los ayudantes de entrada para evitar que el usuario manipule
    la estructura XML
    AuthenticView.DisableElementEntryHelper();
    AuthenticView.DisableAttributeEntryHelper();

    // Y deshabilitamos los botones de marcado por el mismo motivo
    AuthenticView.SetToolBarButtonState( 'AuthenticMarkupSmall',
authenticToolBarButtonDisabled );
    AuthenticView.SetToolBarButtonState( 'AuthenticMarkupLarge',
authenticToolBarButtonDisabled );
    AuthenticView.SetToolBarButtonState( 'AuthenticMarkupMixed',
authenticToolBarButtonDisabled );
}
```

En este ejemplo el estado de los botones de la barra de herramientas **Marcado pequeño**, **Marcado mixto**, **Marcado pequeño** se manipulan con ayuda de los identificadores de botón. Aquí puede ver una [lista completa](#)¹⁴⁸.

4.2.15.1.10 OnMouseEvent

Evento: OnMouseEvent (*nXPos* as Long, *nYPos* as Long, *eMouseEvent* as SPYMouseEvent, *objRange* as AuthenticRange) as Boolean

Entorno de scripting de XMLSpy: VBScript:

```
Function On_AuthenticMouseEvent(nXPos, nYPos, eMouseEvent, objRange)
    ' On_AuthenticMouseEvent = True ' para cancelar la propagación del evento
End Function
```

Entorno de scripting de XMLSpy: JScript:

```
function On_AuthenticMouseEvent(nXPos, nYPos, eMouseEvent, objRange)
{
    // devolver false; /* para cancelar la propagación del evento */
}
```

Complemento de XMLSpy para entornos IDE:

```
IXMLSpyPlugIn.OnEvent (31, ...) // nEventId = 31
```

Descripción

Este evento se desencadena cada vez que se mueve el ratón y con cada mensaje de Windows para los botones del ratón.

El tipo de mensaje y el estado del botón del ratón está disponible en el parámetro *eMouseEvent*. Use las máscaras de bits definidas en el tipo de datos de enumeración *SPYMouseEvent* para probar los diferentes mensajes, el estado del botón y sus combinaciones.

El parámetro *objRange* identifica la parte del documento donde se encuentra el cursor del ratón. Los objetos de intervalo siempre seleccionan una etiqueta entera del documento. (Esto puede cambiar en futuras versiones del producto que ofrezcan un mecanismo de posicionamiento más preciso.) Si en la posición actual del cursor no se encuentra una parte del documento que se pueda seleccionar, el objeto de intervalo es *null*.

NOTA: estos eventos del entorno de scripting y del complemento de Altova XMLSpy para entornos IDE son compatibles pero se vuelven obsoletos con este evento:

```
On_AuthenticMouseMove()           IXMLSpyPlugIn.OnEvent (15, ...) //
nEventId = 15
On_AuthenticButtonUp()           IXMLSpyPlugIn.OnEvent (16, ...) // nEventId =
16
On_AuthenticButtonDown()         IXMLSpyPlugIn.OnEvent (17, ...) //
nEventId = 17
On_AuthenticButtonDoubleClick()  IXMLSpyPlugIn.OnEvent (24, ...) // nEventId =
24
```

Ejemplos

```
' -----
' fragmento de código VB: conectarse a eventos a nivel de objeto
' -----
' acceder a XMLSpy (sin buscar errores)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' esta es la rutina de devolución de llamada
' conectada al evento OnMouseEvent del objeto objView. Si hace clic con el botón
izquierdo del ratón
' mientras pulsa la tecla Ctrl, la selección actual se asignará
' a la etiqueta en la que está el cursor
Private Function objView_OnMouseEvent(ByVal i_nXPos As Long, ByVal i_nYPos As Long, ByVal
i_eMouseEvent As XMLSpyLib.SPYMouseEvent, ByVal i_pRange As XMLSpyLib.IAuthenticRange) As
Boolean
    If (i_eMouseEvent = (XMLSpyLib.spyLeftButtonDownMask Or
XMLSpyLib.spyCtrlKeyDownMask)) Then
        On Error Resume Next
        i_pRange.Select
        objView_OnMouseEvent = True
    Else
        objView_OnMouseEvent = False
    End If
End Function

' use la palabra clave de VBA WithEvents para conectarse al evento a nivel de objeto
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continuar con más código ...
' y servir el bucle de mensajes
```

4.2.15.1.11 OnSelectionChanged

Evento: `OnSelectionChanged` (`objNewSelection` as AuthenticRange)

Entorno de scripting de XMLSpy: VBScript:

```
Function On_AuthenticSelectionChanged (objNewSelection)
End Function
```

Entorno de scripting de XMLSpy: JScript:

```
function On_AuthenticSelectionChanged (objNewSelection)
{
}
```

Complemento de XMLSpy para entornos IDE:

```
IXMLSpyPlugIn.OnEvent (23, ...) // nEventId = 23
```

Descripción

Este evento se desencadena cada vez que cambia la selección en la interfaz del usuario.

Ejemplos

```
' -----
' fragmento de código VB: conectarse a eventos a nivel de objeto
' -----
' acceder a XMLSpy (sin buscar errores)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' esta es la rutina de devolución de llamada
' conectada al evento OnSelectionChanged del objeto objView
Private Sub objView_OnSelectionChanged (ByVal i_ipNewRange As XMLSpyLib.IAuthenticRange)
    MsgBox ("new selection: " & i_ipNewRange.Text)
End Sub

' use la palabra clave de VBA WithEvents para conectarse al evento a nivel de objeto
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continuar con más código ...
' y servir el bucle de mensajes
```

4.2.15.1.12 OnToolBarButtonClicked

Evento: `OnToolBarButtonClicked` (identificador de botón)

Descripción: `OnToolBarButtonClicked` se desencadena cuando el usuario hace clic en un botón de una barra de herramientas. El parámetro identificador del botón ayuda a determinar en qué botón se hizo clic. A continuación puede consultar la lista predefinida de identificadores de botón:

- 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
- AuthenticTablePropiedades
- AuthenticAppendRow
- AuthenticInsertRow
- AuthenticDuplicateRow
- AuthenticMoveRowUp
- AuthenticMoveRowDown
- AuthenticDeleteRow
- AuthenticDefineEntities

Además el usuario puede añadir identificadores para sus propios botones. Es necesario tener cuidado al añadir identificadores porque no se comprueba si son únicos. Se pueden usar los mismos identificadores para identificar los botones de las llamadas `Set/GetToolbarState()` de la API de COM. Si añade código para

cada botón, el usuario tiene la posibilidad de redefinir completamente el comportamiento de la barra de herramientas VistaAuthentic, añadiendo métodos propios para la manipulación de tables, etc.

4.2.15.1.13 OnToolBarButtonExecuted

Evento: `OnToolBarButtonExecuted` (identificador de botón)

Descripción: `OnToolBarButtonClicked` se desencadena cuando el usuario hace clic en un botón de una barra de herramientas. El parámetro `identificador de botones` ayuda a determinar en qué botón se hizo clic. [Consulte aquí](#)¹⁴⁸ la lista predefinida de identificadores de botón.

`OnToolBarButtonExecuted` se desencadena después de que se ejecuta una acción de barra de herramientas. Por ejemplo, es muy útil a la hora de añadir código actualizado, como en el ejemplo siguiente:

