

# Altova StyleVision Server 2024



## Manual del usuario y referencia

# **Altova StyleVision Server 2024**

## **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

<b>1</b>	<b>Introducción</b>	<b>6</b>
<b>2</b>	<b>Funcionamiento</b>	<b>7</b>
2.1	Como módulo de FlowForce Server.....	8
2.2	Como producto servidor independiente.....	9
<b>3</b>	<b>Instalación y asignación de licencias</b>	<b>10</b>
3.1	Instalación y configuración en Windows.....	11
3.1.1	Instalar en Windows.....	11
3.1.2	Instalar en Windows Server Core.....	12
3.1.3	Instalar LicenseServer (Windows).....	14
3.1.4	Asignar licencias a StyleVision Server (Windows).....	15
3.1.5	Notas sobre la configuración (Windows).....	18
3.2	Instalación y configuración en Linux.....	19
3.2.1	Instalar en Linux.....	19
3.2.2	Instalar LicenseServer (Linux).....	21
3.2.3	Asignar licencias a StyleVision Server (Linux).....	22
3.2.4	Notas sobre configuración del entorno (Linux).....	24
3.3	Instalación y configuración en macOS.....	26
3.3.1	Instalar en macOS.....	26
3.3.2	Instalar LicenseServer (macOS).....	27
3.3.3	Asignar licencias a StyleVision Server (macOS).....	28
3.3.4	Notas sobre configuración del entorno (macOS).....	30
3.4	Actualizar StyleVision Server.....	32
3.5	Migrar StyleVision Server a un equipo nuevo.....	33
3.6	Requisitos FOP.....	34

---

<b>4</b>	<b>Línea de comandos de StyleVision Server</b>	<b>35</b>
4.1	assignlicense.....	37
4.2	exportresourcestrings.....	39
4.3	generate.....	41
4.4	help .....	46
4.5	licenseserver.....	47
4.6	pdfdata.....	49
4.7	setdeflang.....	51
4.8	setfopath.....	52
4.9	verifylicense.....	54
4.10	version.....	56
<b>5</b>	<b>API de StyleVision Server</b>	<b>57</b>
5.1	Información sobre la interfaz .NET.....	58
5.2	Información sobre la interfaz COM.....	59
5.3	Información sobre la interfaz Java.....	60
5.4	Ejemplos de código.....	62
5.4.1	C# .....	62
5.4.2	C++ .....	63
5.4.3	Java .....	65
5.4.4	VBScript.....	66
5.4.5	Visual Basic.....	68
5.5	Referencia de la API.....	71
5.5.1	COM y .NET .....	71
5.5.2	Java .....	78
<b>6</b>	<b>Gestor de esquemas</b>	<b>83</b>
6.1	Ejecutar el gestor de esquemas.....	87
6.2	Categorías de estado.....	90
6.3	Aplicar parches o instalar un esquema.....	92
6.4	Desinstalar o restaurar esquemas.....	94
6.5	Interfaz de la línea de comandos (ILC).....	95

6.5.1	help .....	95
6.5.2	info .....	96
6.5.3	initialize.....	96
6.5.4	install .....	97
6.5.5	list .....	97
6.5.6	reset .....	98
6.5.7	uninstall.....	99
6.5.8	update.....	100
6.5.9	upgrade.....	100

## **7 Gestor de taxonomías 102**

7.1	Ejecutar el gestor de taxonomías.....	106
7.2	Categorías de estado.....	109
7.3	Aplicar parches o instalar una taxonomía.....	111
7.4	Desinstalar o restaurar taxonomías.....	113
7.5	Interfaz de la línea de comandos (ILC).....	114
7.5.1	help .....	114
7.5.2	info .....	115
7.5.3	initialize.....	115
7.5.4	install .....	116
7.5.5	list .....	116
7.5.6	reset .....	117
7.5.7	uninstall.....	118
7.5.8	update.....	119
7.5.9	upgrade.....	119

## **Índice 121**

# 1 Introducción

**Altova StyleVision Server** es una implementación del motor de ejecución integrado del producto [StyleVision de Altova](#). Funciona como [producto servidor independiente](#)<sup>9</sup>, pero también está disponible como módulo de [FlowForce Server](#)<sup>8</sup> de Altova.



StyleVision Server ejecuta paquetes de transformación almacenados en disco o implementados en un servidor [FlowForce Server](#)<sup>8</sup>:

- Las funciones de StyleVision Server se pueden invocar [desde la línea de comandos](#)<sup>35</sup> para ejecutar un paquete de transformación y generar los archivos especificados en él.
- [FlowForce Server](#) inicia estas transformaciones en base a desencadenadores temporizados programables, desencadenadores de archivos o desencadenadores remotos definidos en FlowForce Server.

Altova ofrece versiones de 32 y 64 bits de StyleVision Server para equipos Windows. Para más información sobre cómo instalar StyleVision Server y asignarle licencias, consulte los apartados sobre la instalación en [Windows](#)<sup>11</sup>, [Linux](#)<sup>19</sup> y [macOS](#)<sup>26</sup>.

**Nota:** Si en uno de estos archivos PDF faltan partes editables al abrirlo en un sistema macOS, esto puede deberse a que Java 6 no esté instalado en el equipo. Si efectivamente este es el problema, puede descargar Java 6 en <https://support.apple.com/es-es/106384>. Si ya tiene instalada una versión más reciente que Java 6, entonces la instalación de Java 6 no afectará al funcionamiento de la nueva versión, que será la versión predeterminada del sistema.

*Última actualización: 09.04.2024*

## 2 Funcionamiento

StyleVision Server transforma archivos XML en documentos de salida HTML, PDF, RTF y DOCX con ayuda de hojas de estilos XSLT. Estas hojas de estilos XSLT se obtienen a partir de archivos PXF creados con la aplicación de diseño de hojas de estilos [Altova StyleVision](#).

StyleVision Server puede utilizarse de dos maneras:

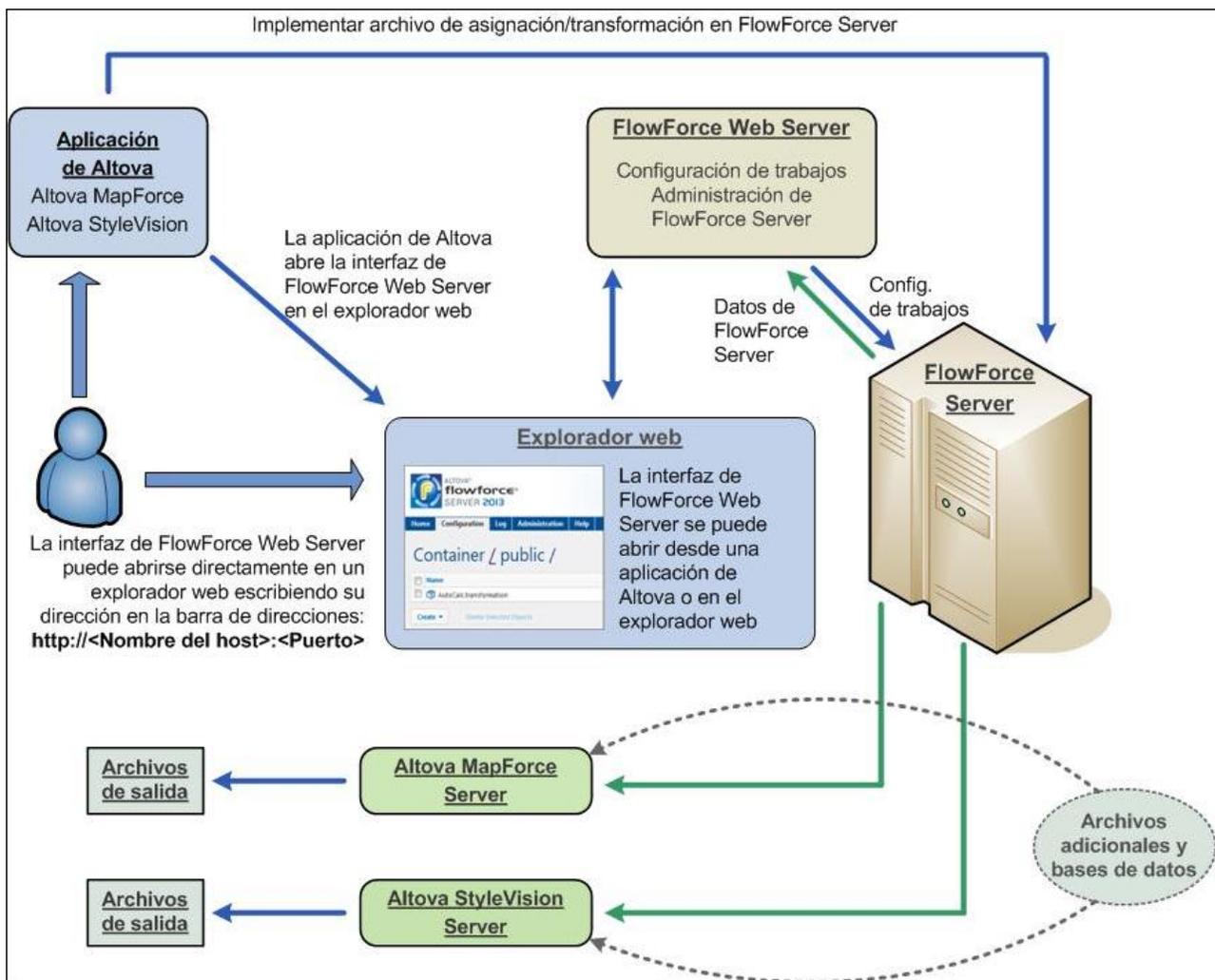
- Como módulo de [Altova FlowForce Server](#)<sup>8</sup>. Visite el [sitio web de Altova](#) para obtener más información sobre Altova FlowForce Server.
- [Como producto servidor independiente](#)<sup>9</sup> al que puede acceder desde su interfaz de la línea de comandos (ILC).

El usuario pasa un archivo XML de entrada y un archivo PXF a StyleVision Server, que genera los documentos de salida necesarios.

## 2.1 Como módulo de FlowForce Server

En [Altova FlowForce Server](#) se crean trabajos de FlowForce. Un trabajo de FlowForce especifica: (i) las entradas y salidas de una transformación de StyleVision Server y (ii) los desencadenadores que inician la ejecución del trabajo (p. ej. la hora del día). En tiempo de ejecución Altova FlowForce Server pasa instrucciones de transformación a StyleVision Server, que ejecuta la transformación.

En el diagrama que aparece a continuación puede ver qué papel desempeña StyleVision Server en el flujo de trabajo de FlowForce Server. El diagrama también incluye MapForce Server porque también se pueden crear trabajos de FlowForce Server que envían asignaciones de [Altova MapForce](#) a [Altova MapForce Server](#) para que las ejecute.



StyleVision Server se puede invocar desde un trabajo de FlowForce Server o desde la línea de comandos. Consulte el apartado [Uso de la línea de comandos](#) <sup>35</sup> para obtener más información.

## 2.2 Como producto servidor independiente

StyleVision Server se puede instalar como producto independiente en sistemas Windows, Linux y macOS. En la versión independiente las funciones de StyleVision Server se invocan desde la línea de comandos. Consulte el apartado [Uso de la línea de comandos](#)<sup>35</sup> para más información.

### 3 Instalación y asignación de licencias

En esta sección se explica cómo instalar su producto, asignarle una licencia y configurarlo. La sección se divide en varios apartados:

- [Instalación en Windows](#) <sup>11</sup>
- [Instalación en Linux](#) <sup>19</sup>
- [Instalación en macOS](#) <sup>26</sup>
- [Actualizar StyleVision Server](#) <sup>32</sup>
- [Migrar StyleVision Server a un equipo nuevo](#) <sup>33</sup>

## 3.1 Instalación y configuración en Windows

Esta sección explica cómo [instalar](#)<sup>11</sup> StyleVision Server y asignarle [licencias](#)<sup>15</sup> en sistemas Windows.

### Requisitos del sistema (Windows)

Tenga en cuenta estos requisitos del sistema:

- Windows 10, Windows 11
- Windows Server 2016 o superior

### Requisitos previos

Tenga en cuenta los siguientes requisitos:

- La instalación debe realizarse como usuario con privilegios de administrador.
- A partir de la versión 2021, la versión de 32 bits de StyleVision Server no se puede instalar sobre una versión de 64 bits o viceversa. Debe (i) eliminar la versión anterior antes de instalar la nueva o (ii) actualizar la versión a una más reciente de la misma versión de bits que la que ya tenía instalada.

### 3.1.1 Instalar en Windows

StyleVision Server se puede instalar en sistemas Windows. A continuación explicamos cómo instalar y configurar su producto. Para información detallada sobre partes específicas del proceso de instalación consulte los apartados correspondientes.

#### Instalar StyleVision Server

Para instalar StyleVision Server en sistemas Windows siga estos pasos:

- Como producto servidor independiente. Para instalar StyleVision Server descargue y ejecute su instalador. Siga las instrucciones que aparecen en pantalla.
- Para instalar StyleVision Server como parte del paquete de [FlowForce](#) Server descargue e instale el instalador de FlowForce Server. Siga las instrucciones que aparecen en pantalla y asegúrese de que marca la casilla para que también se instale StyleVision Server.

Puede encontrar tanto el instalador de StyleVision Server como el de [FlowForce Server](#) en el Centro de descargas de Altova (<http://www.altova.com/download.html>). Puede seleccionar el idioma de instalación en la parte inferior izquierda del asistente de instalación. Tenga en cuenta que el idioma que elija en la instalación también será el idioma predeterminado de StyleVision Server. Puede cambiar el idioma más tarde desde la línea de comandos.

#### Instalar LicenseServer

Para poder utilizar StyleVision Server debe asignarle una registros y licencia con un [Altova LicenseServer](#) de su red. En sistemas Windows puede instalar LicenseServer al mismo tiempo que StyleVision Server o FlowForce Server. Para más detalles consulte [Instalar LicenseServer](#)<sup>14</sup>.

Después de la instalación encontrará el ejecutable de StyleVision Server en esta carpeta:

```
<CarpetaArchivosPrograma>\Altova\StyleVisionServer2024\bin\StyleVisionServer.exe
```

El instalador se encarga de realizar los registros necesarios para usar StyleVision Server a través de una interfaz COM, como interfaz Java y en el entorno .NET.

## Instalación en Windows Server Core

Windows Server Core no tiene ninguna IGU y debe instalarse a través de la línea de comandos. Para más información sobre cómo instalar el equipo, consulte el apartado [Instalar en Windows Server Core](#)<sup>12</sup>.

## Desinstalar StyleVision Server

Para desinstalar StyleVision Server siga estos pasos:

1. Haga clic con el botón derecho en el botón de **Inicio** de Windows y seleccione **Opciones**.
2. Abra el Panel de control (empiece a escribir "Panel de control" y haga clic en la entrada sugerida).
3. En *Programas* haga clic en **Desinstalar un programa**.
4. En el Panel de control seleccione StyleVision Server y haga clic en **Desinstalar**.

## Licencia de evaluación

Durante el proceso de instalación podrá solicitar una licencia de evaluación para StyleVision Server que dura 30 días. Una vez enviada la solicitud recibirá una licencia de evaluación en la dirección de correo electrónico que indicó al registrarse.

### 3.1.2 Instalar en Windows Server Core

Windows Server Core es una instalación mínima de Windows que no usa todas las características de la IGU. Para instalar StyleVision Server en un equipo Windows Server Core:

1. Descargue el instalador de StyleVision Server desde el Centro de descargas de Altova. Este archivo se llama `styleVisionServer<version>.exe`. Asegúrese de que escoge el ejecutable que coincide con la plataforma de su servidor (32 bits o 64 bits).
2. En un equipo Windows estándar (no en el equipo con Windows Server Core) ejecute el comando `styleVisionServer<version>.exe /u`. Este comando abre los archivos `.msi` en el mismo directorio que el ejecutable del instalador.
3. Copie el archivo `.msi` sin descomprimir en el equipo en el que está Windows Server Core.
4. Si está actualizando una versión antigua de StyleVision Server cierre StyleVision Server antes de ejecutar el paso siguiente.
5. Use el archivo `.msi` para la instalación; para ello ejecute el comando `msiexec /i styleVisionServer.msi`. Este comando inicia la instalación en Windows Server Core.

#### Importante: conserve el archivo MSI

Debe tener en cuenta que:

- Mantenga los archivos `.msi` extraídos en un lugar seguro. Lo necesitará más tarde para desinstalar, reparar o modificar su instalación.
- Si quiere cambiar el nombre del archivo MSI debe hacerlo antes de instalar StyleVision Server.

- El nombre del archivo MSI se almacena en el registro, que es donde puede actualizar el nombre si este ha cambiado.

## Registrar StyleVision Server con LicenseServer

Si instala StyleVision Server por primera vez o si está actualizando el programa a la **versión siguiente**, necesita registrar StyleVision Server con Altova LicenseServer en su red. Si está instalando una actualización menor de StyleVision Server, el instalador reconocerá que ya registró el programa con LicenseServer, por lo que no será necesario volver a registrarlo. Sin embargo, si quiere cambiar el LicenseServer que usa StyleVision Server en algún momento, debe registrar StyleVision Server con License Server.

Para registrar StyleVision Server con Altova LicenseServer después de la instalación ejecute el comando de instalación con la propiedad `REGISTER_WITH_LICENSE_SERVER`, como se ve a continuación, e indique el nombre o la dirección del LicenseServer correspondiente como valor de la propiedad, por ejemplo:

```
msiexec /i StyleVisionServer.msi REGISTER_WITH_LICENSE_SERVER="localhost"
```

Para registrar StyleVision Server con un Altova LicenseServer después de instalarlo ejecute este comando:

```
msiexec /r StyleVisionServer.msi REGISTER_WITH_LICENSE_SERVER="<MyLS-IPAddress>"
```

## Comandos útiles

A continuación encontrará varios comandos que pueden serle útiles en el contexto de la instalación.

Para probar el valor de retorno de la instalación puede que quiera ejecutar un script parecido al que mostramos a continuación. El código de retorno estará en la variable de entorno `%errorlevel%`. El código de retorno 0 indica que la operación se ha realizado correctamente.

```
start /wait msiexec /i StyleVisionServer.msi /q
echo %errorlevel%
```

Si prefiere realizar una instalación silenciosa con un código de retorno y un registro del proceso de instalación use este comando:

```
start /wait msiexec /i StyleVisionServer.msi /q /L*v! <pathToInstallLogFile>
```

Para modificar la instalación, ejecute:

```
msiexec /m StyleVisionServer.msi
```

Para reparar la instalación, ejecute:

```
msiexec /r StyleVisionServer.msi
```

Para desinstalar StyleVision Server:

```
msiexec /x StyleVisionServer.msi
```

Para una desinstalación silenciosa de StyleVision Server y obtener un registro detallado del resultado en un archivo de registro:

```
start /wait msiexec /x StyleVisionServer.msi /q /L*v! <pathToUninstallLogFile>
```

Para instalar StyleVision Server en otro idioma (los idiomas disponibles y sus códigos son: Alemán=de; Español=es; Francés=fr):

```
msiexec /i StyleVisionServer.msi INSTALLER_LANGUAGE=<languageCode>
```

**Nota:** En Windows Server Core no están disponibles los gráficos ni los códigos de barras de StyleVision Server.

**Nota:** Para instalar taxonomías use el Gestor de taxonomías desde la línea de comandos. Para más información consulte el Manual del usuario de StyleVision Server.

### 3.1.3 Instalar LicenseServer (Windows)

Para poder utilizar StyleVision Server debe asignarle una licencia con un [Altova LicenseServer](#) de su red. En sistemas Windows puede instalar LicenseServer al mismo tiempo que StyleVision Server o FlowForce Server. Si ya hay un LicenseServer instalado en su red no necesita instalar uno nuevo, a no ser que necesite actualizar la versión de LicenseServer. (Véase el punto siguiente, [versiones de LicenseServer](#).)

Durante el proceso de instalación de StyleVision Server or FlowForce Server, active o desactive la opción para instalar LicenseServer según corresponda. Tenga en cuenta que:

- Si no ha instalado Altova LicenseServer, no cambie las opciones seleccionadas por defecto. El asistente instalará la versión más reciente en el equipo donde se está ejecutando.
- Si no ha instalado Altova LicenseServer y quiere instalarlo en otro equipo, desactive la casilla *Instalar Altova LicenseServer* en este equipo y elija el botón de opción **Registrar más tarde**. En este caso deberá instalar LicenseServer por separado y registrar StyleVision Server por más adelante.
- Si LicenseServer ya está instalado en el equipo pero se trata de una versión anterior a la indicada por el asistente para la instalación, no cambie las opciones seleccionadas por defecto (para la actualización a una nueva versión). En este caso, el asistente de instalación actualizará la versión de su LicenseServer automáticamente. La información del registro y de la licencia se traspasa a la versión más reciente de LicenseServer.
- Si LicenseServer ya está instalado en el equipo o en la red y se trata de la misma versión que la indicada por el asistente para la instalación, entonces siga estos pasos:
  - Desactive la casilla *Instalar Altova LicenseServer en este equipo*.
  - En *Registrar este producto* con elija el LicenseServer con el que quiere registrar StyleVision Server. También puede optar por la opción **Registrar más tarde**. Siempre puede elegir la opción **Registrar más tarde** si no quiere asociar el producto con LicenseServer y continuar con la instalación de StyleVision Server.

Para más información sobre cómo registrar StyleVision Server y asignarle una licencia con [Altova LicenseServer](#) consulte el apartado [Asignar licencias a StyleVision Server](#)<sup>15</sup>.

#### Versiones de LicenseServer

- Los productos servidor de Altova deben tener una licencia (i) con la versión de LicenseServer correspondiente a la versión de StyleVision Server instalada o (ii) con una versión posterior de LicenseServer.
- La versión de LicenseServer correspondiente a la versión actual de StyleVision Server es **3.14**.
- En Windows, puede instalar esta versión de LicenseServer junto con StyleVision Server o puede instalar LicenseServer por separado. En Linux y macOS, tiene que instalar LicenseServer por separado.
- Antes de instalar una versión nueva de LicenseServer, es necesario desinstalar versiones anteriores.
- Cuando se desinstala LicenseServer, todos los datos de registro y asignación de licencias almacenados en la versión antigua de LicenseServer se guardan en una base de datos en el equipo servidor. Estos datos se importan de forma automática a la siguiente versión que se instale en el

- equipo.
- Las versiones de LicenseServer son compatibles con versiones antiguas. Más concretamente, funcionan con versiones más antiguas de StyleVision Server.
- La última versión de LicenseServer está disponible en el sitio web de Altova. Esta versión funcionará con cualquier versión actual o anterior de StyleVision Server.
- El número de versión de LicenseServer siempre aparece al final de la [página de configuración de LicenseServer](#).

### 3.1.4 Asignar licencias a StyleVision Server (Windows)

Para poder trabajar con StyleVision Server es necesario asignarle una licencia con Altova LicenseServer. La asignación de licencias es un proceso de dos pasos:

1. El primero consiste en **registrar StyleVision Server** con LicenseServer desde StyleVision Server.
2. El segundo paso consiste en **asignar una licencia** a StyleVision Server desde LicenseServer. Descargue la versión más reciente de LicenseServer desde [el sitio web de Altova](#) e instálelo en su equipo local o en uno de los equipos de su red.

Más abajo explicamos qué pasos debe seguir. Para información más detallada consulte el [manual del usuario de LicenseServer](#) en [el sitio web de Altova](#).

#### 3.1.4.1 Iniciar LicenseServer, StyleVision Server

Tanto Altova LicenseServer (LicenseServer para abreviar) como StyleVision Server se inician con Altova ServiceController.

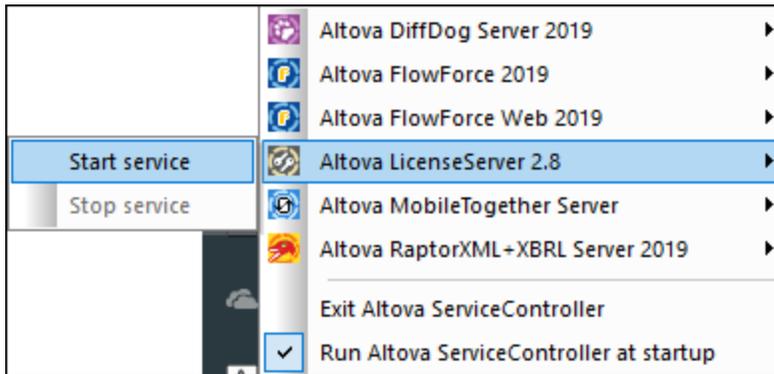
##### Altova ServiceController

Altova ServiceController (en adelante *ServiceController*) es una práctica aplicación que sirve para iniciar, detener y configurar los servicios de Altova **en sistemas Windows**.

ServiceController se instala con Altova LicenseServer y con los productos servidor de Altova que se instalan como servicios (DiffDog Server, FlowForce Server, Mobile Together Server y RaptorXML(+XBRL) Server). Una vez iniciado, podrá acceder a ServiceController desde la bandeja del sistema (*imagen siguiente*).

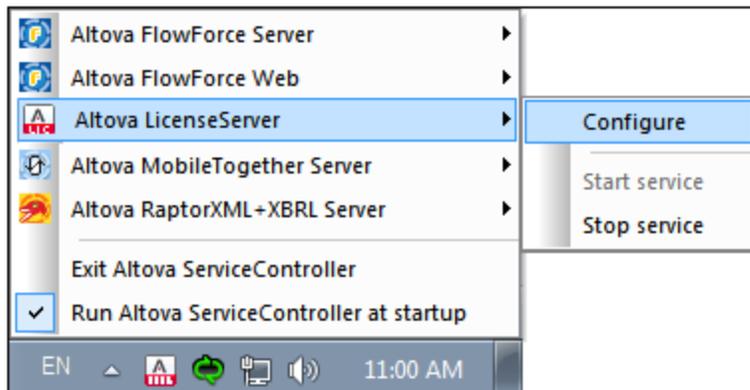


Si quiere que ServiceController se inicie automáticamente nada más iniciar sesión en el sistema, haga clic en el icono de **ServiceController** de la bandeja del sistema para abrir el menú de opciones de **ServiceController** (*imagen siguiente*) y active la opción **Run Altova ServiceController at Startup** (*Ejecutar Altova ServiceController al inicio*), que de todas maneras es la opción predeterminada. Para cerrar ServiceController haga clic en el icono de **ServiceController** de la bandeja del sistema y en el menú haga clic en la opción **Exit Altova ServiceController** (*Salir de Altova ServiceController*).



### Iniciar LicenseServer

Para iniciar LicenseServer haga clic en el icono de **ServiceController** en la bandeja del sistema, pase el puntero del ratón por encima de Altova LicenseServer en el menú que aparece (*imagen siguiente*) y seleccione **Iniciar servicio** en el submenú de LicenseServer. Si LicenseServer ya está en ejecución, la opción *Start Service* se deshabilita. También puede detener el servicio con ServiceController.



### Start StyleVision Server

Para iniciar StyleVision Server haga clic en el icono de **ServiceController** en la bandeja del sistema, pase el puntero del ratón por encima de Altova StyleVision Server en el menú que aparece y seleccione **Iniciar servicio** en el submenú de StyleVision Server. Si StyleVision Server ya está en ejecución, la opción Iniciar servicio se deshabilita. También puede detener el servicio con ServiceController.

### 3.1.4.2 Registrar StyleVision Server

Para poder asignar una licencia a StyleVision Server con Altova LicenseServer debe registrar primero la aplicación con LicenseServer.

