Altova StyleVision Server 2021
User & Reference Manual

All rights reserved. No parts of this work may be reproduced in any form or by any
means - graphic, electronic, or mechanical, including photocopying, recording, taping, or
information storage and retrieval systems - without the written permission of the
publisher.

Products that are referred to in this document may be either trademarks and/or
registered trademarks of the respective owners. The publisher and the author make no
claim to these trademarks.

While every precaution has been taken in the preparation of this document, the
publisher and the author assume no responsibility for errors or omissions, or for
damages resulting from the use of information contained in this document or from the
use of programs and source code that may accompany it. In no event shall the publisher
and the author be liable for any loss of profit or any other commercial damage caused or
alleged to have been caused directly or indirectly by this document.

Published: 2020

© 2020 Altova GmbH
# Table of Contents

1 Introduction ........................................ 6

2 Functionality ....................................... 8
2.1 In the FlowForce Workflow ........................................ 9
2.2 As a Standalone Server ........................................ 10

3 StyleVision Server Setup ...................... 11
3.1 Processor Cores and Licenses ........................................ 12
3.2 Setup on Windows ........................................ 13
   3.2.1 Installation on Windows ........................................ 13
   3.2.2 Licensing on Windows ........................................ 15
3.3 Setup on Linux ........................................ 19
   3.3.1 Installation on Linux ........................................ 19
   3.3.2 Licensing on Linux ........................................ 22
   3.3.3 Notes about Environment ........................................ 23
3.4 Setup on macOS ........................................ 26
   3.4.1 Installation on macOS ........................................ 26
   3.4.2 Licensing on macOS ........................................ 28
   3.4.3 Notes about Environment ........................................ 29
3.5 Additional Setup Notes ........................................ 31

4 StyleVision Server Command Line .......... 32
4.1 assignlicense (Windows only) ........................................ 34
4.2 exportresourcestrings ........................................ 36
4.3 generate ........................................ 38
4.4 help ........................................ 41
4.5 licenseserver ........................................ 42
5 StyleVision Server API

5.1 About the .NET Interface.................................................................52
5.2 About the COM Interface..............................................................53
5.3 About the Java Interface...............................................................54
5.4 Code Examples.............................................................................55
  5.4.1 C#..........................................................................................55
  5.4.2 C++.......................................................................................56
  5.4.3 Java......................................................................................58
  5.4.4 VBScript................................................................................59
  5.4.5 Visual Basic..........................................................................61
5.5 API Reference................................................................................63
  5.5.1 COM and .NET......................................................................63
  5.5.2 Java......................................................................................70

6 XBRL Taxonomy Manager

6.1 Run XBRL Taxonomy Manager.....................................................78
6.2 Install a Taxonomy.........................................................................79
6.3 View Installed Taxonomies............................................................82
6.4 Uninstall a Taxonomy....................................................................83
6.5 Command Line Interface.............................................................84
  6.5.1 help........................................................................................85
  6.5.2 info........................................................................................86
  6.5.3 initialize.................................................................................86
  6.5.4 install....................................................................................87
  6.5.5 list..........................................................................................88
  6.5.6 reset......................................................................................89
  6.5.7 setdeflang............................................................................90
Index

6.5.8 uninstall..................................................................................................................... 90
6.5.9 update ......................................................................................................................... 91
6.5.10 upgrade..................................................................................................................... 92
1 Introduction

Altova StyleVision Server is an implementation of Altova StyleVision's built-in execution engine. It operates as a module of Altova's FlowForce Server, and it is also available as a standalone server product.

StyleVision Server executes transformation packages that have been deployed to a FlowForce Server. These transformations are initiated by FlowForce Server based on a variety of programmable time triggers, file triggers, or remote triggers. Additionally, StyleVision Server functionality can be invoked via the command line.

System requirements, installation and licensing

StyleVision Server is supported on the following operating systems:

- **Windows**
  - Windows 7 SP1 with Platform Update, Windows 8, Windows 10

- **Windows Server**
  - Windows Server 2008 R2 SP1 with Platform Update or newer

- **Linux**
  - CentOS 7 or newer
  - RedHat 7 or newer
  - Debian 8 or newer
  - Ubuntu 16.04 LTS or newer

The following libraries are required as a prerequisite to install and run the application. If the packages below are not already available on your Linux machine, run the command `yum` (or `apt-get` if applicable) to install them.

<table>
<thead>
<tr>
<th>Required by</th>
<th>CentOS, RedHat</th>
<th>Debian</th>
<th>Ubuntu</th>
</tr>
</thead>
<tbody>
<tr>
<td>LicenseServer</td>
<td>krb5-libs</td>
<td>libgssapi-krb5-2</td>
<td>libgssapi-krb5-2</td>
</tr>
<tr>
<td>StyleVision Server</td>
<td>qt5-qtbase-gui, krb5-libs</td>
<td>libqt5gui5, libgssapi-krb5-2</td>
<td>libqt5gui5, libgssapi-krb5-2</td>
</tr>
</tbody>
</table>

**Note:** If you plan to use Altova's Charts functionality, then at least one font must be installed on your system to ensure that charts will be rendered correctly. To list installed fonts, use, for example, the `fc-list` command of the Fontconfig library.

- **macOS**
  - macOS 10.13 or newer
StyleVision Server is available for both 32-bit and 64-bit on Windows machines. For details about installation and licensing, see the setup sections for Windows, Linux, and macOS.

**Note:** If the fillable parts of a fillable PDF are missing when the PDF is opened on a macOS system, one likely cause is that Java 6 is not installed on the machine. If this is the case, you can install Java 6 from [https://support.apple.com/kb/dl1572?locale=en_US](https://support.apple.com/kb/dl1572?locale=en_US). If a version newer than Java 6 has already been installed on your system, then the installation of the older Java 6 version will not affect the working of the newer version, which will be the default version of the system.

*Last updated: 9 October 2020*
2 Functionality

StyleVision Server transforms XML files into output HTML, PDF, RTF, and DOCX documents with the use of XSLT stylesheets. These XSLT stylesheets are obtained from PXF files that have been created in Altova’s stylesheet designer application, Altova StyleVision.

StyleVision Server can be used in two ways:

- As part of the Altova FlowForce workflow. For more information about Altova FlowForce, visit the Altova website.
- As a standalone server product that is accessed via its command line interface (CLI).

An XML input file and a PXF file are passed to StyleVision Server, which produces the required output document/s.
2.1 In the FlowForce Workflow

A FlowForce job is created in Altova FlowForce Server. The FlowForce job specifies: (i) the inputs and outputs of a StyleVision Server transformation; and (ii) the triggers for when the job is to be executed, such as a specific time every day. At execution time, Altova FlowForce Server passes the transformation instructions to StyleVision Server, which then carries out the transformation.

The role of StyleVision Server in the FlowForce workflow is shown in the diagram below. (The role of MapForce Server in the workflow is also displayed since FlowForce jobs can be created that send Altova MapForce mappings to the Altova MapForce Server for execution.)

Additionally to being invoked by a FlowForce job, StyleVision Server can also be invoked via the command line. Usage is described in the section StyleVision Server Command Line.
2.2 As a Standalone Server

StyleVision Server can be installed as a standalone product on Windows, Linux, and macOS systems. In this version its functionality is invoked only via the command line. Usage is described in the section StyleVision Server Command Line.
3 StyleVision Server Setup

This section describes procedures for setting up StyleVision Server. It describes the following:

- Information about processor cores and licenses
- Installation and licensing of StyleVision Server on Windows systems
- Installation and licensing of StyleVision Server on Linux systems
- Installation and licensing of StyleVision Server on macOS systems
3.1 Processor Cores and Licenses

The licensing of Altova server products is based on the number of physical processor cores available on the product machine (as opposed to the number of logical cores). For example, a dual-core processor has two cores, a quad-core processor four cores, a hexa-core processor six cores, and so on. The number of cores licensed for a product must be greater than or equal to the number of cores available on that server machine, whether the server is a physical or virtual machine. For example, if a server has eight cores, you must purchase an 8-core license. You can also combine licenses to achieve the core count. So, two 4-core licenses can be used for an eight-core server instead of one 8-core license.

If you are using a computer server with a large number of CPU cores but only have a low volume to process, you may also create a virtual machine that is allocated a smaller number of cores, and purchase a license for that number. Such a deployment, of course, will have less processing speed than if all available cores of the computer were utilized.

**Note:** Each Altova server product license can be used for only one client machine at a time—the machine on which the Altova server product is installed—even if the license has unused licensing capacity. For example, if a 10-core license is used for a client machine that has 6 CPU cores, then the remaining 4 cores of licensing capacity cannot be used simultaneously for another client machine.

**Single-thread execution**

If an Altova server product allows single-thread execution, an option for Single-thread execution will be available. In these cases, if an Altova server-product license for only one core is available in the license pool, a machine with multiple cores can be assigned this one-core license. In such a case, the machine will run that product on a single core. Processing will therefore be slower, because multi-threading (which is possible on multiple cores) will not be available. The product will be executed in single thread mode on that machine.

To assign a single-core license to a multiple-core machine, in LicenseServer, select the Limit to single thread execution check box for that product.

**Estimate of core requirements**

There are various external factors that influence the data volumes and processing times your server can handle (for example: the hardware, the current load on the CPU, and memory allocation of other applications running on the server). In order to measure performance as accurately as possible, test the applications in your environment with data volumes and in conditions that approximate as closely as possible to real business situations.
3.2 Setup on Windows

This section describes the installation and licensing of StyleVision Server on Windows systems.

**Installation on Windows**
- System requirements
- Installing StyleVision Server
- Altova LicenseServer
- LicenseServer versions
- Trial license
- Application folder location

**Licensing on Windows**
- Start ServiceController
- Start LicenseServer
- Register StyleVision Server
- Assign a license

**Note:** From version 2021 onwards, a 32-bit version of StyleVision Server cannot be installed over a 64-bit version, or a 64-bit version over a 32-bit version. You must either (i) remove the older version before installing the newer version, or (ii) upgrade to a newer version that is the same bit version as your older installation.

3.2.1 Installation on Windows

StyleVision Server is available for installation on Windows systems. Its installation and setup procedure is described below.

**System requirements**
- Windows
  - Windows 7 SP1 with Platform Update, Windows 8, Windows 10
- Windows Server
  - Windows Server 2008 R2 SP1 with Platform Update or newer

**Installing StyleVision Server**

StyleVision Server can be installed on Windows systems as follows:
- As a separate standalone server product called StyleVision Server. To install StyleVision Server, download and run the StyleVision Server installer. Follow the on-screen instructions.
- As part of the FlowForce Server installation package. To install StyleVision Server as part of the FlowForce Server package, download and run the FlowForce Server installer. Follow the on-screen instructions and make sure you check the option for installing StyleVision Server.
The installers of both StyleVision Server and FlowForce Server are available at the Altova Download Center (http://www.altova.com/download.html).

After installation, the StyleVision Server executable will be located by default at:

```
<ProgramFilesFolder>\Altova\StyleVisionServer2021\bin\StyleVisionServer.exe
```

All the necessary registrations to use StyleVision Server via a COM interface, as a Java interface, and in the .NET environment will be done by the installer.