```
//el evento se desencadena cuando se ejecuta una acción de un botón de una barra de
herramientas
function On_AuthenticToolBarButtonExecuted( varBtnIdentifier )
{
    UpdateOwnToolBarButtonStates();
}
```

En este caso `UpdateOwnToolBarButtonStates` es una función del usuario definida en las declaraciones globales.

4.2.15.1.14 OnUserAddedXMLNode

Evento: `OnUserAddedXMLNode` (nodo XML)

Descripción: `OnUserAddedXMLNode` se desencadenará cuando el usuario añada un nodo XML como acción principal. Esto ocurre cuando el usuario hace clic en:

- hipervínculos de adición automática.
- en los comandos **Insertar...**, **Insertar después de...**, **Insertar antes de...** de los menús contextuales.
- en los botones **Anexar fila** e **Insertar fila** de la barra de herramientas.
- en las acciones **Insertar después de...**, **Insertar antes de...** del ayudante de entrada de un elemento (fuera de `StyleVision`).

El evento no se desencadena con el comando **Duplicar fila**, ni cuando se añade el nodo desde fuera de la aplicación (por ejemplo, desde la API de COM), ni con el comando **Aplicar** (por ejemplo, con los iconos de estado del texto), ni cuando se realizan operaciones de tabla XML o de BD.

El parámetro del evento es el objeto nodo XML que se añadió dando el usuario la oportunidad de manipular el nodo XML añadido. En el archivo `OnUserAddedXMLNode.sps` puede encontrar un ejemplo avanzado de un controlador de evento (en la carpeta `Authentic/Scripting` del proyecto `Examples` de la ventana Proyecto).

4.2.15.2 AuthenticView.Application

Propiedad: [Application](#) as [Authentic](#)⁶⁶ (solo lectura)

Descripción

Acceder al objeto de aplicación de .

Errores

- 2000 El objeto `AuthenticView` ya no es válido.
- 2005 Se indicó una dirección no válida para el parámetro de devolución.

4.2.15.3 AuthenticView.AsXMLString

Propiedad: [AsXMLString](#) as `String`

Descripción

Devuelve o configura el contenido del documento como cadena XML. Al asignarse un nuevo valor al contenido no cambia el archivo de esquema ni el archivo SPS. Si el nuevo `XMLString` no cumple con el archivo de esquema, se devuelve el error 2011.

Errores

- 2000 El objeto `AuthenticView` ya no es válido.
- 2011 `AsXMLString` recibe un valor que no es XML válido según el archivo de esquema actual.

4.2.15.4 AuthenticView.ContextMenu

Propiedad: [ContextMenu](#) as `ContextMenu`

Descripción

La propiedad `ContextMenu` permite personalizar el menú contextual. El mejor sitio para hacerlo es el controlador de eventos `OnContextMenuActivated`.

Errores

- 2000 Objeto no válido.
- 2005 Parámetro no válido.

4.2.15.5 AuthenticView.CreateXMLNode

Método: [CreateXMLNode](#) (`nKind` as `SPYXMLDataKind`) as `XMLData`

Valor devuelto

El método devuelve el nuevo objeto [XMLData](#)¹⁶⁸.

Descripción

Para crear un objeto nuevo `XMLData` utilice el método `CreateXMLNode()`.

Errores

- 2000 Objeto no válido.
- 2012 No se puede crear el nodo XML.

4.2.15.6 `AuthenticView.DisableAttributeEntryHelper`

Método: `DisableAttributeEntryHelper()`

Descripción

`DisableAttributeEntryHelper()` deshabilita el ayudante de entrada Atributos de XMLSpy, de Authentic Desktop y del complemento Authentic Browser.

Errores

- 2000 Objeto no válido.

4.2.15.7 `AuthenticView.DisableElementEntryHelper`

Método: `DisableElementEntryHelper()`

Descripción

`DisableElementEntryHelper()` deshabilita el ayudante de entrada Elementos de XMLSpy, de Authentic Desktop y del complemento Authentic Browser.

Errores

- 2000 Objeto no válido.

4.2.15.8 `AuthenticView.DisableEntityEntryHelper`

Método: `DisableEntityEntryHelper()`

Descripción

`DisableEntityEntryHelper()` deshabilita el ayudante de entrada Entidades de XMLSpy, de Authentic Desktop y del complemento Authentic Browser.

Errores

- 2000 Objeto no válido.

4.2.15.9 AuthenticView.DocumentBegin

Propiedad: [DocumentBegin](#) as [AuthenticRange](#)¹⁰⁴ (solo lectura)

Descripción

Recupera un objeto de intervalo que apunta al principio del documento.

Errores

- 2000 El objeto AuthenticView ya no es válido.
- 2005 Se indicó una dirección no válida para el parámetro de devolución.

4.2.15.10 AuthenticView.DocumentEnd

Propiedad: [DocumentEnd](#) as [AuthenticRange](#)¹⁰⁴ (solo lectura)

Descripción

Recupera un objeto de intervalo que apunta al final del documento.

Errores

- 2000 El objeto AuthenticView ya no es válido.
- 2005 Se indicó una dirección no válida para el parámetro de devolución.

4.2.15.11 AuthenticView.DoNotPerformStandardAction

Método: [DoNotPerformStandardAction](#) ()

Descripción

[DoNotPerformStandardAction\(\)](#) sirve de burbuja Cancelar para macros. Detiene la ejecución una vez finalizada la macro.

Errores

- 2000 Objeto no válido.

4.2.15.12 AuthenticView.EvaluateXPath

Método: [EvaluateXPath](#) ([XMLData](#)¹⁶⁸, string expression) as string

Valor devuelto

El método devuelve una cadena de texto.

Descripción

[EvaluateXPath\(\)](#) ejecuta una expresión XPath con el nodo de contexto XML dado. El resultado se devuelve en forma de cadena de texto. Si se trata de una secuencia, se devuelve una cadena separada por espacios.

Errores

- 2000 Objeto no válido.
- 2005 Parámetro no válido.
- 2008 Error interno.
- 2013 Error de XPath.

4.2.15.13 AuthenticView.Event

Propiedad: `Event` as `AuthenticEvent` (solo lectura)

Descripción

Esta propiedad ofrece acceso a los parámetros del último evento, al igual que `OldAuthenticView.event`. Como todos los eventos para el entorno de scripting y clientes externos ahora tienen parámetros, esta propiedad solo se debería usar desde los complementos para entornos IDE.

Errores

- 2000 El objeto `AuthenticView` ya no es válido.
- 2005 Se indicó una dirección no válida para el parámetro de devolución.

4.2.15.14 AuthenticView.EventContext

Propiedad: `EventContext` as `EventContext`

Descripción

`EventContext` ofrece acceso al contexto de las macros que están en ejecución. Consulte la descripción de la interfaz `EventContext` para obtener más información.

Errores

- 2000 Objeto no válido.

4.2.15.15 AuthenticView.GetToolBarButtonState

Método: `GetToolBarButtonState` (`ButtonIdentifier` as `string`) as `AuthenticToolBarButtonState`

Valor devuelto

El método devuelve `AuthenticToolBarButtonState`

Descripción

`GetToolBarButtonState` consulta el estado de un botón de una barra de herramientas y permite al usuario deshabilitarlo o habilitarlo. El botón se identifica por medio de su identificador ([consulte esta lista](#)¹⁴⁸). Por ejemplo, puede usarlo para deshabilitar los botones de las barras de herramientas permanentemente. O puede poner `SetToolBarButtonState` en el controlador de eventos `OnSelectionChanged`, porque los botones de las barras de herramientas se actualizan regularmente cuando en el documento cambia la selección.

Los estados de los botones de las barras de herramientas están en las [enumeraciones](#)¹⁸².

El estado predeterminado significa que la habilitación/deshabilitación del botón está regida por `AuthenticView`. Cuando el usuario configura el estado del botón para habilitarlo/deshabilitarlo, el botón sigue teniendo ese estado hasta que el usuario vuelva a cambiarlo.

Errores

- 2000 Objeto no válido.
- 2005 Parámetro no válido.
- 2008 Error interno.
- 2014 Identificador de botón no válido

4.2.15.16 `AuthenticView.Goto`

Método: `Goto` (*eKind* as `SPYAuthenticElementKind`, *nCount* as `Long`, *eFrom* as `SPYAuthenticDocumentPosition`) as [AuthenticRange](#)¹⁰⁴

Descripción

Recupera un objeto de intervalo que apunta al principio del elemento nº *nCount* de tipo *eKind*. La posición inicial viene definida por el parámetro *eFrom*. Use valores positivo para *nCount* para navegar hasta el final del documento. Por el contrario, para navegar hasta el principio del documento, utilice valores negativos.

Errores

- 2000 El objeto `AuthenticView` ya no es válido.
- 2003 El destino está después del final del documento.
- 2004 El destino está antes del principio del documento.
- 2005 El tipo de elemento indicado no es válido.
La posición desde don se debe comenzar no es `spyAuthenticDocumentBegin` ni `spyAuthenticDocumentEnd`.
Se indicó una dirección no válida para el parámetro de devolución.

Ejemplos