Para registrar StyleVision Server desde la línea de comandos use el comando `licenseserver` e introduzca la dirección del equipo en el que está instalado LicenseServer (*ver más abajo*).

```
StyleVisionServer licenseserver [opciones] Servidor-O-Dirección-IP
```

Por ejemplo, si el nombre del servidor donde está instalado LicenseServer es `localhost`, entonces debe usar este comando:

```
StyleVisionServer licenseserver localhost
```

Si StyleVision Server se instaló como parte de la instalación de [FlowForce Server](#), al registrar FlowForce Server con LicenseServer también se registrará automáticamente StyleVision Server. En resumen: (i) Inicie Altova FlowForce Web con ServiceController (*véase punto anterior*), (ii) introduzca su contraseña para acceder a la página de configuración, (iii) seleccione el nombre o la dirección del LicenseServer y haga clic en **Registrar con LicenseServer**. Para más información consulte [Registrar FlowForce Server](#).

Una vez haya registrado su producto correctamente vaya a [la pestaña Client Management de la página de configuración de LicenseServer](#) y asigne una licencia a StyleVision Server.

Para más información sobre cómo registrar los productos de Altova con LicenseServer consulte el [manual del usuario de LicenseServer](#).

### 3.1.4.3 Asignar licencias a StyleVision Server

Una vez haya registrado StyleVision Server correctamente, la aplicación aparecerá en la pestaña Client Management de la página de configuración de LicenseServer. Ahora puede [asignar una licencia](#) a StyleVision Server.

La asignación de licencias a productos servidor de Altova depende de cuántos núcleos físicos (en contraposición a núcleos lógicos) tiene el procesador del equipo donde se ejecuta el producto servidor de Altova. Por ejemplo, un procesador dual tiene dos núcleos, un procesador *quad* tiene cuatro núcleos, un procesador *hexa-core* tiene seis núcleos, y así sucesivamente. El número de núcleos de la licencia asignada a un producto debe ser mayor o igual al número de núcleos disponibles en dicho equipo servidor, ya sea un servidor físico o un equipo virtual.

Por ejemplo, si un servidor tiene ocho núcleos (un procesador *octa-core*), deberá comprar una licencia para ocho núcleos. También puede combinar varias licencias para alcanzar el número de núcleos necesario. Es decir, puede usar dos licencias para cuatro núcleos para un servidor *octa-core* en lugar de una licencia para ocho núcleos, por ejemplo.

Si usa un equipo servidor con gran cantidad de núcleos, pero tiene un bajo volumen de procesamiento, también puede crear un equipo virtual que tenga adjudicados menos núcleos y comprar una licencia para ese menor número de núcleos. No obstante, dicha implementación sería menos rápida que si utilizaran todos los núcleos disponibles en el servidor.

**Nota:** Cada licencia de los productos servidor de Altova se puede usar de forma simultánea en un equipo como máximo (en el equipo donde está instalado el producto servidor de Altova), incluso si la capacidad de la licencia no está agotada. Por ejemplo, si utiliza una licencia para 10 núcleos para un equipo cliente que tiene 6 núcleos, los 4 núcleos restantes de la licencia no se pueden usar simultáneamente en otro equipo cliente.

### Ejecución por subprocesos simples

Si su producto de Altova admite la ejecución por subprocesos simples verá que hay disponible la opción correspondiente. En estos casos, si en el repertorio de licencias hay una licencia de producto servidor de Altova para un solo núcleo, puede asignársela a un equipo que tenga varios núcleos. En este caso, el equipo ejecutará el producto en un solo núcleo. El procesamiento será lógicamente más lento porque solo se usa un núcleo. Es decir, el producto se ejecutará en modo de subprocesamiento simple.

Para asignar una licencia de un solo núcleo a un equipo con varios núcleos basta con marcar en LicenseServer la casilla *Limit to single thread execution* del producto.

### Estimación del número de núcleos

Existen varios factores externos que suelen influir en los volúmenes y tiempos de procesamiento que su servidor puede manejar (por ejemplo, el hardware, la carga actual de la CPU, la memoria asignada a otras aplicaciones que se estén ejecutando en el servidor). Para poder conseguir un cálculo lo más exacto posible, recomendamos que primero ejecute las herramientas en su entorno para exponerlas a los factores y datos reales concretos de su negocio.

## 3.1.5 Notas sobre la configuración (Windows)

Para poder ejecutar en Windows los ejemplos que vienen incluidos con StyleVision Server y que puede encontrar en la subcarpeta `etc\examples` de la carpeta de la aplicación, StyleVision Server DLL debe estar correctamente registrado en el sistema. Puede generarse un error de registro si ya se han instalado distintas versiones de StyleVision Server (de 32 y 64 bits) en un mismo equipo.

Para registrar correctamente el DLL de StyleVision Server (de 32 o de 64 bits) en equipos Windows, siga estos pasos:

1. Abra la línea de comandos en modo administrador
2. Vaya a la carpeta en la que está el DLL. Para ello puede usar este comando: `cd C:\Program Files\Altova\StyleVisionServer2024\bin`
3. Ejecute este comando para registrar el DLL (versión de 32 o de 64 bits): `regsvr32 styleVisionServer.dll`
4. Debería aparecer un cuadro de diálogo que confirme que el registro se ha efectuado correctamente
5. Abra Visual Studio
6. Cargue el proyecto con `styleVisionServerAPI_Sample.sln`
7. Confirme que las rutas de su archivo `Program.cs` son válidas
8. Ejecute el proyecto con **Ctrl+F5**

**Nota:** La ruta de acceso a la carpeta de la aplicación en sistemas Windows suele ser: `C:\Program Files\Altova\StyleVisionServer2024`.

## 3.2 Instalación y configuración en Linux

Esta sección explica cómo [instalar](#)<sup>19</sup> StyleVision Server y [asignarle licencias](#)<sup>22</sup> en sistemas Linux (Debian, Ubuntu, CentOS, RedHat).

### Requisitos del sistema (Linux)

- Red Hat Enterprise Linux 7 o superior
- CentOS 7, CentOS Stream 8
- Debian 10 o superior
- Ubuntu 20.04, 22.04, 24.04
- AlmaLinux 9.0
- Rocky Linux 9.0

### Requisitos

- Puede instalarlo como usuario raíz o como usuario con privilegios sudo.
- Debe desinstalar la versión anterior de StyleVision Server antes de instalar una nueva.
- si quiere usar la función de gráficos de Altova, entonces debe haber al menos una fuente instalada en su sistema para que los gráficos se muestren correctamente. Para ver una lista de las fuentes instaladas use, por ejemplo, el comando `fc-list` de la [biblioteca Fontconfig](#).
- Estas son las fuentes necesarias como requisito para instalar y ejecutar la aplicación. Si los paquetes que aparecen en esta tabla no están en su equipo Linux, ejecute el comando `yum` (o `apt-get` si procede) para instalarlos.

CentOS, RedHat	Debian	Ubuntu
krb5-libs	libgssapi-krb5-2	libgssapi-krb5-2

### 3.2.1 Instalar en Linux

StyleVision Server se puede instalar en sistemas Linux. Puede instalarlo como usuario `raíz` o como usuario con privilegios `sudo`.

#### Integración de FlowForce Server y otros productos servidor de Altova

Si instala StyleVision Server junto con FlowForce Server, recomendamos que instale primero FlowForce Server. Si instala StyleVision Server antes que FlowForce Server, una vez haya instalado las dos aplicaciones ejecute este comando:

```
cp /opt/Altova/StyleVisionServer2024/etc/*.tool /opt/Altova/FlowForceServer2024/tools
```

Con este comando se copia el archivo `.tools` del directorio `/etc` de StyleVision Server en el directorio `/tools` de FlowForce Server. El archivo `.tool` es necesario para usar FlowForce Server. Este archivo contiene la ruta al ejecutable de StyleVision Server. Si instala FlowForce antes de instalar StyleVision Server no necesita ejecutar este comando.

## Desinstalar StyleVision Server

Tiene que desinstalar cualquier versión anterior de StyleVision Server que tenga instalada antes de poder instalar la versión nueva de la aplicación.

Para comprobar qué productos servidor de Altova están ya instalados, utilice uno de los siguientes comandos:

```
[Debian, Ubuntu]:  dpkg --list | grep Altova
[CentOS, RedHat]:  rpm -qa | grep server
```

Si tiene que desinstalar una versión anterior de StyleVision Server puede usar uno de los siguientes comandos:

```
[Debian, Ubuntu]:  sudo dpkg --remove stylevisionserver
[CentOS, RedHat]:  sudo rpm -e stylevisionserver
```

En sistemas Debian y Ubuntu puede ocurrir que StyleVision Server siga apareciendo en la lista de productos instalados incluso después de haberlo desinstalado. En este caso puede ejecutar el comando `purge` para eliminar StyleVision Server de la lista. También puede usar el comando `purge` en lugar del comando `instead` que mencionamos más arriba.

```
[Debian, Ubuntu]:  sudo dpkg --purge stylevisionserver
```

## Descargar el paquete de instalación de StyleVision Server para Linux

Puede descargar los paquetes de instalación de StyleVision Server para los sistemas Linux siguientes en el [sitio web de Altova](#).

Distribución	Extensión del paquete
Debian	.deb
Ubuntu	.deb
CentOS	.rpm
RedHat	.rpm

Una vez haya descargado el paquete de instalación para Linux, copie el paquete en cualquier directorio del sistema Linux. Como necesitará un [Altova LicenseServer](#) para ejecutar StyleVision Server, puede que quiera descargar LicenseServer del [sitio web de Altova](#) al mismo tiempo que StyleVision Server en lugar de descargarlo más tarde.

## Instalar StyleVision Server

En una ventana de la terminal cambie al directorio donde copió el paquete Linux. Por ejemplo, si lo copió en un directorio del usuario llamado `MiAltova` ubicado en `/home/User`, cambie a ese directorio con esta línea de comandos:

```
cd /home/User/MiAltova
```

Instale StyleVision Server con el comando correspondiente:

```
[Debian]: sudo dpkg --install stylevisionserver-2024-debian.deb
[Ubuntu]: sudo dpkg --install stylevisionserver-2024-ubuntu.deb
[CentOS]: sudo rpm -ivh stylevisionserver-2024-1.x86_64.rpm
[RedHat]: sudo rpm -ivh stylevisionserver-2024-1.x86_64.rpm
```

Quizás sea necesario ajustar el nombre del paquete anterior para que tenga el número de versión o de service pack actual.

El paquete de StyleVision Server se instalará en esta carpeta:

```
/opt/Altova/StyleVisionServer2024
```

### 3.2.2 Instalar LicenseServer (Linux)

Para poder utilizar StyleVision Server debe asignarle una licencia con un [Altova LicenseServer](#) de su red. Descargue el instalador de LicenseServer del [sitio web de Altova](#) y copie el paquete en cualquier directorio. Instálelo tal y como instaló StyleVision Server (consulte [el apartado anterior](#)<sup>19</sup>).

```
[Debian]: sudo dpkg --install licenseserver-3.14-debian.deb
[Ubuntu]: sudo dpkg --install licenseserver-3.14-ubuntu.deb
[CentOS]: sudo rpm -ivh licenseserver-3.14-1.x86_64.rpm
[RedHat]: sudo rpm -ivh licenseserver-3.14-1.x86_64.rpm
```

El paquete de LicenseServer se instalará en la siguiente ruta de acceso:

```
/opt/Altova/LicenseServer
```

Para más información sobre cómo registrar StyleVision Server y asignarle una licencia con [Altova LicenseServer](#) consulte el apartado [Asignar licencias a StyleVision Server](#)<sup>22</sup>. Consulte también [la documentación de LicenseServer](#) para obtener información más detallada.

#### Versiones de LicenseServer

- Los productos servidor de Altova deben tener una licencia (i) con la versión de LicenseServer correspondiente a la versión de StyleVision Server instalada o (ii) con una versión posterior de LicenseServer.
- La versión de LicenseServer correspondiente a la versión actual de StyleVision Server es **3.14**.
- En Windows, puede instalar esta versión de LicenseServer junto con StyleVision Server o puede instalar LicenseServer por separado. En Linux y macOS, tiene que instalar LicenseServer por separado.
- Antes de instalar una versión nueva de LicenseServer, es necesario desinstalar versiones anteriores.
- Cuando se desinstala LicenseServer, todos los datos de registro y asignación de licencias almacenados en la versión antigua de LicenseServer se guardan en una base de datos en el equipo servidor. Estos datos se importan de forma automática a la siguiente versión que se instale en el equipo.
- Las versiones de LicenseServer son compatibles con versiones antiguas. Más concretamente, funcionan con versiones más antiguas de StyleVision Server.
- La última versión de LicenseServer está disponible en el sitio web de Altova. Esta versión funcionará con cualquier versión actual o anterior de StyleVision Server.

- El número de versión de LicenseServer siempre aparece al final de la [página de configuración de LicenseServer](#).

### 3.2.3 Asignar licencias a StyleVision Server (Linux)

Para poder trabajar con StyleVision Server es necesario asignarle una licencia con Altova LicenseServer. La asignación de licencias es un proceso de dos pasos:

1. El primero consiste en **registrar StyleVision Server** con LicenseServer desde StyleVision Server.
2. El segundo paso consiste en **asignar una licencia** a StyleVision Server desde LicenseServer. Descargue la versión más reciente de LicenseServer desde [el sitio web de Altova](#) e instálelo en su equipo local o en uno de los equipos de su red.

Más abajo explicamos qué pasos debe seguir. Para información más detallada consulte el [manual del usuario de LicenseServer](#) en [el sitio web de Altova](#).

#### 3.2.3.1 Iniciar LicenseServer, StyleVision Server

Inicie Altova LicenseServer y StyleVision Server o como usuario `root` o como usuario normal con privilegios `sudo`.

##### Iniciar LicenseServer

Para registrar y asignar una licencia a StyleVision Server con LicenseServer debe ejecutar LicenseServer como demonio. Para ello use este comando:

```
sudo systemctl start licenseserver
```

Si necesita detener LicenseServer en algún momento, reemplace `start` con `stop` en el comando anterior. Por ejemplo:

```
sudo systemctl stop licenseserver
```

#### 3.2.3.2 Registrar StyleVision Server

Para poder asignar una licencia a StyleVision Server con Altova LicenseServer debe registrar primero la aplicación con LicenseServer.

Para registrar StyleVision Server desde la línea de comandos use el comando `licenseserver`:

```
sudo /opt/Altova/StyleVisionServer2024/bin/stylevisionserver licenseserver [opciones]  
Servidor-O-Dirección-IP
```

Por ejemplo, si el nombre del servidor donde está instalado LicenseServer es `localhost`, entonces debe usar este comando:

```
sudo /opt/Altova/StyleVisionServer2024/bin/stylevisionserver licenseserver localhost
```

En el comando anterior, `localhost` es el nombre del servidor en el que está instalado LicenseServer. Observe que el ejecutable de StyleVision Server se encuentra en:

```
/opt/Altova/StyleVisionServer2024/bin/
```

Una vez haya registrado su producto correctamente vaya a [la pestaña Client Management de la página de configuración de LicenseServer](#) y asigne una licencia a StyleVision Server.

Para más información sobre cómo registrar los productos de Altova con LicenseServer consulte el [manual del usuario de LicenseServer](#).

### 3.2.3.3 Asignar licencias a StyleVision Server

Una vez haya registrado StyleVision Server correctamente, la aplicación aparecerá en la pestaña Client Management de la página de configuración de LicenseServer. Ahora puede [asignar una licencia](#) a StyleVision Server.

La asignación de licencias a productos servidor de Altova depende de cuántos núcleos físicos (en contraposición a núcleos lógicos) tiene el procesador del equipo donde se ejecuta el producto servidor de Altova. Por ejemplo, un procesador dual tiene dos núcleos, un procesador *quad* tiene cuatro núcleos, un procesador *hexa-core* tiene seis núcleos, y así sucesivamente. El número de núcleos de la licencia asignada a un producto debe ser mayor o igual al número de núcleos disponibles en dicho equipo servidor, ya sea un servidor físico o un equipo virtual.

Por ejemplo, si un servidor tiene ocho núcleos (un procesador *octa-core*), deberá comprar una licencia para ocho núcleos. También puede combinar varias licencias para alcanzar el número de núcleos necesario. Es decir, puede usar dos licencias para cuatro núcleos para un servidor *octa-core* en lugar de una licencia para ocho núcleos, por ejemplo.

Si usa un equipo servidor con gran cantidad de núcleos, pero tiene un bajo volumen de procesamiento, también puede crear un equipo virtual que tenga adjudicados menos núcleos y comprar una licencia para ese menor número de núcleos. No obstante, dicha implementación sería menos rápida que si utilizaran todos los núcleos disponibles en el servidor.

**Nota:** Cada licencia de los productos servidor de Altova se puede usar de forma simultánea en un equipo como máximo (en el equipo donde está instalado el producto servidor de Altova), incluso si la capacidad de la licencia no está agotada. Por ejemplo, si utiliza una licencia para 10 núcleos para un equipo cliente que tiene 6 núcleos, los 4 núcleos restantes de la licencia no se pueden usar simultáneamente en otro equipo cliente.

#### Ejecución por subprocesos simples

Si su producto de Altova admite la ejecución por subprocesos simples verá que hay disponible la opción correspondiente. En estos casos, si en el repertorio de licencias hay una licencia de producto servidor de Altova para un solo núcleo, puede asignársela a un equipo que tenga varios núcleos. En este caso, el equipo ejecutará el producto en un solo núcleo. El procesamiento será lógicamente más lento porque solo se usa un núcleo. Es decir, el producto se ejecutará en modo de subprocesamiento simple.

Para asignar una licencia de un solo núcleo a un equipo con varios núcleos basta con marcar en LicenseServer

la casilla *Limit to single thread execution* del producto.

#### Estimación del número de núcleos

Existen varios factores externos que suelen influir en los volúmenes y tiempos de procesamiento que su servidor puede manejar (por ejemplo, el hardware, la carga actual de la CPU, la memoria asignada a otras aplicaciones que se estén ejecutando en el servidor). Para poder conseguir un cálculo lo más exacto posible, recomendamos que primero ejecute las herramientas en su entorno para exponerlas a los factores y datos reales concretos de su negocio.

## 3.2.4 Notas sobre configuración del entorno (Linux)

### Carpetas

A continuación enumeramos carpetas importantes de su sistema StyleVision Server.

#### ☐ Directorio raíz de instalación

`/opt/Altova/StyleVisionServer2024/`

#### ☐ Archivos de licencia

`/var/opt/Altova/StyleVisionServer`

#### ☐ Configuración del entorno

`/etc/profile.d/jdbc.sh`

El archivo de parámetros del entorno (por lo general llamado `jdbc.sh`) se ejecuta al iniciar el sistema. Las definiciones que contiene deben ser específicas para un entorno en concreto. La ruta de acceso anterior es un ejemplo solamente.

**Nota:** El archivo de parámetros del entorno establece las variables para **todos los usuarios** del sistema, por lo que debe tener cuidado al modificar las opciones. Por ejemplo, si modifica la ruta de una clase en este archivo, el cambio tendrá lugar en todo el sistema. Si quiere realizar cambios que afecten únicamente a StyleVision Server, recomendamos que considere usar un fichero de unidades (véase la sección *Conexiones JDBC* más abajo).

### Conexiones de BD

En Linux se admiten las siguientes conexiones de BD:

- JDBC: puede usar JDBC para todas las BD compatibles salvo para Microsoft Access
- Conexiones nativas: actualmente disponibles para las bases de datos SQLite y PostgreSQL

Si usa JDBC debe tener en cuenta los siguientes aspectos:

- Debe tener instalado Java Runtime Environment o el kit de desarrollo de software.
- Debe tener instalados controladores JDBC para la base de datos de destino.
- Debe establecer correctamente estas variables de entorno:
  - `CLASSPATH`: para encontrar los archivos jar que se conectan a la base de datos JDBC; los archivos jar se pueden introducir en (i) un archivo ejecutable (como `jdbc.sh`) que se ejecuta al iniciar el sistema o (ii) en un fichero de unidades que se ejecuta cuando StyleVision Server se inicia como servicio. Usar un fichero de unidades para especificar los archivos jar tiene la ventaja de que los

archivos necesarios para las conexiones JDBC de StyleVision Server se ubican sin que tenga que modificar la configuración del sistema. Más abajo se muestra un fichero de unidades.

- o PATH: para buscar el entorno JRE, aunque a veces no es necesaria, dependiendo de la instalación.
- o JAVA\_HOME: a veces no es necesaria, dependiendo de la instalación.

## Lista de archivos importantes

Puede copiar este script de shell (o archivo de unidades) en la carpeta `/opt/Altova/StyleVisionServer/etc` para no sobrescribir los archivos de configuración actuales. Realice los cambios que necesite en el script. Consulte también el apartado anterior *Conexiones JDBC*. Las partes que aparecen resaltadas en azul son propias del entorno y deberán ajustarse.

### Script de shell (fichero de unidades)

```
#- jdbc - environment -
export PATH=/usr/local/jdk1.7.0_17/bin:/usr/lib64/qt-
3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/qa/bin
export JAVA_HOME=/usr/local/jdk1.7.0_17
export
CLASSPATH=/usr/local/jdbc/oracle/ojdbc6.jar:/usr/local/jdbc/oracle/xdm.jar:/usr/local/j
dbc/oracle/xmlparserv2.jar:/usr/local/jdbc/postgre/postgresql-9.0-
801.jdbc4.jar:/usr/local/jdbc/mssql/sqljdbc4.jar:/usr/local/jdbc/ibseries/lib/jt400.jar:
/usr/local/jdbc/mysql/mysql-connector-java-5.1.16-
bin.jar:/usr/local/jdbc/sqlite/sqlitejdbc-
v056.jar:/usr/local/jdbc/Informix_JDBC_Driver/lib/ifxjdbc.jar:/usr/local/jdbc/sybase/jc
onn7/jconn4.jar:/usr/local/jdbc/db2/db2jcc.jar:/usr/local/jdbc/db2/db2jcc_license_cu.ja
r:./:
```

## 3.3 Instalación y configuración en macOS

Esta sección explica cómo [instalar](#)<sup>26</sup> StyleVision Server y [asignarle licencias](#)<sup>28</sup> en sistemas macOS.

### Requisitos del sistema (macOS)

Tenga en cuenta estos requisitos del sistema:

- macOS 12 o superior

### Requisitos

Tenga en cuenta los siguientes requisitos:

- Asegúrese de que Altova LicenseServer está instalado y ejecutándose.
- Puede instalarlo como usuario `raíz` o como usuario con privilegios `sudo`.
- Debe desinstalar la versión anterior de StyleVision Server antes de instalar una nueva.
- Si tiene pensado usar la función de gráficos de Altova, debe tener instalada al menos una fuente en su sistema para garantizar que los gráficos se visualizan correctamente. Para ver una lista de las fuentes instaladas use, por ejemplo, el comando `fc-list` de la [biblioteca Fontconfig](#).
- El equipo macOS debe estar configurado de forma que su nombre se resuelva en una dirección de IP. Esto significa que debe poder hacerle ping al nombre de host desde la terminal con el comando `ping <hostname>`.

### 3.3.1 Instalar en macOS

Esta sección explica cómo instalar y configurar StyleVision Server en sistemas macOS.

#### Integración con FlowForce

Si instala StyleVision Server junto con FlowForce Server, recomendamos que instale primero FlowForce Server. Si instala StyleVision Server antes que FlowForce Server, entonces una vez haya instalado las dos aplicaciones ejecute este comando:

```
cp /usr/local/Altova/StyleVisionServer2024/etc/*.tool /usr/local/Altova/FlowForceServer2024/tools
```

Con este comando se copia el archivo `.tools` del directorio `/etc` de StyleVision Server en el directorio `/tools` de FlowForce Server. El archivo `.tool` es necesario para usar FlowForce Server. Este archivo contiene la ruta al ejecutable de StyleVision Server. Si instala FlowForce antes de instalar StyleVision Server no necesita ejecutar este comando.

#### Desinstalar StyleVision Server

En la carpeta Aplicaciones, haga clic con el botón derecho en el icono de StyleVision Server y seleccione **Mover a la papelera**. La aplicación se moverá a la papelera. pero debe quitar la aplicación de la carpeta `usr`. Para ello use el comando:

```
sudo rm -rf /usr/local/Altova/StyleVisionServer2024/
```

Si tiene que desinstalar una versión antigua de Altova License Server primer tiene que detener su ejecución como servicio. Para ello use el comando:

```
sudo launchctl unload /Library/LaunchDaemons/com.altova.LicenseServer.plist
```

Para comprobar si el servicio se detuvo o no, abra la terminal del Monitor de actividad en Finder y confirme que LicenseServer no está en la lista. Ahora procesa a desinstalar la aplicación con los mismos pasos que hemos explicado antes para StyleVision Server.

## Instalar StyleVision Server

Para instalar StyleVision Server siga estos pasos:

1. Descargue el archivo de imagen de disco (`.dmg`) de StyleVision Server desde el sitio web de Altova (<http://www.altova.com/es/download.html>).
2. Haga clic en el archivo para abrir la imagen de disco (`.dmg`). El instalador de StyleVision Server aparece en el equipo como otra unidad de disco más.
3. En el disco virtual haga doble clic en el paquete del instalador (`.pkg`).
4. Siga los pasos del asistente de instalación. En uno de los pasos debe aceptar el contrato de licencia para poder seguir con la instalación. Consulte también [Asignar licencias a StyleVision Server](#)<sup>28</sup>.
5. Para expulsar el disco después de la instalación haga clic en él con el botón derecho y seleccione **Expulsar**.

El paquete de StyleVision Server se instalará en la carpeta:

```
/usr/local/Altova/StyleVisionServer2024 (archivos binarios de la aplicación)  
/var/Altova/StyleVisionServer (archivos de datos: BD y registros)
```

El servidor demonio de StyleVision Server se inicia automáticamente después de la instalación y de reiniciar el equipo. Siempre puede iniciar StyleVision Server como demonio con este comando:

```
sudo launchctl load /Library/LaunchDaemons/com.altova.StyleVisionServer2024.plist
```

## 3.3.2 Instalar LicenseServer (macOS)

Puede descargar Altova LicenseServer desde el sitio web de Altova (<http://www.altova.com/es/download.html>). Ahora proceda al proceso de instalación descrito [aquí](#)<sup>26</sup>.

El paquete de LicenseServer se instalará en la siguiente carpeta:

```
/usr/local/Altova/LicenseServer
```

Para más información sobre cómo registrar StyleVision Server con [Altova LicenseServer](#) y asignarle una licencia consulte [Asignar licencias en macOS](#)<sup>28</sup>.

## Versiones de LicenseServer

- Los productos servidor de Altova deben tener una licencia (i) con la versión de LicenseServer

correspondiente a la versión de StyleVision Server instalada o (ii) con una versión posterior de LicenseServer.

- La versión de LicenseServer correspondiente a la versión actual de StyleVision Server es **3.14**.
- En Windows, puede instalar esta versión de LicenseServer junto con StyleVision Server o puede instalar LicenseServer por separado. En Linux y macOS, tiene que instalar LicenseServer por separado.
- Antes de instalar una versión nueva de LicenseServer, es necesario desinstalar versiones anteriores.
- Cuando se desinstala LicenseServer, todos los datos de registro y asignación de licencias almacenados en la versión antigua de LicenseServer se guardan en una base de datos en el equipo servidor. Estos datos se importan de forma automática a la siguiente versión que se instale en el equipo.
- Las versiones de LicenseServer son compatibles con versiones antiguas. Más concretamente, funcionan con versiones más antiguas de StyleVision Server.
- La última versión de LicenseServer está disponible en el sitio web de Altova. Esta versión funcionará con cualquier versión actual o anterior de StyleVision Server.
- El número de versión de LicenseServer siempre aparece al final de la [página de configuración de LicenseServer](#).