**Altova LicenseServer**

- In order for StyleVision Server to work, it must be licensed via an Altova LicenseServer on your network.
- When you install StyleVision Server or FlowForce Server on Windows systems, an option is available that allows you to download and install Altova LicenseServer together with StyleVision Server or FlowForce Server.
- If an Altova LicenseServer is already installed on your network, you do not need to install another one—unless a newer version of Altova LicenseServer is required. (See next point, LicenseServer versions.)
- During the installation process of StyleVision Server or FlowForce Server, check or uncheck the option for installing Altova LicenseServer as appropriate. Note the following points:
  a. If you haven't installed Altova LicenseServer yet, leave the default settings as is. The wizard will install the latest version of Altova LicenseServer on the computer where you are running the wizard.
  b. If you haven't installed Altova LicenseServer yet and want to install Altova LicenseServer on another computer, clear the Install Altova LicenseServer on this machine check box, and then choose Register Later. In this case, you will need to install Altova LicenseServer and register StyleVision Server separately.
  c. If Altova LicenseServer has already been installed on your computer but it has a lower version than the one indicated by the installation wizard, leave the default settings as is. In this case, the installation wizard will automatically upgrade your LicenseServer version with the one indicated on the dialog box. Note that the existing registration and licensing information will be preserved after the upgrade.
  d. If Altova LicenseServer has already been installed on your computer or network, and if it has the same version as the one indicated by the wizard, do the following:
    i. Clear the Install Altova LicenseServer on this machine check box.
    ii. Under Register this product with, choose the Altova LicenseServer instance on which you want to register StyleVision Server, or choose Register later. Note that you can always select Register Later if you want to ignore the LicenseServer associations and carry on with the installation of StyleVision Server.

See the section, Licensing on Windows, for more information about how to register and license StyleVision Server with Altova LicenseServer.

**LicenseServer versions**

- Altova server products must be licensed either with the version of LicenseServer that is appropriate to the installed StyleVision Server version, or with a later version of LicenseServer.
- The LicenseServer version that is appropriate for a particular version of StyleVision Server is
displayed during the installation of StyleVision Server. You can install this version of LicenseServer along with StyleVision Server, or you can install LicenseServer separately.

- Before installing a newer version of LicenseServer, any older one must be de-installed. The LicenseServer installer will do this automatically if it detects an older version.
- LicenseServer versions are backwards compatible. They will work with older versions of StyleVision Server.
- If you install a new version of StyleVision Server and if your installed LicenseServer version is older than the appropriate LicenseServer, install the latest version available on the Altova website.
- At the time of LicenseServer de-installation, all registration and licensing information held in the older version of LicenseServer will be saved to a database on your server machine. This data will be imported automatically into the newer version when the newer version is installed.
- The version number of the currently installed LicenseServer is given at the bottom of the LicenseServer configuration page (all tabs).

**Current version: 3.5**

▼ Trial license
During the installation process, you will be given the option of requesting a 30-day trial license for StyleVision Server. After submitting the request, a trial license will be sent to the email address you registered.

▼ Application folder location
The application will be installed in the following folder:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows 7, 8, 10</td>
<td>C:\Program Files\Altova\</td>
</tr>
<tr>
<td>32 bit Version on 64-bit OS</td>
<td>C:\Program Files (x86)\Altova\</td>
</tr>
</tbody>
</table>

### 3.2.2 Licensing on Windows

StyleVision Server must be licensed with Altova LicenseServer. Licensing is a two-step process:

1. **Register StyleVision Server** with LicenseServer. Registration is done from StyleVision Server.
2. **Assign a license** to StyleVision Server from LicenseServer. Download the latest version of LicenseServer from the [Altova website](https://www.altova.com), and install it on your local machine or a machine on your network.

The steps to carry out are given below in brief. For detailed information, see the [LicenseServer user manual](https://www.altova.com) at the [Altova website](https://www.altova.com).

▼ Start ServiceController
Altova ServiceController is started in order to start Altova LicenseServer.

Altova ServiceController (ServiceController for short) is an application for conveniently starting, stopping
and configuring Altova services on Windows systems.

ServiceController is installed with Altova LicenseServer and with Altova server products that are installed as services (FlowForce Server, RaptorXML(+XBRL) Server, and Mobile Together Server). It can be started by clicking Start | Altova LicenseServer | Altova ServiceController. (This command is also available in the Start menu folders of Altova server products that are installed as services (FlowForce Server, RaptorXML(+XBRL) Server, and Mobile Together Server).) After ServiceController has been started, it can be accessed via the system tray (screenshot below).

To specify that ServiceController starts automatically on logging in to the system, click the ServiceController icon in the system tray to display the ServiceController menu (screenshot below), and then toggle on the command Run Altova ServiceController at Startup. (This command is toggled on by default.) To exit ServiceController, click the ServiceController icon in the system tray and, in the menu that appears (see screenshot below), click Exit Altova ServiceController.

Start LicenseServer

To start LicenseServer, click the ServiceController icon in the system tray, hover over Altova LicenseServer in the menu that pops up (see screenshot below), and then select Start Service from the LicenseServer submenu. If LicenseServer is already running, the Start Service option will be disabled.
Register StyleVision Server

To register StyleVision Server from the command line interface, use the `licenseserver` command:

```
StyleVisionServer licenseserver [options] ServerName-Or-IP-Address
```

For example, if `localhost` is the name of the server on which LicenseServer is installed:

```
StyleVisionServer licenseserver localhost
```

If StyleVision Server was installed as part of a FlowForce Server installation, registering FlowForce Server with LicenseServer will automatically also register StyleVision Server. Essentially: (i) Start Altova FlowForce Web as a service via ServiceController (see previous point); (ii) Enter your password to access the Setup page; (iii) Select the LicenseServer name or address and click Register with LicenseServer. For more information, see Register FlowForce Server.

After successful registration, go to the Client Management tab of LicenseServer's configuration page to assign a license to StyleVision Server.

Assign a license

After successfully registering StyleVision Server, it will be listed in the Client Management tab of the configuration page of LicenseServer. Go there and assign a license to StyleVision Server.

The licensing of Altova server products is based on the number of processor cores available on the product machine. For example, a dual-core processor has two cores, a quad-core processor four cores, a hexa-core processor six cores, and so on. The number of cores licensed for a product must be greater than or equal to the number of cores available on that server machine, whether the server is a physical or virtual machine. For example, if a server has eight cores (an octa-core processor), you must purchase at least one 8-core license. You can also combine licenses to achieve the core count. So, two 4-core licenses can also be used for an octa-core server instead of one 8-core license.

If you are using a computer server with a large number of CPU cores but only have a low volume to process, you may also create a virtual machine that is allocated a smaller number of cores, and purchase a license for that number. Such a deployment, of course, will have less processing speed than if all available cores on the server were utilized.
Note: Each Altova server product license can be used for only one client machine—the machine on which the Altova server product is installed—at a time, even if the license has unused licensing capacity. For example, if a 10-core license is used for a client machine that has 6 CPU cores, then the remaining 4 cores of licensing capacity cannot be used simultaneously for another client machine.
3.3 Setup on Linux

This section describes the installation and licensing of StyleVision Server on Linux systems (Debian, Ubuntu, CentOS, RedHat).

**Installation on Linux**
- System requirements
- Uninstall old versions of Altova server products
- Download the Linux package
- Install StyleVision Server
- Altova LicenseServer
- LicenseServer versions

**Licensing on Linux**
- Start LicenseServer
- Register StyleVision Server
- Assign a license

**Notes about Environment**

### 3.3.1 Installation on Linux

StyleVision Server is available for installation on Linux systems. Its installation and setup procedure is described below.

▲ System requirements

#### Linux
- CentOS 7 or newer
- RedHat 7 or newer
- Debian 8 or newer
- Ubuntu 16.04 LTS or newer

The following libraries are required as a prerequisite to install and run the application. If the packages below are not already available on your Linux machine, run the command `yum` (or `apt-get` if applicable) to install them.

<table>
<thead>
<tr>
<th>Required by</th>
<th>CentOS, RedHat</th>
<th>Debian</th>
<th>Ubuntu</th>
</tr>
</thead>
<tbody>
<tr>
<td>LicenseServer</td>
<td>krb5-libs</td>
<td>libgssapi-krb5-2</td>
<td>libgssapi-krb5-2</td>
</tr>
<tr>
<td>StyleVision Server</td>
<td>qt5-qtxbase-gui, krb5-libs</td>
<td>libqt5gui5, libgssapi-krb5-2</td>
<td>libqt5gui5, libgssapi-krb5-2</td>
</tr>
</tbody>
</table>

**Note:** If you plan to use Altova's Charts functionality, then at least one font must be installed on your system to ensure that charts will be rendered correctly. To list installed fonts, use, for example, the `fc-list` command of the Fontconfig library.
FlowForce Server integration

If you are installing StyleVision Server together with FlowForce Server, it is recommended that you install FlowForce Server first. Otherwise, after having installed both StyleVision Server and FlowForce Server, run the following command:

```
cp /opt/Altova/StyleVisionServer2021/etc/*.tool /opt/Altova/FlowForceServer2021/tools
```

This command copies the `.tool` file from `/etc` directory of StyleVision Server to the FlowForce Server `/tools` directory. The `.tool` file is required by FlowForce Server; it contains the path to the StyleVision Server executable. You do not need to run this command if you install FlowForce Server before installing StyleVision Server.

Uninstall old versions of Altova server products

If you need to uninstall a previous version, do this as follows. On the Linux command line interface (CLI), you can check which Altova server products are installed with the following command:

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

If StyleVision Server is not installed, go ahead with the installation as documented below in Installing StyleVision Server.

If StyleVision Server is installed and you wish to install a newer version of StyleVision Server, uninstall the old version with the command:

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

If you need to uninstall an old version of Altova LicenseServer, do this with the following command:

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

Download the Linux package

StyleVision Server installation packages for the following Linux systems are available at the Altova website.

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Package extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debian</td>
<td>.deb</td>
</tr>
<tr>
<td>Ubuntu</td>
<td>.deb</td>
</tr>
<tr>
<td>CentOS</td>
<td>.rpm</td>
</tr>
<tr>
<td>RedHat</td>
<td>.rpm</td>
</tr>
</tbody>
</table>
After downloading the Linux package, copy it to any directory on the Linux system. Since you will need an Altova LicenseServer in order to run StyleVision Server, you may want to download LicenseServer from the Altova website at the same time as you download StyleVision Server, rather than download it at a later time.

Install StyleVision Server

In a terminal window, switch to the directory where you have copied the Linux package. For example, if you copied it to a user directory called MyAltova (that is located, say, in the /home/User directory), then switch to this directory as follows:

```
cd /home/User/MyAltova
```

Install StyleVision Server with the following command:

- **Debian**: `sudo dpkg --install stylevisionserver-2021-debian.deb`
- **Ubuntu**: `sudo dpkg --install stylevisionserver-2021-ubuntu.deb`
- **CentOS**: `sudo rpm -ivh stylevisionserver-2021-1.x86_64.rpm`
- **RedHat**: `sudo rpm -ivh stylevisionserver-2021-1.x86_64.rpm`

The StyleVision Server package will be installed in the folder:

```
/opt/Altova/StyleVisionServer2021
```

Altova LicenseServer

In order for any Altova Server product—including StyleVision Server—to run, that server product must be licensed via an Altova LicenseServer on your network.

On Linux systems, Altova LicenseServer will need to be installed separately. Download LicenseServer from the Altova website and copy the package to any directory on the Linux system. Install it just like you installed StyleVision Server (see previous step).

- **Debian**: `sudo dpkg --install licenseserver-3.5-debian.deb`
- **Ubuntu**: `sudo dpkg --install licenseserver-3.5-ubuntu.deb`
- **CentOS**: `sudo rpm -ivh licenseserver-3.5-1.x86_64.rpm`
- **RedHat**: `sudo rpm -ivh licenseserver-3.5-1.x86_64.rpm`

The LicenseServer package will be installed in:

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

For information about how to register StyleVision Server with Altova LicenseServer and license it, see the section, Licensing on Linux. Also see the LicenseServer documentation for more detailed information.