```
Dim objAuthenticView
Set objAuthenticView = objPlugin.AuthenticView

On Error Resume Next
Dim objRange
' ir al principio de la primera tabla del documento
Set objRange = objAuthenticView.Goto (spyAuthenticTable, 1, spyAuthenticDocumentBegin)
If (Err.number = 0) Then
    objRange.Select()
Else
    MsgBox "No se encontró ninguna tabla en el documento"
End If
```

4.2.15.17 `AuthenticView.IsRedoEnabled`

Propiedad: `IsRedoEnabled` as Boolean (solo lectura)

Descripción

True si se puede rehacer la acción y [Redo](#) es posible.

Errores

- 2000 El objeto AuthenticView ya no es válido.
- 2005 Se indicó una dirección no válida para el parámetro de devolución.

4.2.15.18 AuthenticView.IsUndoEnabled

Propiedad: [IsUndoEnabled](#) as Boolean (solo lectura)

Descripción

True si se puede deshacer la acción y [Undo](#) es posible.

Errores

- 2000 El objeto AuthenticView ya no es válido.
- 2005 Se indicó una dirección no válida para el parámetro de devolución.

4.2.15.19 AuthenticView.MarkupVisibility

Propiedad: [MarkupVisibility](#) as [SPYAuthenticMarkupVisibility](#)

Descripción

Establecer u obtener la visibilidad actual del marcado.

Errores

- 2000 El objeto AuthenticView ya no es válido.
- 2005 Se indicó un valor de enumeración no válido.
Se indicó una dirección no válida para el parámetro de devolución.

4.2.15.20 AuthenticView.Parent

Propiedad: [Parent](#) as [Authentic](#)⁶⁶ (solo lectura)

Descripción

Acceder a la aplicación.

Errores

- 2000 El objeto AuthenticView ya no es válido.
- 2005 Se indicó una dirección no válida para el parámetro de devolución.

4.2.15.21 AuthenticView.Print

Método: `Print` (*bWithPreview* as `Boolean`, *bPromptUser* as `Boolean`)

Descripción

Imprimir el documento que aparece en la vista. Si *bWithPreview* recibe el valor `True`, aparece el cuadro de diálogo de vista previa de impresión. Si *bPromptUser* recibe el valor `True`, aparece el cuadro de diálogo "Imprimir". Si ambos parámetros reciben el valor `False`, el documento se imprime directamente.

Errores

2000 El objeto `AuthenticView` ya no es válido.

4.2.15.22 AuthenticView.Redo

Método: `Redo` () as `Boolean`

Descripción

Rehacer la modificación deshecha con el último comando `Deshacer`.

Errores

2000 El objeto `AuthenticView` ya no es válido.

2005 Se indicó una dirección no válida para el parámetro de devolución.

4.2.15.23 AuthenticView.Selection

Propiedad: `Selection` as `AuthenticRange`¹⁰⁴

Descripción

Establecer u obtener la selección de texto actual de la interfaz del usuario.

Errores

2000 El objeto `AuthenticView` ya no es válido.

2002 No hay activa ninguna selección de texto.

2005 Se indicó una dirección no válida para el parámetro de devolución.

Ejemplos

```
'-----  
'                               VBScript  
'-----  
  
Dim objAuthenticView  
Set objAuthenticView = objPlugin.AuthenticView  
  
' si estamos al final del documento, volver a comenzar desde el principio  
If (objAuthenticView.Selection.IsEqual(objAuthenticView.DocumentEnd)) Then  
    objAuthenticView.Selection = objAuthenticView.DocumentBegin  
Else  
    ' objAuthenticView.Selection = objAuthenticView.Selection.GotoNextCursorPosition()
```

```
' o más breve:  
    objAuthenticView.Selection.GotoNextCursorPosition().Select  
End If
```

4.2.15.24 AuthenticView.SetToolBarButtonState

Método: `SetToolBarButtonState` (ButtonIdentifier as string, AuthenticToolBarButtonState state)

Descripción

`SetToolBarButtonState` consulta el estado de un botón de una barra de herramientas y permite al usuario deshabilitarlo o habilitarlo. El botón se identifica por medio de su identificador ([consulte esta lista](#)¹⁴⁸). Por ejemplo, puede usarlo para deshabilitar los botones de las barras de herramientas permanentemente. O puede poner `SetToolBarButtonState` en el controlador de eventos `OnSelectionChanged`, porque los botones de las barras de herramientas se actualizan regularmente cuando en el documento cambia la selección.

Los estados de los botones de las barras de herramientas están en las [enumeraciones](#)¹⁸².

El estado predeterminado significa que la habilitación/deshabilitación del botón está regida por `AuthenticView`. Cuando el usuario configura el estado del botón para habilitarlo/deshabilitarlo, el botón sigue teniendo ese estado hasta que el usuario vuelva a cambiarlo.

Errores

- 2000 Objeto no válido.
- 2008 Error interno.
- 2014 Identificador de botón no válido

4.2.15.25 AuthenticView.Undo

Método: `Undo` () as Boolean

Descripción

Deshacer la última modificación del documento desde dentro de esta vista.

Errores

- 2000 El objeto `AuthenticView` ya no es válido.
- 2005 Se indicó una dirección no válida para el parámetro de devolución.

4.2.15.26 AuthenticView.UpdateXMLInstanceEntities

Método: `UpdateXMLInstanceEntities` ()

Descripción

Actualiza la representación interna de las entidades declaradas y vuelve a rellenar el ayudante de entrada. Además, el validador se vuelve a cargar, permitiendo validar correctamente el archivo XML. Recuerde que esto también puede hacer que se vuelvan a cargar los archivos de esquema.

Errores

El método no devuelve nunca un error.

Ejemplo