### 3.3.3 Asignar licencias a StyleVision Server (macOS)

Para poder trabajar con StyleVision Server es necesario asignarle una licencia con Altova LicenseServer. La asignación de licencias es un proceso de dos pasos:

1. El primero consiste en **registrar StyleVision Server** con LicenseServer desde StyleVision Server.
2. El segundo paso consiste en **asignar una licencia** a StyleVision Server desde LicenseServer. Descargue la versión más reciente de LicenseServer desde [el sitio web de Altova](#) e instálelo en su equipo local o en uno de los equipos de su red.

Más abajo explicamos qué pasos debe seguir. Para información más detallada consulte el [manual del usuario de LicenseServer](#) en [el sitio web de Altova](#).

#### 3.3.3.1 Iniciar LicenseServer, StyleVision Server

Inicie Altova LicenseServer y StyleVision Server o como usuario `root` o como usuario normal con privilegios `sudo`.

##### Iniciar LicenseServer

Para registrar y asignar una licencia a StyleVision Server con LicenseServer debe ejecutar LicenseServer como demonio. Para iniciar LicenseServer como demonio, ejecute este comando:

```
sudo launchctl load /Library/LaunchDaemons/com.altova.LicenseServer.plist
```

Si necesita detener LicenseServer en algún momento, reemplace `load` con `unload` en el comando anterior.

### 3.3.3.2 Registrar StyleVision Server

Para poder asignar una licencia a StyleVision Server con Altova LicenseServer debe registrar primero la aplicación con LicenseServer.

Para registrar StyleVision Server desde la línea de comandos use el comando `licenseserver`:

```
sudo /usr/local/Altova/StyleVisionServer2024/bin/StyleVisionServer licenseserver  
[opciones] Servidor-O-Dirección-IP
```

Por ejemplo, si el nombre del servidor donde está instalado LicenseServer es `localhost`, entonces debe usar este comando:

```
sudo /usr/local/Altova/StyleVisionServer2024/bin/StyleVisionServer licenseserver  
localhost
```

En el comando anterior, `localhost` es el nombre del servidor en el que está instalado LicenseServer. Observe que el ejecutable de StyleVision Server se encuentra en:

```
/usr/local/Altova/StyleVisionServer2024/bin/
```

Una vez haya registrado su producto correctamente vaya a [la pestaña Client Management de la página de configuración de LicenseServer](#) y asigne una licencia a StyleVision Server.

Para más información sobre cómo registrar los productos de Altova con LicenseServer consulte el [manual del usuario de LicenseServer](#).

### 3.3.3.3 Asignar licencias a StyleVision Server

Una vez haya registrado StyleVision Server correctamente, la aplicación aparecerá en la pestaña Client Management de la página de configuración de LicenseServer. Ahora puede [asignar una licencia](#) a StyleVision Server.

La asignación de licencias a productos servidor de Altova depende de cuántos núcleos físicos (en contraposición a núcleos lógicos) tiene el procesador del equipo donde se ejecuta el producto servidor de Altova. Por ejemplo, un procesador dual tiene dos núcleos, un procesador *quad* tiene cuatro núcleos, un procesador *hexa-core* tiene seis núcleos, y así sucesivamente. El número de núcleos de la licencia asignada a un producto debe ser mayor o igual al número de núcleos disponibles en dicho equipo servidor, ya sea un servidor físico o un equipo virtual.

Por ejemplo, si un servidor tiene ocho núcleos (un procesador *octa-core*), deberá comprar una licencia para ocho núcleos. También puede combinar varias licencias para alcanzar el número de núcleos necesario. Es decir, puede usar dos licencias para cuatro núcleos para un servidor *octa-core* en lugar de una licencia para ocho núcleos, por ejemplo.

Si usa un equipo servidor con gran cantidad de núcleos, pero tiene un bajo volumen de procesamiento, también puede crear un equipo virtual que tenga adjudicados menos núcleos y comprar una licencia para ese menor número de núcleos. No obstante, dicha implementación sería menos rápida que si utilizaran todos los núcleos disponibles en el servidor.

**Nota:** Cada licencia de los productos servidor de Altova se puede usar de forma simultánea en un equipo como máximo (en el equipo donde está instalado el producto servidor de Altova), incluso si la capacidad de la licencia no está agotada. Por ejemplo, si utiliza una licencia para 10 núcleos para un equipo cliente que tiene 6 núcleos, los 4 núcleos restantes de la licencia no se pueden usar simultáneamente en otro equipo cliente.

#### Ejecución por subprocesos simples

Si su producto de Altova admite la ejecución por subprocesos simples verá que hay disponible la opción correspondiente. En estos casos, si en el repertorio de licencias hay una licencia de producto servidor de Altova para un solo núcleo, puede asignársela a un equipo que tenga varios núcleos. En este caso, el equipo ejecutará el producto en un solo núcleo. El procesamiento será lógicamente más lento porque solo se usa un núcleo. Es decir, el producto se ejecutará en modo de subprocesamiento simple.

Para asignar una licencia de un solo núcleo a un equipo con varios núcleos basta con marcar en LicenseServer la casilla *Limit to single thread execution* del producto.

#### Estimación del número de núcleos

Existen varios factores externos que suelen influir en los volúmenes y tiempos de procesamiento que su servidor puede manejar (por ejemplo, el hardware, la carga actual de la CPU, la memoria asignada a otras aplicaciones que se estén ejecutando en el servidor). Para poder conseguir un cálculo lo más exacto posible, recomendamos que primero ejecute las herramientas en su entorno para exponerlas a los factores y datos reales concretos de su negocio.

## 3.3.4 Notas sobre configuración del entorno (macOS)

### Carpetas

A continuación enumeramos carpetas importantes de su sistema StyleVision Server.

#### ▣ Directorio raíz de instalación

`/usr/local/Altova/StyleVisionServer2024/`

#### ▣ Archivos de licencias

`/var/Altova/StyleVisionServer`

#### ▣ Parámetros del entorno

`/Library/LaunchDaemons/com.altova.StyleVisionServer.plist`

El archivo de parámetros del entorno debe definirse en función del entorno de cada usuario. La ruta de acceso anterior es un ejemplo solamente.

**Nota:** Estas variables del entorno solamente se establecen para el proceso de StyleVision Server y por tanto no afectan a los demás usuarios.

### Conexiones de BD

En macOS se admiten las siguientes conexiones de BD:

- JDBC: puede usar JDBC para todas las BD compatibles salvo para Microsoft Access
- Conexiones nativas: actualmente disponibles para las bases de datos SQLite y PostgreSQL

Si usa JDBC debe tener en cuenta estos aspectos:

- Debe tener instalado Java Runtime Environment o el kit de desarrollo de software.
- Debe tener instalados controladores JDBC para la base de datos de destino.
- Debe establecer correctamente estas variables de entorno:
  - **CLASSPATH**: para encontrar los archivos jar: la ruta de la clase se determina en el archivo `plist`.
  - **PATH**: para buscar el entorno JRE, aunque a veces no es necesaria, dependiendo de la instalación.
  - **JAVA\_HOME**: a veces no es necesaria, dependiendo de la instalación.

## Java 6 para formularios PDF editables

Si en uno de estos archivos PDF faltan partes editables al abrirlo en un sistema Mac OS, esto puede deberse a que Java 6 no esté instalado en el equipo. Si es así, puede instalar Java 6 desde [este enlace](#). Si ya tiene instalada una versión más reciente que Java 6, entonces la instalación de Java 6 no afectará al funcionamiento de la nueva versión, que será la versión predeterminada del sistema.

## 3.4 Actualizar StyleVision Server

La forma más fácil de transferir una licencia de una versión anterior de StyleVision Server a la más reciente es seguir los pasos del proceso de instalación. Estos son los pasos principales que debe llevar a cabo durante la instalación:

1. Registre la versión nueva de StyleVision Server con el LicenseServer en el que está la licencia que usaba la versión anterior de StyleVision Server.
2. Acepte el acuerdo de la licencia de StyleVision Server. (Si no lo hace no se instalará la versión más reciente.)

**Nota:** Si no registra StyleVision Server con LicenseServer durante el proceso de instalación, podrá hacerlo más adelante y después asignarle una licencia.

## 3.5 Migrar StyleVision Server a un equipo nuevo

Si quiere migrar StyleVision Server de un equipo a otro (también entre plataformas compatibles) siga las instrucciones de más abajo.

Migrar StyleVision Server a otro equipo consiste en asignar la licencia del equipo antiguo al equipo nuevo. Para ello siga estas instrucciones:

1. Instale StyleVision Server en el equipo nuevo. Si ya instaló el producto como parte de la instalación de FlowForce Server, ignore este paso.
2. En el equipo nuevo registre StyleVision Server con Altova LicenseServer.
3. En el equipo antiguo asegúrese de que ningún cliente está usando el servidor .
4. Abra la página de administración de Altova LicenseServer. Desactive la licencia de StyleVision Server en el equipo antiguo y vuelva a asignarla en el equipo nuevo.

## 3.6 Requisitos FOP

StyleVision Server usa por defecto el procesador FO del [proyecto Apache FOP](#) al generar documentos PDF con el comando `generate`. (Puede seleccionar otro procesador FO con el comando `setfopath`.)

Al instalar StyleVision Server se instala también FOP. En sistemas Windows el FOP se instala en la carpeta `C:\ProgramData\Altova\SharedBetweenVersions`. En sistemas Linux y macOS se instala en una subcarpeta de `styleVisionServer2024`.

Para más información sobre ejecutar el FOP 2.9, consulte el [apartado al respecto](#) en el sitio web Apache.org. (Si está usando otra versión del FOP, vaya a la página correspondiente.)

### Entorno de ejecución de Java

Para crear PDFs con el FOP debe tener instalado en su equipo un entorno de ejecución de java (JRE). Compruebe en el [sitio web Apache.org](#) cuál es la versión mínima de JRE que necesita para su versión de FOP. Por ejemplo, para la versión 2.8 de FOP necesita un JRE versión 1.8 o superior. Si intenta generar un PDF con el FOP sin tener un JRE instalado, generará un mensaje de error.

#### Configurar OpenJDK

Hemos probado StyleVision Server de forma regular con la versión más reciente disponible de [OpenJDK](#). En el momento de redactar este manual, la versión más reciente es 19.0.2. Una vez haya descargado y descomprimido el paquete OpenJDK, asegúrese (i) de que añade la ruta `java\bin` a las rutas del sistema y (ii) de configurar la variable de entorno `JAVA_HOME`.

Para configurar la variable de entorno `JAVA_HOME` en Windows siga estos pasos:

1. En la barra de tareas de Windows, haga clic con el botón derecho en el icono de Windows y seleccione *Sistema*.
2. En la ventana de configuración, en *Vínculos relacionados*, haga clic en *Configuración avanzada del sistema*.
3. En la pestaña *Opciones avanzadas*, haga clic en *Variables de entorno*.
4. Haga clic en *Nueva* para crear una variable de entorno nueva y llámela `JAVA_HOME`.
5. Configure el valor de la variable de entorno. Por ejemplo, para `C:\Programs\openjdk\bin\java.exe` defina el valor como `C:\Programs\openjdk`.

## 4 Línea de comandos de StyleVision Server

### Ubicación predeterminada del ejecutable de StyleVision Server

A continuación se muestra la ubicación predeterminada del ejecutable de StyleVision Server en las plataformas compatibles:

<i>Linux</i>	<code>/opt/Altova/StyleVisionServer2024/bin/<b>stylevisionserver</b></code>
<i>Mac</i>	<code>/usr/local/Altova/StyleVisionServer2024/bin/<b>stylevisionserver</b></code>
<i>Windows</i>	<code>&lt;CarpetaArchivosPrograma&gt;\Altova\StyleVisionServer2024\bin\<b>StyleVisionServer.exe</b></code>

### Uso de la línea de comandos y lista de comandos

La sintaxis de la línea de comandos es:

```
stylevisionserver --h | --help | --version | <comando> [opciones] [argumentos]
```

- `--help` (`--h` en versión corta) muestra el texto de ayuda del comando dado. Si no se indica ningún comando, entonces se enumeran todos los comandos del ejecutable, cada uno con una breve descripción.
- `--version` muestra el número de versión de StyleVision Server.
- `<command>` es el comando que se ejecuta. Los comandos se describen en las subsecciones de este apartado (véase la lista más abajo).
- `[opciones]` son las opciones de un comando; se enumeran y describen con sus comandos correspondientes.
- `[argumentos]` son los argumentos de un comando; se enumeran y describen con sus comandos correspondientes.

#### ▼ Uso de mayúsculas/minúsculas y de barras diagonales en la línea de comandos

`StyleVisionServer` en Windows  
`stylevisionserver` en Unix (Linux, Mac)

\* Las minúsculas (`stylevisionserver`) funcionan en todas las plataformas (Windows, Linux y Mac), mientras que la mezcla de mayúsculas y minúsculas (`StyleVisionServer`) sólo funciona en Windows y Mac.

\* En Linux y Mac utilice barras diagonales. En Windows utilice barras diagonales inversas.

### Comandos CLI

A continuación se enumeran los comandos disponibles, que se explican en las subsecciones de este apartado.

- [assignlicense](#)<sup>37</sup>: carga una licencia a LicenseServer y asigna esa licencia a StyleVision Server.
- [exportresourcestrings](#)<sup>39</sup>: exporta todas las cadenas de los recursos de la aplicación a un archivo XML.
- [generate](#)<sup>41</sup>: genera uno o varios documentos a partir de un archivo XML de entrada y una hoja de estilos XSLT en el archivo PXF de entrada.

- [help](#)<sup>46</sup>: muestra información sobre el comando que se envía en el argumento (o sobre todos los comandos si no se envía ningún argumento).
- [licenseserver](#)<sup>47</sup>: registra StyleVision Server con un LicenseServer de la red local.
- [pdfdata](#)<sup>49</sup>: genera datos de formularios a partir de un archivo PDF en un archivo FDF o XML.
- [setdeflang](#)<sup>51</sup>: establece el idioma predeterminado de StyleVision Server.
- [setfopath](#)<sup>52</sup>: selecciona un procesador FO alternativo para generar archivos PDF.
- [verifylicense](#)<sup>54</sup>: comprueba si StyleVision Server tiene una licencia y si usa la clave de licencia dada.
- [version](#)<sup>56</sup>: muestra el número de versión de StyleVision Server.

## 4.1 assignlicense

### Sintaxis y descripción

Al ejecutarse el comando `assignlicense` carga el archivo de licencia indicado por el argumento `ARCHIVO` en el servidor `LicenseServer` con el que está registrado `StyleVision Server` (véase el comando `licenseserver`) y asigna esa licencia a `StyleVision Server` en este equipo. El argumento `ARCHIVO` toma la ruta de acceso del archivo de licencia. El comando también permite comprobar la validez de las licencias.

```
stylevisionserver assignlicense [opciones] ARCHIVO
```

- El argumento `FILE` toma la ruta al archivo de la licencia.
- La opción `--test-only` sirve para cargar la licencia a `LicenseServer` y validarla sin asignarla primero a `StyleVision Server`.

Para más información sobre el proceso de asignación de licencias consulte la documentación de Altova `LicenseServer` (<https://www.altova.com/manual/es/licenseserver/3.14/>).

#### ▼ Mayúsculas/minúsculas y barras en la línea de comandos

`stylevisionserver` en Windows

`stylevisionserver` en Windows y Unix (Linux, Mac)

\* Las minúsculas (`stylevisionserver`) funcionan en todas las plataformas (Windows, Linux y Mac), mientras que las mayúsculas (`StyleVisionServer`) funcionan solamente en Windows y Mac.

\* Use la barra diagonal en Linux y Mac y la barra diagonal inversa en Windows.

#### ▼ Barra diagonal inversa y espacios en sistemas Windows

En sistemas Windows: si hay espacios o caracteres especiales (por ejemplo en los nombres de archivos o de carpetas, empresas, personas o productos) debe usar comillas: por ejemplo, `"Mi archivo"`. Sin embargo, debe tener en cuenta que una barra diagonal inversa seguida de comillas dobles (por ejemplo, `"C:\Mi Directorio\"`) es posible que no se lea correctamente. Esto se debe a que la barra diagonal inversa también se usa para empezar secuencias de escape y la secuencia de escape `\"` también usa las comillas dobles. Para evitar secuencia de caracteres puede añadir otra barra diagonal inversa: `\\\"`. En resumen: si necesita escribir una ruta que contenga espacios y una barra diagonal inversa, esta es la mejor forma de hacerlo: `"C:\Mi Directorio\\"`.

## Ejemplos

Ejemplos del comando `assignlicense`:

```
stylevisionserver assignlicense C:\licensepool\mylicensekey.altova_licenses
stylevisionserver assignlicense --test-only=true C:
\licensepool\mylicensekey.altova_licenses
```

- El primer ejemplo carga la licencia indicada a `LicenseServer` y se la asigna a `StyleVision Server`.
- El tercer comando carga la licencia indicada a `LicenseServer` y la valida, sin asignársela a `StyleVision Server`.

## Opciones

Las opciones se enumeran en versión corta (si existe) y larga. Puede usar una o dos barras tanto para la versión corta como para la larga. Las opciones pueden tomar un valor o no hacerlo. Si lo hacen, se escribe así: `--option=value`. Los valores se pueden indicar sin comillas, salvo en dos casos: (i) cuando la cadena del valor contiene espacios o (ii) cuando se indica de forma explícita en la descripción de la opción que las comillas son necesarias. Si una opción toma un valor booleano y no se especifica ningún valor, entonces el valor predeterminado de la opción es `TRUE`. Use la opción `--h, --help` para mostrar información sobre el comando.

### ▼ test-only [t]

`--t, --test-only = true|false`

Son valores permitidos: `true|false`. Si su valor es `true`, el archivo de licencia solamente se carga y valida en LicenseServer, pero no se asigna:

## 4.2 exportresourcestrings

### Sintaxis y descripción

El comando `exportresourcestrings` genera un archivo XML que contiene todas las cadenas de recursos de la aplicación StyleVision Server en el idioma indicado. Los idiomas en los que se puede generar el archivo son inglés (`en`), español (`es`), francés (`fr`), alemán (`de`) y japonés (`ja`).

```
stylevisionserver exportresourcestrings [opciones] CódigoIdioma XMLOutputFile
```

- El argumento *LanguageCode* indica el idioma de las cadenas de recursos del archivo XML de salida; se trata del *lenguaje de exportación*. Se admiten estos idiomas de exportación (se indica el código correspondiente entre paréntesis): inglés (`en`), alemán, (`de`), español (`es`), francés (`fr`) y japonés (`ja`).
- El argumento *XMLOutputFile* indica la ruta y el nombre del archivo XML de salida.

A continuación explicamos cómo localizar.

#### ▼ Mayúsculas/minúsculas y barras en la línea de comandos

`stylevisionserver` en Windows

`stylevisionserver` en Windows y Unix (Linux, Mac)

\* Las minúsculas (`stylevisionserver`) funcionan en todas las plataformas (Windows, Linux y Mac), mientras que las mayúsculas (`StyleVisionServer`) funcionan solamente en Windows y Mac.

\* Use la barra diagonal en Linux y Mac y la barra diagonal inversa en Windows.

#### ▼ Barra diagonal inversa y espacios en sistemas Windows

En sistemas Windows: si hay espacios o caracteres especiales (por ejemplo en los nombres de archivos o de carpetas, empresas, personas o productos) debe usar comillas: por ejemplo, "`Mi archivo`". Sin embargo, debe tener en cuenta que una barra diagonal inversa seguida de comillas dobles (por ejemplo, "`C:\Mi Directorio\`") es posible que no se lea correctamente. Esto se debe a que la barra diagonal inversa también se usa para empezar secuencias de escape y la secuencia de escape `\` también usa las comillas dobles. Para evitar secuencia de caracteres puede añadir otra barra diagonal inversa: `\\`". En resumen: si necesita escribir una ruta que contenga espacios y una barra diagonal inversa, esta es la mejor forma de hacerlo: "`C:\Mi Directorio\\`".

### Ejemplos

Ejemplos del comando `exportresourcestrings`:

```
stylevisionserver exportresourcestrings de c:\Strings.xml
```

- El comando crea un archivo llamado `Strings.xml` en la carpeta `c:\`; ese archivo contiene las cadenas de los recursos de **StyleVision Server** en inglés.

### Localizar StyleVision Server en otros idiomas

Si quiere puede localizar StyleVision Server en cualquier idioma. Altova ya ofrece la aplicación en cinco idiomas: inglés, español, francés, alemán y japonés (todos los archivos están en la carpeta `C:\Archivos de`

programa (x86)\Altova\StyleVisionServer2024\bin) pero puede localizarla en cualquier otro idioma.

Siga estos pasos para localizar la aplicación:

1. Genere un archivo XML con las cadenas de recursos usando el comando `exportresourcestrings` (ver *más arriba*). Las cadenas de recursos de este archivo XML puede estar en uno de estos idiomas: inglés (`en`), español (`es`), francés (`fr`), alemán (`de`) o japonés (`ja`), dependiendo del argumento `códigoIdioma` que utilice con el comando.
2. Traduzca las cadenas de recursos al idioma de destino. Las cadenas de recursos son el contenido de los elementos `<string>` del archivo XML. No traduzca las variables que aparecen entre llaves, p.ej. `{option}` o `{product}`.
3. Póngase en contacto con [el equipo de soporte técnico de Altova](#), que le ayudarán a generar un archivo DLL localizado de StyleVision Server a partir de su archivo XML traducido.
4. Cuando reciba el archivo DLL localizado del equipo de [soporte técnico de Altova](#), guárdelo en la carpeta `C:\Archivos de programa (x86)\Altova\StyleVisionServer2024\bin`. El DLL tendrá un nombre similar a este `StyleVisionServer2024_ci.dll`. La parte `_ci` del nombre contiene el código del idioma. Por ejemplo, en `StyleVisionServer2024_de.dll`, la parte `de` es el código del idioma alemán (Deutsch).
1. Ejecute el comando `setdeflang` para establecer el archivo DLL localizado como aplicación StyleVision Server predeterminada. Use el código de idioma del nombre del archivo DLL como argumento del comando `setdeflang`

**Nota:** Altova ya ofrece StyleVision Server en estos cinco idiomas: inglés, español, francés, alemán y japonés. Para usar uno de estos idiomas como idioma predeterminado, use el comando `setdeflang` de StyleVision Server.

## 4.3 generate

### Sintaxis y descripción

El comando `generate` (`gen` en versión abreviada) genera uno o más archivos de salida (HTML, PDF, RTF y/o DOCX) al transformar el archivo XML de entrada con los documentos XSLT que contiene el archivo PXF de entrada.

```
stylevisionserver generate | gen --inputxml=Filename [opciones adicionales] InputPXF
```

- La opción `--inputxml` es obligatoria y da la ruta de acceso del archivo XML.
- El argumento `InputPXF` indica la ruta de acceso al archivo PXF que contiene los documentos XSLT que se usan para generar los documentos de salida. Los archivos PXF se crean con la [aplicación de Altova StyleVision](#).
- Cada formato de salida se genera indicando la opción correspondiente para ese resultado (véase la *lista Opciones más abajo*). El valor de cada una de las opciones es una ruta que indica dónde se debe generar el resultado.

**Nota:** StyleVision Server usa [Apache FOP](#), el procesador FO del proyecto Apache, para generar archivos PDF a partir de FO. Apache FOP se instala automáticamente con StyleVision Server en estas ubicaciones: en sistemas Windows, en `ProgramData\Altova\SharedBetweenVersions`; en sistemas Linux y macOS, en una subcarpeta de `StyleVisionServer2024`. Recuerde que para poder usar Apache FOP es necesario tener instalado **Java Runtime Environment 1.8 o superior** en el equipo de StyleVision Server. Debe instalar la versión de Java que corresponda a su versión de StyleVision Server (32 o 64 bits, respectivamente). Para más información sobre cómo configurar FOP consulte el apartado [Requisitos para FOP](#) <sup>34</sup>.

#### ▼ AltovaFOPServer para acelerar la generación de PDF

Si quiere ejecutar múltiples transformaciones a PDF, especialmente de documentos de gran tamaño, puede que desee usar AltovaFOPServer para realizar el trabajo más rápidamente. La mayor velocidad se debe a que AltovaFOPServer mantiene cargados los archivos Java VM y FOP jar después de la primera transformación, ahorrando así el tiempo para cargar estos archivos de nuevo cada vez. AltovaFOPServer se instalará junto con su paquete StyleVision. Para poder utilizarlo para transformaciones, primero debe iniciar AltovaFOPServer a través de la línea de comandos.

#### Iniciar AltovaFOPServer

La sintaxis del comando para iniciar AltovaFOPServer es:

```
java -cp "{classPath}" {className} --pid "{pidFile}" --port {portNumbers}
```

Un comando de la línea de comandos para iniciar AltovaFOPServer podría tener este aspecto:

```
java -cp "C:\Archivos de programa\Altova\Common2024\jar\gson\gson.jar;C:\Archivos de programa\Altova\Common2024\jar\AltovaFOPServer.jar" com.altova.stylevision.fopserver.Principal --pid "C:\Usuarios\usuario\AppData\Local\Temp\AltovaFopServer.pid" --puerto 9090-9103,9999
```

El argumento `classPath` debe contener las rutas de acceso a los archivos `gson.jar` y `AltovaFOPServer.jar`. Introduzca la ubicación correcta para encontrar estos archivos.

### Llamar a AltovaFOPServer en StyleVision Server

Una vez haya iniciado AltovaFOPServer, puede llamarlo desde StyleVision Server usando el comando `generate` o con la opción `altova-fopserver-pid-file` o `altova-fopserver-port` (ver *Opciones más abajo*).

#### ▼ Mayúsculas/minúsculas y barras en la línea de comandos

`stylevisionserver` en Windows

`stylevisionserver` en Windows y Unix (Linux, Mac)

\* Las minúsculas (`stylevisionserver`) funcionan en todas las plataformas (Windows, Linux y Mac), mientras que las mayúsculas (`StyleVisionServer`) funcionan solamente en Windows y Mac.

\* Use la barra diagonal en Linux y Mac y la barra diagonal inversa en Windows.

#### ▼ Barra diagonal inversa y espacios en sistemas Windows

En sistemas Windows: si hay espacios o caracteres especiales (por ejemplo en los nombres de archivos o de carpetas, empresas, personas o productos) debe usar comillas: por ejemplo, "Mi archivo". Sin embargo, debe tener en cuenta que una barra diagonal inversa seguida de comillas dobles (por ejemplo, "C:\Mi Directorio\") es posible que no se lea correctamente. Esto se debe a que la barra diagonal inversa también se usa para empezar secuencias de escape y la secuencia de escape `\ "` también usa las comillas dobles. Para evitar secuencia de caracteres puede añadir otra barra diagonal inversa: `\\ "`. En resumen: si necesita escribir una ruta que contenga espacios y una barra diagonal inversa, esta es la mejor forma de hacerlo: `"C:\Mi Directorio\\"`.