LicenseServer versions

- Altova server products must be licensed either with the version of LicenseServer that is appropriate to the installed StyleVision Server version, or with a later version of LicenseServer.
- The LicenseServer version that is appropriate for a particular version of StyleVision Server is displayed during the installation of StyleVision Server. You can install this version of LicenseServer along with StyleVision Server, or you can install LicenseServer separately.
- Before installing a newer version of LicenseServer, any older one must be de-installed. The LicenseServer installer will do this automatically if it detects an older version.
• LicenseServer versions are backwards compatible. They will work with older versions of StyleVision Server.
• If you install a new version of StyleVision Server and if your installed LicenseServer version is older than the appropriate LicenseServer, install the latest version available on the Altova website.
• At the time of LicenseServer de-installation, all registration and licensing information held in the older version of LicenseServer will be saved to a database on your server machine. This data will be imported automatically into the newer version when the newer version is installed.
• The version number of the currently installed LicenseServer is given at the bottom of the LicenseServer configuration page (all tabs).

Current version: 3.5

3.3.2 Licensing on Linux

StyleVision Server must be licensed with Altova LicenseServer. Licensing is a two-step process:

1. Register StyleVision Server with LicenseServer. Registration is done from StyleVision Server.
2. Assign a license to StyleVision Server from LicenseServer. Download the latest version of LicenseServer from the Altova website, and install it on your local machine or a machine on your network.

The steps to carry out are given below in brief. For detailed information, see the LicenseServer user manual at the Altova website.

Start LicenseServer
To correctly register and license StyleVision Server with LicenseServer, LicenseServer must be running as a daemon on the network. Start LicenseServer as a daemon with the following command:

```
[< Debian 8]  sudo /etc/init.d/licenseserver start
[≥ Debian 8]  sudo systemctl start licenseserver
[< CentOS 7]  sudo initctl start licenseserver
[≥ CentOS 7]  sudo systemctl start licenseserver
[< Ubuntu 15] sudo initctl start licenseserver
[≥ Ubuntu 15] sudo systemctl start licenseserver
[RedHat]      sudo initctl start licenseserver
```

If at any time you need to stop LicenseServer, replace start with stop in the above commands. For example:

```
sudo /etc/init.d/licenseserver stop
```

Register StyleVision Server
To register StyleVision Server from the command line interface, use the licenseserver command:
sudo /opt/Altova/StyleVisionServer2021/bin/stylevisionserver licenseserver [options] ServerName-Or-IP-Address

For example, if localhost is the name of the server on which LicenseServer is installed:

sudo /opt/Altova/StyleVisionServer2021/bin/stylevisionserver licenseserver localhost

In the command above, localhost is the name of the server on which LicenseServer is installed. Notice also that the location of the StyleVision Server executable is:

/opt/Altova/StyleVisionServer2021/bin/

After successful registration, go to the Client Management tab of LicenseServer's configuration page to assign a license to StyleVision Server.

Assign a license

After successfully registering StyleVision Server, it will be listed in the Client Management tab of the configuration page of LicenseServer. Go there and assign a license to StyleVision Server.

The licensing of Altova server products is based on the number of processor cores available on the product machine. For example, a dual-core processor has two cores, a quad-core processor four cores, a hexa-core processor six cores, and so on. The number of cores licensed for a product must be greater than or equal to the number of cores available on that server machine, whether the server is a physical or virtual machine. For example, if a server has eight cores (an octa-core processor), you must purchase at least one 8-core license. You can also combine licenses to achieve the core count. So, two 4-core licenses can also be used for an octa-core server instead of one 8-core license.

If you are using a computer server with a large number of CPU cores but only have a low volume to process, you may also create a virtual machine that is allocated a smaller number of cores, and purchase a license for that number. Such a deployment, of course, will have less processing speed than if all available cores on the server were utilized.

Note: Each Altova server product license can be used for only one client machine—the machine on which the Altova server product is installed—at a time, even if the license has unused licensing capacity. For example, if a 10-core license is used for a client machine that has 6 CPU cores, then the remaining 4 cores of licensing capacity cannot be used simultaneously for another client machine.

### 3.3.3 Notes about Environment

**Folders**

Given below is a list of important folders in your StyleVision Server setup.

- **Installation root**
  
  /opt/Altova/StyleVisionServer2021/

- **License Files**
Environment settings

/etc/profile.d/jdbc.sh

The environment settings file (typically named `jdbc.sh`) is executed at system start. The definitions in it must be specific to your particular environment. The example path above serves only as a general guide.

**Note:** The environment settings file sets the variables for all users on the system, so you must be careful when modifying settings. For example, if you modify a class path in this file, then the modifications will be applied across the system. If you wish to make changes for StyleVision Server only, you might want to consider using a unit file (explained in the section **JDBC Connections** below).

Database connections

On Linux, the following database connections are supported:

- **JDBC** — You can use JDBC for all supported databases except Microsoft Access
- **Native connections** — Currently available for SQLite and PostgreSQL databases

If you are using JDBC, note the following points:

- The Java Runtime Environment or SDK must be installed.
- The JDBC drivers for the target database must be installed.
- The following environment variables must be set correctly for your environment:
  - **CLASSPATH**: to find the jar-files that connect to the JDBC database; the jar-files can be entered either in (i) an executable script (like `jdbc.sh`) that is executed on system start, or (ii) a unit file that is executed when StyleVision Server is started as a service. Using a unit file to specify the jar-files has the advantage that the files required for StyleVision Server's JDBC connections will be located without you having to modify the existing system configuration. A unit file is listed below.
  - **PATH**: to find the JRE, but might not be necessary depending on the installation
  - **JAVA_HOME**: if necessary, depending on the installation

Listing of important files

The following shell script (or unit file) is copied to the folder `/opt/Altova/StyleVisionServer/etc` so as not to overwrite already existing configuration files. Make the necessary changes as required. Also see the section **JDBC Connections** above. The parts highlighted in blue are environment-specific and will need to be adjusted to match your environment:

**Shell script (unit file)**

```bash
#!/bin/bash

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
3.4  Setup on macOS

This section describes the installation and licensing of StyleVision Server on macOS systems.

**Installation on macOS**
- System requirements
- Uninstall old versions of Altova server products
- Download the macOS package
- Install StyleVision Server
- Altova LicenseServer
- LicenseServer versions

**Licensing on macOS**
- Start LicenseServer
- Register StyleVision Server
- Assign a license

**Notes about Environment**

### 3.4.1 Installation on macOS

StyleVision Server is available for installation on macOS systems. Its installation and setup procedure is described below.

- **System requirements**
  - macOS 10.13 or newer

- **FlowForce Server integration**
  If you install StyleVision Server together with FlowForce Server, it is recommended that you install FlowForce Server first. If you install StyleVision Server before FlowForce Server, then, after having installed both StyleVision Server and FlowForce Server, run the following command:

  ```bash
  cp /usr/local/Altova/StyleVisionServer2021/etc/*.tool /usr/local/Altova/FlowForceServer2021/tools
  ```

  This command copies the `.tool` file from `/etc` directory of StyleVision Server to the FlowForce Server `/tools` directory. The `.tool` file is required by FlowForce Server; it contains the path to the StyleVision Server executable. You do not need to run this command if you install FlowForce Server before installing StyleVision Server.

- **Uninstall old versions of Altova server products**
In the Applications folder in Finder, right-click the StyleVision Server icon and select **Move to Trash**. The application will be moved to Trash. You will, however, still need to remove the application from the `usr` folder. Do this with the command:

```
sudo rm -rf /usr/local/Altova/StyleVisionServer2021/
```

If you need to uninstall an old version of Altova LicenseServer, you must first stop it running as a service. Do this with the following command:

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

To check whether the service has been stopped, open the Activity Monitor in Finder and make sure that LicenseServer is not in the list. Then proceed to uninstall in the same way as described above for StyleVision Server.

▼ Download the disk image file

Download the disk image (.dmg) file from the Altova website ([http://www.altova.com/download.html](http://www.altova.com/download.html)).

▼ Install StyleVision Server

Click to open the downloaded disk image (.dmg). This causes the StyleVision Server installer to appear as a new virtual drive on your computer. On the new virtual drive, double-click the installer package (.pkg). Go through the successive steps of the installer wizard. These are self-explanatory and include one step in which you have to agree to the license agreement before being able to proceed. To eject the drive after installation, right-click it and select **Eject**.

The StyleVision Server package will be installed in the folder:

- `/usr/local/Altova/StyleVisionServer2021` (application binaries)
- `/var/Altova/StyleVisionServer` (data files: database and logs)

▼ Altova LicenseServer

In order for any Altova Server product—including StyleVision Server—to run, that server product must be licensed via an Altova LicenseServer on your network.

The Altova LicenseServer installation package is available on the virtual drive you have mounted in the previous step. To install Altova LicenseServer, double-click the installer package included on the virtual drive and follow the on-screen instructions. You will need to accept the license agreement for installation to proceed.

Altova LicenseServer can also be downloaded and installed separately from the Altova website ([http://www.altova.com/download.html](http://www.altova.com/download.html)).

The LicenseServer package will be installed in the folder:

- `/usr/local/Altova/LicenseServer`

For information about how to register StyleVision Server with Altova LicenseServer and license it, see the section, **Licensing on macOS**. 

© 2020 Altova GmbH

Altova StyleVision Server 2021
LicenseServer versions

- Altova server products must be licensed either with the version of LicenseServer that is appropriate to the installed StyleVision Server version, or with a later version of LicenseServer.
- The LicenseServer version that is appropriate for a particular version of StyleVision Server is displayed during the installation of StyleVision Server. You can install this version of LicenseServer along with StyleVision Server, or you can install LicenseServer separately.
- Before installing a newer version of LicenseServer, any older one must be de-installed. The LicenseServer installer will do this automatically if it detects an older version.
- LicenseServer versions are backwards compatible. They will work with older versions of StyleVision Server.
- If you install a new version of StyleVision Server and if your installed LicenseServer version is older than the appropriate LicenseServer, install the latest version available on the Altova website.
- At the time of LicenseServer de-installation, all registration and licensing information held in the older version of LicenseServer will be saved to a database on your server machine. This data will be imported automatically into the newer version when the newer version is installed.
- The version number of the currently installed LicenseServer is given at the bottom of the LicenseServer configuration page (all tabs).

Current version: 3.5

### 3.4.2 Licensing on macOS

StyleVision Server must be licensed with Altova LicenseServer. Licensing is a two-step process:

1. **Register StyleVision Server** with LicenseServer. Registration is done from StyleVision Server.
2. **Assign a license** to StyleVision Server from LicenseServer. Download the latest version of LicenseServer from the Altova website, and install it on your local machine or a machine on your network.

The steps to carry out are given below in brief. For detailed information, see the LicenseServer user manual at the Altova website.

#### Start LicenseServer

To correctly register and license StyleVision Server with LicenseServer, LicenseServer must be running as a daemon. Start LicenseServer as a daemon with the following command:

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

If at any time you need to stop LicenseServer, replace `load` with `unload` in the above command:

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

#### Register StyleVision Server

To register StyleVision Server from the command line interface, use the `license` command:

```
sudo /usr/local/Altova/StyleVisionServer2021/bin/StyleVisionServer license [options] ServerName-Or-IP-Address
```
For example, if `localhost` is the name of the server on which LicenseServer is installed:

```bash
sudo /usr/local/Altova/StyleVisionServer2021/bin/StyleVisionServer licenseserver localhost
```

In the command above, `localhost` is the name of the server on which LicenseServer is installed. Notice also that the location of the StyleVision Server executable is:

```bash
/usr/local/Altova/StyleVisionServer2021/bin/
```

After successful registration, go to the Client Management tab of LicenseServer's configuration page to assign a license to StyleVision Server.

**Assign a license**

After successfully registering StyleVision Server, it will be listed in the Client Management tab of the configuration page of LicenseServer. Go there and assign a license to StyleVision Server.

The licensing of Altova server products is based on the number of processor cores available on the product machine. For example, a dual-core processor has two cores, a quad-core processor four cores, a hexa-core processor six cores, and so on. The number of cores licensed for a product must be greater than or equal to the number of cores available on that server machine, whether the server is a physical or virtual machine. For example, if a server has eight cores (an octa-core processor), you must purchase at least one 8-core license. You can also combine licenses to achieve the core count. So, two 4-core licenses can also be used for an octa-core server instead of one 8-core license.

If you are using a computer server with a large number of CPU cores but only have a low volume to process, you may also create a virtual machine that is allocated a smaller number of cores, and purchase a license for that number. Such a deployment, of course, will have less processing speed than if all available cores on the server were utilized.

**Note:** Each Altova server product license can be used for only one client machine—the machine on which the Altova server product is installed—at a time, even if the license has unused licensing capacity. For example, if a 10-core license is used for a client machine that has 6 CPU cores, then the remaining 4 cores of licensing capacity cannot be used simultaneously for another client machine.

### 3.4.3 Notes about Environment

**Folders**

Given below is a list of important folders in your StyleVision Server setup.

- **Installation root**
  ```bash
  /usr/local/Altova/StyleVisionServer2021/
  ```

- **License Files**
  ```bash
  /var/Altova/StyleVisionServer```
Environment settings

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

The environment settings file must be defined according to your specific environment. The example path above serves only as a general guide.