```
// -----  
//           JavaScript  
// -----  
var objDocType;  
objDocType = objPlugin.XMLRoot.GetFirstChild(10);  
  
if(objDocType)  
{  
    var objEntity = objPlugin.CreateChild(14);  
    objEntity.Name = "child";  
    objEntity.TextValue = "SYSTEM \"child.xml\"";  
    objDocType.AppendChild(objEntity);  
  
    objPlugin.AuthenticView.UpdateXMLInstanceEntities();  
}
```

4.2.15.27 AuthenticView.WholeDocument

Propiedad: [WholeDocument](#) as [AuthenticRange](#)¹⁰⁴ (solo lectura)

Descripción

Recupera un objeto de intervalo que selecciona todo el documento.

Errores

- 2000 El objeto AuthenticView ya no es válido.
- 2005 Se indicó una dirección no válida para el parámetro de devolución.

4.2.15.28 AuthenticView.XMLDataRoot

Propiedad: [XMLDataRoot](#) as XMLData (solo lectura)

Descripción

Devuelve o establece el elemento de nivel superior XMLData del documento actual. Este elemento por lo general describe la estructura del documento y sería de tipo spyXMLDataXMLDocStruct, spyXMLDataXMLEntityDocStruct o spyXMLDataDTDDocStruct.

Errores

- 2000 El objeto AuthenticView ya no es válido.
- 2005 Se indicó una dirección no válida para el parámetro de devolución.

4.2.16 AuthenticXMLTableCommands

Métodos

[Insert](#) ¹⁶³
[Delete](#) ¹⁶²
[AppendRow](#) ¹⁶²
[InsertRow](#) ¹⁶³
[DeleteRow](#) ¹⁶³
[AppendColumn](#) ¹⁶²
[InsertColumn](#) ¹⁶³
[DeleteColumn](#) ¹⁶²
[JoinLeft](#) ¹⁶⁴
[JoinRight](#) ¹⁶⁴
[JoinUp](#) ¹⁶⁴
[JoinDown](#) ¹⁶⁴
[SplitHorizontal](#) ¹⁶⁸
[SplitVertical](#) ¹⁶⁸
[AlignVerticalTop](#) ¹⁶²
[AlignVerticalCenter](#) ¹⁶²
[AlignVerticalBottom](#) ¹⁶¹
[AlignHorizontalLeft](#) ¹⁶¹
[AlignHorizontalRight](#) ¹⁶¹
[AlignHorizontalCenter](#) ¹⁶¹
[AlignHorizontalJustify](#) ¹⁶¹
[EditPropiedades](#) ¹⁶³

Propiedades

[MayInsert](#) ¹⁶⁶
[MayDelete](#) ¹⁶⁵
[MayAppendRow](#) ¹⁶⁵
[MayInsertRow](#) ¹⁶⁶
[MayDeleteRow](#) ¹⁶⁵
[MayAppendCol](#) ¹⁶⁵
[MayInsertCol](#) ¹⁶⁶
[MayDeleteCol](#) ¹⁶⁵
[MayJoinLeft](#) ¹⁶⁷
[MayJoinRight](#) ¹⁶⁷
[MayJoinUp](#) ¹⁶⁷
[MayJoinDown](#) ¹⁶⁶
[MaySplitHorizontal](#) ¹⁶⁷
[MaySplitVertical](#) ¹⁶⁷
[MayAlignVertical](#) ¹⁶⁴
[MayAlignHorizontal](#) ¹⁶⁴
[MayEditPropiedades](#) ¹⁶⁶

4.2.16.1 AuthenticXMLTableCommands.AlignHorizontalCenter

Declaración [AlignHorizontalCenter\(\)](#)

Descripción

Asigna el valor `center` al atributo de alineación horizontal o borra el atributo si ya tenía asignado el valor `center`.

4.2.16.2 AuthenticXMLTableCommands.AlignHorizontalJustify

Declaración [AlignHorizontalJustify\(\)](#)

Descripción

Asigna el valor `justify` al atributo de alineación horizontal o borra el atributo si ya tenía asignado el valor `justify`.

4.2.16.3 AuthenticXMLTableCommands.AlignHorizontalLeft

Declaración [AlignHorizontalLeft\(\)](#)

Descripción

Asigna el valor `left` al atributo de alineación horizontal o borra el atributo si ya tenía asignado el valor `left`.

4.2.16.4 AuthenticXMLTableCommands.AlignHorizontalRight

Declaración [AlignHorizontalRight\(\)](#)

Descripción

Asigna el valor `right` al atributo de alineación horizontal o borra el atributo si ya tenía asignado el valor `right`.

4.2.16.5 AuthenticXMLTableCommands.AlignVerticalBottom

Declaración [AlignVerticalBottom\(\)](#)

Descripción

Asigna el valor `bottom` al atributo de alineación vertical o borra el atributo si ya tenía asignado el valor `bottom`.

4.2.16.6 AuthenticXMLTableCommands.AlignVerticalCenter

Declaración [AlignVerticalCenter\(\)](#)

Descripción

Asigna el valor `center` al atributo de alineación vertical o borra el atributo si ya tenía asignado el valor `center`.

4.2.16.7 AuthenticXMLTableCommands.AlignVerticalTop

Declaración [AlignVerticalTop\(\)](#)

Descripción

Asigna el valor `top` al atributo de alineación vertical o borra el atributo si ya tenía asignado el valor `top`.

4.2.16.8 AuthenticXMLTableCommands.AppendCol

Declaración [AppendCol\(\)](#)

Descripción

Anexa una columna a la tabla actual.

4.2.16.9 AuthenticXMLTableCommands.AppendRow

Declaración [AppendRow\(\)](#)

Descripción

Anexa una fila a la tabla actual.

4.2.16.10 AuthenticXMLTableCommands.Delete

Declaración [Delete\(\)](#)

Descripción

Si el usuario confirmó el cuadro de diálogo, el método elimina la tabla que está seleccionada.

4.2.16.11 AuthenticXMLTableCommands.DeleteCol

Declaración [DeleteCol\(\)](#)

Descripción

El método elimina la columna que está seleccionada.

Si solamente queda una columna y el usuario confirma el cuadro de diálogo, se elimina la tabla entera.

4.2.16.12 AuthenticXMLTableCommands.DeleteRow

Declaración [DeleteRow\(\)](#)**Descripción**

El método elimina la fila que está seleccionada.

Si solamente queda una fila y el usuario confirma el cuadro de diálogo, se elimina la tabla entera.

4.2.16.13 AuthenticXMLTableCommands.EditProperties

Declaración [EditPropiedades\(\)](#)**Descripción**

Muestra el cuadro de diálogo "Propiedades" de la celda/tabla seleccionada.

4.2.16.14 AuthenticXMLTableCommands.Insert

Declaración [Insert\(\)](#)**Descripción**

Muestra un cuadro de diálogo e inserta una tabla nueva en el XML actual.

4.2.16.15 AuthenticXMLTableCommands.InsertCol

Declaración [InsertCol\(\)](#)**Descripción**

Inserta una columna nueva antes de la columna que está seleccionada.

4.2.16.16 AuthenticXMLTableCommands.InsertRow

Declaración [InsertRow\(\)](#)**Descripción**

Inserta una fila nueva antes de la fila que está seleccionada.

4.2.16.17 AuthenticXMLTableCommands.JoinDown

Declaración [JoinDown\(\)](#)

Descripción

Combina la celda actual con la celda situada justo debajo.

4.2.16.18 AuthenticXMLTableCommands.JoinLeft

Declaración [JoinLeft\(\)](#)

Descripción

Combina la celda actual con la celda situada a su izquierda.

4.2.16.19 AuthenticXMLTableCommands.JoinRight

Declaración [JoinRight\(\)](#)

Descripción

Combina la celda actual con la celda situada a su derecha.

4.2.16.20 AuthenticXMLTableCommands.JoinUp

Declaración [JoinUp\(\)](#)

Descripción

Combina la celda actual con la celda situada justo encima.

4.2.16.21 AuthenticXMLTableCommands.MayAlignHorizontal

Declaración [MayAlignHorizontal](#) as [Boolean](#)

Descripción

True si se puede establecer el valor de los atributos de alineación horizontal para la celda actual.

4.2.16.22 AuthenticXMLTableCommands.MayAlignVertical

Declaración [MayAlignVertical](#) as [Boolean](#)

Descripción

True si se puede establecer el valor de los atributos de alineación vertical para la celda actual.

4.2.16.23 AuthenticXMLTableCommands.MayAppendCol

Declaración `MayAppendCol` as `Boolean`

Descripción

True si se puede anexar una columna.

4.2.16.24 AuthenticXMLTableCommands.MayAppendRow

Declaración `MayAppendRow` as `Boolean`

Descripción

True si se puede anexar una fila.

4.2.16.25 AuthenticXMLTableCommands.MayDelete

Declaración `MayDelete` as `Boolean`

Descripción

True si hay seleccionada una tabla.

4.2.16.26 AuthenticXMLTableCommands.MayDeleteCol

Declaración `MayDeleteCol` as `Boolean`

Descripción

True si la columna actual se puede eliminar.

4.2.16.27 AuthenticXMLTableCommands.MayDeleteRow

Declaración `MayDeleteRow` as `Boolean`

Descripción

True si la fila actual se puede eliminar.

4.2.16.28 AuthenticXMLTableCommands.MayEditProperties

Declaración `MayEditPropiedades` as `Boolean`

Descripción

True si el cuadro de diálogo "Propiedades" está disponible para la celda/tabla seleccionada.

4.2.16.29 AuthenticXMLTableCommands.MayInsert

Declaración `MayInsert` as `Boolean`

Descripción

True si se puede insertar una tabla nueva en la selección actual.

4.2.16.30 AuthenticXMLTableCommands.MayInsertCol

Declaración `MayInsertCol` as `Boolean`

Descripción

True si se puede insertar una columna.

4.2.16.31 AuthenticXMLTableCommands.MayInsertRow

Declaración `MayInsertRow` as `Boolean`

Descripción

True si se puede insertar una fila.

4.2.16.32 AuthenticXMLTableCommands.MayJoinDown

Declaración `MayJoinDown` as `Boolean`

Descripción

True si la celda actual se puede combinar con la celda situada justo debajo.

4.2.16.33 AuthenticXMLTableCommands.MayJoinLeft

Declaración `MayJoinLeft` as `Boolean`

Descripción

True si la celda actual se puede combinar con la celda situada a la izquierda.

4.2.16.34 AuthenticXMLTableCommands.MayJoinRight

Declaración `MayJoinRight` as `Boolean`

Descripción

True si la celda actual se puede combinar con la celda situada a la derecha.

4.2.16.35 AuthenticXMLTableCommands.MayJoinUp

Declaración `MayJoinUp` as `Boolean`

Descripción

True si la celda actual se puede combinar con la celda situada justo encima.

4.2.16.36 AuthenticXMLTableCommands.MaySplitHorizontal

Declaración `MaySplitHorizontal` as `Boolean`

Descripción

True si la celda actual se puede dividir horizontalmente.

4.2.16.37 AuthenticXMLTableCommands.MaySplitVertical

Declaración `MaySplitVertical` as `Boolean`

Descripción

True si la celda actual se puede dividir verticalmente.

4.2.16.38 AuthenticXMLTableCommands.SplitHorizontal

Declaración [SplitHorizontal\(\)](#)

Descripción

El método divide la celda actual horizontalmente.

4.2.16.39 AuthenticXMLTableCommands.SplitVertical

Declaración [SplitVertical\(\)](#)

Descripción

El método divide la celda actual verticalmente.

4.2.17 XMLData

Métodos

[InsertChild](#)¹⁷⁵
[AppendChild](#)¹⁶⁹
[EraseAllChildren](#)¹⁷⁰
[EraseCurrentChild](#)¹⁷⁰
[GetCurrentChild](#)¹⁷²
[GetFirstChild](#)¹⁷³
[GetNextChild](#)¹⁷³
[GetChild](#)¹⁷¹
[GetChildKind](#)¹⁷²
[IsSameNode](#)¹⁷⁶
[HasChildrenKind](#)¹⁷⁴
[CountChildren](#)¹⁶⁹
[CountChildrenKind](#)¹⁶⁹

Propiedades

[Name](#)¹⁷⁶
[TextValue](#)¹⁷⁷
[HasChildren](#)¹⁷⁴
[MayHaveChildren](#)¹⁷⁶
[Kind](#)¹⁷⁶
[Parent](#)¹⁷⁶

Descripción

Puede usar la interfaz XMLData para manipular el contenido del XML que está visible. Esta interfaz es un equivalente COM más sencillo de la implementación utilizada dentro del complemento y de Altova XMLSpy.

Para crear un objeto XMLData nuevo utilice el método CreateChild() de la interfaz del complemento.

4.2.17.1 XMLData.AppendChild

Declaración `AppendChild(pNewData as XMLData168)`

Descripción

`AppendChild` anexa `pNewData` como último elemento secundario en el objeto `XMLData`. Consulte también [Usar XMLData⁵⁷](#).

Ejemplo