## Ejemplos

Ejemplos del comando `generate`:

```
stylevisionserver generate --inputxml=C:\MyFiles\ExpReport.xml --html=Test.html
ExpReport.pxf
stylevisionserver generate --inputxml=C:\ExpReport.pxf | zip\ExpReport.xml --
html=Test.html ExpReport.pxf
stylevisionserver generate --inputxml=altova://packagedfile/ExpReport.xml --
html=Test.html ExpReport.pxf
stylevisionserver generate --inputxml=ExternalXML.xml --html=Test.html Test.pxf
```

- Estos ejemplos utilizan la opción obligatoria `--inputxml`, el argumento `PXFEntrada` (`Test.pxf`) y un mínimo de una **opción de formato de salida** (`--html` en todos los ejemplos mencionados).
- El archivo XML de entrada que se debe usar puede encontrarse dentro del archivo PXF (ver el segundo y tercer ejemplo más arriba) o puede ser un archivo XML externo (ubicado fuera del archivo PXF, como en el primer y cuarto ejemplo).
- El modificador `--inputxml` se ignora si el esquema fuente principal es de BD o BD-XML, pero debe estar presente por motivos sintácticos y será necesario algo como `--inputxml=baseDedatos`.
- Si la opción de formato de salida `--html` toma una ruta de acceso relativa, como en los ejemplos más arriba, entonces la ubicación del archivo de salida será relativa a la carpeta donde está guardado el archivo PXF.

## Opciones

### inputxml [xml]

`--xml, --inputxml = PathToXMLFile`

Esta opción es obligatoria. Indica la ruta de acceso del archivo XML que se quiere procesar. El archivo XML se puede encontrar dentro o fuera del archivo PXF. Para indicar archivos XML que estén dentro de un archivo PXF debe usar el localizador `|zip` (véase la parte resaltada en los ejemplos anteriores). La opción `--inputxml` se omite si la fuente principal del esquema del archivo PXF de entrada es una BD o BD-XML.

### dbwhere [dbw]

`--dbw, --dbwhere = WHEREClause`

Una cláusula SQL `WHERE` que determina qué filas de una fuente BD-XML se deben procesar.

### param [p]

`--p, --param = $ParamName:ParamValue`

Asigna un valor a un parámetro definido en el archivo PXF. El conmutador `--param` se debe usar antes de cada parámetro. Si `NombreParám` o `ValorParám` contiene un espacio, utilice comillas. Ejemplo: `--p=$company:"Nanonull Inc"`

### prohibit-output-outside-target-folder

`--prohibit-output-outside-target-folder = true|false`

Los valores son `true|false`. Si su valor es `true`, no permite crear documentos de salida en ninguna otra carpeta que no sea la carpeta donde se crea el archivo de salida principal (HTML, PDF, RTF, DOCX, FO). De esta forma se pueden proteger otras carpetas si hace falta. El valor predeterminado es `false`.

### outhtml [html]

`--html, --outhtml = FilePath`

La ruta de acceso al archivo HTML que se quiere generar.

### outpdf [pdf]

`--pdf, --outpdf = FilePath`

La ruta de acceso al archivo PDF que se quiere generar.

### outrtf [rtf]

`--rtf, --outrtf = FilePath`

La ruta de acceso al archivo RTF que se quiere generar.

### outdocx [docx]

`--docx, --outdocx = FilePath`

La ruta de acceso al archivo DOCX que se quiere generar.

### outtext [text]

`--text, --outtext = FilePath`

La ruta de acceso al archivo de texto que se quiere generar.

## ▼ outfo [fo]

```
--fo, --outfo = FilePath
```

La ruta de acceso al archivo FO que se quiere generar.

## ▼ generate-html-output-as-mime

```
--generate-html-output-as-mime = true|false
```

Los valores son `true|false`. Si no se indica esta opción, el valor predeterminado es `false`. Si se indica pero no se le asigna ningún valor, entonces se lee como `true`. Si su valor es `true`, entonces el resultado HTML se genera como una cadena de texto tipo MIME.

## ▼ altova-fopserver-pid-file

```
--altova-fopserver-pid-file = PathToFile
```

La ruta de acceso al archivo PID que se usa para conectarse a un AltovaFOPServer en ejecución. Puede conectarse a un AltovaFOPDServer a través del archivo PID del servidor o de un puerto. Es decir que puede usar esta opción de PID como alternativa a usar un puerto (*véase la siguiente opción*). Para obtener más información sobre cómo usar AltovaFOPServer, consulte la descripción general del comando `generate` más arriba.

## ▼ altova-fopserver-port

```
--altova-fopserver-port = Value
```

Un número de puerto o una lista de números de puerto que se usan para conectarse a un AltovaFOPServer en ejecución. Los números de puerto de una lista se separan por comas. En lugar de un único número en la lista, también puede utilizar un rango de números. Por tanto, puede introducir `9090-9100,9999` como valor de esta opción. Recuerde que puede conectarse a un AltovaFOPDServer a través del archivo PID del servidor AltovaFOPServer o de un puerto. Es decir que puede usar esta opción de puerto como alternativa a usar un archivo PID (*véase la opción anterior*). Para obtener más información sobre cómo usar AltovaFOPServer, consulte la descripción general del comando `generate` más arriba.

## ▼ taxonomy-package

```
--taxonomy-package = FilePath
```

La ruta de acceso a otro paquete de taxonomía. Agregue esta opción varias veces para indicar más de un paquete de taxonomía.

## ▼ taxonomy-packages-config-file

```
--taxonomy-packages-config-file = FilePath
```

La ruta de acceso al archivo `TaxonomyPackagesConfig.json`, que es un archivo común que usan los productos de Altova XMLSpy, MapForce y StyleVision para configurar taxonomías XBRL que se usan en todos estos productos.

## ▼ verbose [v]

```
--v, --verbose = true|false
```

Los valores son `true|false`. Activa o desactiva todos los mensajes que aparecen. Si no incluye esta opción, el valor predeterminado es `false`. Si incluye la opción pero no indica ningún valor, entonces se usa `true`.

## ▼ lang [l]

```
--l, --lang = en|de|es|fr|ja
```

El idioma utilizado para la presentación de los mensajes.

Use la opción `--h, --help` para mostrar información sobre el comando.

Las opciones se enumeran en versión corta (si existe) y larga. Puede usar una o dos barras tanto para la versión corta como para la larga. Las opciones pueden tomar un valor o no hacerlo. Si lo hacen, se escribe así: `--option=value`. Los valores se pueden indicar sin comillas, salvo en dos casos: (i) cuando la cadena del valor contiene espacios o (ii) cuando se indica de forma explícita en la descripción de la opción que las comillas son necesarias. Si una opción toma un valor booleano y no se especifica ningún valor, entonces el valor predeterminado de la opción es `TRUE`. Use la opción `--h, --help` para mostrar información sobre el comando.

## Catálogos

Si usa el mecanismo de catalogación de Altova encontrará los archivos relevantes en la carpeta `etc` de la carpeta de la aplicación StyleVision Server. Para leer información más detallada consulte [la sección Catálogos del manual de Altova StyleVision](#).

Puede crear el archivo `CustomCatalog.xml` a partir del archivo de plantilla `CustomCatalog_template.xml`. Asegúrese de que cambia el nombre del archivo de plantilla a `CustomCatalog.xml`, ya que es este el que se usa en el mecanismo de catalogación (y no la plantilla).

Tenga en cuenta que:

- Durante una instalación nueva de la misma versión principal (misma u otra subversión), el archivo de plantilla se sustituye por uno nuevo, pero `CustomCatalog.xml` no cambia.
- Sin embargo, si instala una versión principal nueva sobre una versión principal anterior, la carpeta de la instalación anterior se borra junto con su archivo `CustomCatalog.xml`. Por tanto, si quiere seguir usando el `CustomCatalog.xml` de la versión anterior debe guardarlo en una ubicación segura. Después de instalar la versión principal nueva puede copiar el archivo `CustomCatalog.xml` guardado en la carpeta `etc` de la versión principal nueva y editarlo allí como necesite.

## 4.4 help

### Sintaxis y descripción

El comando `help` toma un único argumento (`Command`), que es el nombre del comando para el que necesita la ayuda, y muestra la sintaxis del comando, sus opciones y otra información relevante. Si no se especifica el comando `Command`, entonces se enumeran todos los comandos del ejecutable, cada uno con una breve descripción.

```
stylevisionserver help Command
```

#### ▼ Mayúsculas/minúsculas y barras en la línea de comandos

`StyleVisionServer` en Windows

`stylevisionserver` en Windows y Unix (Linux, Mac)

\* Las minúsculas (`stylevisionserver`) funcionan en todas las plataformas (Windows, Linux y Mac), mientras que las mayúsculas (`StyleVisionServer`) funcionan solamente en Windows y Mac.

\* Use la barra diagonal en Linux y Mac y la barra diagonal inversa en Windows.

### Ejemplo

Ejemplo del comando `help` para mostrar información sobre el comando `licenseserver`:

```
stylevisionserver help licenseserver
```

### La opción `--help`

También puede acceder a la información de ayuda sobre un comando usando la opción `--help` del comando para el que necesita la ayuda. Los comandos siguientes obtienen el mismo resultado:

```
stylevisionserver licenseserver --help
```

El comando anterior usa la opción `--help` del comando `licenseserver`.

```
stylevisionserver help licenseserver
```

El comando `help` toma como argumento `licenseserver`.

En ambos casos, aparece información de ayuda sobre el comando `licenseserver`.

## 4.5 licenseserver

### Sintaxis y descripción

Al ejecutarse, el comando `licenseserver` registra StyleVision Server con el servidor LicenseServer indicado por el argumento *Servidor-O-Dirección-IP*. Para que el comando `licenseserver` se ejecute correctamente, los dos servidores (StyleVision Server y LicenseServer) deben estar conectados en la red y LicenseServer debe estar en ejecución. Además debe tener privilegios de administrador para poder registrar StyleVision Server con LicenseServer.

```
stylevisionserver licenseserver [opciones] Server-Or-IP-Address
```

- El argumento *Server-Or-IP-Address* toma el nombre o la dirección IP del equipo en el que se está ejecutando LicenseServer.

Cuando StyleVision Server se registre con LicenseServer, recibirá un mensaje de confirmación. El mensaje incluirá la URL del servidor LicenseServer. Ahora puede usar la URL para ir a LicenseServer y asignarle una licencia a StyleVision Server. Consulte la documentación de Altova LicenseServer para obtener más información (<https://www.altova.com/manual/es/licenseserver/3.14/>).

#### ▼ Mayúsculas/minúsculas y barras en la línea de comandos

`stylevisionserver` en Windows

`stylevisionserver` en Windows y Unix (Linux, Mac)

\* Las minúsculas (`stylevisionserver`) funcionan en todas las plataformas (Windows, Linux y Mac), mientras que las mayúsculas (`StyleVisionServer`) funcionan solamente en Windows y Mac.

\* Use la barra diagonal en Linux y Mac y la barra diagonal inversa en Windows.

#### ▼ Barra diagonal inversa y espacios en sistemas Windows

En sistemas Windows: si hay espacios o caracteres especiales (por ejemplo en los nombres de archivos o de carpetas, empresas, personas o productos) debe usar comillas: por ejemplo, `"Mi archivo"`. Sin embargo, debe tener en cuenta que una barra diagonal inversa seguida de comillas dobles (por ejemplo, `"C:\Mi Directorio\"`) es posible que no se lea correctamente. Esto se debe a que la barra diagonal inversa también se usa para empezar secuencias de escape y la secuencia de escape `\"` también usa las comillas dobles. Para evitar secuencia de caracteres puede añadir otra barra diagonal inversa: `\\\"`. En resumen: si necesita escribir una ruta que contenga espacios y una barra diagonal inversa, esta es la mejor forma de hacerlo: `"C:\Mi Directorio\\"`.

### Ejemplo

Ejemplo del comando `licenseserver`:

```
stylevisionserver licenseserver DOC.altova.com
stylevisionserver licenseserver localhost
stylevisionserver licenseserver 127.0.0.1
```

Estos comandos indican respectivamente que el equipo que ejecuta el servidor Altova LicenseServer es un

equipo llamado `DOC.altova.com` y el equipo del usuario (`localhost` y `127.0.0.1`). En cada caso el comando registra StyleVision Server con el servidor LicenseServer del equipo correspondiente. El último comando indica al ejecutable del servidor que ejecute el comando.

## Opciones

Las opciones se enumeran en versión corta (si existe) y larga. Puede usar una o dos barras tanto para la versión corta como para la larga. Las opciones pueden tomar un valor o no hacerlo. Si lo hacen, se escribe así: `--option=value`. Los valores se pueden indicar sin comillas, salvo en dos casos: (i) cuando la cadena del valor contiene espacios o (ii) cuando se indica de forma explícita en la descripción de la opción que las comillas son necesarias. Si una opción toma un valor booleano y no se especifica ningún valor, entonces el valor predeterminado de la opción es `TRUE`. Use la opción `--h, --help` para mostrar información sobre el comando.

### ▼ json [j]

`--j, --json = true|false`

Usa los valores `true|false`. Si se establece en `true` imprime el resultado del intento de registro como objeto JSON analizable por máquina.

## 4.6 pdfdata

### Sintaxis y descripción

El comando `pdfdata` genera un archivo FDF o XML a partir del archivo PDF dado como argumento *InputPDF*.

```
stylevisionserver pdfdata [opciones] InputPDF
```

- El argumento *InputPDF* indica la ruta de acceso al archivo PDF a partir del cual se genera el archivo FDF o XML de salida. Si el archivo PDF no contiene datos de formulario, tampoco los contiene el archivo generado.
- Use la opción `--outfdf` para indicar la ubicación del archivo PDF generado y la opción `--outxml` para indicar la ubicación del archivo XML.

Para más información sobre cómo configurar FOP, que StyleVision Server usa por defecto para generar archivos PDF, consulte el apartado [Requisitos para FOP](#)<sup>34</sup>.

Para más información sobre archivos FDF y los formularios PDF para rellenar consulte la documentación de [Altova StyleVision \(Enterprise Edition\)](#).

#### ▼ Mayúsculas/minúsculas y barras en la línea de comandos

`stylevisionserver` en Windows

`stylevisionserver` en Windows y Unix (Linux, Mac)

\* Las minúsculas (`stylevisionserver`) funcionan en todas las plataformas (Windows, Linux y Mac), mientras que las mayúsculas (`StyleVisionServer`) funcionan solamente en Windows y Mac.

\* Use la barra diagonal en Linux y Mac y la barra diagonal inversa en Windows.

#### ▼ Barra diagonal inversa y espacios en sistemas Windows

En sistemas Windows: si hay espacios o caracteres especiales (por ejemplo en los nombres de archivos o de carpetas, empresas, personas o productos) debe usar comillas: por ejemplo, `"Mi archivo"`. Sin embargo, debe tener en cuenta que una barra diagonal inversa seguida de comillas dobles (por ejemplo, `"C:\Mi Directorio\"`) es posible que no se lea correctamente. Esto se debe a que la barra diagonal inversa también se usa para empezar secuencias de escape y la secuencia de escape `\"` también usa las comillas dobles. Para evitar secuencia de caracteres puede añadir otra barra diagonal inversa: `\\\"`. En resumen: si necesita escribir una ruta que contenga espacios y una barra diagonal inversa, esta es la mejor forma de hacerlo: `"C:\Mi Directorio\\\"`.

### Ejemplos

Ejemplos de uso del comando `pdfdata`:

```
stylevisionserver pdfdata --outfdf=C:\test\formularios\DatosFDF.fdf C:\test\formularios\TestFormulario.pdf
stylevisionserver pdfdata --outxml=C:\test\formularios\DatosXML.xml C:\test\formularios\TestFormulario.pdf
```

Estos ejemplos crean, respectivamente, un archivo FDF y un archivo XML a partir del mismo archivo PDF de entrada.

## Opciones

### ▼ outfdf

`--outfdf = FilePath`

La ruta de acceso al archivo FDF generado.

### ▼ outxml

`--outxml = FilePath`

La ruta de acceso al archivo XML generado.

### ▼ verbose [v]

`--v, --verbose = true|false`

Los valores son `true|false`. Activa o desactiva todos los mensajes que aparecen. Si no incluye esta opción, el valor predeterminado es `false`. Si incluye la opción pero no indica ningún valor, entonces se usa `true`.

Use la opción `--h, --help` para mostrar información sobre el comando.

Las opciones se enumeran en versión corta (si existe) y larga. Puede usar una o dos barras tanto para la versión corta como para la larga. Las opciones pueden tomar un valor o no hacerlo. Si lo hacen, se escribe así: `--option=value`. Los valores se pueden indicar sin comillas, salvo en dos casos: (i) cuando la cadena del valor contiene espacios o (ii) cuando se indica de forma explícita en la descripción de la opción que las comillas son necesarias. Si una opción toma un valor booleano y no se especifica ningún valor, entonces el valor predeterminado de la opción es `TRUE`. Use la opción `--h, --help` para mostrar información sobre el comando.

## 4.7 setdeflang

### Sintaxis y descripción

El comando `setdeflang` (`sdl` en versión corta) establece el idioma predeterminado de StyleVision Server. Los idiomas disponibles son inglés (`en`), alemán (`de`), español (`es`), francés (`fr`) y japonés (`ja`). El comando toma el argumento obligatorio `LanguageCode`.

```
stylevisionserver setdeflang [opciones] CódigoIdioma
```

- El argumento `CódigoIdioma` es obligatorio y define el idioma predeterminado de StyleVision Server. Los valores correspondientes son: `en`, `de`, `es`, `fr`, `ja`.
- Use la opción `--h`, `--help` para mostrar información sobre el comando.

#### ▼ Mayúsculas/minúsculas y barras en la línea de comandos

`stylevisionServer` en Windows

`stylevisionserver` en Windows y Unix (Linux, Mac)

\* Las minúsculas (`stylevisionserver`) funcionan en todas las plataformas (Windows, Linux y Mac), mientras que las mayúsculas (`StylevisionServer`) funcionan solamente en Windows y Mac.

\* Use la barra diagonal en Linux y Mac y la barra diagonal inversa en Windows.

### Ejemplo

Ejemplos del comando `setdeflang` (`sdl`):

```
stylevisionserver sdl de
stylevisionserver setdeflang es
```

- El primer comando define el alemán como idioma predeterminado de StyleVision Server.
- El segundo comando define el español como idioma predeterminado de StyleVision Server.

### Opciones

Use la opción `--h`, `--help` para mostrar información sobre el comando.

## 4.8 setfopath

### Sintaxis y descripción

El comando `setfopath` (`sfp` en versión abreviada) indica la ruta de acceso a un procesador Apache FOP distinto al que viene incluido en el paquete de StyleVision Server.

```
stylevisionserver setfopath | sfp [options] Path
```

- El procesador Apache FOP que viene incluido con StyleVision Server se usa por defecto para procesar los documentos FO y generar archivos PDF de salida. Si quiere usar una instancia de procesador Apache FOP distinta a la que viene integrada en StyleVision Server use el comando `setfopath` con el argumento `Path` e indique la ruta de acceso al procesador FO que quiere usar.
- Una vez haya indicado un procesador FO alternativo con el comando `setfopath`, ese será el procesador que se use al generar archivos PDF más adelante con el comando [generate](#)<sup>41</sup>. Para volver a cambiar el procesador solo tiene que volver a usar el comando `setfopath`. Para que StyleVision Server vuelva a usar el procesador FOP debe ubicar la carpeta FOP en su sistema y usar esa ruta de acceso como argumento para `setfopath`.
- En sistemas Windows la carpeta FOP que se instala con StyleVision Server está en `ProgramData\Altova\SharedBetweenVersions`; en sistemas Linux y macOS está en una carpeta bajo la carpeta `StyleVisionServer2024`.

Para más información sobre cómo configurar FOP, que StyleVision Server usa por defecto para generar archivos PDF, consulte el apartado [Requisitos para FOP](#)<sup>34</sup>.

Para más información sobre archivos FDF y los formularios PDF para rellenar consulte la documentación de [Altova StyleVision \(Enterprise Edition\)](#).

#### ▼ Mayúsculas/minúsculas y barras en la línea de comandos

`stylevisionserver` en Windows

`stylevisionserver` en Windows y Unix (Linux, Mac)

\* Las minúsculas (`stylevisionserver`) funcionan en todas las plataformas (Windows, Linux y Mac), mientras que las mayúsculas (`StyleVisionServer`) funcionan solamente en Windows y Mac.

\* Use la barra diagonal en Linux y Mac y la barra diagonal inversa en Windows.

#### ▼ Barra diagonal inversa y espacios en sistemas Windows

En sistemas Windows: si hay espacios o caracteres especiales (por ejemplo en los nombres de archivos o de carpetas, empresas, personas o productos) debe usar comillas: por ejemplo, `"Mi archivo"`. Sin embargo, debe tener en cuenta que una barra diagonal inversa seguida de comillas dobles (por ejemplo, `"C:\Mi Directorio\"`) es posible que no se lea correctamente. Esto se debe a que la barra diagonal inversa también se usa para empezar secuencias de escape y la secuencia de escape `\"` también usa las comillas dobles. Para evitar secuencia de caracteres puede añadir otra barra diagonal inversa: `\\\"`. En resumen: si necesita escribir una ruta que contenga espacios y una barra diagonal inversa, esta es la mejor forma de hacerlo: `"C:\Mi Directorio\\"`.

## Ejemplos

Una vez haya ejecutado el comando `setfopath` puede usar el comando `generate`<sup>41</sup> para generar archivos PDF usando el procesador FO que acaba de especificar.

```
stylevisionserver setfopath C:\FOP\FOP.bat
stylevisionserver generate --inputxml=Test.xml --pdf=Test.pdf Test.pxf
```

Estos comandos hacen lo siguiente:

1. El comando `setfopath` indica que el procesador ubicado en `C:\FOP\FOP.bat` debe usarse para generar archivos PDF más adelante con el comando correspondiente.
2. El comando `generate` genera un archivo PDF a partir de los datos XML de entrada indicados, para lo que usa los archivos de transformación que contiene el archivo PXF. El procesador FO indicado en el comando anterior se usa para generar el PDF.

## Opciones

Use la opción `--h, --help` para mostrar información sobre el comando.

Las opciones se enumeran en versión corta (si existe) y larga. Puede usar una o dos barras tanto para la versión corta como para la larga. Las opciones pueden tomar un valor o no hacerlo. Si lo hacen, se escribe así: `--option=value`. Los valores se pueden indicar sin comillas, salvo en dos casos: (i) cuando la cadena del valor contiene espacios o (ii) cuando se indica de forma explícita en la descripción de la opción que las comillas son necesarias. Si una opción toma un valor booleano y no se especifica ningún valor, entonces el valor predeterminado de la opción es `TRUE`. Use la opción `--h, --help` para mostrar información sobre el comando.

## 4.9 verifylicense

### Sintaxis y descripción

El comando `verifylicense` comprueba si un producto tiene licencia. La opción `--license-key` también permite comprobar si ya se ha asignado al producto una clave de licencia.

```
stylevisionserver verifylicense [opciones]
```

- Para comprobar si una licencia en concreto está asignada a StyleVision Server debe indicar la clave de licencia como valor de la opción `--license-key`.

Para más información sobre el proceso de asignación de licencias consulte la documentación de LicenseServer (<https://www.altova.com/manual/es/licenseserver/3.14/>).

#### ▼ Mayúsculas/minúsculas y barras en la línea de comandos

`stylevisionserver` en Windows

`stylevisionserver` en Windows y Unix (Linux, Mac)

\* Las minúsculas (`stylevisionserver`) funcionan en todas las plataformas (Windows, Linux y Mac), mientras que las mayúsculas (`StyleVisionServer`) funcionan solamente en Windows y Mac.

\* Use la barra diagonal en Linux y Mac y la barra diagonal inversa en Windows.

### Ejemplos

Ejemplos del comando `verifylicense`:

```
stylevisionserver verifylicense
stylevisionserver verifylicense --license-key=ABCD123-ABCD123-ABCD123-ABCD123-ABCD123-ABCD123
```

- El primer comando comprueba si StyleVision Server tiene licencia.
- El segundo comando comprueba si StyleVision Server usa la clave de licencia indicada en la opción `--license-key`.

### Opciones

Las opciones se enumeran en versión corta (si existe) y larga. Puede usar una o dos barras tanto para la versión corta como para la larga. Las opciones pueden tomar un valor o no hacerlo. Si lo hacen, se escribe así: `--option=value`. Los valores se pueden indicar sin comillas, salvo en dos casos: (i) cuando la cadena del valor contiene espacios o (ii) cuando se indica de forma explícita en la descripción de la opción que las comillas son necesarias. Si una opción toma un valor booleano y no se especifica ningún valor, entonces el valor predeterminado de la opción es `TRUE`. Use la opción `--h, --help` para mostrar información sobre el comando.

#### ▼ license-key [l]

```
--l, --license-key = Value
```

Comprueba si StyleVision Server usa la clave de licencia indicada como valor de esta opción.



## 4.10 version

### Sintaxis y descripción

El comando `version` muestra el número de versión de StyleVision Server.

```
stylevisionserver version
```

#### ▼ Mayúsculas/minúsculas y barras en la línea de comandos

`styleVisionServer` en Windows

`stylevisionserver` en Windows y Unix (Linux, Mac)

\* Las minúsculas (`stylevisionserver`) funcionan en todas las plataformas (Windows, Linux y Mac), mientras que las mayúsculas (`styleVisionServer`) funcionan solamente en Windows y Mac.

\* Use la barra diagonal en Linux y Mac y la barra diagonal inversa en Windows.

### Ejemplo

Ejemplo del comando `version`:

```
stylevisionserver version
```

## 5 API de StyleVision Server

StyleVision Server ofrece una interfaz de programación de aplicaciones (API) a la que puede acceder mediante programación desde su código .NET, Java o COM.

En esta sección de referencia encontrará:

- [Información sobre la interfaz .NET](#) <sup>58</sup>
- [Información sobre la interfaz COM](#) <sup>59</sup>
- [Información sobre la interfaz Java](#) <sup>60</sup>
- [Ejemplos de código](#) <sup>62</sup>
- [Una referencia de la API](#) <sup>71</sup>

## 5.1 Información sobre la interfaz .NET

La interfaz .NET está construida como contenedor alrededor de la interfaz COM. Se ofrece como ensamblado de interoperabilidad primario firmado por Altova y utiliza el espacio de nombres `Altova.StyleVisionServer`.

Durante el proceso de instalación, StyleVision Server se registra automáticamente como objeto de servidor COM, por lo que no es necesario registrarlo a mano. Si recibe un error de acceso, abra los Servicios de componentes y otorgue permiso a la misma cuenta que ejecuta el grupo de aplicaciones que incluye StyleVision Server.

Para poder usar StyleVision Server en su proyecto .NET basta con añadir una referencia al archivo `Altova.StyleVisionServer.dll` (ver instrucciones más abajo). El archivo `Altova.StyleVisionServer.dll` está en la carpeta `bin` de la carpeta de instalación de StyleVision Server. Este archivo DLL se añade automáticamente al caché global de ensamblados (GAC) durante la instalación de StyleVision Server (el GAC suele estar en la carpeta `C:\WINDOWS\assembly`).

Si StyleVision Server está registrado como objeto de servidor COM y el archivo `Altova.StyleVisionServer.dll` está a disposición de la interfaz .NET, las funciones de la API de StyleVision Server estarán disponibles para el proyecto .NET.

**Nota:** Si instaló una versión de 64 bits de StyleVision Server, los archivos de la versión de 32 bits de `Altova.StyleVisionServer.dll` estarán en la carpeta `bin\API_32bit`. Ocurre igual al contrario, si instaló una versión de 32 bits de StyleVision Server, los archivos de la versión de 64 bits de `Altova.StyleVisionServer.dll` estarán en la carpeta `bin\API_64bit`.

### Añadir una referencia al archivo DLL de StyleVision Server en un proyecto Visual Studio .NET

1. Abra el proyecto .NET en Visual Studio y haga clic en **Proyecto | Agregar referencia**. Aparece el cuadro de diálogo "Agregar referencia".
2. Abra la pestaña *Examinar*, navegue hasta la carpeta `<carpeta de aplicación de StyleVisionServer>/bin`, seleccione el archivo `Altova.StyleVisionServer.dll` y haga clic en **Aceptar**.