Note: These environment variables are only set for the StyleVision Server process and do not have an impact on other users.

Database connections

On MacOS, the following database connections are supported:

- JDBC — You can use JDBC for all supported databases except Microsoft Access
- Native connections — Currently available for SQLite and PostgreSQL databases

If you are using JDBC, note the following points:

- The Java Runtime Environment or SDK must be installed.
- The JDBC-Connects for the target database must be installed.
- The following environment variables must be set correctly for your environment:
  - **CLASSPATH**: to find the jar-files; the class path is set in the Plist file.
  - **PATH**: to find the JRE, but might not be necessary depending on the installation
  - **JAVA_HOME**: if necessary, depending on the installation

Java 6 for fillable PDF forms

If the fillable parts of a fillable PDF are missing when the PDF is opened on a Mac OS system, one likely cause is that Java 6 is not installed on the machine. If this is the case, you can install Java 6 from https://support.apple.com/kb/dl1572?locale=en_US. If a version newer than Java 6 has already been installed, then the installation of the older Java 6 version will not affect the working of the newer version, which will be the default version of the system.
3.5 Additional Setup Notes

In order to run the Windows examples that are packaged with StyleVision Server in the etc\examples sub-folder of the application folder, the StyleVision Server DLL must be correctly registered with the system. A registration error typically occurs if you have, over time, installed different bit-versions of StyleVision Server (32-bit and 64-bit) on a single machine.

To correctly register the StyleVision Server DLL (either the 32-bit or 64-bit version) on Windows machines, do the following:

1. Open a command prompt in administrator mode
2. Switch to the folder in which the DLL is located. The command to do this would be: `cd C:\Program Files\Altova\StyleVisionServer2021\bin`
3. Run the following command to register the DLL (either 32-bit or 64-bit): `regsvr32 StyleVisionServer.dll`
4. Ensure that you get a popup saying that the registration succeeded
5. Open Visual Studio
6. Load the project using `StyleVisionServerAPI_Sample.sln`
7. Confirm that your `Program.cs` file contains valid pathways
8. Run the project by using `Ctrl+F5`

Note: The path to the application folder on Windows systems is typically: `C:\Program Files\Altova\StyleVisionServer2021`. 
4 StyleVision Server Command Line

Default location of StyleVision Server executable
Given below are the default locations of the StyleVision Server executable:

- **Linux**: `/opt/Altova/StyleVisionServer2021/bin/stylevisionserver`
- **Mac**: `/usr/local/Altova/StyleVisionServer2021/bin/stylevisionserver`
- **Windows**: `<ProgramFilesFolder>\Altova\StyleVisionServer2021\bin\StyleVisionServer.exe`

Usage and list of CLI commands
The command line syntax is:

```
stylevisionserver --h | --help | --version | <command> [options] [arguments]
```

- `--help` (short form `--h`) displays the help text of the given command. If no command is named, then all commands of the executable are listed, each with a brief description of the command.
- `--version` displays the version number of StyleVision Server.
- `<command>` is the command to execute. Commands are described in the sub-sections of this section (see list below).
- `[options]` are the options of a command; they are listed and described with their respective commands.
- `[arguments]` are the arguments of a command; they are listed and described with their respective commands.

▼ Casing and slashes on the command line
- `StyleVisionServer` on Windows
- `stylevisionserver` on Windows and Unix (Linux, Mac)

* Note that lowercase (`stylevisionserver`) works on all platforms (Windows, Linux, and Mac), while upper-lower (`StyleVisionServer`) works only on Windows and Mac.
* Use forward slashes on Linux and Mac, backslashes on Windows.

CLI commands
Available commands are listed below and are explained in the sub-sections of this section.

- `assignlicense` : Uploads a license to LicenseServer and assigns this license to StyleVision Server.
- `exportresourcestrings` : Exports all application resource strings to an XML file.
- `generate` : Generates one or several documents from an input XML file and an XSLT stylesheet in the input PXF file.
- `help` : Displays information about the command that is submitted in the argument (or about all commands if no argument is submitted).
- `licenseserver` : Registers StyleVision Server with a LicenseServer on the local network.
- `pdfdata` : Generates form data from a PDF file to an FDF or XML file.
StyleVision Server Command Line

- **setdeflang**: Sets the default language of StyleVision Server.
- **setfopath**: Selects an alternative FO processor for subsequent PDF generation.
- **verifylicense**: Checks if current StyleVision Server is licensed and, optionally, whether it is licensed with the given license key.
- **version**: Displays the version number of StyleVision Server.
4.1 assignlicense (Windows only)

Syntax and description
The assignlicense command is available on Windows only (not on Linux or Mac systems). It uploads a license file to the Altova LicenseServer with which StyleVision Server is registered (see the licenseserver command), and assigns the license to StyleVision Server. It takes the path of a license file as its argument. The command also allows you to test the validity of a license.

```
stylevisionserver assignlicense [options] FILE
```

- The FILE argument takes the path of the license file.
- The --test-only option uploads the license file to LicenseServer and validates the license, but does not assign the license to StyleVision Server.

For details about licensing, see the LicenseServer documentation (https://www.altova.com/manual/AltovaLicenseServer/).

Casing and slashes on the command line
StyleVisionServer on Windows
stylevisionserver on Windows and Unix (Linux, Mac)

* Note that lowercase (stylevisionserver) works on all platforms (Windows, Linux, and Mac), while upper-lower (StyleVisionServer) works only on Windows and Mac.
* Use forward slashes on Linux and Mac, backslashes on Windows.

Backslashes, spaces, and special characters on Windows systems
On Windows systems: When spaces or special characters occur in strings (for example in file or folder names, or company, person or product names), use quotes: for example, "My File". Note, however, that a backslash followed by a double-quotation mark (for example, "C:\My directory") might not be read correctly. This is because the backslash character is also used to indicate the start of an escape sequence, and the escape sequence \" stands for the double-quotation mark character. If you want to escape this sequence of characters, use a preceding backslash, like this: \\". To summarize: If you need to write a file path that contains spaces or an end backslash, write it like this: "C:\My Directory\".

Examples
Examples of the assignlicense command:

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

- The first command above uploads the specified license to LicenseServer and assigns it to StyleVision Server.
- The last command uploads the specified license to LicenseServer and validates it, without assigning it to StyleVision Server.
Options
Options are listed in short form (if available) and long form. You can use one or two dashes for both short and long forms. An option may or may not take a value. If it takes a value, it is written like this: \texttt{--option=value}. Values can be specified without quotes except in two cases: (i) when the value string contains spaces, or (ii) when explicitly stated in the description of the option that quotes are required. If an option takes a Boolean value and no value is specified, then the option's default value is \texttt{TRUE}. Use the \texttt{--h, --help} option to display information about the command.

\texttt{test-only \,[t]}  
\hspace{1em} \texttt{--t, --test-only = true|false}  
Values are \texttt{true|false}. If \texttt{true}, then the license file is uploaded to LicenseServer and validated, but not assigned.
4.2 exportresourcestrings

Syntax and description

The `exportresourcestrings` command outputs an XML file containing the resource strings of the StyleVision Server application in the specified language. Available export languages are English (en), German (de), Spanish (es), French (fr), and Japanese (ja).

```
stylevisionserver exportresourcestrings [options] LanguageCode XMLOutputFile
```

- The `LanguageCode` argument gives the language of the resource strings in the output XML file; this is the export language. Allowed export languages (with their language codes in parentheses) are: English (en), German (de), Spanish (es), French (fr), and Japanese (ja).
- The `XMLOutputFile` argument specifies the path and name of the output XML file.

How to create localizations is described below.

Casing and slashes on the command line

- **StyleVisionServer on Windows**
  - `stylevisionserver` on Windows and Unix (Linux, Mac)

  * Note that lowercase (`stylevisionserver`) works on all platforms (Windows, Linux, and Mac), while upper-lower (`StyleVisionServer`) works only on Windows and Mac.
  * Use forward slashes on Linux and Mac, backslashes on Windows.

Backslashes, spaces, and special characters on Windows systems

- On Windows systems: When spaces or special characters occur in strings (for example in file or folder names, or company, person or product names), use quotes: for example, "My File". Note, however, that a backslash followed by a double-quotation mark (for example, "C:\My directory") might not be read correctly. This is because the backslash character is also used to indicate the start of an escape sequence, and the escape sequence \" stands for the double-quotation mark character. If you want to escape this sequence of characters, use a preceding backslash, like this: \\". To summarize: If you need to write a file path that contains spaces or an end backslash, write it like this: "C:\My Directory\\".

Examples

Examples of the `exportresourcestrings` command:

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

- The command above creates a file called `Strings.xml` at `c:\` that contains the resource strings of StyleVision Server in German.

Creating localized versions of StyleVision Server

You can create a localized version of StyleVision Server for any language of your choice. Five localized versions (English, German, Spanish, French, and Japanese) are already available in the C:\Program Files
Create a localized version as follows:

1. Generate an XML file containing the resource strings by using the `exportresourcestrings` command (see command syntax above). The resource strings in this XML file will be one of the five supported languages: English (en), German (de), Spanish (es), French (fr), or Japanese (ja), according to the `LanguageCode` argument used with the command.

2. Translate the resource strings from one of the five supported languages into the target language. The resource strings are the contents of the `<string>` elements in the XML file. Do not translate variables in curly brackets, such as `{option}` or `{product}`.

3. Contact Altova Support to generate a localized StyleVision Server DLL file from your translated XML file.

4. After you receive your localized DLL file from Altova Support, save the DLL in the `C:\Program Files (x86)\Altova\StyleVisionServer2021\bin` folder. Your DLL file will have a name of the form `StyleVisionServer2021_lc.dll`. The `_lc` part of the name contains the language code. For example, in `StyleVisionServer2021_de.dll`, the `de` part is the language code for German (Deutsch).

5. Run the `setdeflang` command to set your localized DLL file as the StyleVision Server application to use. For the argument of the `setdeflang` command, use the language code that is part of the DLL name.