```
Dim objCurrentParent
Dim objNewChild

Set objNewChild = objPlugIn.CreateChild(spyXMLDataElement)
Set objCurrentParent = objPlugIn.XMLRoot

objCurrentParent.AppendChild objNewChild

Set objNewChild = Nothing
```

4.2.17.2 XMLData.CountChildren

Declaración `CountChildren` as long

Descripción

`CountChildren` obtiene el número de elementos secundarios.

Disponible con TypeLibrary versión 1.6

Errores

1500 El objeto `XMLData` ya no es válido.

4.2.17.3 XMLData.CountChildrenKind

Declaración `CountChildrenKind (nKind as SPYXMLDataKind182)` as long

Descripción

`CountChildrenKind` obtiene el número de secundarios del tipo indicado.

Disponible con TypeLibrary versión 1.6

Errores

1500 El objeto `XMLData` ya no es válido.

4.2.17.4 XMLData.EraseAllChildren

Declaración [EraseAllChildren](#)

Descripción

EraseAllChildren elimina todos los secundarios asociados del objeto XMLData.

Ejemplo

En el ejemplo se eliminan todos los elementos del documento activo.

```
Dim objCurrentParent

Set objCurrentParent = objPlugIn.XMLRoot
objCurrentParent.EraseAllChildren
```

4.2.17.5 XMLData.EraseChild

Método: [EraseChild](#) (Nodo secundario y nodo nuevo como [XMLData](#)¹⁶⁸)

Descripción

Elimina el nodo secundario dado.

Errores

- 1500 Objeto no válido.
- 1506 XML de entrada no válido
- 1510 Parámetro no válido.

4.2.17.6 XMLData.EraseCurrentChild

Declaración [EraseCurrentChild](#)

Descripción

EraseCurrentChild elimina el objeto secundario XMLData actual. Antes de llamar a EraseCurrentChild debe inicializar un iterador interno con [XMLData.GetFirstChild](#)¹⁷³.

Ejemplo

Este ejemplo de JavaScript elimina todos los elementos llamados "Bórrame". En el código puede ver que se puede llamar a EraseCurrentChild y a GetNextChild dentro del mismo bucle para seguir recorriendo los elementos secundarios.