Para ver la estructura del ensamblado `Altova.StyleVisionServer` utilice el Examinador de objetos de Visual Studio (haciendo clic en **Ver | Examinador de objetos**).

## 5.2 Información sobre la interfaz COM

StyleVision Server se registra automáticamente como objeto de servidor COM durante la instalación. Para comprobar si se registró correctamente, abra el Editor del Registro (p. ej. escriba el comando `regedit.exe` en la línea de comandos). Si el proceso de registro funcionó, el registro incluirá dos clases `StyleVision.Server` (suelen estar en `HKEY_LOCAL_MACHINE\SOFTWARE\Classes`).

Una vez registrado como objeto de servidor COM, podrá invocarlo desde otras aplicaciones y lenguajes de scripting que sean compatibles con llamadas COM. Si quiere cambiar la ubicación de la carpeta de instalación de StyleVision Server, lo mejor es desinstalar StyleVision Server y volver a instalarlo en la ubicación deseada. De esta manera el programa de instalación se encargará de los procesos de anulación de registro y registro automáticamente.

## 5.3 Información sobre la interfaz Java

La API consiste en un archivo JAR (`styleVisionServer.jar`) y un archivo JNI (`styleVisionServer.dll`). Estos dos archivos, así como otros archivos relacionados con la API, están disponibles en la carpeta `bin` de la carpeta de instalación de StyleVision Server. Puede hacer referencia a esos archivos desde su ubicación original o copiarlos a otra ubicación si eso encaja en su configuración del proyecto. (En sistemas Windows necesitará derechos de administrador para ejecutar el programa desde su ubicación original.)

**Nota:** Si instaló una versión de 64 bits de StyleVision Server, los archivos de la versión de 32 bits de `styleVisionServer.jar` y (`styleVisionServer.dll`) estarán en la carpeta `bin\API_32bit` de la carpeta de instalación de StyleVision Server. Necesitará estos archivos si está usando una versión de 32 bits. Ocurre igual al contrario, si instaló una versión de 32 bits de StyleVision Server, los archivos de la versión de 64 bits de `APPNAME-UL%>.jar` y (`styleVisionServer.dll`) estarán en la carpeta `bin\API_64bit`. Necesitará estos archivos si está usando una versión de 64 bits.

Para acceder al código Java de la API de StyleVision Server agregue estas referencias al archivo `.classpath` de su proyecto Java.

<code>styleVisionServer.jar</code>	La biblioteca que se comunica con StyleVision Server
<code>styleVisionServer_JavaDoc.zip</code>	Documentación de la API de StyleVision Server

Además, `java.library.path` debe incluir la carpeta en la que se encuentra el archivo de biblioteca JNI (`styleVisionServer.dll`).

Si implementa su proyecto en una aplicación de servidor, asegúrese de que `styleVisionServer.jar` y `styleVisionServer.dll` están configurados correctamente con Java en el equipo servidor.

Para ver un ejemplo de cómo usar los archivos de biblioteca de la API consulte el archivo `buildAndRun.bat` (véase *más abajo*), que se encuentra en la carpeta `etc\Examples\Java` de la carpeta de instalación de StyleVision Server.

### Generar y ejecutar un programa Java con la API

Para aprender a generar y ejecutar un programa Java que use la API de StyleVision Server consulte el archivo de ejemplo `buildAndRun.bat`. Puede reutilizar este archivo para ejecutar sus propios programas de Java; para ello modifíquelo como necesite.

Inicie el archivo de lote en una interfaz de la línea de comandos con este comando:

```
buildAndRun "path_to_Java_bin_folder"
```

**Nota:** Para comprobar si Java es su classpath puede ejecutar el comando `java --version`. Si Java no es su classpath debe proporcionar la ruta de acceso relevante como parámetro del comando `buildAndRun`. Si la ruta contiene espacios debe usar comillas entorno a la ruta.

#### Extracto de `buildAndRun.bat`

```
@echo off
if %1.==. goto error
```

```
REM The location of the JAVA API binaries, the JAR file and the JNI library.
REM Adapt to your needs.
SETLOCAL
Set JavaAPIBinPath=%PROGRAMFILES%\Altova\StyleVisionServer2024\bin

REM Compile sample java
REM The -cp option (classpath) needs to point to the installed jar file (here, in its
original location)
REM "Program.java" is the Java program you want to compile
%1\javac.exe -cp "%JavaAPIBinPath%\StyleVisionServer.jar" Program.java

REM Run sample java
REM The -cp option (classpath) needs to point to the StyleVisionServer.jar file
REM The java.library.path needs to include the folder where the JNI library
StyleVisionServer.dll is located.
%1\java.exe -cp "%JavaAPIBinPath%\StyleVisionServer.jar;." -Djava.library.path="%
JavaAPIBinPath%" Program

@echo off
goto end

:error
echo Usage: buildAndRun "<path_to_java_bin_folder>"

:end
```

## Añadir las referencias de la biblioteca de StyleVision Server en Eclipse

En Eclipse puede añadir las referencias editando las propiedades del proyecto Java, tal y como se explica en las siguientes instrucciones (solo relevantes para Eclipse 4.4).

1. Abra el proyecto en Eclipse, haga clic en **Project | Properties** y después seleccione Java Build Path.
2. En la pestaña *Libraries* de bibliotecas haga clic en el botón **Add External JARs** para añadir archivos JAR externos y navegue hasta el archivo `StyleVisionServer.jar` (situado en la carpeta de instalación de StyleVision Server).
3. En el panel *JARs and class folders on the build path*, expanda el nodo `StyleVisionServer.jar` y después haga doble clic en el registro `Javadoc location: (None).Ç`
4. Compruebe que están marcadas las opciones **Javadoc in archive** y **External file** y en el campo *Archive path* seleccione el archivo `StyleVisionServer_JavaDoc.zip` (situado en la carpeta de instalación de StyleVision Server).
5. Haga clic en **OK** para terminar. La referencia a la biblioteca de StyleVision Server y el archivo de almacenamiento Javadoc se añade ahora al archivo `.classpath` del proyecto.

## 5.4 Ejemplos de código

En este apartado encontrará ejemplos para estos lenguajes de programación:

- [C#](#) <sup>62</sup>
- [C++](#) <sup>63</sup>
- [Java](#) <sup>65</sup>
- [VBScript](#) <sup>66</sup>
- [Visual Basic](#) <sup>68</sup>

Después de haber instalado StyleVision Server, copie la carpeta `etc/Examples` en su directorio principal u otra ubicación adecuada.

### 5.4.1 C#

Este ejemplo explica cómo usar código C# para generar un archivo de salida RTF con un archivo PXF y un archivo XML de entrada.

Antes de empezar a probar el ejemplo, compruebe que cumple con estos requisitos:

- StyleVision Server está instalado y tiene asignada una licencia válida.
- StyleVision Server está disponible como objeto de servidor COM (este proceso suele tener lugar automáticamente durante la instalación de StyleVision Server. Consulte el apartado [Información sobre la interfaz COM](#) <sup>59</sup> para obtener más información). Consulte también [Información sobre la interfaz .NET](#) <sup>58</sup>.

```
namespace StyleVisionServerAPI_sample
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                //Create a StyleVision Server object
                Altova.StyleVisionServer.Server objSVS = new
Altova.StyleVisionServer.Server();

                //Set a working directory - used for output and for intermediate files
                objSVS.WorkingDirectory = "..\\..\\..";

                //Default path to the StyleVision Server executable is the installation path
                (same dir with the StyleVisionServer.dll)
                //In case you moved the binaries on the disk, you need to explicitly set the
                path to the .exe file
                //objSVS.ServerPath = "C:\\Program Files (x86)\\Altova\\
\\StyleVisionServer2024\\bin\\StyleVisionServer.exe";

                //Prepare the name of the working XML
```



```

// The following import statements require the corresponding C++ tool-chain to be selected
// in the project configuration file.
#ifdef _WIN64
// 32-bit StyleVisionServer
#import "progid:StyleVision.Server"
#else
// 64-bit StyleVisionServer
#import "progid:StyleVision_x64.Server"
#endif

int _tmain(int argc, _TCHAR* argv[])
{
    CoInitialize( NULL );

    try
    {
        //Create a StyleVision Server object
        StyleVisionServerLib::IServerPtr pSVS;
        CoCreateInstance( __uuidof( StyleVisionServerLib::Server ), NULL, CLSCTX_ALL,
            __uuidof( StyleVisionServerLib::IServer ), reinterpret_cast< void** >( &pSVS ) );

        //Set a working directory - used for output and for intermediate files
        pSVS->WorkingDirectory = ".."; // this is relative to this applications' working
        directory (the project folder)

        //Default path to the StyleVision Server executable is the installation path (same dir
        with the StyleVisionServer.dll)
        //In case you moved the binaries on the disk, you need to explicitly set the path to
        the .exe file
        //pSVS->ServerPath = "C:\\Program Files (x86)\\Altova\\StyleVisionServer2024\\bin\\
        \\StyleVisionServer.exe";
        //pSVS->ServerPath = "C:\\Program Files\\Altova\\StyleVisionServer2024\\bin\\
        \\StyleVisionServer.exe";

        //Prepare the name of the working XML
        // This can be an absolute/relative path if the file is stored externally (not
        inside PXF)
        // pSVS->InputXML = "ExpReport.xml";
        // Or it can contain the path INSIDE the PXF
        // pSVS->InputXML = "ExpReport.pxf|zip\\ExpReport.xml";
        // Easiest way is to refer to the file as being embedded in the transformation file
        pSVS->InputXML = "altova://packagedfile/ExpReport.xml";

        //Add output paths (absolute or relative to WorkingDirectory) for all formats that
        should be generated
        pSVS->OutputRTF = "ExpReport.rtf";
        pSVS->OutputPDF = "ExpReport.pdfrtf";
        pSVS->OutputHTML = "ExpReport.html";

        //Prepare the parameters, if your design uses parameters
    }
}

```

```

//pSVS->AddParameter( "testparam1", "value 1" );

//Run the transformation; the output will be stored at C:\temp\ExpReport.rtf
// NOTE Please adapt the path to the input file in order to run the sample
if (pSVS->Generate("ExpReport.pxf"))
{
    std::cout << pSVS->LastExecutionMessage << std::endl;
    std::cout << "Success - finished execution" << std::endl;
}
else
    std::cout << pSVS->LastExecutionMessage << std::endl;
}
catch (_com_error& err )
{
    BSTR bstrMessage;
    (err).ErrorInfo()->GetDescription( &bstrMessage );
    std::cout << "Exception occurred: " << _com_util::ConvertBSTRToString( bstrMessage )
<< std::endl;
}

CoUninitialize();
return 0;
}

```

### 5.4.3 Java

Este ejemplo muestra cómo usar código Java para generar un archivo RTF de salida con un archivo PFX y un archivo XML de entrada. Antes de empezar a probar el ejemplo, asegúrese de que StyleVision Server está instalado, tiene asignada una licencia válida y está disponible como objeto de servidor. Puede registrarlo como objeto servidor al instalar StyleVision Server. El programa de ejemplo que puede ver más abajo se puede generar y ejecutar con el archivo por lotes `buildAndRun.bat`, que se encuentra en la carpeta `etc\Examples\Java` de su carpeta de instalación de StyleVision Server.

Consulte el apartado [Información sobre la interfaz Java](#)<sup>60</sup> para obtener más información.

**Nota:** Es recomendable que copie la carpeta Examples a su directorio principal u otra ubicación adecuada, y después entre en la carpeta Java para acceder a `Program.java` y `buildAndRun.bat`.

#### Program.java

```

public class Program
{

    public static void main(String[] args)
    {
        com.altova.stylevisionserver.StyleVisionServer objSVS
        try
        {
            //Create a StyleVision Server object
            objSVS = new com.altova.stylevisionserver.StyleVisionServer();

```

```

library
    //The default location of server binary is the folder containing the Java native
    //Select a different server binary with the following line:
    //objSVS.setServerPath(strServerPath);

    //The sample data is located in the parent folder of the Java sample code
    //Set this parent folder to be the working directory:
    objSVS.setWorkingDirectory( ".." );

    System.out.println("Running " + objSVS.getProductAndVersion());

    //Prepare the name of the working XML
    //This can be an absolute/relative path if the file is external (not inside PXF)
    //  objSVS.setInputXML( "ExpReport.xml" );
    //Or it can contain the path INSIDE the PXF
    //  objSVS.setInputXML( "C:\\Program Files (x86)\\Altova\\StyleVisionServer" +
majorVersionYear + "\\etc\\Examples\\ExpReport.pxf|zip\\ExpReport.xml" );
    //The easiest way is to refer to the file as being embedded in the transformation
file
    objSVS.setInputXML( "altova://packagedfile/ExpReport.xml" );

    //Add output paths (absolute or relative to WorkingDirectory) for all formats
that should be generated
    objSVS.setOutputRTF( "ExpReport.rtf" );

    //Prepare the parameters, if your design uses parameters
    //objSVS.AddParameter( "testparam1", "value 1" );

    //Run the transformation; the output will be stored at C:\\temp\\ExpReport.rtf
    // NOTE Please adapt the path to the input file in order to run the sample
    if ( objSVS.generate( "ExpReport.pxf" ) )
        System.out.println( "Success" + objSVS.getLastExecutionMessage() );
    else
        System.out.println( objSVS.getLastExecutionMessage() );
}
catch ( Exception e)
{
    System.out.println(e.getMessage());
}
}
}

```

### 5.4.4 VBScript

Este ejemplo explica cómo usar código VB Script para generar un archivo de salida RTF con un archivo PXF y un archivo XML de entrada.

Antes de empezar a probar el ejemplo, compruebe que cumple con estos requisitos:

- StyleVision Server está instalado y tiene asignada una licencia válida.
- StyleVision Server está disponible como objeto de servidor COM (este proceso suele tener lugar automáticamente durante la instalación de StyleVision Server. Consulte el apartado [Información sobre la interfaz COM](#)<sup>59</sup> para obtener más información).

#### Option Explicit

```
'Create a StyleVision Server object; use "StyleVision_x64.Server" if you want to use the
64-bit installation
Dim objSVS
' Since we load a COM-DLL we need care about the process architecture
On Error Resume Next ' ignore any COM errors avoiding uncontrolled script termination
Dim WshShell
Dim WshProcEnv
Set WshShell = CreateObject("WScript.Shell")
Set WshProcEnv = WshShell.Environment("Process")
Dim process_architecture
process_architecture= WshProcEnv("PROCESSOR_ARCHITECTURE")
If process_architecture = "x86" Then
    Set objSVS = WScript.GetObject( "", "StyleVision.Server" )
    If Err.Number <> 0 then
        WScript.Echo("You are running in a 32-bit process but StyleVision Server COM-API 32-
bit seems not to be installed on your system.")
        WScript.Quit -1
    End If
Else
    Set objSVS = WScript.GetObject( "", "StyleVision_x64.Server" )
    If Err.Number <> 0 then
        WScript.Echo("You are running in a 64-bit process but StyleVision Server COM-API 64-
bit seems not to be installed on your system.")
        WScript.Echo("If you have installed 32-bit StyleVision Server consider calling your
script from the 32-bit console 'C:\Windows\SysWOW64\cmd.exe.'")
        WScript.Quit -1
    End If
End If
On Error Goto 0 ' re-enable default error promotion

'Set a working directory - used for input, output and for intermediate files
'objSVS.WorkingDirectory = "C:\Program Files (x86)
\Altova\StyleVisionServer2020\etc\examples"
objSVS.WorkingDirectory = ".."

'Default path to the StyleVision Server executable is the installation path (same dir with
the StyleVisionServer.dll)
'In case you moved the binaries on the disk, you need to explicitly set the path to the
.exe file
'objSVS.ServerPath = "C:\Program Files (x86)
\Altova\StyleVisionServer2020\bin\StyleVisionServer_DebugDLL.exe"

' The Generate method will return 'True' if generation was successful otherwise 'False'.
' In the case of fundamental errors like termination of the server process a COM error will
be raised which
```

```

' can be handled using the VBScript Err object.
On Error Resume Next ' ignore any COM errors avoiding uncontrolled script termination
Err.Clear

WScript.Echo("Running " & objSVS.ProductNameAndVersion & vbCrlf)

'Prepare the name of the working XML
' This can be an absolute/relative path if the file is stored externally (not inside
PXF)
' objSVS.InputXML = "ExpReport.xml"
' or it can contain the path INSIDE the PXF
objSVS.InputXML = "ExpReport.pxf|zip\ExpReport.xml"
' or refer to the file as being embedded in the transformation file
'objSVS.InputXML = "altova://packagedfile/ExpReport.xml"

'Add output paths (absolute or relative to WorkingDirectory) for all formats that should be
generated
' make sure you have write permissions
'objSVS.OutputRTF = "C:\tmp\ExpReport.rtf"
objSVS.OutputPDF = "C:\tmp\ExpReport.pdf"
'objSVS.OutputHTML = "C:\tmp\ExpReport.html"

'Prepare the parameters, if your design uses parameters
'Call objSVS.AddParameter( "testparam1", "value_1" )

' Run the transformation. The PXF file path can be relative to the working folder or
absolute.
WScript.Echo("Generating output from ExpReport.pxf...")
If ( objSVS.Generate( "ExpReport.pxf" ) ) Then
    WScript.Echo( objSVS.LastExecutionMessage )
    WScript.Echo( "Success - finished execution" )
Else
    WScript.Echo( objSVS.LastExecutionMessage )
End If

' handle COM errors
If Err.Number <> 0 Then
    WScript.Echo("Internal error - " & Err.Description )
    WScript.Quit -1
End If

On Error Goto 0 ' re-enable default error promotion

```

## 5.4.5 Visual Basic

Este ejemplo muestra cómo usar código Visual Basic para generar un archivo RTF de salida con un archivo PXF y un archivo XML de entrada.

Antes de empezar a probar el ejemplo, compruebe que cumple con estos requisitos:

- StyleVision Server está instalado y tiene asignada una licencia válida.
- Su proyecto de Visual Studio [incluye una referencia al DLL](#) <sup>58</sup> de StyleVision Server.

Option Explicit On

Module Program

```
Sub Main()

    Try
        'Create a StyleVision Server object
        Dim objSVS As Altova.StyleVisionServer.Server = New
Altova.StyleVisionServer.Server

        'Set a working directory - used for output and for intermediate files
        objSVS.WorkingDirectory = "C:\Program Files (x86)
\Altova\MapForceServer2020\etc\Examples"
        objSVS.WorkingDirectory = "..\..\.."

        'Default path to the StyleVision Server executable is the installation path
(same dir with the StyleVisionServer.dll)
        'In case you moved the binaries on the disk, you need to explicitly set the
path to the .exe file
        objSVS.ServerPath = "C:\Program Files (x86)
\Altova\StyleVisionServer2020\bin\StyleVisionServer.exe"
        objSVS.ServerPath = "C:\Program
Files\Altova\StyleVisionServer2020\bin\StyleVisionServer.exe"

        'Prepare the name of the working XML
        ' This can be an absolute/relative path if the file is stored externally
(not inside PXF)
        objSVS.InputXML = "ExpReport.xml"
        ' Or it can contain the path INSIDE the PXF
objSVS.InputXML = "ExpReport.pxf|zip\ExpReport.xml"
        ' Easiest way is to refer to the file as being embedded in the
transformation file
        objSVS.InputXML = "altova://packagedfile/ExpReport.xml"

        'Add output paths (absolute or relative to WorkingDirectory) for all formats
that should be generated
objSVS.OutputRTF = "C:\tmp\ExpReport.rtf"
objSVS.OutputPDF = "C:\tmp\ExpReport.pdf"
objSVS.OutputHTML = "C:\tmp\ExpReport.html"

        'Prepare the parameters, if your design uses parameters
objSVS.AddParameter( "testparam1", "value 1" )

        ' Run the transformation; the output will be stored at C:\temp
If (objSVS.Generate("ExpReport.pxf")) Then
    System.Console.WriteLine(objSVS.LastExecutionMessage)
    System.Console.WriteLine("Success - finished execution")
Else
```

```
        System.Console.WriteLine(objSVS.LastExecutionMessage)
    End If

    Catch ex As Exception
        System.Console.WriteLine("Internal Error - " & ex.Message())
    End Try

End Sub

End Module
```

## 5.5 Referencia de la API

Este apartado ofrece una referencia del usuario de la API de StyleVision Server para estas plataformas:

- [COM y .NET](#) <sup>71</sup>
- [Java](#) <sup>78</sup>

### 5.5.1 COM y .NET

La API de StyleVision Server expone la **interfaz `IServer`**, que crea una instancia de objeto StyleVision Server nueva y ofrece acceso a StyleVision Server.

La interfaz `IServer` tiene los métodos y propiedades que aparecen a continuación.

#### Métodos

### ▼ AddParameter

Asigna un valor a un parámetro definido en el archivo PXF.

#### ▣ C#

```
void AddParameter(string bstrName, string bstrValue)
```

#### ▣ C++

```
HRESULT AddParameter([in] BSTR bstrName, [in] BSTR bstrValue );
```

#### ▣ VB

```
Sub AddParameter(ByVal bstrName As String, ByVal bstrValue As String)
```

### ▼ ClearParameterList

Borra la lista de parámetros.

#### ▣ C#

```
void ClearParameterList()
```

#### ▣ C++

```
HRESULT ClearParameterList();
```

#### ▣ VB

```
Sub ClearParameterList()
```

### ▼ Generate

Genera archivos de salida (HTML, PDF, RTF o DOCX) con el archivo PXF indicado por `TransfPath`. Transforma el archivo XML de entrada (el archivo XML de trabajo del archivo PXF) con el documento XSLT que incluye el archivo PXF. Devuelve `TRUE` si la generación finaliza correctamente y `FALSE` si se produce un error.

#### ▣ C#

```
bool Generate(string bstrTransfPath)
```

#### ▣ C++

```
HRESULT Generate( [in] BSTR bstrTransfPath, [out, retval] VARIANT_BOOL*  
pbSuccess );
```

#### ▣ VB

```
Function Generate(ByVal bstrTransfPath As String) As Boolean
```

## Propiedades

#### ▼ APIMajorVersion

Obtiene la versión principal de la API de StyleVision Server. Puede ser distinta a la versión del producto si la API está conectada a otro servidor.

##### ▣ C#

```
int APIMajorVersion { get; }
```

##### ▣ C++

```
HRESULT APIMajorVersion([out, retval] INT* pRetVal);
```

##### ▣ VB

```
ReadOnly Property APIMajorVersion As Integer
```

#### ▼ APIMinorVersion

Obtiene la versión secundaria de la API de StyleVision Server. Puede ser distinta a la versión del producto si la API está conectada a otro servidor.

##### ▣ C#

```
int APIMinorVersion { get; }
```

##### ▣ C++

```
HRESULT APIMinorVersion([out, retval] INT* pRetVal);
```

##### ▣ VB

```
ReadOnly Property APIMinorVersion As Integer
```

#### ▼ APIServicePackVersion

Obtiene la versión de service pack de la API de StyleVision Server. Puede ser distinta a la versión del producto si la API está conectada a otro servidor.

##### ▣ C#

```
int APIServicePackVersion { get; }
```

##### ▣ C++

```
HRESULT APIServicePackVersion([out, retval] INT* pRetVal);
```

##### ▣ VB

```
ReadOnly Property APIServicePackVersion As Integer
```

#### ▼ InputXML

Define la ruta de acceso y el nombre del archivo XML que se debe procesar (el archivo XML de trabajo del archivo PXF).

##### ▣ C#

```
string InputXML { set; }
```

## ▣ C++

```
HRESULT InputXML([in] BSTR bstrPath );
```

## ▣ VB

```
Property InputXML As String
```

## ▼ Is64Bit

Devuelve `TRUE` si el motor de StyleVision Server es un ejecutable de 64 bits.

## ▣ C#

```
bool Is64Bit { get; }
```

## ▣ C++

```
HRESULT Is64Bit([out, retval] VARIANT_BOOL* pbVal);
```

## ▣ VB

```
ReadOnly Property Is64Bit As Boolean
```

## ▼ LastExecutionMessage

Obtiene el mensaje recibido durante la última ejecución del comando `Generate`.

## ▣ C#

```
string LastExecutionMessage { get; }
```

## ▣ C++

```
HRESULT LastExecutionMessage([out, retval] BSTR* pbstrResult );
```

## ▣ VB

```
ReadOnly Property LastExecutionMessage As String
```

## ▼ MajorVersion

Obtiene la versión principal de StyleVision Server.

## ▣ C#

```
int MajorVersion { get; }
```

## ▣ C++

```
HRESULT MajorVersion([out, retval] INT* pnVal);
```

## ▣ VB

```
ReadOnly Property MajorVersion As Integer
```

## ▼ MinorVersion

Obtiene la versión secundaria de StyleVision Server.

- ▣ C#  
`int MinorVersion { get; }`
- ▣ C++  
`HRESULT MinorVersion([out, retval] INT* pnVal);`
- ▣ VB  
`ReadOnly Property MinorVersion As Integer`

#### ▼ OutputDOCX

Establece la ruta de acceso y el nombre del archivo DOCX de salida.

- ▣ C#  
`string OutputDOCX { set; }`
- ▣ C++  
`HRESULT OutputDOCX([in] BSTR bstrPath );`
- ▣ VB  
`Property OutputDOCX As String`

#### ▼ OutputFO

Establece la ruta de acceso y el nombre del archivo FO de salida.

- ▣ C#  
`string OutputFO { set; }`
- ▣ C++  
`HRESULT OutputFO([in] BSTR bstrPath );`
- ▣ VB  
`Property OutputFO As String`

#### ▼ OutputHTML

Establece la ruta de acceso y el nombre del archivo HTML de salida.

- ▣ C#  
`string OutputHTML { set; }`
- ▣ C++  
`HRESULT OutputHTML([in] BSTR bstrPath );`
- ▣ VB  
`Property OutputHTML As String`

### ▼ OutputPDF

Establece la ruta de acceso y el nombre del archivo PDF de salida.

#### ▣ C#

```
string OutputPDF { set; }
```

#### ▣ C++

```
HRESULT OutputPDF([in] BSTR bstrPath );
```

#### ▣ VB

```
Property OutputPDF As String
```

### ▼ OutputRTF

Establece la ruta de acceso y el nombre del archivo RTF de salida.

#### ▣ C#

```
string OutputRTF { set; }
```

#### ▣ C++

```
HRESULT OutputRTF([in] BSTR bstrPath );
```

#### ▣ VB

```
Property OutputRTF As String
```

### ▼ OutputText

Establece la ruta de acceso y el nombre del archivo de texto de salida.

#### ▣ C#

```
string OutputText { set; }
```

#### ▣ C++

```
HRESULT OutputText([in] BSTR bstrPath );
```

#### ▣ VB

```
Property OutputText As String
```

### ▼ ProductName

Obtiene el nombre del producto: "StyleVision Server"

#### ▣ C#

```
string ProductName { get; }
```

#### ▣ C++

```
HRESULT ProductName([out, retval] BSTR* pstrVal);
```

VB

```
ReadOnly Property ProductName As String
```

▼ ProductNameAndVersion

Obtiene el nombre completo del producto, incluido el número de versión: "StyleVision Server 2014r2 sp1 (x64)".

C#

```
string ProductNameAndVersion { get; }
```

C++

```
HRESULT ProductNameAndVersion([out, retval] BSTR* pstrVal);
```

VB

```
ReadOnly Property ProductNameAndVersion As String
```

▼ ServerPath

Obtiene o establece la ruta de acceso del ejecutable de StyleVision Server.