**Note:** Altova StyleVision Server is delivered with support for five languages: English, German, Spanish, French, and Japanese. So you do not need to create a localized version of these languages. To set any of these languages as the default language, use StyleVision Server's `setdeflang` command.
4.3 generate

Syntax and description
The `generate` command (short form `gen`) generates one or more output files (HTML, PDF, RTF, and/or DOCX) by transforming the input XML file using the XSLT document/s contained in the input PXF file.

```
stylevisionserver generate | gen --inputxml=Filename [additional options] InputPXF
```

- The `--inputxml` option is mandatory; it gives the path to the XML file.
- The `InputPXF` argument specifies the path to the PXF file which contains the XSLT document/s that will be used to generate the output document/s. PXF files are created with Altova's StyleVision application.
- Each output format is generated by specifying an option for that output (see Options list below). The value of each option is a path that specifies where the output is to be generated.

Note: StyleVision Server uses Apache FOP, the FO processor of the Apache Project, to generate PDF files from FO. Apache FOP is installed with StyleVision Server at the following location: On Windows systems, ProgramData\Altova\SharedBetweenVersions; on Linux and macOS systems, in a descendant folder of the StyleVisionServer2021 folder. Note that Apache FOP requires that Java Runtime Environment 1.6 or later be installed on the StyleVision Server machine. For 32-bit StyleVision Server, install the 32-bit Java; for 64-bit StyleVision Server, install the 64-bit Java.

Casing and slashes on the command line
- `StyleVisionServer` on Windows
- `stylevisionserver` on Windows and Unix (Linux, Mac)

  * Note that lowercase (`stylevisionserver`) works on all platforms (Windows, Linux, and Mac), while upper-lower (`StyleVisionServer`) works only on Windows and Mac.
  * Use forward slashes on Linux and Mac, backslashes on Windows.

Backslashes, spaces, and special characters on Windows systems
On Windows systems: When spaces or special characters occur in strings (for example in file or folder names, or company, person or product names), use quotes: for example, "My File". Note, however, that a backslash followed by a double-quotation mark (for example, "C:\My directory\") might not be read correctly. This is because the backslash character is also used to indicate the start of an escape sequence, and the escape sequence \" stands for the double-quotation mark character. If you want to escape this sequence of characters, use a preceding backslash, like this: \". To summarize: If you need to write a file path that contains spaces or an end backslash, write it like this: "C:\My Directory\".

Examples
Examples of the `generate` command:
```
stylevisionserver generate --inputxml=C:\MyFiles\ExpReport.xml --html=Test.html ExpReport.pxf
```
StyleVision Server Command Line

```plaintext
stylevisionserver generate --inputxml=ExternalXML.xml --html=Test.html Test.pxf
```

- The commands above contain the mandatory `--inputxml` option, the `InputPXF` argument (`Test.pxf`), and a minimum of one output-creation option (`--html` in all the examples above).
- The input XML file to use can be located inside the PXF file (see second and third examples above) or it can be an external XML file (located outside the PXF file; see first and fourth examples above).
- The `--inputxml` switch is ignored if the main schema source is DB or DB-XML, but it must be present for syntactical reasons, and you should use something like `--inputxml=database`.
- If the output-creation option `--html` takes a relative path, as in the examples above, then the output file's location will be relative to the folder in which the PXF file is.

### Options

#### `inputxml [xml]`

```plaintext
--xml, --inputxml = PathToXMLFile
```

This option is mandatory. It specifies the path to the XML file to process. The XML file can be located inside or outside the PXF file. To target XML files inside a PXF file, use the `|zip` locator (see the highlighted part in the examples above). The `--inputxml` option is ignored if the main schema source of the input PXF is a DB or DB-XML.

#### `dbwhere [dbw]`

```plaintext
--dbw, --dbwhere = WHEREClause
```

An SQL `WHERE` clause that determines what rows of a DB-XML source to process.

#### `param [p]`

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

Assigns a value to a parameter defined in the PXF file. The `--param` switch must be used before each parameter. Use quotes if `ParamName` or `ParamValue` contains a space. Example:

```plaintext
--p=$company:"Nanonull Inc"
```

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

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

Values are `true`|`false`. If `true`, does not allow the creation of output in any folder other than that in which the main output file (HTML, PDF, RTF, DOCX, FO) is created. This provides protection for other folders if needed. Default is `false`.

#### `outhtml [html]`

```plaintext
--html, --outhtml = FilePath
```

Path to the HTML file to generate.

#### `outpdf [pdf]`

```plaintext
--pdf, --outpdf = FilePath
```

Path to the PDF file to generate.
- `outrtf [rtf]`
  
  `--rtf, --outrtf = FilePath`
  
  Path to the RTF file to generate.

- `outdocx [docx]`
  
  `--docx, --outdocx = FilePath`
  
  Path to the DOCX file to generate.

- `outfo [fo]`
  
  `--fo, --outfo = FilePath`
  
  Path to the FO file to generate.

- `generate-html-output-as-mime`
  
  `--generate-html-output-as-mime = true|false`
  
  Values are true|false. If the option is not specified, default is false, if specified with no value, then true. If true, HTML output is generated as a mime stream.

- `verbose [v]`
  
  `--v, --verbose = true|false`
  
  Values are true|false. Turns the display of all messages, respectively, on or off. Default is false if the option is not provided, true if provided without a value.

- `lang [l]`
  
  `--l, --lang = en|de|es|fr|ja`
  
  The language used for displaying messages.

Use the `--h, --help` option to display information about the command.

Options are listed in short form (if available) and long form. You can use one or two dashes for both short and long forms. An option may or may not take a value. If it takes a value, it is written like this: `--option=value`. Values can be specified without quotes except in two cases: (i) when the value string contains spaces, or (ii) when explicitly stated in the description of the option that quotes are required. If an option takes a Boolean value and no value is specified, then the option's default value is true. Use the `--h, --help` option to display information about the command.
4.4 help

Syntax and description

The help command takes a single argument (Command), which is the name of the command for which help is required. It displays the command's syntax, its options, and other relevant information. If the Command argument is not specified, then all commands of the executable are listed, with each having a brief text description.

```
stylevisionserver help Command
```

-Casing and slashes on the command line

```
StyleVisionServer on Windows
stylevisionserver on Windows and Unix (Linux, Mac)
```

* Note that lowercase (stylevisionserver) works on all platforms (Windows, Linux, and Mac), while upper-lower (StyleVisionServer) works only on Windows and Mac.
* Use forward slashes on Linux and Mac, backslashes on Windows.

Example

Example of the help command to display information about the licenserver command:

```
stylevisionserver help licenseserver
```

The --help option

Help information about a command is also available by using the --help option of the command for which help information is required. The two commands below produce the same results:

```
stylevisionserver licenseserver --help
```

The command above uses the --help option of the licenseserver command.

```
stylevisionserver help licenseserver
```

The help command takes licenseserver as its argument.

Both commands display help information about the licenseserver command.
4.5 licenseserver

Syntax and description

The `licenseserver` command registers StyleVision Server with the Altova LicenseServer specified by the `Server-Or-IP-Address` argument. For the `licenseserver` command to be executed successfully, the two servers (StyleVision Server and LicenseServer) must be on the same network and LicenseServer must be running. You must also have administrator privileges in order to register StyleVision Server with LicenseServer.

```bash
stylevisionserver licenseserver [options] Server-Or-IP-Address
```

- The `Server-Or-IP-Address` argument takes the name or IP address of the LicenseServer machine.
- To request an evaluation license for StyleVision Server, submit appropriate values for the `--company`, `--email`, and `--name` options. To request an evaluation license for an additional Altova product (that is, in addition to StyleVision Server), add the `--request-eval-license` option with a value that is the name of that product. See the list of options below for more information.

Once StyleVision Server has been successfully registered with LicenseServer, you will receive a message to this effect. The message will also display the URL of the LicenseServer. You can now go to LicenseServer to assign StyleVision Server a license. For details about licensing, see the LicenseServer documentation (https://www.altova.com/manual/AltovaLicenseServer/).

▼ Casing and slashes on the command line

- **StyleVisionServer** on Windows
- **stylevisionserver** on Windows and Unix (Linux, Mac)

  * Note that lowercase (`stylevisionserver`) works on all platforms (Windows, Linux, and Mac), while upper-lower (`StyleVisionServer`) works only on Windows and Mac.
  * Use forward slashes on Linux and Mac, backslashes on Windows.

▼ Backslashes, spaces, and special characters on Windows systems

On Windows systems: When spaces or special characters occur in strings (for example in file or folder names, or company, person or product names), use quotes: for example, "My File". Note, however, that a backslash followed by a double-quotation mark (for example, “C:\My directory\”) might not be read correctly. This is because the backslash character is also used to indicate the start of an escape sequence, and the escape sequence \" stands for the double-quotation mark character. If you want to escape this sequence of characters, use a preceding backslash, like this: \". To summarize: If you need to write a file path that contains spaces or an end backslash, write it like this: "C:\My Directory\".

Examples

Examples of the `licenseserver` command:

- `stylevisionserver licenseserver DOC.altova.com`
- `stylevisionserver licenseserver localhost`
- `stylevisionserver licenseserver 127.0.0.1`
The commands above specify, respectively, the machine named `DOC.altova.com`, and the user's machine (`localhost` and `127.0.0.1`) as the machine running Altova LicenseServer. In each case, the command registers StyleVision Server with the LicenseServer on the machine specified. The last command calls the server-executable to execute the command.

**Options**

Options are listed in short form (if available) and long form. You can use one or two dashes for both short and long forms. An option may or may not take a value. If it takes a value, it is written like this: `--option=value`. Values can be specified without quotes except in two cases: (i) when the value string contains spaces, or (ii) when explicitly stated in the description of the option that quotes are required. If an option takes a Boolean value and no value is specified, then the option's default value is `TRUE`. Use the `--h, --help` option to display information about the command.

```
\> json [j]
   --j, --json = true|false
   Values are true|false. If true, prints the result of the registration attempt as a machine-parsable JSON object.
```
4.6 pdfdata

Syntax and description

The pdfdata command generates an FDF file or XML file from the PDF file that is submitted as the InputPDF argument.

```
stylevisionserver pdfdata [options] InputPDF
```

- The InputPDF argument specifies the path to the PDF file, from which the output FDF or XML file will be generated. If the PDF file does not have any form data, the generated file will contain no form data.
- Use the --outfdf option to specify the location of the generated FDF file or the --outxml option to specify the location of the generated XML file.

For more information about FDF files and designing fillable PDF forms, see the Altova StyleVision (Enterprise Edition) documentation.

▼ Casing and slashes on the command line

**StyleVisionServer on Windows**

**stylevisionserver on Windows and Unix (Linux, Mac)**

- Note that lowercase (stylevisionserver) works on all platforms (Windows, Linux, and Mac), while upper-lower (StyleVisionServer) works only on Windows and Mac.
- Use forward slashes on Linux and Mac, backslashes on Windows.

▼ Backslashes, spaces, and special characters on Windows systems

On Windows systems: When spaces or special characters occur in strings (for example in file or folder names, or company, person or product names), use quotes: for example, "My File". Note, however, that a backslash followed by a double-quotation mark (for example, "C:\My directory\") might not be read correctly. This is because the backslash character is also used to indicate the start of an escape sequence, and the escape sequence \" stands for the double-quotation mark character. If you want to escape this sequence of characters, use a preceding backslash, like this: \\". To summarize: If you need to write a file path that contains spaces or an end backslash, write it like this: "C:\My Directory\".

Examples

Examples of the pdfdata command:

```
stylevisionserver pdfdata --outfdf=C:\test\forms\FDFData.fdf C:\test\forms\TestForm.pdf
stylevisionserver pdfdata --outxml=C:\test\forms\XMLData.xml C:\test\forms\TestForm.pdf
```

The examples above create, respectively, and FDF file and an XML file from the same PDF input.

Options
StyleVision Server Command Line

- outfdf
  --outfdf = FilePath
  The path to the generated FDF file.

- outxml
  --outxml = FilePath
  The path to the generated XML file.