```
function DeleteXMLElements(objXMLData)
{
    if(objXMLData == null)
        return;

    if(objXMLData.HasChildren) {
```

```
var objChild;
objChild = objXMLData.GetFirstChild(-1);

while(objChild) {
    DeleteXMLElements(objChild);

    try {
        if(objChild.Name == "Bórrame")
            objXMLData.EraseCurrentChild();

        objChild = objXMLData.GetNextChild();
    }
    catch(Err) {
        objChild = null;
    }
}
}
```

4.2.17.7 XMLData.GetChild

Declaración `GetChild` (*position* as long) as [XMLData](#)¹⁶⁸

Valor devuelto

Devuelve un elemento XML como objeto XMLData.

Descripción

`GetChild()` devuelve una referencia al secundario en el índice dado (basado en cero).

Disponible con TypeLibrary versión 1.6

Errores

- 1500 El objeto XMLData ya no es válido.
- 1510 Se indicó una dirección no válida para el parámetro de devolución.

4.2.17.8 XMLData.GetChildAttribute

Método: `GetChildAttribute` (Nombre como cadena) como objeto XMLData (NULL cuando haya error)

Descripción

Recupera el atributo que tiene el nombre dado.

Errores

- 1500 Objeto no válido.
- 1510 Parámetro no válido.

4.2.17.9 XMLData.GetChildElement

Método: `GetChildElement` (Nombre como cadena, posición como entero) como objeto `XMLData` (`NULL` cuando haya error)

Descripción

Recupera el elemento secundario `n` que tiene el nombre dado.

Errores

- 1500 Objeto no válido.
- 1510 Parámetro no válido.

4.2.17.10 XMLData.GetChildKind

Declaración `GetChildKind` (`position` as long, `nKind` as `SPYXMLDataKind`¹⁸²) as `XMLData`¹⁶⁸

Valor devuelto

Devuelve un elemento XML como objeto `XMLData`.

Descripción

`GetChildKind()` devuelve una referencia a un secundario de su tipo en el índice dado (basado en cero). El parámetro de posición es relativo al número de secundarios del tipo indicado y no a todos los secundarios del objeto.

Disponible con TypeLibrary versión 1.6

Errores

- 1500 El objeto `XMLData` ya no es válido.
- 1510 Se indicó una dirección no válida para el parámetro de devolución.

4.2.17.11 XMLData.GetCurrentChild

Declaración `GetCurrentChild` as `XMLData`¹⁶⁸

Valor devuelto

Devuelve un elemento XML como objeto `XMLData`.

Descripción

`GetCurrentChild` obtiene el secundario actual. Antes de llamar a `GetCurrentChild` debe inicializar un iterador interno con `XMLData.GetFirstChild`¹⁷³.

4.2.17.12 XMLData.GetFirstChild

Declaración `GetFirstChild(nKind as SPYXMLDataKind182) as XMLData168`

Valor devuelto

Devuelve un elemento XML como objeto XMLData.

Descripción

GetFirstChild inicializa un iterador nuevo y devuelve el primer secundario. Asigne el valor `nKind = -1` para obtener un iterador para todos los tipos de secundarios.

Ejemplo

Consulte el ejemplo disponible en [XMLData.GetNextChild](#)¹⁷³.

4.2.17.13 XMLData.GetNamespacePrefixForURI

Método: `GetNamespacePrefixForURI` (URI como cadena de texto) Prefijo como cadena de texto

Descripción

Devuelve el prefijo de espacio de nombres del URI dado.

Errores

- 1500 Objeto no válido.
- 1510 Parámetro no válido.

4.2.17.14 XMLData.GetNextChild

Declaración `GetNextChild` as [XMLData](#)¹⁶⁸

Valor devuelto

Devuelve un elemento XML como objeto XMLData.

Descripción

GetNextChild pasa al siguiente secundario de este elemento. Antes de llamar a GetNextChild debe inicializar un iterador interno con [XMLData.GetFirstChild](#)¹⁷³.

Busque el último secundario del elemento tal y como muestra el ejemplo siguiente.

Ejemplo

```
On Error Resume Next
Set objParent = objPlugin.XMLRoot

'obtener elementos de todos los tipos
```

```
Set objCurrentChild = objParent.GetFirstChild(-1)