C#

```
string ServerPath { set; get; }
```

C++

```
HRESULT ServerPath([in] BSTR bstrServerFile );  
HRESULT ServerPath([out, retval] BSTR* pbstrServerFile );
```

VB

```
Property ServerPath As String
```

▼ ServicePackVersion

Obtiene la versión de service pack de StyleVision Server (p. ej.: **1** si el nombre completo del producto es Altova StyleVision Server 2014 r2 sp1 (x64).)

▣ C#

```
int ServicePackVersion { get; }
```

▣ C++

```
HRESULT ServicePackVersion([out, retval] INT* pnVal);
```

▣ VB

```
ReadOnly Property ServicePackVersion As Integer
```

▼ WhereClause

Establece una cláusula `WHERE` de SQL que determina qué filas se deben procesar de la fuente de esquema XML de BD.

▣ C#

```
string WhereClause { set; }
```

▣ C++

```
HRESULT WhereClause([in] BSTR bstrPath );
```

▣ VB

```
Property WhereClause As String
```

▼ WorkingDirectory

Obtiene o establece el directorio actual para los trabajos en ejecución. Las rutas de acceso relativas se evalúan con respecto al directorio de trabajo.

▣ C#

```
string WorkingDirectory { set; get; }
```

▣ C++

```
HRESULT WorkingDirectory([in] BSTR bstrWorkingDirectory );
HRESULT WorkingDirectory([out, retval] BSTR* pbstrWorkingDirectory );
```

▣ VB

```
Property WorkingDirectory As String
```

## 5.5.2 Java

El paquete `com.altova.stylevisionserver` está compuesto por estas dos clases:

- `public class StyleVisionServer` (ver descripción más abajo)
- `public class StyleVisionServerException` extends [Exception](#)

### clase `StyleVisionServer`

La clase `StyleVisionServer` crea una instancia de objeto `StyleVision Server` nueva y ofrece acceso a `StyleVision Server`. Los métodos de la clase `StyleVisionServer` se describen a continuación.

## Métodos de la clase `StyleVisionServer`

A continuación se enumeran por orden alfabético todos los métodos de la clase `StyleVisionServer`.

### ▼ `addParameter`

```
public void addParameter(String nombre, String valor)
```

Añade el nombre y el valor de un parámetro nuevo. Cada parámetro y su valor se especifican en una llamada distinta al método. Los parámetros se deben declarar en el documento XSLT.

Parámetros:

`nombre`: almacena el nombre del parámetro en forma de cadena de texto

`valor`: almacena el valor del parámetro en forma de cadena de texto

### ▼ `clearParameterList`

```
public void clearParameterList()
```

Borra la lista de parámetros.

### ▼ `generate`

```
public boolean generate(String transfPath)
```

Procesa el archivo PXF indicado por `transfPath`. Emite `StyleVisionServerException`.

Parámetros:

`transfPath`: dirección URL absoluta que da la ubicación del archivo PXF.

Devuelve:

`true()` si la ejecución finaliza correctamente

`false()` si se produce un error de ejecución

Si se produce un error, utilice `getLastExecutionMessage()`

### ▼ `getAPIMajorVersion`

```
public int getAPIMajorVersion()
```

Obtiene la versión principal de la API de `StyleVision Server`. Puede ser distinta a la versión del producto si la API está conectada a otro servidor.

### ▼ `getAPIMinorVersion`

```
public int getAPIMinorVersion()
```

Obtiene la versión secundaria de la API de `StyleVision Server`. Puede ser distinta a la versión del producto si la API está conectada a otro servidor.

### ▼ `getAPIServicePackVersion`

```
public int getAPIServicePackVersion()
```

Obtiene la versión de service pack de la API de StyleVision Server. Puede ser distinta a la versión del producto si la API está conectada a otro servidor.

▼ `getLastExecutionMessage`

```
public String getLastExecutionMessage()
```

Obtiene el mensaje recibido durante la última ejecución del comando `generate`.

▼ `getMajorVersion`

```
public int getMajorVersion()
```

Obtiene la versión principal de la aplicación.

▼ `getMinorVersion`

```
public int getMinorVersion()
```

Obtiene la versión secundaria de la aplicación.

▼ `getProductName`

```
public String getProductName()
```

Obtiene el nombre del producto.

▼ `getProductNameAndVersion`

```
public String getProductNameAndVersion()
```

Obtiene el nombre completo del producto y su número de versión.

▼ `getServerPath`

```
public String getServerPath()
```

Obtiene la ruta de acceso al archivo binario ejecutable del servidor.

▼ `getServicePackVersion`

```
public int getServicePackVersion()
```

Obtiene la versión de service pack de la API de StyleVision Server.

▼ `getWorkingDirectory`

```
public String getWorkingDirectory()
```

Obtiene el directorio de trabajo actual.

▼ `is64bit`

```
public boolean is64bit()
```

Comprueba si el ejecutable es de 64 bits.

Devuelve:

`true()` para StyleVision Server (x64), de lo contrario devuelve `false()`.

▼ `setInputXML`

```
public void setInputXML(String rutaAcceso)
```

Establece qué archivo XML se debe procesar. Debe ser la ruta de acceso del archivo XML de trabajo indicado en el archivo PXF.

Parámetros:

rutaAcceso: almacena la ruta de acceso del archivo XML de trabajo del archivo PXF.

▼ **setOutputDOCX**

```
public void setOutputDOCX(String rutaAcceso)
```

Establece la ruta de acceso y el nombre del archivo DOCX que se debe generar.

Parámetros:

rutaAcceso: ruta de acceso y nombre del archivo DOCX que se debe generar.

▼ **setOutputFO**

```
public void setOutputFO(String rutaAcceso)
```

Establece la ruta de acceso y el nombre del archivo FO que se debe generar.

Parámetros:

rutaAcceso: la ruta de acceso y el nombre del archivo FO que se debe generar.

▼ **setOutputHTML**

```
public void setOutputHTML(String rutaAcceso)
```

Establece la ruta de acceso y el nombre del archivo HTML que se debe generar.

Parámetros:

rutaAcceso: ruta de acceso y nombre del archivo HTML que se debe generar.

▼ **setOutputPDF**

```
public void setOutputPDF(String rutaAcceso)
```

Establece la ruta de acceso y el nombre del archivo PDF que se debe generar.

Parámetros:

rutaAcceso: ruta de acceso y nombre del archivo PDF que se debe generar.

▼ **setOutputRTF**

```
public void setOutputRTF(String rutaAcceso)
```

Establece la ruta de acceso y el nombre del archivo RTF que se debe generar.

Parámetros:

rutaAcceso: ruta de acceso y nombre del archivo RTF que se debe generar.

▼ **setOutputText**

```
public void setOutputText(String path)
```

Establece la ruta de acceso y el nombre del archivo de texto que se debe generar.

Parámetros:

path: Thede acceso y nombre del archivo de texto que se debe generar.

▼ **setServerPath**

```
public void setServerPath(String archivoServidor)
```

Establece la ruta de acceso del ejecutable de StyleVisionServer.

Parámetros:

archivoServidor: ruta de acceso del ejecutable de StyleVisionServer.

▼ **setWhereClause**

```
public void setWhereClause(String cláusulaWhere)
```

Establece una cláusula `WHERE` de SQL que determina qué filas se deben procesar de una fuente de esquema XML de BD.

Parámetros:

`cláusulaWhere`: la cláusula `WHERE` de SQL que determina qué filas se deben procesar de una fuente de esquema XML de BD.

▼ **setWorkingDirectory**

```
public void setWorkingDirectory(String directorioTrabajo)
```

Establece un directorio predeterminado. Las rutas de acceso relativas se resuelven en relación a este directorio.

Parámetros:

`directorioTrabajo`: ruta de acceso del directorio predeterminado (de trabajo).

▼ **stop**

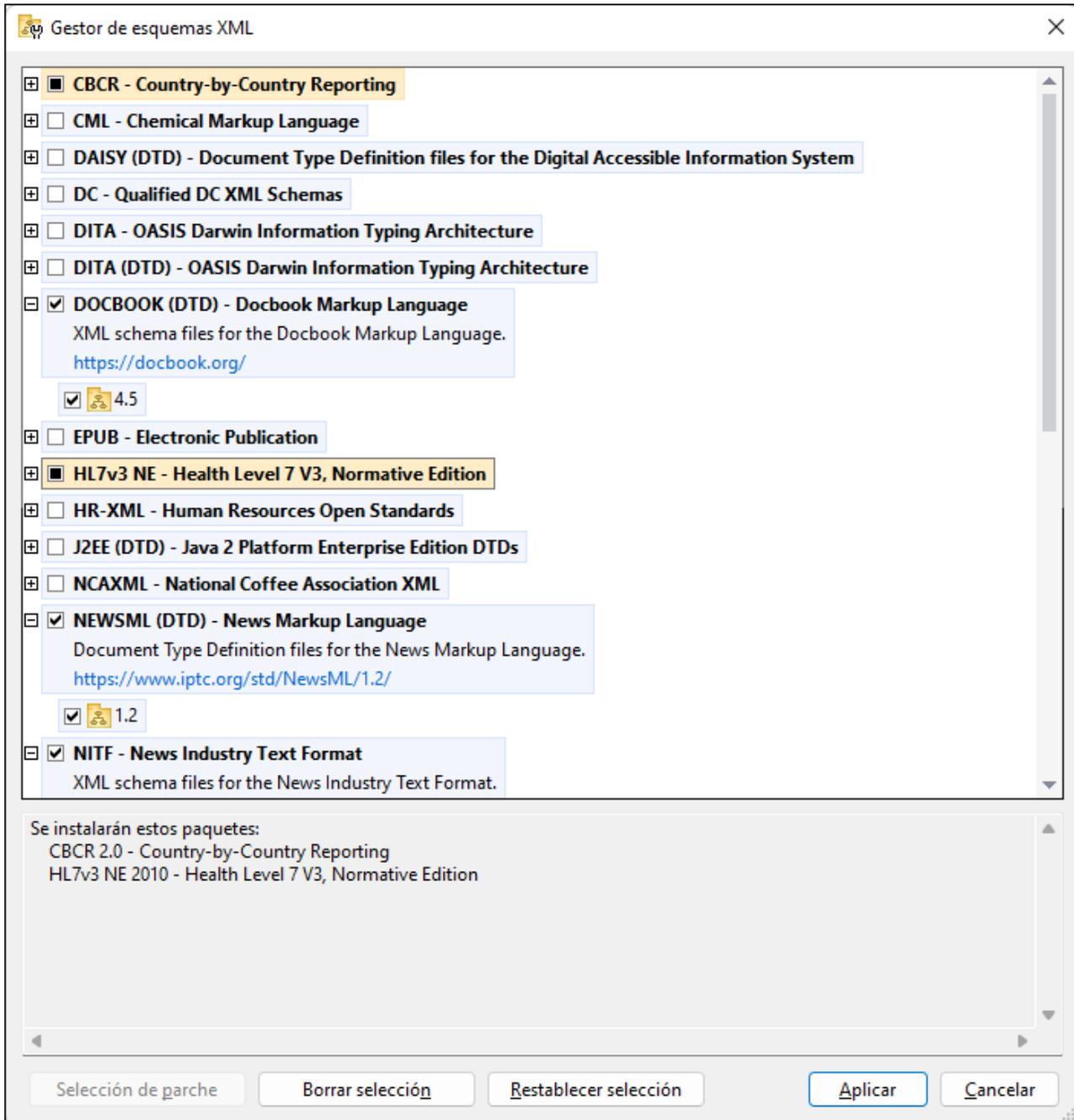
```
public void stop()
```

Detiene el proceso del servidor.

## 6 Gestor de esquemas

El Gestor de esquemas XML es una herramienta que ofrece una forma centralizada de instalar y administrar esquemas XML (DTDs para XML y esquemas XML) para usarlos en todas las aplicaciones de Altova compatibles con XML Schema, incluido **StyleVision Server**.

- En Windows, el gestor tiene una interfaz gráfica del usuario (*imagen siguiente*) a la que también puede acceder desde la línea de comandos. (Las aplicaciones de escritorio de Altova solo están disponibles para Windows; *consulte la lista siguiente*.)
- En Linux y macOS el Gestor de esquemas solo está disponible en la línea de comandos. (Las aplicaciones de escritorio de Altova están disponibles para Windows, Linux y macOS; *consulte la lista siguiente*.)



*Aplicaciones de Altova que funcionan con el Gestor de esquemas*

Aplicaciones de escritorio (solo para Windows)	Aplicaciones de servidor (Windows, Linux, macOS)
XMLSpy (todas las ediciones)	RaptorXML Server, RaptorXML+XBRL Server
MapForce (todas las ediciones)	StyleVision Server

StyleVision (todas las ediciones)	
-----------------------------------	--

## Instalación y desinstalación del Gestor de esquemas

El Gestor de esquemas se instala automáticamente al instalar cualquiera de las aplicaciones de Altova compatibles con XML o el Altova Mission Kit (véase *la tabla de más arriba*).

También se elimina automáticamente si desinstala todas las aplicaciones de Altova compatibles con XML del equipo.

## Características de Gestor de esquemas

El Gestor de esquemas permite:

- Ver los esquemas XML que hay instaladas en su equipo y comprobar si hay versiones nuevas para descargar.
- Descargar las versiones más recientes de los esquemas XML independientemente del ciclo de versiones de Altova. Altova guarda todos los esquemas en un sistema de almacenamiento en línea al que tiene acceso el Gestor de esquemas y desde donde puede descargarlas tan pronto como estén disponibles.
- Instalar o desinstalar cualquiera de las múltiples versiones de un esquema en concreto (o todas ellas, si las necesita).
- Un solo esquema XML representa un "paquete", pero puede tener dependencias en otros esquemas. Al instalar o desinstalar un esquema, se detectan e instalan o desinstalan también automáticamente todas sus dependencias. La interfaz gráfica del usuario (o la línea de comandos, en su caso) le informa cuando se añaden o eliminan esquemas.
- Los esquemas XML administradas con el Gestor de esquemas pueden usar el [catálogo XML](#), que permite resolver referencias a URI en documentos de instancia o esquema desde archivos locales, en vez de a través de Internet.
- Todos los esquemas principales están incluidos en Gestor de esquemas y se actualizan de forma periódica a la versión más reciente. De esta forma puede administrar todos los esquemas desde un punto común y tenerlos siempre listos para las aplicaciones de Altova que los usan.
- Los cambios que se realizan en el Gestor de esquemas afectan a todos los productos de Altova que estén instalados en ese equipo.
- En los productos de Altova, si intenta validar con un esquema que no está instalado pero sí disponible con el Gestor de esquemas, este se instala automáticamente. Sin embargo, si el paquete de esquemas que quiere instalar contiene asignaciones de espacios de nombres, no puede instalarse automáticamente, sino que debe ejecutar Gestor de esquemas, seleccionar qué paquetes quiere instalar y ejecutar la instalación. Si después de instalar los paquetes la aplicación de Altova que está abierta no se reinicia automáticamente, debe reiniciarla manualmente.

## Funcionamiento

Altova mantiene un almacenamiento en línea donde guarda todos los esquemas XML de los productos de Altova. Este almacenamiento se actualiza de forma periódica, por ejemplo, poco después de que las organizaciones correspondientes publiquen las versiones nuevas de los esquemas respectivos. Al ejecutar Gestor de esquemas desde la interfaz gráfica del usuario aparece información sobre los esquemas más recientes disponibles en un cuadro de diálogo en el que puede visualizarlos, instalarlos, actualizarlos o desinstalarlos.

También puede instalar los esquemas de otra manera. En el sitio web de Altova (<https://www.altova.com/schema-manager>) puede seleccionar el esquema y los esquemas dependientes de

este que quiere instalar. El sitio web prepara un archivo de tipo `.altova_xmlschemas` que puede descargar y que contiene la información sobre los esquemas seleccionados. Al hacer doble clic en este archivo o pasarlo a **Gestor de esquemas** desde la línea de comandos como argumento del comando `install`<sup>97</sup>, Gestor de esquemas instala los esquemas que contiene.

*Memoria caché local: seguimiento de esquemas*

Independientemente de cómo se instalen los esquemas, toda la información sobre los esquemas instalados se almacena en una ubicación centralizada de su equipo, el directorio caché. El directorio caché local está en:

<i>Windows</i>	C:\ProgramData\Altova\pkgs\.cache
<i>Linux</i>	/var/opt/Altova/pkgs\.cache
<i>macOS</i>	/var/Altova/pkgs

El directorio caché local se actualiza automáticamente de vez en cuando para que el estado más actual del equipo corresponda con el del almacenamiento en línea. Más concretamente, el caché se actualiza:

- al ejecutar el Gestor de esquemas.
- al ejecutar StyleVision Server por primera vez en un mismo día natural.
- si StyleVision Server ya se está ejecutando, el directorio caché se actualiza cada 24 horas.
- también puede actualizar el caché local desde el almacenamiento en línea manualmente ejecutando el comando de actualización `update`<sup>100</sup> desde la línea de comandos.

Si instala o desinstala esquemas, el directorio caché local se actualiza automáticamente con información sobre los esquemas disponibles e instalados, además de con los propios archivos de esquema.

### No modifique la memoria caché manualmente

El directorio caché local se mantiene automáticamente en base a los esquemas que instale o desinstale; no debe modificarlo ni eliminarlo manualmente. Si necesita restaurar el Gestor de esquemas a su estado original, ejecute el comando `reset`<sup>98</sup> desde la línea de comandos y después ejecute el comando `initialize`<sup>96</sup>.

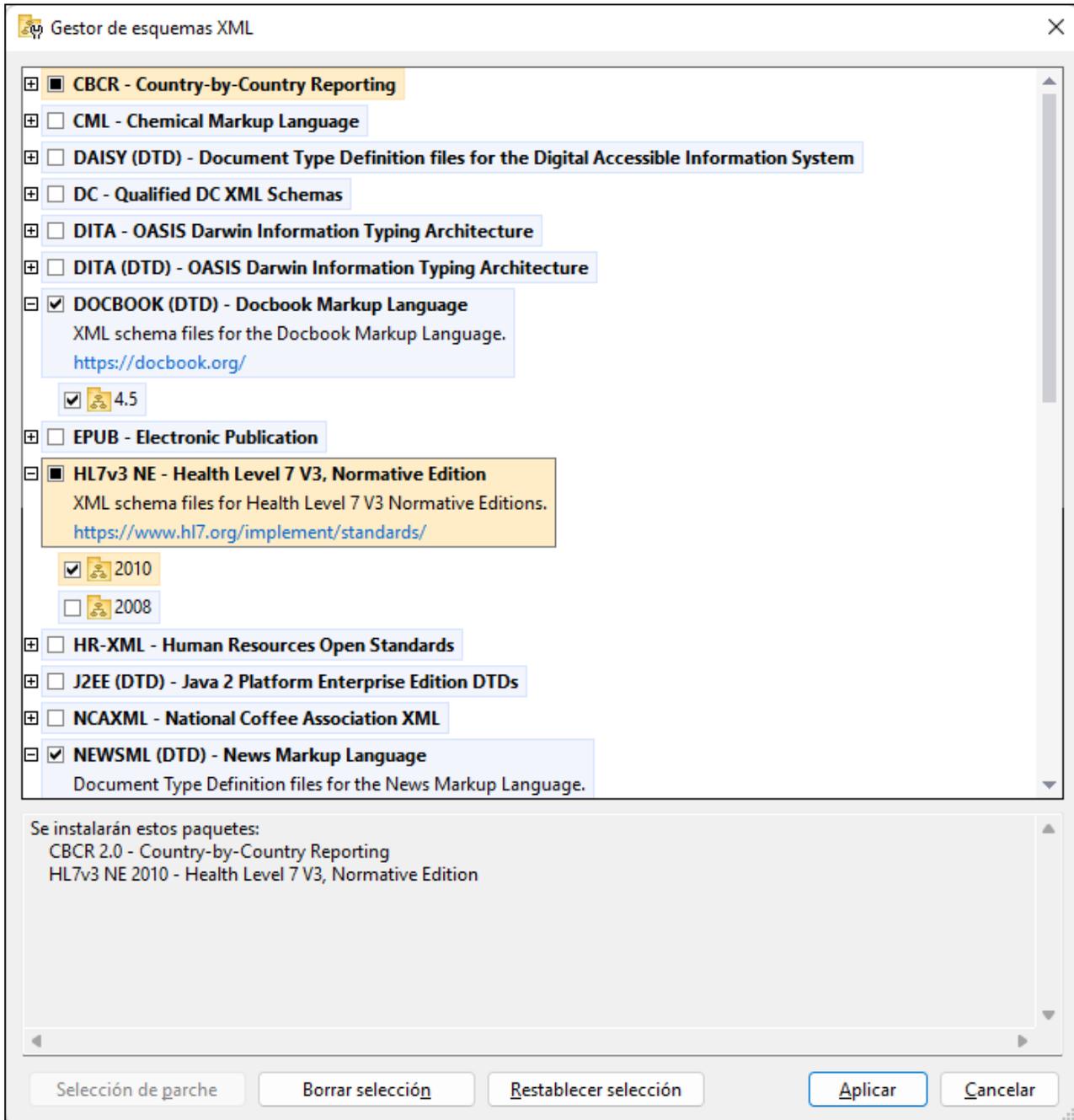
## 6.1 Ejecutar el gestor de esquemas

### Interfaz gráfica del usuario

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

- *Durante la instalación de StyleVision Server:* al final del proceso de instalación, seleccione la casilla *Invocar al Gestor de esquemas* para acceder directamente a la IGU del gestor de esquemas XML. Con él puede instalar esquemas durante el proceso de instalación de su aplicación de Altova.
- Mediante el archivo `.altova_schemas` que descargó del [sitio web de Altova](#): haga doble clic en el archivo para ejecutar Gestor de esquemas, que instalará los esquemas que haya seleccionado.

Cuando se abra la IGU del Gestor de esquemas (*imagen siguiente*) podrá ver en ella los esquemas que ya se han instalado. Si quiere instalar más solo tiene que seleccionarlas, y al contrario si quiere desinstalar alguna. Una vez haya terminado, puede aplicar los cambios. los esquemas que se vayan a instalar o desinstalar aparecerán resaltados y un mensaje le avisará de los cambios que está a punto de hacer en la ventana *Mensajes*, en la parte inferior de la ventana de **Gestor de esquemas** (véase *imagen*).



## Interfaz de la línea de comandos

Para ejecutar el Gestor de esquemas desde una interfaz de la línea de comandos debe usar su archivo ejecutable, `xmlschemamanager.exe`.

Puede encontrar este archivo:

- *en Windows*: C:\ProgramData\Altova\SharedBetweenVersions
- *en Linux o macOS (solo para aplicaciones de servidor)*: %INSTALLDIR%/bin, donde %INSTALLDIR% es el directorio de instalación del programa.

Puede usar cualquiera de los comandos de la [referencia de la línea de comandos](#) <sup>95</sup>, a continuación.

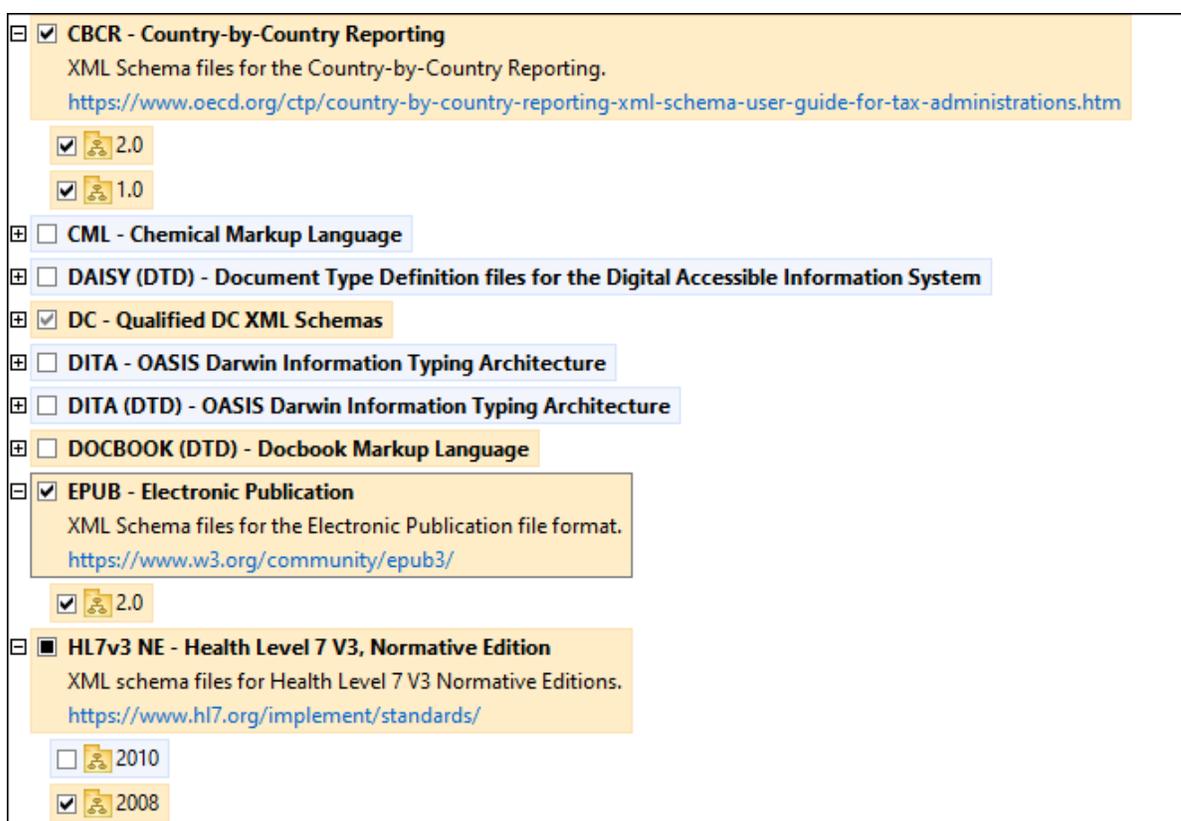
Para mostrar al ayuda de los comandos ejecute:

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

## 6.2 Categorías de estado

Gestor de esquemas diferencia los esquemas que administra entre:

- *Esquemas instalados:* estos aparecen en la IGU con sus casillas marcadas (*en la imagen siguiente las versiones marcadas de los esquemas EPUB y HL7v3 NE son las que están instaladas*). Si se seleccionan todas las versiones de un esquema, en la casilla del esquema aparece una marca de verificación. Si hay al menos un esquema sin seleccionar, en la casilla del esquema aparece un cuadrado negro. Para **desinstalar** un esquema debe desmarcar la casilla correspondiente (*en la imagen siguiente, el DTD DocBook está instalado y su casilla se ha desmarcado, es decir, se va a desinstalar*).
- *Esquemas disponibles no instalados:* estos aparecen en la IGU con las casillas correspondientes sin seleccionar. Para **instalar** esquemas, marque la casilla correspondiente.



- *Esquemas que pueden actualizarse:* son los que han sido revisados por sus emisores. Aparecen indicados en la IGU con el icono  (*imagen anterior*). Puede aplicar **parches** al esquema seleccionado con la revisión que esté disponible.

### Puntos importantes

- En la imagen anterior se han marcado los esquemas CBCR. Las que tienen un fondo azul ya están instaladas. Las que tienen el fondo amarillo no están instaladas pero se han seleccionado para instalarlas. Observe que el esquema HL7v3 NE 2010 no está instalada ni se ha seleccionado para instalarlo.