Use the --h, --help option to display information about the command.

Options are listed in short form (if available) and long form. You can use one or two dashes for both short and long forms. An option may or may not take a value. If it takes a value, it is written like this: --option=value. Values can be specified without quotes except in two cases: (i) when the value string contains spaces, or (ii) when explicitly stated in the description of the option that quotes are required. If an option takes a Boolean value and no value is specified, then the option's default value is true. Use the --h, --help option to display information about the command.
4.7 setdeflang

Syntax and description
The `setdeflang` command (short form is `sdl`) sets the default language of StyleVision Server. Available languages are English (`en`), German (`de`), Spanish (`es`), French (`fr`), and Japanese (`ja`). The command takes a mandatory `LanguageCode` argument.

```
stylevisionserver setdeflang [options] LanguageCode
```

- The `LanguageCode` argument is required and sets the default language of StyleVision Server. The respective values to use are: `en`, `de`, `es`, `fr`, `ja`.
- Use the `--h`, `--help` option to display information about the command.

Casing and slashes on the command line
```
StyleVisionServer on Windows
stylevisionserver on Windows and Unix (Linux, Mac)
```
* Note that lowercase (`stylevisionserver`) works on all platforms (Windows, Linux, and Mac), while upper-lower (`StyleVisionServer`) works only on Windows and Mac.
* Use forward slashes on Linux and Mac, backslashes on Windows.

Examples
Examples of the `setdeflang` (`sdl`) command:
```
stylevisionserver sdl de
stylevisionserver setdeflang es
```
- The first command sets the default language of StyleVision Server to German.
- The second command sets the default language of StyleVision Server to Spanish.

Options
Use the `--h`, `--help` option to display information about the command.
### 4.8 setfopath

#### Syntax and description

The `setfopath` command (short form is `sfp`) specifies the path to an Apache FOP processor other than that included in the StyleVision Server package.

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

- By default the Apache FOP processor that is included with StyleVision Server is used for processing FO documents and generating PDF output. If you wish to use some other Apache FOP processor instance than the processor supplied with StyleVision Server, use the `setfopath` command with the `Path` argument giving the path to the FO processor you want to use.
- After an alternative FO processor has been specified with the `setfopath` command, it is this processor that will be used when PDF is generated with subsequent `generate` commands. To change processors again, use the `setfopath` command again. To switch back to StyleVision Server's FOP processor, locate the FOP folder on your system and use this path as the argument of `setfopath`.
- On Windows systems, the FOP folder that was installed with StyleVision Server will be located under `ProgramData\Altova\SharedBetweenVersions`; on Linux and macOS systems in a descendant folder of the `StyleVisionServer2021` folder.

For more information about FDF files and designing fillable PDF forms, see the [Altova StyleVision (Enterprise Edition)](https://www.altova.com/stylevision-enterprise-edition) documentation.

▼ **Casing and slashes on the command line**

- **StyleVisionServer** on Windows
- **stylevisionserver** on Windows and Unix (Linux, Mac)

  * Note that lowercase (`stylevisionserver`) works on all platforms (Windows, Linux, and Mac), while upper-lower (`StyleVisionServer`) works only on Windows and Mac.
  * Use forward slashes on Linux and Mac, backslashes on Windows.

▼ **Backslashes, spaces, and special characters on Windows systems**

On Windows systems: When spaces or special characters occur in strings (for example in file or folder names, or company, person or product names), use quotes: for example, "My File". Note, however, that a backslash followed by a double-quotation mark (for example, "C:\My directory\") might not be read correctly. This is because the backslash character is also used to indicate the start of an escape sequence, and the escape sequence " stands for the double-quotation mark character. If you want to escape this sequence of characters, use a preceding backslash, like this: \". To summarize: If you need to write a file path that contains spaces or an end backslash, write it like this: "C:\My Directory\".

**Examples**

After running the `setfopath` command, you can use the `generate` command to generate a PDF using the just-specified FO processor:

```
stylevisionserver setfopath C:\FOP\FOP.bat
```
The commands above do the following:

1. The `setfopath` command specifies that the FO processor at the location `c:\FOP\FOP.bat` is to be used to generate PDF in subsequent PDF-generation commands.
2. The `generate` command generates a PDF file from the specified input XML, using transformation files contained in the PXF file. The FO processor specified in the previous command is used for generating the PDF.

**Options**

Use the `--h, --help` option to display information about the command.

Options are listed in short form (if available) and long form. You can use one or two dashes for both short and long forms. An option may or may not take a value. If it takes a value, it is written like this: `--option=value`. Values can be specified without quotes except in two cases: (i) when the value string contains spaces, or (ii) when explicitly stated in the description of the option that quotes are required. If an option takes a Boolean value and no value is specified, then the option's default value is `TRUE`. Use the `--h, --help` option to display information about the command.
4.9  verifylicense (Windows only)

Syntax and description
The verifylicense command checks whether the current product is licensed. Additionally, the --license-key option enables you to check whether a specific license key is already assigned to the product. This command is supported only on Windows systems. It is not supported on Linux or Mac systems.

    stylevisionserver verifylicense [options]

- To check whether a specific license is assigned to StyleVision Server, supply the license key as the value of the --license-key option.

For details about licensing, see the LicenseServer documentation (https://www.altova.com/manual/AltovaLicenseServer/).

▼ Casing and slashes on the command line
    StyleVisionServer on Windows
    stylevisionserver on Windows and Unix (Linux, Mac)

* Note that lowercase (stylevisionserver) works on all platforms (Windows, Linux, and Mac), while upper-lower (StyleVisionServer) works only on Windows and Mac.
* Use forward slashes on Linux and Mac, backslashes on Windows.

Examples
Example of the verifylicense command:

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

- The first command checks whether StyleVision Server is licensed.
- The second command checks whether StyleVision Server is licensed with the license key specified with the --license-key option.

Options
Options are listed in short form (if available) and long form. You can use one or two dashes for both short and long forms. An option may or may not take a value. If it takes a value, it is written like this: --option=value. Values can be specified without quotes except in two cases: (i) when the value string contains spaces, or (ii) when explicitly stated in the description of the option that quotes are required. If an option takes a Boolean value and no value is specified, then the option's default value is TRUE. Use the --h, --help option to display information about the command.

▼ license-key [l]
    --l, --license-key = Value
    Checks whether StyleVision Server is licensed with the license key specified as the value of this option.
4.10 version

Syntax and description
The version command displays the version number of StyleVision Server.

```
stylevisionserver version
```

Casing and slashes on the command line
- StyleVisionServer on Windows
- stylevisionserver on Windows and Unix (Linux, Mac)

* Note that lowercase (stylevisionserver) works on all platforms (Windows, Linux, and Mac), while upper-lower (StyleVisionServer) works only on Windows and Mac.
* Use forward slashes on Linux and Mac, backslashes on Windows.

Example
Example of the version command:

```
stylevisionserver version
```
5 StyleVision Server API

StyleVision Server provides an application programming interface (API) that you can access programmatically from your .NET, COM, or Java-based code.

This reference section is organized as follows:

- About the .NET Interface
- About the COM Interface
- About the Java Interface
- Code Examples
- API Reference
5.1 About the .NET Interface

The .NET interface is built as a wrapper around the COM interface. It is provided as a primary interop assembly signed by Altova and uses the namespace `Altova.StyleVisionServer`.

During installation, StyleVision Server will be registered automatically as a COM server object, so there is no need for a manual registration. If you receive an access error, open the Component Services and give permissions to the same account that runs the application pool containing StyleVision Server.

In order to use StyleVision Server in your .NET project, add a reference to the `Altova.StyleVisionServer.dll` file (see the instructions below). The `Altova.StyleVisionServer.dll` is located in the `bin` folder of the StyleVision Server installation folder. This `.dll` file is automatically added to the global assembly cache (GAC) during StyleVision Server installation (the GAC is typically located in the `C:\WINDOWS\assembly` folder).

Once StyleVision Server has been registered as a COM server object, and the `Altova.StyleVisionServer.dll` is available to the .NET interface, StyleVision Server API functionality becomes available in your .NET project.

To add a reference to the StyleVision Server DLL in a Visual Studio .NET project

1. With the .NET project open in Visual Studio, click **Project | Add Reference**. The Add Reference dialog box pops up.
2. On the **Browse** tab, browse for the folder `<StyleVisionServer application folder>/bin`, select `Altova.StyleVisionServer.dll`, and click **OK**.

You can view the structure of the `Altova.StyleVisionServer` assembly using the Visual Studio Object Browser (to display the Object Browser, click **Object Browser** on the **View** menu).
5.2 About the COM Interface

StyleVision Server is automatically registered as a COM server object during installation. To check whether the registration was successful, open the Registry Editor (for example, by typing `regedit.exe` command at the command line). If registration was successful, the Registry will contain the classes `StyleVision.Server`. These two classes will typically be found under `HKEY_LOCAL_MACHINE\SOFTWARE\Classes`.

Once the COM server object is registered, you can invoke it from within applications and scripting languages that have programming support for COM calls. If you wish to change the location of the StyleVision Server installation package, it is best to uninstall StyleVision Server and then reinstall it at the required location. In this way, the necessary de-registration and registration are carried out by the installer process.
5.3 About the Java Interface

To access the StyleVision Server API from Java code, add the following references to the .classpath file of your Java project.

<table>
<thead>
<tr>
<th>Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StyleVisionServer.jar</td>
<td>The library that communicates with StyleVision Server</td>
</tr>
<tr>
<td>StyleVisionServer_JavaDoc.zip</td>
<td>Documentation of the StyleVision Server API</td>
</tr>
</tbody>
</table>

Both files are available in the bin folder of the StyleVision Server installation folder. You can either reference the file from their original location or copy them to another location if this fits your project setup.

Adding library references in Eclipse

In Eclipse, you can add the classpath references by editing the properties of the Java project. The sample instructions below apply to Eclipse 4.4.

1. With the project open in Eclipse, on the **Project** menu, click **Properties**.
2. On the **Libraries** tab, click **Add External JARs**, and then browse for the StyleVisionServer.jar file located in the StyleVision Server installation folder.
3. Under **JARs and class folders on the build path**, expand the StyleVisionServer.jar record, and then double-click the Javadoc location: (None) record.
4. Ensure that the Javadoc in archive and External file options are selected, and then browse for the StyleVisionServer_JavaDoc.zip file located in the StyleVision Server installation folder.
5. Click **OK**. The reference to the StyleVision Server library and Javadoc archive is added to the .classpath file of the project.
5.4 Code Examples

The examples in this section are for the following programming languages:

- C++
- C#
- VBScript
- Visual Basic
- Java

5.4.1 C#

The example below shows how to use C# code to generate an output RTF file using a PXF file and an input XML file. Ensure that StyleVision Server is installed and licensed and that it is available as a COM server object. Registration as a COM server object usually takes place during installation of StyleVision Server. To check if registration was successful, see About the COM Interface.

```csharp
namespace StyleVisionServerAPI_sample
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                // Create a StyleVision Server object

                // 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\StyleVisionServer2021\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
```

© 2020 Altova GmbH

Altova StyleVision Server 2021
5.4.2 C++

The example below shows how to use C++ code to generate an output RTF file using a PXF file and an input XML file. Ensure that StyleVision Server is installed and licensed and that it is available as a COM server object. Registration as a COM server object usually takes place during installation of StyleVision Server. To check if registration was successful, see About the COM Interface.

```c++
#include <iostream>
#include "atlbase.h"

// 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[]) {
```

// Add output paths (absolute or relative to WorkingDirectory) for all formats that should be generated
objSVS.OutputRTF = "C:\\tmp\\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.Console.WriteLine("Success - finished execution");
else
    System.Console.WriteLine(objSVS.LastExecutionMessage);
}
```
{  
    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\StyleVisionServer2021\bin\StyleVisionServer.exe";
        //pSVS->ServerPath = "C:\Program Files\Altova\StyleVisionServer2021\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
        // 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;
    }
}

5.4.3 Java

The example below shows how to use Java code to generate an output RTF file using a PXF file and an input XML file. Ensure that StyleVision Server is installed and licensed and that it is available as a server object. Registration as a server object usually takes place during installation of StyleVision Server. To check if registration was successful, see About the Java Interface.