Do
    'hacer algo útil con el secundario

    'pasar al siguiente secundario
    Set objCurrentChild = objParent.GetNextChild
Loop Until (Err.Number - vbObjectError = 1503)
```

4.2.17.15 XMLData.GetTextValueXMLDecoded

Método: `GetTextValueXMLDecoded` as string

Descripción

Obtiene el valor de texto codificado del XML.

Errores

- 1500 Objeto no válido.
- 1510 Parámetro no válido.

4.2.17.16 XMLData.HasChildren

Declaración `HasChildren` as Boolean

Descripción

La propiedad es true si el objeto es primario de otros objetos XMLData.

Esta propiedad es de solo lectura.

4.2.17.17 XMLData.HasChildrenKind

Declaración `HasChildrenKind` (*nKind* as [SPYXMLDataKind](#)¹⁸²) as Boolean

Descripción

El método devuelve true si el objeto es el primario de otros objetos XMLData del mismo tipo.

Disponible con TypeLibrary versión 1.6

Errores

- 1500 El objeto XMLData ya no es válido.
- 1510 Se indicó una dirección no válida para el parámetro de devolución.

4.2.17.18 XMLData.InsertChild

Declaración `InsertChild(pNewData as XMLData168)`

Descripción

InsertChild inserta el secundario nuevo antes del secundario actual.

Consulte también [XMLData.GetFirstChild¹⁷³](#), [XMLData.GetNextChild¹⁷³](#) para establecer el secundario actual.

4.2.17.19 XMLData.InsertChildAfter

Método: `InsertChildAfter` (Nodo secundario y nodo nuevo como XMLData)

Descripción

Inserta un nodo XML nuevo después del nodo dado.

Errores

- 1500 Objeto no válido.
- 1506 XML de entrada no válido
- 1507 No se admiten elementos secundarios
- 1510 Parámetro no válido.
- 1512 El secundario ya se añadió
- 1514 Tipo no válido en la posición

4.2.17.20 XMLData.InsertChildBefore

Método: `InsertChildBefore` (Nodo secundario y nodo nuevo como XMLData)

Descripción

Inserta un nodo XML nuevo antes del nodo dado.

Errores

- 1500 Objeto no válido.
- 1506 XML de entrada no válido
- 1507 No se admiten elementos secundarios
- 1510 Parámetro no válido.
- 1512 El secundario ya se añadió
- 1514 Tipo no válido en la posición

4.2.17.21 XMLData.IsSameNode

Declaración `IsSameNode(pNodeToCompare as XMLData168)` as `Boolean`

Descripción

Devuelve true si pNodeToCompare hace referencia al mismo nodo que el objeto mismo.

4.2.17.22 XMLData.Kind

Declaración `Kind` as `SPYXMLDataKind`¹⁸²

Descripción

El tipo de este objeto XMLData.

Esta propiedad es de solo lectura.

4.2.17.23 XMLData.MayHaveChildren

Declaración `MayHaveChildren` as `Boolean`

Descripción

Comprueba si se pueden añadir secundarios a este objeto XMLData.

Esta propiedad es de solo lectura.

4.2.17.24 XMLData.Name

Declaración `Name` as `String`

Descripción

Se utiliza para modificar y obtener el nombre del objeto XMLData.

4.2.17.25 XMLData.Parent

Declaración `Parent` as `XMLData`¹⁶⁸

Valor devuelto

El primario como como objeto XMLData.

Nada (o NULL) si no hay elemento primario.

Descripción

Elemento primario de este elemento.

Esta propiedad es de solo lectura.

4.2.17.26 XMLData.SetTextValueXMLDecoded

Método: [SetTextValueXMLDecoded](#) (valor de cadena)

Descripción

Establece el valor de texto codificado del XML.

Errores

- 1500 Objeto no válido.
- 1513 La modificación no está permitida.

4.2.17.27 XMLData.TextValue

Declaración [TextValue](#) as [String](#)

Descripción

Se utiliza para modificar y para obtener el valor de texto de este objeto XMLData. Consulte también [Usar XMLData](#)⁵⁷.

4.3 Enumeraciones

Esta sección enumera y describe las enumeraciones de Authentic Browser.

4.3.1 SPYAuthenticActions

Descripción

Acciones que se pueden llevar a cabo en objetos [AuthenticRange](#)¹⁰⁴.

Valores posibles:

spyAuthenticInsertAt	= 0
spyAuthenticApply	= 1
spyAuthenticClearSurr	= 2
spyAuthenticAppend	= 3
spyAuthenticInsertBefore	= 4
spyAuthenticRemove	= 5

4.3.2 SPYAuthenticCommand

Descripción

Enumeración de todos los comandos disponibles

Valores posibles:

// CommandGroupMain

k_CommandSeparator	= 0
k_CommandSave	= 1
k_CommandPrint	= 2
k_CommandPrintPreview	= 3
k_CommandValidate	= 4
k_CommandUndo	= 5
k_CommandRedo	= 6

// CommandGroupEdit

k_CommandEditCut	= 7
k_CommandEditCopy	= 8
k_CommandEditPaste	= 9
k_CommandEditFind	= 10

k_CommandEditRepeat = 11
k_CommandEditReplace = 12

// CommandGroupMarkup

k_CommandMarkupHide = 13
k_CommandMarkupLarge = 14

// CommandGroupRow

k_CommandRowAppend = 15
k_CommandRowInsert = 16
k_CommandRowDuplicate = 17
k_CommandRowMoveUp = 18
k_CommandRowMoveDown = 19
k_CommandRowDelete = 20

// CommandGroupXMLTables

k_CommandXMLTableInsert = 21
k_CommandXMLTableDelete = 22
k_CommandXMLTableAppendRow = 23
k_CommandXMLTableInsertRow = 24
k_CommandXMLTableDeleteRow = 25
k_CommandXMLTableAppendCol = 26
k_CommandXMLTableInsertCol = 27
k_CommandXMLTableDeleteCol = 28
k_CommandXMLTableJoinRight = 29
k_CommandXMLTableJoinLeft = 30
k_CommandXMLTableJoinUp = 31
k_CommandXMLTableJoinDown = 32
k_CommandXMLTableSplitHorz = 33
k_CommandXMLTableSplitVert = 34
k_CommandXMLTableVAlignTop = 35
k_CommandXMLTableVAlignCenter = 36
k_CommandXMLTableVAlignBottom = 37
k_CommandXMLTableAlignLeft = 38
k_CommandXMLTableAlignCenter = 39
k_CommandXMLTableAlignRight = 40
k_CommandXMLTableAlignJustify = 41
k_CommandXMLTableEditProperties = 42

```
// since TypeLib 1.2
k_CommandCheckSpelling      = 43
k_CommandAbout              = 44
k_CommandPackageManagement = 45
```

4.3.3 SPYAuthenticCommandGroup

Descripción

Grupos a los que puede pertenecer un comando.

Valores posibles:

```
k_CommandGroupMain      = 0
k_CommandGroupEdit      = 1
k_CommandGroupMarkup    = 2
k_CommandGroupRow       = 3
k_CommandGroupXMLTables = 4
```

4.3.4 SPYAuthenticDocumentPosition

Descripción

Posiciones relativas y absolutas utilizadas para navegar con objetos [AuthenticRange](#)¹⁰⁴.

Valores posibles:

```
spyAuthenticDocumentBegin = 0
spyAuthenticDocumentEnd   = 1
spyAuthenticRangeBegin    = 2
spyAuthenticRangeEnd      = 3
```

4.3.5 SPYAuthenticElementActions

Descripción

Acciones que se pueden usar con [GetAllowedElements](#)⁷⁷.

Valores posibles:

```
k_ActionInsertAt      = 0
k_ActionApply         = 1
k_ActionClearSurr     = 2
k_ActionAppend        = 3
```

k_ActionInsertBefore = 4
k_ActionRemove = 5

4.3.6 SPYAuthenticElementKind

Descripción

Enumeración de los diferentes tipos de elemento utilizados para navegación y selección dentro de objetos [AuthenticRange](#)¹⁰⁴ y [AuthenticView](#)¹³⁹.

Valores posibles:

spyAuthenticChar = 0
spyAuthenticWord = 1
spyAuthenticLine = 3
spyAuthenticParagraph = 4
spyAuthenticTag = 6
spyAuthenticDocument = 8
spyAuthenticTable = 9
spyAuthenticTableRow = 10
spyAuthenticTableColumn = 11

4.3.7 SPYAuthenticEntryHelperWindows

Descripción

Identificación de las ventanas de ayudantes de entrada disponibles.

Valores posibles:

k_Elements = 1
k_Attributes = 2
k_Entities = 4

4.3.8 SPYAuthenticMarkupVisibility

Descripción

Valores de enumeración para personalizar la visibilidad del marcado.

Valores posibles:

spyAuthenticMarkupHidden = 0
spyAuthenticMarkupSmall = 1

spyAuthenticMarkupLarge = 2
 spyAuthenticMarkupMixed = 3

4.3.9 SPYAuthenticToolBarAlignment

Descripción

Valores para especificar la alineación de la barra de herramientas.

Valores posibles:

k_ToolbarAlignTop = 0
 k_ToolbarAlignLeft = 1
 k_ToolbarAlignBottom = 2
 k_ToolbarAlignRight = 3

4.3.10 SPYAuthenticToolBarButtonState

Descripción

Estados de los botones de la barra de herramientas de Authentic:

Valores posibles:

authenticToolBarButtonDefault = 0
 authenticToolBarButtonEnabled = 1
 authenticToolBarButtonDisabled = 2

4.3.11 SPYXMLDataKind

Descripción

Los diferentes tipos de elementos XMLData disponibles para documentos XML.

Valores posibles:

spyXMLDataXMLDocStruct = 0
 spyXMLDataXMLEntityDocStruct = 1
 spyXMLDataDTDDocStruct = 2
 spyXMLDataXML = 3
 spyXMLDataElement = 4
 spyXMLDataAttr = 5
 spyXMLDataText = 6
 spyXMLDataCDATA = 7
 spyXMLDataComment = 8

spyXMLDataP	= 9
spyXMLDataDefDoctype	= 10
spyXMLDataDefExternalID	= 11
spyXMLDataDefElement	= 12
spyXMLDataDefAttlist	= 13
spyXMLDataDefEntity	= 14
spyXMLDataDefNotation	= 15
spyXMLDataKindsCount	= 16

5 Información sobre licencias

En esta sección encontrará información sobre:

- 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](#).

5.1 **Altova Authentic: contrato de licencia para el usuario final**

- Encontrará el Contrato de licencia de Altova para el usuario final de Authentic (en inglés) en: <https://www.altova.com/es/legal/authentic-eula>
- Encontrará la Política de privacidad de Altova en: <https://www.altova.com/es/privacy>

Índice

A

Archivo CAB, 10, 13, 36

Archivo XPI, 10, 13, 36

Archivos SPS basados en BD,

requisitos para trabajar con Authentic Browser, 18

Atributo CODEBASE, 22

Authentic, 85

- ApplyTextState, 68
- attachCallBack, 68
- ControlInitialized, 71
- CreateChild, 71
- CurrentSelection, 71
- DesignDataLoadObject, 72
- EditClear, 72
- EditCopy, 72
- EditCut, 73
- EditPaste, 73
- EditRedo, 73
- EditSelectAll, 73
- EditUndo, 73
- event, 75
- FindDialog, 75
- FindNext, 75
- GetAllAttributes, 75
- GetAllowedElements, 77
- GetFileVersion, 78
- GetNextVisible, 79
- GetPreviousVisible, 79
- IsEditClearEnabled, 79
- IsEditCopyEnabled, 79
- IsEditCutEnabled, 79
- IsEditPasteEnabled, 80
- IsEditRedoEnabled, 80
- IsEditUndoEnabled, 80
- IsFindNextEnabled, 80
- IsRowAppendEnabled, 80
- IsRowDeleteEnabled, 81
- IsRowDuplicateEnabled, 81
- IsRowInsertEnabled, 81
- IsRowMoveDownEnabled, 81
- IsRowMoveUpEnabled, 82

- IsTextStateApplied, 82
- IsTextStateEnabled, 82
- LoadXML, 82
- MarkupView, 82
- Print, 83
- PrintPreview, 83
- ReplaceDialog, 84
- Reset, 84
- RowAppend, 84
- RowDelete, 84
- RowInsert, 85
- RowMoveDown, 85
- RowMoveUp, 85
- Save, 85
- SavePOST, 86
- SaveXML, 87
- SchemaLoadObject, 87
- SelectionChanged, 87
- SelectionMoveTabOrder, 88
- SelectionSet, 88
- StartEditing, 88
- ValidateDocument, 90
- validationBadData, 91
- validationMessage, 91
- XMLDataLoadObject, 91
- XMLDataSaveUrl, 91
- XMLRoot, 92

Authentic Browser, 66

- archivo del complemento, 10, 13, 36
- configuración de red, 8
- control de eventos, 26
- descarga de archivo en servidor, 10, 13, 36
- IDs de clase, 10, 22
- resumen, 8
- subrutinas, 26
- Tipos MIME, 10
- ventajas, 7
- versión, 22
- versiones, 10
- y archivos SPS basados en BD, 18

Authentic Row Duplicate, 85

AuthenticDataTransfer,

- dropEffect, 96
- getData, 96
- ownDrag, 96
- type, 96

AuthenticEvent,

AuthenticEvent,

- altKey, 97
- altLeft, 97
- button, 97
- cancelBubble, 98
- clientX, 98
- clientY, 98
- ctrlKey, 98
- ctrlLeft, 99
- dataTransfer, 99
- fromElement, 99
- keyCode, 99
- propertyName, 99
- repeat, 100
- returnValue, 100
- shiftKey, 100
- shiftLeft, 100
- srcElement, 100
- type, 101

AuthenticRange, 104

- AppendRow, 106
- Application, 107
- CanPerformAction, 107
- CanPerformActionWith, 107
- Close, 108
- CollapsToBegin, 108
- CollapsToEnd, 108
- Copy, 109
- Cut, 109
- Delete, 109
- DeleteRow, 110
- DuplicateRow, 110
- ExpandTo, 111
- FirstTextPosition, 111
- FirstXMLData, 112
- FirstXMLDataOffset, 113
- GetElementAttributeNames, 114
- GetElementAttributeValue, 115
- GetElementHierarchy, 115
- GetEntityNames, 116
- Goto, 116
- GotoNext, 117
- GotoNextCursorPosition, 117
- GotoPrevious, 118
- GotoPreviousCursorPosition, 119
- HasElementAttribute, 119
- InsertEntity, 119

- InsertRow, 120
- IsEmpty, 121
- IsEqual, 121
- IsInDynamicTable, 122
- LastTextPosition, 123
- LastXMLData, 124
- LastXMLDataOffset, 125
- MoveBegin, 126
- MoveEnd, 126
- MoveRowDown, 127
- MoveRowUp, 127
- Parent, 127
- Paste, 128
- PerformAction, 128
- Select, 129
- SelectNext, 129
- SelectPrevious, 130
- SetElementAttributeValue, 131
- SetFromRange, 132
- Text, 133

AuthenticView, 69, 139, 158

- Application, 151
- DocumentBegin, 153
- DocumentEnd, 153
- Goto, 155
- MarkupVisibility, 156
- Parent, 156
- Print, 157
- Redo, 157
- Selection, 157
- Undo, 158
- WholeDocument, 159

Ayudantes de entrada, 51**B****Botones de barras de herramientas,**

- cambiar el comportamiento de, 47

Botones de estado del texto, 51**Buscar y reemplazar, 49****C****Configuración de red, 8**

Configuración del servidor, 13

Contenido,

acceder y modificar, 49

Contenido del documento,

acceder y modificar, 49

Contrato de licencia para el usuario final, 184

Control de eventos, 26

Controladores de eventos, 46, 47

ControlInitialized, 46, 71

Corrección ortográfica, 52

D

Distribución,

de productos de software de Altova, 184

DOM,

y XMLData, 60

E

elemento OBJECT,

en página HTML para IE, 22

elemento SCRIPT,

en página HTML para IE, 26

Eventos,

referencia, 48

I

IDs de clase, 10, 22

Información legal, 184

Información sobre derechos de autor, 184

Internet Information Service para el servidor, 14

L

Licencia,

información sobre, 184

Licencia para la edición Enterprise, 21

O

Objeto AuthenticRange, 49

Objeto Authentic, 49

Objeto AuthenticView, 49

Operaciones de edición, 49

Operaciones en filas de tablas, 50

P

Página HTML,

resumen, 20

Página HTML para IE,

ejemplo básico, 27

ejemplo: ordenar una tabla, 28

Elemento OBJECT, 22

Elemento SCRIPT, 26

resumen, 22, 31

Paquetes, 52

Período de evaluación,

de los productos de software de Altova, 184

Puntos de conexión (eventos), 46

R

Reemplazar, 49

Referencia,

eventos, 48

Referencia del usuario, 45

Requisitos del sistema, 8

S

selectionchanged, 46

Servicio de explorador para el servidor, 14

Subrutinas, 26

T

Tablas dinámicas, 50

Teclas de acceso rápido, 50

Tipos MIME, 10

X

XMLData, 57

GetChild, 171

GetChildKind, 172

HasChildrenKind, 174

IsSameNode, 176

y DOM, 60

XMLSpyLib,

AuthenticDataTransfer, 95

AuthenticEvent, 97

XMLSpyXMLData, 168

XMLSPYPLUGINLib,

Authentic, 66

XMLSpyXMLLoadSave, 104

XMLSpyXMLData,

AppendChild, 169

EraseAllChildren, 170

EraseCurrentChild, 170

GetCurrentChild, 172

GetFirstChild, 173

GetNextChild, 173

HasChildren, 174

InsertChild, 175

Kind, 176

MayHaveChildren, 176

Name, 176

Parent, 176

TextValue, 177

XMLSpyXMLLoadSave,

String, 104

URL, 104