- Al ejecutar el Gestor de esquemas desde la línea de comandos puede usar el comando `list` con distintas opciones para ver distintas categorías de esquemas:

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

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

## 6.3 Aplicar parches o instalar un esquema

### Aplicar un parche a un esquema instalado

A veces los emisores de los esquemas XML generan parches. Cuando el Gestor de esquemas XML detecta que hay parches disponibles, estos aparecen en las listas de esquemas, desde donde puede instalarlos.

#### En la IGU

Los parches se indican con el icono . (Consulte también el apartado anterior sobre [categorías de esquemas](#)<sup>90</sup>.) Si hay parches disponibles se habilita el botón **Seleccionar parches**. Haga clic en él para seleccionar y preparar los parches. En la IGU, el icono de los esquemas correspondientes cambia de  a  y el cuadro de diálogo le informa de qué parches se van a aplicar. Las listas del panel principal y las del panel Mensajes están ordenadas alfabéticamente, por lo que puede ver y revisar los esquemas antes de aplicar los parches. Una vez esté listo para instalar los parches seleccionados, haga clic en **Aplicar**.

#### En la línea de comandos

Para aplicar un parche desde la línea de comandos:

1. Ejecute el comando `list -u`<sup>97</sup>. Aparece una lista con los esquemas para las que hay parches disponibles.
1. Ejecute el comando `upgrade`<sup>100</sup> para instalar todos los parches.

### Instalar un esquema disponible

Para instalar esquemas puede usar la IGU del Gestor de esquemas o enviar las instrucciones al Gestor de esquemas desde la línea de comandos.

**Nota:** si el esquema actual tiene dependencias en otros esquemas, también se instalan (o desinstalan, según el caso) los esquemas dependientes.

#### En la IGU

Para instalar esquemas con la IGU del Gestor de esquemas, seleccione los esquemas que quiere instalar y haga clic en **Aplicar**.

También puede seleccionar los esquemas que quiere instalar en el [sitio web de Altova](#) y generar desde allí un archivo `.altova_schemas`. Al hacer doble clic en este archivo se abre el Gestor de esquemas con los esquemas que indicó preseleccionados. Solo tiene que hacer clic en **Aplicar**.

#### En la línea de comandos

Para instalar esquemas desde la línea de comandos ejecute el comando `install`:

```
xmlschemamanager.exe install [opciones] Schema+
```

donde **FILTER** es el esquema (o los esquemas) que quiere instalar o un archivo `.altova_schemas`. Para hacer referencia a un esquema se usa un identificador con el formato `<nombre>-<versión>` que aparece junto a cada esquema que muestra el comando `list`<sup>97</sup>. Puede introducir tantos esquemas como quiera. Para más detalles consulte la descripción del comando `install`<sup>97</sup>.

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

*Instalar un esquema requerido*

Si ejecuta un comando en StyleVision Server y StyleVision Server descubre que uno de los esquemas que necesita para ejecutar el comando falta o está incompleta, el Gestor de esquemas incluirá información sobre ese componente de esquema que falta. Entonces puede aplicar el parche indicado y/o instalar el esquema que falta.

Siempre puede ver todos los esquemas instalados previamente ejecutando el Gestor de esquemas desde **Herramientas | Gestor de esquemas**.

## 6.4 Desinstalar o restaurar esquemas

### Desinstalar un esquema

Para desinstalar esquemas puede usar la IGU del Gestor de esquemas o enviar las instrucciones al Gestor de esquemas desde la línea de comandos.

**Nota:** si el esquema actual tiene dependencias en otros esquemas, también se instalan (o desinstalan, según el caso) los esquemas dependientes.

#### En la IGU

Para desinstalar esquemas con la IGU del Gestor de esquemas, seleccione los esquemas que quiere desinstalar y haga clic en **Aplicar**. Los esquemas seleccionadas y sus dependencias se desinstalarán.

Para desinstalar todos los esquemas haga clic en **Deseleccionar todas** y haga clic en **Aplicar**.

#### En la línea de comandos

Para desinstalar esquemas desde la línea de comandos ejecute el comando `uninstall`<sup>99</sup>:

```
xmlschemamanager.exe uninstall [options] FILTER+
```

donde **FILTER** es el esquema (o los esquemas) que quiere desinstalar o un archivo `.altova_schemas`. Para hacer referencia a un esquema se usa un identificador con el formato `<nombre>-<versión>` que aparece junto a cada esquema que muestra el comando `list`<sup>97</sup>. Puede introducir tantos esquemas como quiera. Para más detalles consulte la descripción del comando `uninstall`<sup>99</sup>.

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

### Restaurar el Gestor de esquemas

Puede restaurar el Gestor de esquemas, es decir, eliminar todos los esquemas instaladas, así como el directorio caché.

- En la IGU, haga clic en **Restaurar selección**.
- En la línea de comandos, use el comando `reset`<sup>98</sup>.

Una vez haya ejecutado este comando, asegúrese de que ejecuta también el comando `initialize`<sup>96</sup> para recrear el directorio caché. También puede ejecutar el comando `reset`<sup>98</sup> con la opción `-i`.

Recuerde que `reset -i`<sup>98</sup> restaura la instalación original del producto, por lo que es recomendable ejecutar el comando `update`<sup>100</sup> después de restaurar el gestor. Puede ejecutar el comando `reset`<sup>98</sup> con las opciones `-i` o `-u`.

## 6.5 Interfaz de la línea de comandos (ILC)

Para llamar a Gestor de esquemas desde la línea de comandos necesita saber la ruta del ejecutable. Por defecto, el ejecutable del Gestor de esquemas se encuentra en:

<i>Windows</i>	C:\ProgramData\Altova\SharedBetweenVersions\XMLSchemaManager.exe
<i>Linux</i>	/opt/Altova/StyleVisionServer2024/bin/xmlschemamanager
<i>macOS</i>	/usr/local/Altova/StyleVisionServer2024/bin/xmlschemamanager

**Nota:** en los sistemas Linux y macOS una vez haya cambiado el directorio al que contiene el ejecutable, puede llamar al ejecutable con `sudo ./xmlschemamanager`. El prefijo `./` indica que el ejecutable está en el directorio actual. El prefijo `sudo` indica que el comando se debe ejecutar con derechos de administrador.

### Sintaxis de la línea de comandos

La sintaxis general para usar la línea de comandos es:

```
<exec> -h | --help | --version | <command> [opciones] [argumentos]
```

En el código anterior la barra vertical `|` separa elementos que se excluyen mutuamente. Los corchetes `[]` indican elementos opcionales. Básicamente, puede teclear la ruta del ejecutable seguida por las opciones `--h`, `--help` o `--version`, o por un comando. Cada comando puede tener opciones y argumentos. Los comandos se describen en los apartados siguientes.

### 6.5.1 help

Este comando ofrece ayuda contextual sobre los comandos del ejecutable del Gestor de esquemas.

#### Sintaxis

```
<exec> help [command]
```

Donde `[command]` es un argumento opcional que indica cualquier nombre válido de comando.

Tenga en cuenta que:

- Puede invocar la ayuda tecleando un comando seguido por `--h` o `--help`, por ejemplo: `<exec> list -h`
- Puede invocar la ayuda general tecleando `--h` o `--help` directamente después del ejecutable, por ejemplo:

#### Ejemplo

Este comando muestra la ayuda del comando `list`:

```
xmlschemamanager help list
```

## 6.5.2 info

Este comando muestra información detallada sobre cada uno de los esquemas dados como argumento. Esta información incluye el título, la versión, la descripción, el editor y las referencias de las dependencias.

### Sintaxis

```
<exec> info [options] Schema+
```

- El argumento `schema` es el nombre de un esquema o parte del nombre de un esquema. (Para ver el ID de un paquete de esquemas y la información relativa a su estado de instalación use el comando [list](#)<sup>97</sup>.)
- Use `<exec> info -h` para ver la ayuda sobre este comando en la línea de comandos..

### Ejemplo

Este comando muestra información detallada sobre los esquemas `DocBook-DTD` y `NITF`:

```
xmlschemamanager info doc nitf
```

## 6.5.3 initialize

Este comando inicializa el entorno del Gestor de esquemas y crea un directorio caché donde se guardan todos los esquemas localmente. El Gestor de esquemas se inicializa automáticamente la primera vez que instale una aplicación de Altova compatible con él, por lo que normalmente no es necesario ejecutar este comando. Por lo general solo es necesario ejecutarlo después de haber ejecutado el comando `reset`.

### Sintaxis

```
<exec> initialize | init [opciones]
```

#### Opciones

Estas son las opciones del comando `initialize`:

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

### Ejemplo

Este comando inicializa el Gestor de esquemas:

```
xmlschemamanager initialize
```

## 6.5.4 install

Este comando instala una o más esquemas.

### Sintaxis

```
<exec> install [options] Schema+
```

Para indicar varios esquemas, repita el argumento `Schema` tantas veces como sea necesario.

El argumento de `Schema` puede ser:

1. Un identificador de esquema en el formato `<name>-<version>`, por ejemplo: `cocr-2.10`. Para ver todos los identificadores de esquemas y sus versiones ejecute el comando [list](#)<sup>97</sup>. También puede usar el nombre de el esquema abreviado, si este es único, por ejemplo `docbook`. Si usa una abreviación del nombre se desinstalan todos los esquemas que contengan esa abreviación.
2. La ruta de acceso a un archivo `.altova_schemas` descargado desde el sitio web de Altova. Para más información sobre estos archivos consulte la [Introducción al Gestor de esquemas: funcionamiento](#)<sup>83</sup>.

### Opciones

Estas son las opciones del comando `install`:

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

### Ejemplo

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

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

## 6.5.5 list

Use este comando para ver los esquemas del Gestor de esquemas; tiene varias opciones:

- lista de todos los esquemas disponibles
- lista de esquemas específicos
- lista de los esquemas instalados
- lista de los esquemas que se pueden actualizar.

## Sintaxis

```
<exec> list | ls [options] Schema?
```

Si no se indica ningún argumento `schema` la lista incluye todos los esquemas. De lo contrario la lista incluye los esquemas indicados en las opciones (véase el ejemplo de más abajo). Recuerde que puede usar el argumento `schema` tantas veces como quiera.

### Opciones

Estas son las opciones del comando `list`:

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

## Ejemplos

- Para ver todos los esquemas disponibles ejecute: `xmlschemamanager list`
- Para ver solamente los esquemas instaladas ejecute: `xmlschemamanager list -i`
- Para ver todos los esquemas cuyos nombres contienen "doc" o "nitf" ejecute: `xmlschemamanager list doc nitf`

## 6.5.6 reset

Este comando elimina todos los esquemas instalados, así como el directorio de caché. Este comando elimina todos los esquemas instalados y su información. Una vez haya ejecutado este comando, asegúrese de que ejecuta el comando [initialize](#)<sup>96</sup> para volver a crear el directorio de caché. También puede ejecutar el comando `reset` con la opción `-i`. Tenga en cuenta que `reset -i` restaura la instalación original del producto, por lo que se recomienda ejecutar también el comando [update](#)<sup>100</sup> después de una restauración. También puede ejecutar el comando `reset` con las opciones `-i` y `-u`.

## Sintaxis

```
<exec> reset [opciones]
```

### Opciones

Estas son las opciones del comando `reset`:

<code>--help, --h</code>	Muestra la ayuda sobre este comando en la línea de comandos.
<code>--init, --i</code>	Inicializa el entorno del Gestor de esquemas XML después de una restauración. El valor predeterminado es <code>false</code> .

<code>--silent, --s</code>	Muestra solamente los mensajes de error. El valor predeterminado es <code>false</code> .
<code>--update, --u</code>	Inicializa y actualiza el entorno del Gestor de esquemas XML después de una restauración. El valor predeterminado es <code>false</code> .
<code>--verbose, --v</code>	Muestra información suplementaria durante la ejecución. El valor predeterminado es <code>false</code> .

## Ejemplos

- Para restaurar el Gestor de esquemas, ejecute: `xmlschemamanager reset`
- Para restaurar el Gestor de esquemas e inicializarlo, ejecute: `xmlschemamanager reset -i`
- Para restaurar el Gestor de esquemas, inicializarlo y actualizar la lista de esquemas, ejecute: `xmlschemamanager reset -i -u`

## 6.5.7 uninstall

Este comando desinstala una o más esquemas. Por defecto, cualquier esquema a la que haga referencia el esquema actual también se desinstala. Para desinstalar solamente el esquema actual y mantener aquellas a las que se hace referencia, use la opción `--k`.

### Sintaxis

```
<exec> uninstall [opciones] Schema+
```

Para indicar varios esquemas, repita `FILTER` tantas veces como sea necesario.

El argumento de `schema` puede ser:

- Un identificador de esquema en el formato `<name>-<version>`, por ejemplo: `eba-2.10`). Para ver todos los identificadores de esquemas y sus versiones ejecute el comando `list -i`<sup>97</sup>. También puede usar el nombre del esquema abreviado, si este es único, por ejemplo `eba`. Si usa una abreviación del nombre se desinstalan todos los esquemas que contengan esa abreviación.
- La ruta de acceso a un archivo `.altova_taxonomies` descargado desde el sitio web de Altova. Para más información sobre estos archivos consulte la [Introducción al Gestor de esquemas: funcionamiento](#)<sup>83</sup>.

### Opciones

Estas son las opciones del comando `uninstall`:

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

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

## Ejemplo

Este comando desinstala los esquemas CBCR 2.0 y EPUB 2.0:

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

Este comando desinstala el esquema `eba-2.10` pero no los esquemas a los que hace referencia:

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

## 6.5.8 update

Este comando consulta la lista de esquemas disponibles en el almacenamiento en línea y actualiza el directorio de caché local. Esta información se actualiza de forma implícita, por lo que no es necesario ejecutar este comando a no ser que haya ejecutado [reset](#)<sup>98</sup> e [initialize](#)<sup>96</sup>.

### Sintaxis

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

#### Opciones

Estas son las opciones del comando `update`:

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

## Ejemplo

Este comando actualiza la lista de esquemas:

```
xmlschemamanager update
```

## 6.5.9 upgrade

Este comando actualiza todos los esquemas aptos para la versión *patche* más reciente disponible. Puede identificar cuáles lo son con el comando `list -u`.

**Nota:** el comando `upgrade` eliminaría un esquema obsoleto si no hay ninguna versión disponible.

### Sintaxis

```
<exec> upgrade [opciones]
```

Opciones

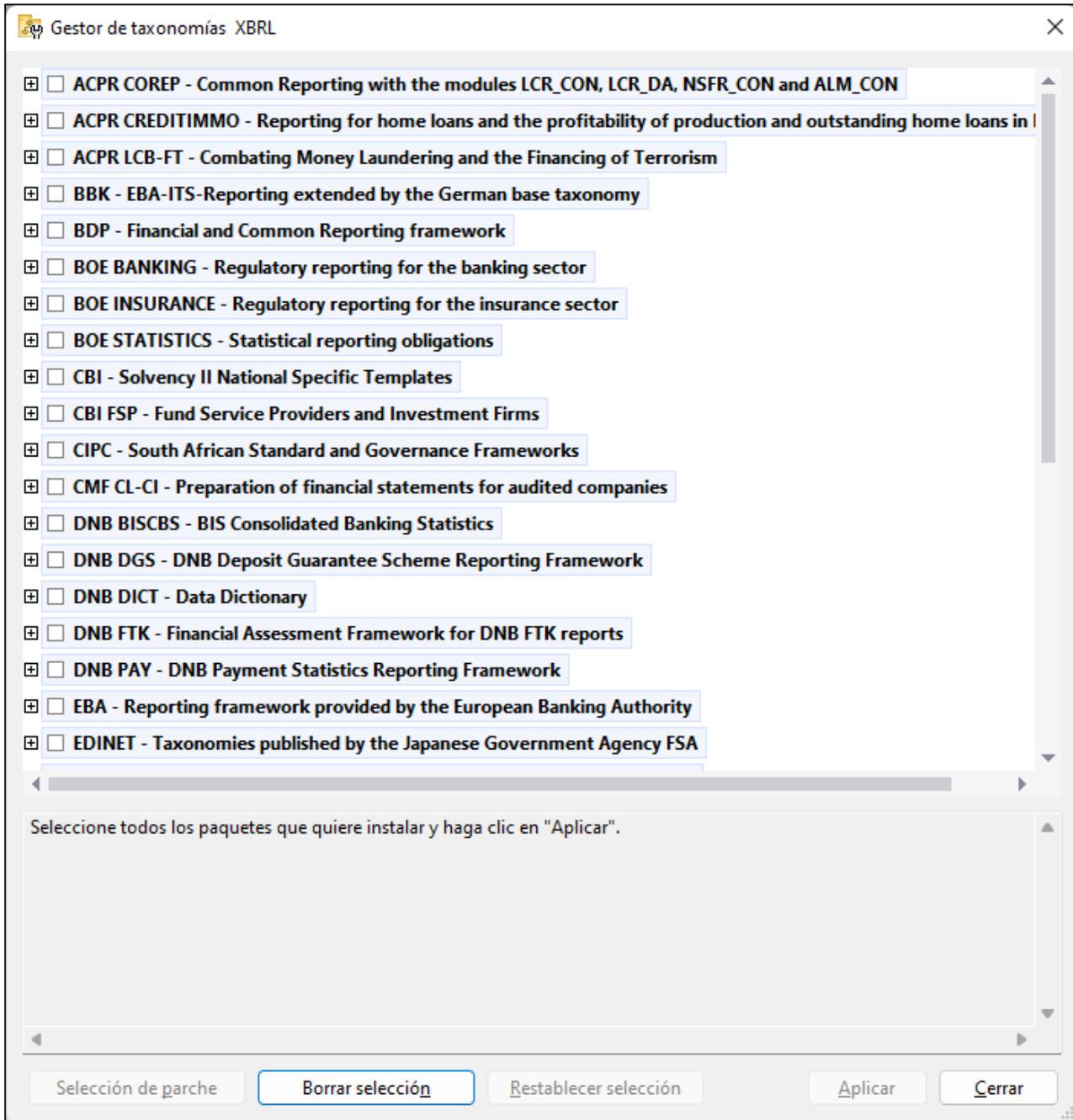
Estas son las opciones del comando `upgrade`:

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

## 7 Gestor de taxonomías

El Gestor de taxonomías XBRL es una herramienta que ofrece una forma centralizada de instalar y administrar taxonomías XBRL para usarlas en todas las aplicaciones de Altova compatibles con XBRL, incluido **StyleVision Server**.

- En Windows, el Gestor de taxonomías tiene una interfaz gráfica del usuario (*imagen siguiente*) a la que también puede acceder desde la línea de comandos. (Las aplicaciones de escritorio de Altova sólo están disponibles para Windows; *consulte la lista siguiente*.)
- En Linux y macOS el Gestor de taxonomías sólo está disponible en la línea de comandos. (Las aplicaciones de escritorio de Altova están disponibles para Windows, Linux y macOS; *consulte la lista siguiente*.)



*Aplicaciones de Altova compatibles con XBRL*

Aplicaciones de escritorio (sólo para Windows)	Aplicaciones de servidor (Windows, Linux, macOS)
Complementos XBRL de Altova para Excel (EBA, ESEF, Solvency II, WIP)	MapForce Server (ediciones Standard y Advanced)

MapForce Enterprise Edition	RaptorXML+XBRL Server
StyleVision Enterprise Edition	StyleVision Server
XMLSpy Enterprise Edition	

### Instalación y desinstalación del Gestor de taxonomías

El Gestor de taxonomías se instala automáticamente al instalar cualquiera de las aplicaciones de Altova compatibles con XBRL o el Altova Mission Kit Enterprise Edition (véase *la tabla de más arriba*). También se elimina automáticamente si desinstala todas las aplicaciones de Altova compatibles con XBRL o el Altova Mission Kit Enterprise Edition del equipo.

### Características de Gestor de taxonomías

El Gestor de taxonomías permite:

- Ver las taxonomías XBRL que hay instaladas en su equipo y comprobar si hay versiones nuevas para descargar.
- Descargar las versiones más recientes de las taxonomías XBRL independientemente del ciclo de versiones de Altova. Altova guarda todas las taxonomías en un sistema de almacenamiento en línea al que tiene acceso el Gestor de taxonomías y desde donde puede descargarlas tan pronto como estén disponibles.
- Instalar o desinstalar cualquiera de las múltiples versiones de una taxonomía en concreto (o todas ellas, si las necesita).
- Una sola taxonomía XBRL representa un "paquete", pero puede tener dependencias en otras taxonomías. Al instalar o desinstalar una taxonomía, se detectan e instalan o desinstalan también automáticamente todas sus dependencias. La interfaz gráfica del usuario (o la línea de comandos, en su caso) le informa cuando se añaden o eliminan taxonomías.
- Las taxonomías XBRL administradas con el Gestor de taxonomías pueden usar el [catálogo XML](#), que permite resolver referencias a URI en documentos de instancia o esquema desde archivos locales, en vez de a través de Internet.
- Todas las taxonomías principales están incluidas en el Gestor de taxonomías y se actualizan de forma periódica a la versión más reciente. De esta forma puede administrar todas las taxonomías desde un punto común y tenerlas siempre listas para las aplicaciones de Altova que las usan.
- Los cambios que se realizan en el Gestor de taxonomías afectan a todos los productos de Altova que estén instalados en ese equipo.

### Taxonomías XBRL personales

Si necesita trabajar con taxonomías XBRL que no estén incluidas en el Gestor de taxonomías, puede agregarlas a un paquete personal al que el StyleVision Server podrá hacer referencia más adelante. Puede habilitar las taxonomías XBRL así:

- *Desde las aplicaciones de escritorio de Altova:* ejecute el comando de menú **Herramientas | Opciones** y después vaya a la página de opciones *XBRL | Paquetes de taxonomía*. Desde allí navegue hasta el paquete .zip de la taxonomía XBRL personal que quiera usar. Para más información consulte la descripción de este comando en la documentación del producto correspondiente.
- *Desde las aplicaciones servidor de Altova:* desde una línea de comandos compatible con XBRL proporcione las opciones `-taxonomy-package 0 ---taxonomy-package-config-file`. Por ejemplo, estas opciones son compatibles con los comandos de validación XBRL como `valxbr1` o `valxbr1taxonomy` en RaptorXML+XBRL Server, o el comando `run` en MapForce Server.

## Funcionamiento

Altova mantiene un almacenamiento en línea donde guarda todas las taxonomías XBRL de los productos de Altova. Este almacenamiento se actualiza de forma periódica, por ejemplo, poco después de que las organizaciones correspondientes publiquen las versiones nuevas de las taxonomías respectivas. Al ejecutar Gestor de taxonomías desde la interfaz gráfica del usuario aparece información sobre las taxonomías más recientes disponibles en un cuadro de diálogo en el que puede visualizarlas, instalarlas, actualizarlas o desinstalarlas. También puede llevar a cabo las mismas acciones desde la línea de comandos.

También puede instalar las taxonomías ejecutando los archivos `.altova_taxonomies` descargados desde el [sitio web de Altova](#). El sitio web prepara un archivo de tipo `.altova_taxonomies` que puede descargar y que contiene la información sobre las taxonomías seleccionadas. Al hacer doble clic en este archivo o pasarlo al **Gestor de taxonomías** desde la línea de comandos como argumento del comando `install`<sup>116</sup>, el Gestor de taxonomías instala las taxonomías que contiene.

*Memoria caché local: seguimiento de taxonomías*

Independientemente de cómo se instalen las taxonomías, toda la información sobre las taxonomías instaladas se almacena en una ubicación centralizada de su equipo, el directorio caché. El directorio caché local está en:

<i>Windows</i>	C:\ProgramData\Altova\pkgs\.cache
<i>Linux</i>	/var/opt/Altova/pkgs\.cache
<i>macOS</i>	/var/Altova/pkgs

El directorio caché local se actualiza automáticamente de vez en cuando para que el estado más actual del equipo corresponda con el del almacenamiento en línea. Más concretamente, la memoria caché se actualiza:

- Al ejecutar el Gestor de taxonomías.
- Al ejecutar StyleVision Server por primera vez en un mismo día natural.
- Si StyleVision Server ya se está ejecutando, el directorio caché se actualiza cada 24 horas.
- También puede actualizar el caché local desde el almacenamiento en línea manualmente ejecutando el comando de actualización `update`<sup>119</sup> desde la línea de comandos.

Si instala o desinstala taxonomías, el directorio caché local del Gestor de taxonomías se actualiza automáticamente con información sobre las taxonomías disponibles e instaladas, además de con los propios archivos de taxonomía.

### No modifique la memoria caché manualmente.

El directorio caché local se mantiene automáticamente en base a las taxonomías que instale o desinstale; no debe modificarlo ni eliminarlo manualmente. Si necesita restaurar el Gestor de taxonomías a su estado original, ejecute el comando `reset`<sup>117</sup> desde la línea de comandos y después ejecute el comando `initialize`<sup>115</sup>. (También puede ejecutar el comando `reset` con la opción `--i`.)

## Proxy de HTTP

Puede usar un proxy de HTTP para las conexiones del Gestor de taxonomías. Se utilizará la configuración proxy del sistema.

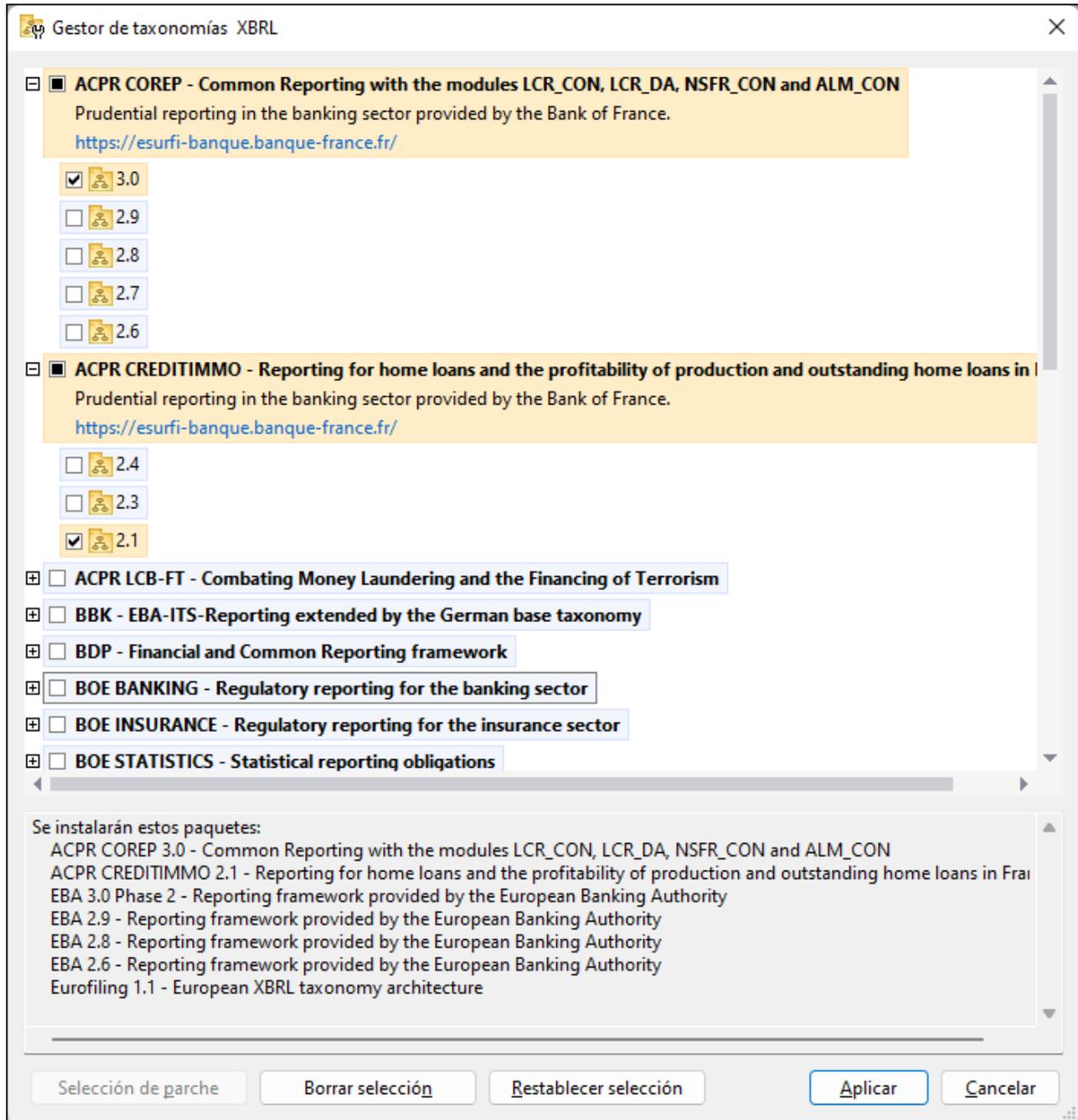
## 7.1 Ejecutar el gestor de taxonomías

### Interfaz gráfica del usuario

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

- *Durante la instalación de StyleVision Server:* al final del proceso de instalación, seleccione la casilla Invocar al Gestor de taxonomías para acceder directamente al gestor de taxonomías XBRL. Con él puede instalar taxonomías durante el proceso de instalación de su aplicación de Altova.
- Mediante el archivo `.altova_taxonomies` que descargó del [Centro de descargas de taxonomías XBRL de Altova](#): haga doble clic en el archivo para ejecutar Gestor de taxonomías, que instalará las taxonomías que haya seleccionado.