```java
public class Program {
    public static void main(String[] args) {
        try {
            //Create a StyleVision Server object

            //Set a working directory - used for output and for intermediate files
            objSVS.setWorkingDirectory( "C:\\temp" );

            //Default path to the StyleVision Server executable is the installation path
            //In case you copied the JAR file to a new location, you need to explicitly set
            //the path to the .exe file
            //objSVS.setServerPath( "C:\Program Files (x86)\\Altova\\StyleVisionServer2021\\bin\\StyleVisionServer.exe" );

            //Prepare the name of the working XML
            objSVS.setInputXML( "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
        }
    }
}
```
Code Examples

5.4.4 VBScript

The example below shows how to use VB Script code to generate an output RTF file using a PXF file and an input XML file. Ensure that StyleVision Server is installed and licensed and that it is available as a COM server object. Registration as a COM server object usually takes place during installation of StyleVision Server. To check if registration was successful, see About the COM Interface.

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.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

The example below shows how to use Visual Basic code to generate an output RTF file using a PXF file and an input XML file. Ensure that StyleVision Server is installed and licensed and that it is available as a COM server object. Registration as a COM server object usually takes place during installation of StyleVision Server. To check if registration was successful, see About the COM Interface.

Option Explicit On

Module Program

    Sub Main()

        Try

            'Create a StyleVision Server object

            '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

        Catch ex As Exception
            ' handle exception

        End Try

    End Sub

End Module
' 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
End Try

End Sub

End Module
5.5 API Reference

This section is a user's reference for the StyleVision Server API.

- COM and .NET
- Java

5.5.1 COM and .NET

The StyleVisionServer API exposes the `IServer` interface, which creates a new StyleVision Server object instance, and provides access to StyleVision Server.

The `IServer` interface has the following methods and properties.

Methods
AddParameter
Assigns a value to a parameter defined in the PXF file.

- C#
  ```csharp
  void AddParameter(string bstrName, string bstrValue)
  ```
- C++
  ```cpp
  HRESULT AddParameter([in] BSTR bstrName, [in] BSTR bstrValue );
  ```
- VB
  ```vb
  Sub AddParameter(ByVal bstrName As String, ByVal bstrValue As String)
  ```

ClearParameterList
Clears the list of parameters.

- C#
  ```csharp
  void ClearParameterList()
  ```
- C++
  ```cpp
  HRESULT ClearParameterList();
  ```
- VB
  ```vb
  Sub ClearParameterList()
  ```

Generate
Generates one or more output files (HTML, PDF, RTF, and/or DOCX) by using the PXF file specified with TransfPath. It transforms the input XML file (Working XML File in the PXF file) using the XSLT document contained in the PXF file. Returns TRUE in case of success; FALSE otherwise.

- C#
  ```csharp
  bool Generate(string bstrTransfPath)
  ```
- C++
  ```cpp
  HRESULT Generate( [in] BSTR bstrTransfPath, [out, retval] VARIANT_BOOL* pbSuccess );
  ```
- VB
  ```vb
  Function Generate(ByVal bstrTransfPath As String) As Boolean
  ```

Properties
### API Major Version

Gets the major version of the StyleVision Server API. It can be different from the product version if the API is connected to another server.

- **C#**
  ```csharp
  int APIMajorVersion { get; }
  ```

- **C++**
  ```cpp
  HRESULT APIMajorVersion([out, retval] INT* pnVal);
  ```

- **VB**
  ```vb
  ReadOnly Property APIMajorVersion As Integer
  ```

### API Minor Version

Gets the minor version of the StyleVision Server API. It can be different from the product version if the API is connected to another server.

- **C#**
  ```csharp
  int APIMinorVersion { get; }
  ```

- **C++**
  ```cpp
  HRESULT APIMinorVersion([out, retval] INT* pnVal);
  ```

- **VB**
  ```vb
  ReadOnly Property APIMinorVersion As Integer
  ```

### API Service Pack Version

Gets the service pack version of the StyleVision Server API. It can be different from the product version if the API is connected to another server.

- **C#**
  ```csharp
  int APIServicePackVersion { get; }
  ```

- **C++**
  ```cpp
  HRESULT APIServicePackVersion([out, retval] INT* pnVal);
  ```

- **VB**
  ```vb
  ReadOnly Property APIServicePackVersion As Integer
  ```

### InputXML

Sets the path and name of the XML file to be processed (the Working XML File in the PXF file).

- **C#**
  ```csharp
  string InputXML { set; }
  ```
C++

HRESULT InputXML([in] BSTR bstrPath);

VB

Property InputXML As String

▼ Is64Bit

Returns TRUE if the StyleVision Server engine is a 64-bit executable.

C#

bool Is64Bit { get; }

C++

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

VB

ReadOnly Property Is64Bit As Boolean

▼ LastExecutionMessage

Gets the message received during the last Generate command.

C#

string LastExecutionMessage { get; }

C++

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

VB

ReadOnly Property LastExecutionMessage As String

▼ MajorVersion

Gets the major version of StyleVision Server.

C#

int MajorVersion { get; }

C++

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

VB

ReadOnly Property MajorVersion As Integer

▼ MinorVersion
Gets the minor version of StyleVision Server.

- **C#**
  ```csharp
  int MinorVersion { get; }
  ```

- **C++**
  ```c++
  HRESULT MinorVersion([out, retval] INT* pnVal);
  ```

- **VB**
  ```vbnet
  ReadOnly Property MinorVersion As Integer
  ```

### OutputDOCX

Sets the path and name of the output `.docx` file.

- **C#**
  ```csharp
  string OutputDOCX { set; }
  ```

- **C++**
  ```c++
  HRESULT OutputDOCX([in] BSTR bstrPath );
  ```

- **VB**
  ```vbnet
  Property OutputDOCX As String
  ```

### OutputFO

Sets the path and name of the output `.foo` file.

- **C#**
  ```csharp
  string OutputFO { set; }
  ```

- **C++**
  ```c++
  HRESULT OutputFO([in] BSTR bstrPath );
  ```

- **VB**
  ```vbnet
  Property OutputFO As String
  ```

### OutputHTML

Sets the path and name of the output `.html` file.

- **C#**
  ```csharp
  string OutputHTML { set; }
  ```

- **C++**
  ```c++
  HRESULT OutputHTML([in] BSTR bstrPath );
  ```

- **VB**
  ```vbnet
  Property OutputHTML As String
  ```
Property OutputHTML As String

OutputPDF
Sets the path and name of the output PDF file.

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

OutputRTF
Sets the path and name of the output RTF file.

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

ProductName
Gets the name of the product: "StyleVision Server"

- C#
  string ProductName { get; }
- C++
  HRESULT ProductName([out, retval] BSTR* pstrVal);
- VB
  ReadOnly Property ProductName As String

ProductNameAndVersion
Gets the complete name of the product, including the version number: "StyleVision Server 2014r2 sp1 (x64)".
string ProductNameAndVersion { get; }

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

ReadOnly Property ProductNameAndVersion As String

ServerPath

Gets or sets the path to the StyleVision Server executable.

string ServerPath { set; get; }

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

Property ServerPath As String

ServicePackVersion

Gets the service pack version of StyleVision Server (for example: 1 for Altova StyleVision Server 2014 r2 sp1 (x64).)

int ServicePackVersion { get; }

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

ReadOnly Property ServicePackVersion As Integer

WhereClause

Sets an SQL WHERE clause that determines the rows of a DB-XML schema source to process.

string WhereClause { set; }

HRESULT WhereClause([in] BSTR bstrPath );
VB

    Property WhereClause As String

WorkingDirectory

Gets or sets the current directory for running jobs. Relative paths are evaluated against the working
directory.

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

The package com.altova.stylevisionserver consists of the following classes:

- public class StyleVisionServer (described below)
- public class StyleVisionServerException extends Exception

StyleVisionServer class

The StyleVisionServer class creates a new StyleVision Server object instance, and provides access to
StyleVision Server. The methods of the StyleVisionServer interface are described below.

Methods of StyleVisionServer class

The methods of the StyleVisionServer class are listed alphabetically below.

addParameter

    public void addParameter(String name, String value)

Adds the name and value of a new parameter. Each parameter and its value is specified in a separate call
to the method. Parameters must be declared in the XSLT document.

Parameters:

- name: Holds the name of the parameter as a string.
- value: Holds the value of the parameter as a string.

clearParameterList
public void clearParameterList()
Clears the list of parameters.

generate
public boolean generate(String transfPath)
Processes the PXF file specified in transfPath. Throws StyleVisionServerException.
Parameters:
transfPath: An absolute URL giving the location of the PXF file.
Returns:
true() if execution is successful
false() if execution fails
In case of an error, use getLastExecutionMessage()

getAPIMajorVersion
public int getAPIMajorVersion()
Gets the major version of the StyleVision Server API. It can be different from the product version if the API is connected to another server.

getAPIMinorVersion
public int getAPIMinorVersion()
Gets the minor version of the StyleVision Server API. It can be different from the product version if the API is connected to another server.

getAPIServicePackVersion
public int getAPIServicePackVersion()
Gets the service pack version of the StyleVision Server API. It can be different from the product version if the API is connected to another server.

getLastExecutionMessage
public String getLastExecutionMessage()
Gets the message received during the last generate command.

getMajorVersion
public int getMajorVersion()
Gets the major version of the application.

getMinorVersion
public int getMinorVersion()
Gets the minor version of the application.

getProductName
public String getProductName()
Gets the product name.

getProductNameAndVersion
public String getProductNameAndVersion()
Gets the complete name and version number of the product.

-servicePackVersion
public int getServicePackVersion()
Gets the service pack version of the StyleVision Server.

-is64bit
public boolean is64bit()
Checks whether the executable is 64-bit.
Returns:
true() for StyleVision Server (x64), false() otherwise.

-inputXML
public void setInputXML(String path)
Sets the XML file to process. This must be the path of the Working XML File that is specified in the PXF file.
Parameters:
path: Holds the path of the Working XML file in the PXF file.

-outputDOCX
public void setOutputDOCX(String path)
Sets the path and name of the DOCX file to generate.
Parameters:
path: The path and name of the DOCX file to generate.

-outputFO
public void setOutputFO(String path)
Sets the path and name of the FO file to generate.
Parameters:
path: The path and name of the FO file to generate.

-outputHTML
public void setOutputHTML(String path)
Sets the path and name of the HTML file to generate.
Parameters:
path: The path and name of the HTML file to generate.

-outputPDF
public void setOutputPDF(String path)
Sets the path and name of the PDF file to generate.
Parameters:
path: The path and name of the PDF file to generate.

-outputRTF
public void setOutputRTF(String path)
Sets the path and name of the RTF file to generate.

**Parameters:**
- path: The path and name of the RTF file to generate.

### setServerPath

```java
public void setServerPath(String serverFile)
```

Sets the path of the StyleVisionServer executable.

**Parameters:**
- serverFile: The path of the StyleVisionServer executable.

### setWhereClause

```java
public void setWhereClause(String whereClause)
```

Sets an SQL WHERE clause that determines the rows of a DB-XML schema source to process.

**Parameters:**
- whereClause: The SQL WHERE clause that determines the rows of a DB-XML schema source to process.

### setWorkingDirectory