Cuando se abra la IGU del Gestor de taxonomías (*imagen siguiente*) podrá ver en ella las taxonomías que ya se han instalado. Si quiere instalar más sólo tiene que seleccionarlas, y al contrario si quiere desinstalar alguna. Una vez haya terminado, puede aplicar los cambios. Las taxonomías que se vayan a instalar o desinstalar aparecerán resaltadas y un mensaje le avisará de los cambios que está a punto de hacer en la ventana *Mensajes*, en la parte inferior de la ventana del **Gestor de taxonomías** (*imagen siguiente*).



## Interfaz de la línea de comandos

Para ejecutar el Gestor de taxonomías desde una interfaz de la línea de comandos debe usar su archivo ejecutable, `taxonomymanager.exe`.

Puede encontrar este archivo:

- *En Windows:* C:\ProgramData\Altova\SharedBetweenVersions
- *En Linux o macOS (sólo para aplicaciones de servidor):* %INSTALLDIR%/bin, donde %INSTALLDIR% es el directorio de instalación del programa.

Puede usar cualquiera de los comandos de la referencia de la línea de comandos, a continuación.

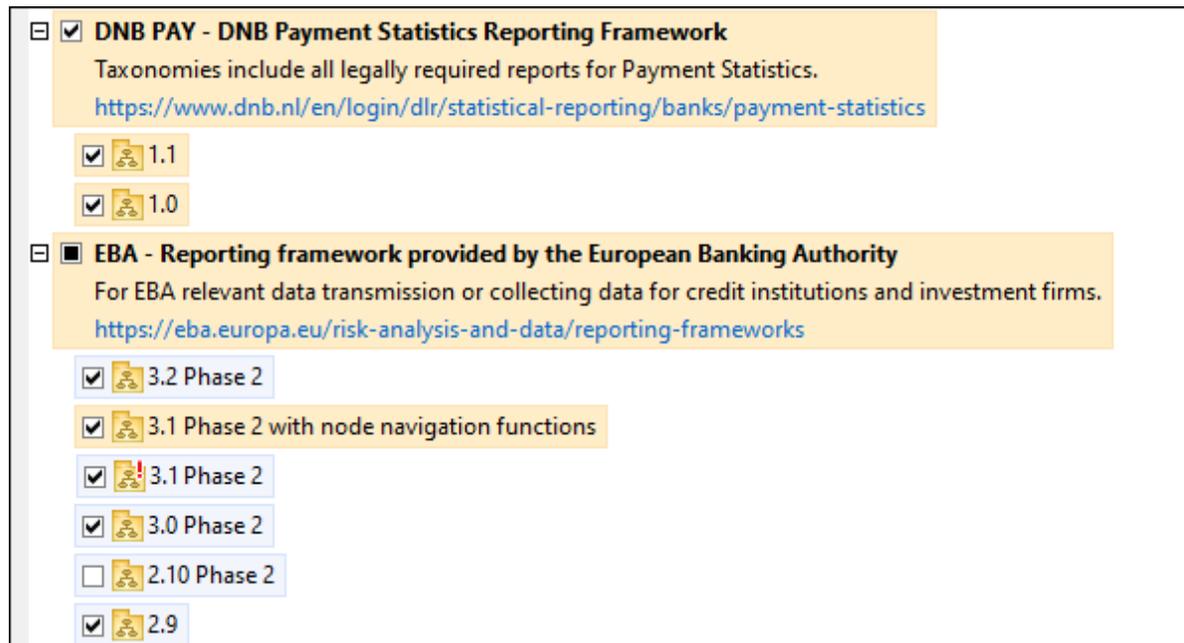
Para mostrar al ayuda de los comandos ejecute:

- *En Windows:* `taxonomymanager.exe --help`
- *En Linux o macOS (sólo para aplicaciones de servidor):* `sudo ./taxonomymanager --help`

## 7.2 Categorías de estado

El Gestor de taxonomías diferencia las taxonomías que administra entre:

- *Taxonomías instaladas*: estas aparecen en la IGU con sus casillas marcadas (*en la imagen siguiente las versiones marcadas de la taxonomía EBA son las que están instaladas*). Si se seleccionan todas las versiones de una taxonomía, en la casilla de la taxonomía aparece una marca de verificación. Si hay al menos una taxonomía sin seleccionar, en la casilla de la taxonomía aparece un cuadrado negro. Para **desinstalar** una taxonomía debe desmarcar la casilla correspondiente.
- *Taxonomías disponibles no instaladas*: Estas aparecen en la IGU con las casillas correspondientes sin seleccionar. Para **instalar** taxonomías, marque la casilla correspondiente.



- *Taxonomías que pueden actualizarse*: son las que han sido revisadas por sus emisores. Aparecen indicadas en la IGU con el icono  (*imagen anterior*). Puede aplicar **parches** a la taxonomía seleccionada con la revisión que esté disponible.

### Notas importantes

- En la imagen anterior se han marcado tanto las taxonomías DNB como las EBA. Las que tienen un fondo azul ya están instaladas. Las que tienen el fondo amarillo no están instaladas pero se han seleccionado para instalarlas. Observe que (i) la taxonomía EBA 2.10 Phase 2 no está instalada ni se ha seleccionado para instalarla, (ii) la taxonomía EBA 3.1 Phase 2 se ha instalado pero su emisor ha generado un parche que no se ha instalado aún.
- Al ejecutar el Gestor de taxonomías desde la línea de comandos puede usar el comando `list`<sup>116</sup> con distintas opciones para ver distintas categorías de taxonomías:

<code>taxonomymanager.exe list</code>	Muestra todas las taxonomías instaladas y disponibles; también indica qué taxonomías se pueden actualizar
---------------------------------------	---

<code>taxonomymanager.exe list -i</code>	Muestra sólo las taxonomías instaladas; también indica qué taxonomías se pueden actualizar
<code>taxonomymanager.exe list -u</code>	Muestra qué taxonomías se pueden actualizar

**Nota:** En Linux y macOS use `sudo ./taxonomymanager list`

## 7.3 Aplicar parches o instalar una taxonomía

### Aplicar un parche a una taxonomía instalada

A veces los emisores de las taxonomías XBRL generan parches. Cuando el Gestor de taxonomías detecta que hay parches disponibles, estos aparecen en las listas del Gestor de taxonomías, desde donde puede instalarlos.

#### En la IGU

Los parches se indican con el icono . (Consulte también el apartado anterior sobre las [categorías de estado](#) <sup>109</sup>.) Si hay parches disponibles se habilita el botón **Seleccionar parches**. Haga clic en él para seleccionar y preparar los parches. En la IGU, el icono de las taxonomías correspondientes cambia de  a  y el cuadro de diálogo le informa de qué parches se van a aplicar. Las listas del panel principal y las del panel Mensajes están ordenadas alfabéticamente, por lo que puede ver y revisar las taxonomías antes de aplicar los parches. Una vez esté listo para instalar los parches seleccionados, haga clic en **Aplicar**.

#### En la línea de comandos

Para aplicar un parche desde la línea de comandos:

1. Ejecute el comando `list -u` <sup>116</sup>. Aparece una lista con las taxonomías para las que hay parches disponibles.
2. Ejecute el comando `upgrade` <sup>119</sup> para instalar todos los parches.

### Instalar una taxonomía disponible

Para instalar taxonomías puede usar la IGU del Gestor de taxonomías o enviar las instrucciones al Gestor de taxonomías desde la línea de comandos.

**Nota:** Si la taxonomía actual tiene dependencias en otras taxonomías, también se instalan (o desinstalan, según el caso) las taxonomías dependientes.

#### En la IGU

Para instalar taxonomías con la IGU del Gestor de taxonomías, seleccione las taxonomías que quiere instalar y haga clic en **Aplicar**.

También puede seleccionar las taxonomías que quiere instalar en el [sitio web de Altova](#) y generar desde allí un archivo `.altova_taxonomies`. Al hacer doble clic en este archivo se abre el Gestor de taxonomías con las taxonomías que indicó preseleccionadas. Sólo tiene que hacer clic en **Aplicar**.

#### En la línea de comandos

Para instalar taxonomías desde la línea de comandos ejecute el comando `install` <sup>116</sup>:

```
taxonomymanager.exe install [opciones] FILTER+
```

donde **FILTER** es la taxonomía (o las taxonomías) que quiere instalar o un archivo `.altova_taxonomies`. Para hacer referencia a una taxonomía se usa un identificador con el formato `<nombre>-<versión>` que aparece junto a cada taxonomía que muestra el comando `list` <sup>116</sup>. Puede introducir tantas taxonomías como quiera. Para más detalles consulte la descripción del comando `install` <sup>116</sup>.

**Nota:** En Linux o macOS, use el comando `sudo ./taxonomymanager`.

*Instalar una taxonomía requerida*

Si ejecuta un comando en StyleVision Server y StyleVision Server descubre que una de las taxonomías que necesita para ejecutar el comando falta o está incompleta, el Gestor de taxonomías incluirá información sobre ese componente de taxonomía que falta. Entonces puede aplicar el parche indicado y/o instalar la taxonomía que falta.

Siempre puede ver todas las taxonomías instaladas previamente ejecutando el Gestor de taxonomías desde **Herramientas | Gestor de taxonomías**.

## 7.4 Desinstalar o restaurar taxonomías

### Desinstalar una taxonomía

Para desinstalar taxonomías puede usar la IGU del Gestor de taxonomías o enviar las instrucciones al Gestor de taxonomías desde la línea de comandos.

**Nota:** Si la taxonomía actual tiene dependencias en otras taxonomías, también se instalan (o desinstalan, según el caso) las taxonomías dependientes.

#### En la IGU

Para desinstalar taxonomías con la IGU del Gestor de taxonomías, seleccione las taxonomías que quiere desinstalar y haga clic en **Aplicar**. Las taxonomías seleccionadas y sus dependencias se desinstalarán.

Para desinstalar todas las taxonomías haga clic en **Deseleccionar todas** y haga clic en **Aplicar**.

#### En la línea de comandos

Para desinstalar taxonomías desde la línea de comandos ejecute el comando `uninstall`:

```
taxonomymanager.exe uninstall [options] FILTER+
```

donde **FILTER** es la taxonomía (o las taxonomías) que quiere desinstalar o un archivo `.altova_taxonomies`. Para hacer referencia a una taxonomía se usa un identificador con el formato `<nombre>-<versión>` que aparece junto a cada taxonomía que muestra el comando [list](#)<sup>116</sup>. Puede introducir tantas taxonomías como quiera. Para más detalles consulte la descripción del comando [uninstall](#)<sup>118</sup>.

**Nota:** En Linux o macOS, use el comando `sudo ./taxonomymanager`.

### Restaurar el Gestor de taxonomías

Puede restaurar el Gestor de taxonomías.

- En la IGU, haga clic en **Restaurar selección**. Esto reinicia la IGU para mostrar qué taxonomías están actualmente instaladas. Se cancelarán todas las opciones que el usuario haya seleccionado o deseleccionado en la sesión actual.
- En la línea de comandos, use el comando [reset](#)<sup>117</sup>. Esto eliminará todas las taxonomías instaladas y el directorio de caché.

Una vez haya ejecutado este comando, asegúrese de que ejecuta también el comando [initialize](#)<sup>115</sup> para recrear el directorio caché. También puede ejecutar el comando [reset](#)<sup>117</sup> con la opción `-i`.

Recuerde que [reset -i](#)<sup>117</sup> restaura la instalación original del producto, por lo que es recomendable ejecutar el comando [update](#)<sup>119</sup> después de restaurar el gestor. Puede ejecutar el comando [reset](#)<sup>117</sup> con las opciones `-i 0 -u`.

## 7.5 Interfaz de la línea de comandos (ILC)

Para llamar al Gestor de taxonomías desde la línea de comandos necesita saber la ruta del ejecutable. Por defecto, el ejecutable del Gestor de taxonomías se encuentra en:

<i>Windows</i>	C:\ProgramData\Altova\SharedBetweenVersions\TaxonomyManager.exe
<i>Linux</i>	/opt/Altova/StyleVisionServer2024/bin/taxonomymanager
<i>macOS</i>	/usr/local/Altova/StyleVisionServer2024/bin/taxonomymanager

**Nota:** En los sistemas Linux y macOS una vez haya cambiado el directorio al que contiene el ejecutable, puede llamar al ejecutable con `sudo ./taxonomymanager`. El prefijo `./` indica que el ejecutable está en el directorio actual. El prefijo `sudo` indica que el comando se debe ejecutar con derechos de administrador.

### Sintaxis de la línea de comandos

La sintaxis general para usar la línea de comandos es:

```
<exec> -h | --help | --version | <command> [opciones] [argumentos]
```

En el código anterior la barra vertical `|` separa elementos que se excluyen mutuamente. Los corchetes `[]` indican elementos opcionales. Básicamente, puede teclear la ruta del ejecutable seguida por las opciones `--h`, `--help` o `--version`, o por un comando. Cada comando puede tener opciones y argumentos. Los comandos se describen en los apartados siguientes.

### 7.5.1 help

Este comando ofrece ayuda contextual sobre los comandos del ejecutable del Gestor de taxonomías.

#### Sintaxis

```
<exec> help [command]
```

Donde `[command]` es un argumento opcional que indica cualquier nombre válido de comando.

Tenga en cuenta que:

- Puede invocar la ayuda tecleando un comando seguido por `--h` or `--help`, por ejemplo: `<exec> list -h`
- Puede invocar la ayuda general (no para el comando) tecleando `--h` o `--help` directamente después del ejecutable, por ejemplo: `<exec> -h list`

#### Ejemplo

Este comando muestra la ayuda del comando `list`:

```
taxonomymanager help list
```

## 7.5.2 info

Este comando muestra información detallada sobre cada una de las taxonomías dadas como argumento. Esta información incluye el título, la versión, la descripción, el editor y las referencias de las dependencias.

### Sintaxis

```
<exec> info [opciones] Taxonomy+
```

- El argumento **Taxonomy** es el nombre de una taxonomía o parte del nombre de una taxonomía. (Para ver el ID de un paquete de taxonomías y la información relativa a su estado de instalación use el comando [list](#)<sup>116</sup>.)
- Use `<exec> info -h` para ver la ayuda sobre este comando en la línea de comandos.

### Ejemplo

Este comando muestra información detallada sobre las taxonomías `eba-2.10` y `us-gaap-2020`:

```
taxonomymanager info eba-2.10 us-gaap-2020.0
```

## 7.5.3 initialize

Este comando inicializa el entorno del Gestor de taxonomías y crea un directorio caché donde se guardan todas las taxonomías localmente. El Gestor de taxonomías se inicializa automáticamente la primera vez que instale una aplicación de Altova compatible con él, por lo que normalmente no es necesario ejecutar este comando. Por lo general sólo es necesario ejecutarlo después de haber ejecutado el comando [reset](#)<sup>117</sup>.

### Sintaxis

```
<exec> initialize | init [opciones]
```

#### Opciones

Estas son las opciones del comando `initialize`:

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

### Ejemplo

Este comando inicializa el Gestor de taxonomías:

```
taxonomymanager initialize
```

## 7.5.4 install

Este comando instala una o más taxonomías.

### Sintaxis

```
<exec> install [options] Taxonomy+
```

Para indicar varias taxonomías, repita `FILTER` tantas veces como sea necesario.

El argumento de `Taxonomy` puede ser:

1. Un identificador de taxonomía en el formato `<name>-<version>`, por ejemplo: `eba-2.10`. Para ver todos los identificadores de taxonomías y sus versiones ejecute el comando `list`<sup>116</sup>. También puede usar el nombre de la taxonomía abreviado, si este es único, por ejemplo `eba`. Si usa una abreviación del nombre se desinstalan todas las taxonomías que contengan esa abreviación.
2. La ruta de acceso a un archivo `.altova_taxonomies` descargado desde el sitio web de Altova. Para más información sobre estos archivos consulte la [Introducción al Gestor de taxonomías: Funcionamiento](#)<sup>102</sup>.

### Opciones

Estas son las opciones del comando `install`:

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

### Ejemplo

Este comando instala las taxonomías `eba` (Autoridad Bancaria Europea) y `us-gaap` (principios de contabilidad generalmente aceptados en EE UU):

```
taxonomymanager install eba us-gaap
```

## 7.5.5 list

Use este comando para ver las taxonomías del Gestor de taxonomías; tiene varias opciones:

- Lista de todas las taxonomías disponibles
- Lista de taxonomías específicas
- Lista de las taxonomías instaladas

- Lista de las taxonomías que se pueden actualizar

## Sintaxis

```
<exec> list | ls [options] Taxonomy?
```

Si no se indica ningún argumento `Taxonomy` la lista incluye todas las taxonomías. De lo contrario la lista incluye las taxonomías indicadas en las opciones (véase el ejemplo de más abajo). Recuerde que puede usar el argumento `Taxonomy` tantas veces como quiera.

### Opciones

Estas son las opciones del comando `list`:

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

## Ejemplos

- Para ver todas las taxonomías disponibles ejecute: `taxonomymanager list`
- Para ver solamente las taxonomías instaladas ejecute: `taxonomymanager list -i`
- Para ver todas las taxonomías cuyos nombres contienen "eba" o "us-gaap" ejecute: `taxonomymanager list eba us-gaap`

## 7.5.6 reset

Este comando elimina todas las taxonomías instaladas, así como el directorio de caché. Este comando elimina todas las taxonomías instaladas y su información. Una vez haya ejecutado este comando, asegúrese de que ejecuta el comando [initialize](#)<sup>115</sup> para volver a crear el directorio de caché. También puede ejecutar el comando `reset` con la opción `-i`. Tenga en cuenta que `reset -i` restaura la instalación original del producto, por lo que se recomienda ejecutar también el comando [update](#)<sup>119</sup> después de una restauración. También puede ejecutar el comando `reset` con las opciones `-i` y `-u`.

## Sintaxis

```
<exec> reset [opciones]
```

### Opciones

Estas son las opciones del comando `reset`:

<code>--help, --h</code>	Muestra la ayuda sobre este comando en la línea de comandos.
--------------------------	--

<code>--init, --i</code>	Inicializa el entorno del Gestor de taxonomías después de una restauración. El valor predeterminado es <code>false</code> .
<code>--silent, --s</code>	Muestra solamente los mensajes de error. El valor predeterminado es <code>false</code> .
<code>--update, --u</code>	Inicializa y actualiza el entorno del Gestor de taxonomías XBRL después de una restauración. El valor predeterminado es <code>false</code> .
<code>--verbose, --v</code>	Muestra información suplementaria durante la ejecución. El valor predeterminado es <code>false</code> .

## Ejemplos

- Para restaurar el Gestor de taxonomías, ejecute: `taxonomymanager reset`
- Para restaurar el Gestor de taxonomías e inicializarlo, ejecute: `taxonomymanager reset -i`
- Para restaurar el Gestor de taxonomías, inicializarlo y actualizar la lista de taxonomías, ejecute: `taxonomymanager reset -i -u`

## 7.5.7 uninstall

Este comando desinstala una o más taxonomías. Por defecto, cualquier taxonomía a la que haga referencia la taxonomía actual también se desinstala. Para desinstalar solamente la taxonomía actual y mantener aquellas a las que se hace referencia, use la opción `--k`.

### Sintaxis

```
<exec> uninstall [opciones] Taxonomy+
```

Para indicar varias taxonomías, repita `FILTER` tantas veces como sea necesario.

El argumento de `Taxonomy` puede ser:

1. Un identificador de taxonomía en el formato `<name>-<version>`, por ejemplo: `eba-2.10`). Para ver todos los identificadores de taxonomías y sus versiones ejecute el comando `list -i`<sup>116</sup>. También puede usar el nombre de la taxonomía abreviado, si este es único, por ejemplo `eba`. Si usa una abreviación del nombre se desinstalan todas las taxonomías que contengan esa abreviación.
2. La ruta de acceso a un archivo `.altova_taxonomies` descargado desde el sitio web de Altova. Para más información sobre estos archivos consulte la [Introducción al Gestor de taxonomías: Funcionamiento](#)<sup>102</sup> ..

### Opciones

Estas son las opciones del comando `uninstall`:

<code>--help, --h</code>	Muestra la ayuda sobre este comando en la línea de comandos.
<code>--keep-references, --k</code>	Si usa esta opción, las taxonomías referenciadas no se desinstalan. El valor predeterminado es <code>false</code> .

<code>--silent, --s</code>	Muestra solamente los mensajes de error. El valor predeterminado es <code>false</code> .
<code>--verbose, --v</code>	Muestra información suplementaria durante la ejecución. El valor predeterminado es <code>false</code> .

## Ejemplo

Este comando desinstala las taxonomías `eba-2.10` y `us-gaap-2020.0` y sus dependencias:

```
taxonomymanager uninstall eba-2.10 us-gaap-2020.0
```

Este comando desinstala la taxonomía `eba-2.10` pero no las taxonomías a las que hace referencia:

```
taxonomymanager uninstall --k eba-2.10
```

## 7.5.8 update

Este comando consulta la lista de taxonomías disponibles en el almacenamiento en línea y actualiza el directorio de caché local. Esta información se actualiza de forma implícita, por lo que no es necesario ejecutar este comando a no ser que haya ejecutado [reset](#)<sup>117</sup> e [initialize](#)<sup>115</sup>.

### Sintaxis

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

#### Opciones

Estas son las opciones del comando `update`:

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

## Ejemplo

Este comando actualiza la lista de taxonomías en la memoria caché local:

```
taxonomymanager update
```

## 7.5.9 upgrade

Este comando actualiza todas las taxonomías aptas para la versión parche más reciente disponible. En otras palabras, solamente actualiza a nivel de los parches de un release específico. Se recomienda ejecutar este comando slo si hay taxonomías por actualizar. Puede identificar cuáles lo son con el comando [list -u](#)<sup>116</sup>.

**Nota:** El comando `upgrade` eliminaría una taxonomía obsoleta si no hay ninguna versión disponible.

## Sintaxis

<exec> upgrade [opciones]

### Opciones

Estas son las opciones del comando `upgrade`:

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

# Índice

## A

- Actualizar StyleVision Server en Windows, 32**
- Altova ServiceController, 15**
- Asignar licencias a StyleVision Server en Linux, 22**
- Asignar licencias a StyleVision Server en macOS, 28**
- Asignar licencias a StyleVision Server en Windows, 15**
- Asignar una licencia a StyleVision Server en Linux, 23**
- Asignar una licencia a StyleVision Server en macOS, 29**
- Asignar una licencia a StyleVision Server en Windows, 17**
- Asignar una licencia para StyleVision Server,**
  - asignar licencias en Linux, 23
  - asignar licencias en macOS, 29
  - asignar licencias en Windows, 17

## C

- Conexiones de base de datos en Linux, 24**
- Configuración,**
  - en Linux, 19
- Configuración del entorno en Linux, 24**

## D

- Desinstalación, 11**
- Desinstalar, 11**

## F

- FlowForce Server,**
  - y StyleVision Server, 8
- FOP requirements, 34**

## G

- Gestor de esquemas,**
  - aplicar un parche a un esquema, 92
  - desinstalar un esquema, 94
  - estado de los esquemas en el, 90
  - funcionamiento del, 87
  - help (comando ILC), 95
  - info (comando ILC), 96
  - initialize (comando ILC), 96
  - instalar un esquema, 92
  - install (comando ILC), 97
  - introducción, 83
  - introducción a la ILC, 95
  - list (comando ILC), 97
  - lista de esquemas por estado, 90
  - reset (comando ILC), 98
  - restaurar, 94
  - uninstall (comando ILC), 99
  - update (comando ILC), 100
  - upgrade (comando ILC), 100
- Gestor de taxonomías,**
  - actualizar una taxonomía, 111
  - aplicar un parche a una taxonomía, 111
  - desinstalar una taxonomía, 113
  - estado de las taxonomías en el, 109
  - funcionamiento, 106
  - help (comando ILC), 114
  - info (comando ILC), 115
  - initialize (comando ILC), 115
  - instalar una taxonomía, 111
  - install (comando ILC), 116
  - introducción, 102
  - introducción a la ILC, 114
  - list (comando ILC), 116
  - lista de taxonomías por estado, 109
  - reset (comando ILC), 117
  - restaurar, 113
  - uninstall (comando ILC), 118
  - update (comando ILC), 119
  - upgrade (comando ILC), 119

## I

**Iniciar LicenseServer en Linux, 22**  
**Iniciar LicenseServer en macOS, 28**  
**Iniciar LicenseServer en Windows, 15**  
**Iniciar StyleVision Server en Linux, 22**  
**Iniciar StyleVision Server en macOS, 28**  
**Iniciar StyleVision Server en Windows, 15**  
**Instalación en Linux, 19**  
**Instalación en macOS, 26**  
**Instalación en Windows, 11**  
**Instalación en Windows Server Core, 12**  
**Instalación y configuración,**  
    en macOS, 26  
    en Windows, 11  
**Instalar LicenseServer en Linux, 21**  
**Instalar LicenseServer en macOS, 27**  
**Instalar LicenseServer en Windows, 14**

## J

**JRE, 34**

## L

**Linux,**  
    asignar una licencia a StyleVision Server en, 22  
    Instalación en, 19

## M

**macOS,**  
    asignar una licencia a StyleVision Server en, 28  
    Instalación en, 26  
**Migrar StyleVision Server a un equipo nuevo, 33**

## O

**OpenJDK, 34**

## R

**Registrar StyleVision Server con LicenseServer en Linux, 22**  
**Registrar StyleVision Server con LicenseServer en macOS, 29**  
**Registrar StyleVision Server con LicenseServer en Windows, 16**

## S

**StyleVision Server,**  
    características, 7  
    como módulo de FlowForce Server, 8  
    funciones, 7  
    migrar a un equipo nuevo, 33

## V

**Versiones de LicenseServer, 14, 21, 27**

## W

**Windows,**  
    actualizar StyleVision Server en, 32  
    asignar una licencia a StyleVision Server en, 15  
    Instalación en, 11