```java
public void setWorkingDirectory(String workingDirectory)
```

Sets a default directory. Relative paths are resolved relative to this directory.

**Parameters:**
- workingDirectory: The path of the default (working) directory.
6 XBRL Taxonomy Manager

XBRL Taxonomy Manager is a tool that provides a centralized way to install and manage XBRL taxonomies for use across all Altova XBRL-enabled applications, including StyleVision Server. On Windows, XBRL Taxonomy Manager has a graphical user interface and is also available at the command line. On Linux and Mac*, the tool is available at the command line only.

* The Linux and macOS operating systems are applicable only if you are running XBRL Taxonomy Manager on those operating systems in conjunction with Altova cross-platform server applications such as MapForce Server, StyleVision Server, or RaptorXML+XBRL Server.

XBRL Taxonomy Manager provides the following features:

- View XBRL taxonomies installed on your computer, and check whether new versions are available for download.
- Download newer versions of XBRL taxonomies independently of the Altova product release cycle. All taxonomies are maintained by Altova on an online-based storage accessible to XBRL Taxonomy Manager, and you can download them as soon as they become available.
- Install or uninstall any of the multiple versions of a given taxonomy (or all versions if necessary).
- A single XBRL taxonomy represents a "package" but it may have dependencies on other taxonomies. Whenever you choose to install or uninstall a particular taxonomy, any dependent taxonomies are detected and also installed or removed automatically. The graphical user interface (or the command line if applicable) informs you when dependencies are being added or removed.
- XBRL taxonomies maintained through XBRL Taxonomy Manager benefit from the XML catalog mechanism that enables URI references in instance or schema documents to be resolved from local files, as opposed to being retrieved from the Internet. This is extremely important in the case of big XBRL taxonomies where schema resolution from remote URIs is not practical or even recommended, mainly for performance reasons.

XBRL Taxonomy Manager provides a way to administer any of the XBRL taxonomies required for use in any one of the Altova XBRL-enabled applications. These include the European Banking Authority Reporting Framework taxonomies, US-GAAP Financial Reporting taxonomies, and various other country- or domain-specific XBRL taxonomies. To view the full list, either run XBRL Taxonomy Manager or run the list command at the command line.

Custom XBRL Taxonomies

If you need to work with custom XBRL taxonomies that are not included with XBRL Taxonomy Manager, you can enable those as follows:

- From your Altova XBRL-enabled desktop application, run the Tools | Options menu command, and then go to the XBRL | Taxonomy Packages settings page. From the settings page, browse for the .zip package of your custom XBRL taxonomy.
- From your Altova XBRL-enabled server application, provide the --taxonomy-package or --taxonomy-package-config-file options when running commands or API methods that support them. For example, these options are supported by XBRL validation commands such as valxbrl or valxbrltaxonomy in RaptorXML+XBRL Server, or by the run command (or the equivalent API method) in MapForce Server.
Altova XBRL-enabled applications

The following Altova applications are XBRL-enabled and thus benefit from the features provided by XBRL Taxonomy Manager:

- Altova XBRL Add-ins for Excel (EBA, Solvency II)
- MapForce Enterprise Edition
- MapForce Server
- MapForce Server Advanced Edition
- RaptorXML+XBRL Server
- StyleVision Server
- StyleVision Enterprise Edition
- XMLSpy Enterprise Edition

Changes in XBRL taxonomies using XBRL Taxonomy Manager take effect for all the applications listed above if they are installed on the same computer.

Installation

XBRL Taxonomy Manager is installed automatically whenever you install any Altova XBRL-enabled application or the Altova Mission Kit Enterprise Edition. Likewise, it is removed automatically when you uninstall the last Altova XBRL-enabled application from your computer or the Altova Mission Kit Enterprise Edition.

How it works

Altova maintains an online taxonomy storage where all XBRL taxonomies used in Altova products are stored. This taxonomy storage is updated on a periodical basis, for example, shortly after new versions of relevant taxonomies are released by their issuing organizations.

Whenever you run XBRL Taxonomy Manager at the graphical user interface, information about the latest available taxonomies is displayed in a dialog box, where you can view, install, upgrade or uninstall them. You can also perform the same actions at the command line interface.
You may also install taxonomies by running .altova_taxonomies files downloaded from the Altova website (https://www.altova.com/download). In addition, when you take an action that requires loading XBRL taxonomies, StyleVision Server prompts you to install the missing XBRL taxonomies.

Regardless of the manner in which taxonomies were installed, all information about installed taxonomies is tracked in a centralized location on your computer, also known as the local cache directory. The local cache directory contains information about Altova packages (except for the actual taxonomy files, which are installed on demand). The local cache directory is at the following path:

<table>
<thead>
<tr>
<th>Platform</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>/var/opt/Altova/pkgs</td>
</tr>
<tr>
<td>macOS</td>
<td>/var/Altova/pkgs</td>
</tr>
</tbody>
</table>
The local cache directory gets updated automatically from time to time, so as to propagate the latest state of the online storage to the local computer. More specifically, the cache is updated as follows:

- When you run the XBRL Taxonomy Manager.
- When you run StyleVision Server for the first time in the same calendar day.
- If StyleVision Server is already running, the cache directory gets updated every 24 hours.
- You can also update the local cache from the online storage on demand, by running the update command at the command line interface.

As you install or uninstall taxonomies, the local cache directory gets automatically updated with information about the available and installed taxonomies, as well as the taxonomy files themselves.

The local cache directory is maintained automatically based on the taxonomies you install or uninstall; it should not be altered or deleted manually. If you ever need to reset XBRL Taxonomy Manager to the original “pristine” state, run the reset command of the command line interface, and then run the initialize command. (Alternatively, run the reset command with the -i option.)
6.1 Run XBRL Taxonomy Manager

You can run XBRL Taxonomy Manager by using the approaches listed below, as applicable.

Graphical user interface (Windows only)
If StyleVision Server runs on Windows, you can run XBRL Taxonomy Manager as follows:

- Run (double-click) a file with .altova_taxonomies extension downloaded from the Altova website.
- From the Windows Control Panel, right-click the Altova Taxonomy Manager entry and select Change or Uninstall from the context menu.

In addition, the check box Invoke Altova Taxonomy Manager is available on the last page of the installation wizard, after you complete the installation of StyleVision Server.

Command line interface (Windows)
To run XBRL Taxonomy Manager from a command line interface:

1. Open a command prompt window and change directory to C:\ProgramData\Altova\SharedBetweenVersions.
2. To display help at the command line, run:
   
   ```
   TaxonomyManager.exe --help
   ```

Command line interface (Linux, macOS)
To run XBRL Taxonomy Manager from a command line interface:

1. Open a terminal window and change directory to %INSTALLDIR%\bin, where %INSTALLDIR% is the program's installation directory.
2. To display help at the command line, run:

   ```
   sudo ./taxonomymanager --help
   ```
6.2 Install a Taxonomy

You can install a taxonomy by using the approaches listed below, as applicable.

Note the following:

- Installing or uninstalling a taxonomy from XBRL Taxonomy Manager takes effect for all users accounts on the same computer.
- Installing or uninstalling a taxonomy from XBRL Taxonomy Manager takes effect in all Altova XBRL-enabled applications installed on the same computer.
- If the current taxonomy has dependencies on other taxonomies, the dependent taxonomies are also installed (or uninstalled, as applicable).

Graphical user interface (Windows only)

To install a taxonomy:

1. Run XBRL Taxonomy Manager.
2. Select the check box next to the taxonomies or taxonomy versions you want to install, and click Apply.

Alternatively, if you have downloaded a file with .altova_taxonomies extension from the Altova website, double-click the .altova_taxonomies file to run it. XBRL Taxonomy Manager opens when you run the .altova_taxonomies file.

Command line interface (Windows)

To install a taxonomy, run:

```
TaxonomyManager.exe install FILTER...
```

Where FILTER means one of the following:

1. A taxonomy identifier in the format `<name>-<version>`, for example: `eba-2.10, us-gaap-2020.0`. To view all the available taxonomy identifiers and versions, run the list command.
2. An `.altova_taxonomies` file downloaded from the Altova website.

Command line interface (Linux, macOS)

To install a taxonomy, run:

```
sudo ./taxonomymanager install FILTER...
```

Where FILTER means one of the following:

1. A taxonomy identifier in the format `<name>-<version>`, for example: `eba-2.10, us-gaap-2020.0`. To view all the available taxonomy identifiers and versions, run the list command.
2. An `.altova_taxonomies` file downloaded from the Altova website.
Installing taxonomies on demand
Whenever StyleVision Server detects that certain XBRL taxonomies must be installed on your computer, you may be prompted to install taxonomies on demand. For example, if you run a command that requires loading XBRL taxonomies, and if no XBRL taxonomies are currently installed, the command is cancelled and an error message similar to the following appears:

Unable to load a schema with target namespace [...] from [...].
Details:
I/O Error: File [...] is part of missing taxonomy [...]. Call 'taxonomymanager install [...]'.

To install the missing taxonomy, run the install command as suggested by the error message. You can always view all of the previously installed taxonomies by running the list command.

Applying patches
Occasionally, XBRL taxonomies may receive patches from their issuers. When the XBRL Taxonomy Manager detects that patches are available, the following happens:

- If you use XBRL Taxonomy Manager through the Windows graphical user interface, the respective XBRL taxonomies are shown with the icon.
- If you use the command line or a Linux/macOS system, any XBRL taxonomies that have patches are listed when you run the executable with the list -u command.

To apply a patch on Windows:

1. Click the Patch Selection button. The icon of each XBRL taxonomy that qualifies changes from to , and the dialog box informs you about the patches that are to be applied, for example:
Note: The Patch Selection button is enabled only when there are patches available for any of the currently installed XBRL taxonomies.

2. Click Apply.

To apply a patch at the command line interface:

1. Run the list -u command. This lists any taxonomies where patch upgrades are available.
2. Run the upgrade command to install the patches.

For more information, see the reference to the Command Line Interface.
6.3 View Installed Taxonomies

You can view the currently installed taxonomies by using the approaches listed below, as applicable.

Graphical user interface (Windows only)
To view all installed taxonomies from a graphical user interface, run XBRL Taxonomy Manager. A selected check box next to a taxonomy (or a taxonomy version) indicates that that taxonomy is installed.

Command line interface (Windows)
To view all available taxonomies from a command line interface, run:

```command
TaxonomyManager.exe list
```

To view only installed taxonomies, run:

```command
TaxonomyManager.exe list -i
```

To view only taxonomies where a newer version is available, run:

```command
TaxonomyManager.exe list -u
```

Command line interface (Linux, macOS)
To view all available taxonomies, run:

```command
sudo ./taxonomymanager list
```

To view only installed taxonomies, run:

```command
sudo ./taxonomymanager list -i
```

To view only taxonomies where a newer version is available, run:

```command
sudo ./taxonomymanager list -u
```
6.4 Uninstall a Taxonomy

You can uninstall a taxonomy by using the approaches listed below, as applicable.

Graphical user interface (Windows only)

To uninstall a taxonomy:

1. Run XBRL Taxonomy Manager.
2. Clear the check box next to the taxonomies or taxonomy versions you want to uninstall. If the selected taxonomy is dependent on other taxonomy packages, a dialog box opens, informing you that the dependencies will be removed as well, for example:

```
Package Dependencies

The following packages depend on the selected package and therefore have to be removed too:
US GAAP 2020 - Financial reporting taxonomy provided by the US Security and Exchange Commission

OK Cancel
```
3. Click **Apply** to remove the taxonomy and its dependencies.

Command line interface (Windows)

To uninstall a taxonomy, run:

```
TaxonomyManager.exe uninstall FILTER...
```

Where **FILTER** means one of the following:

1. A taxonomy identifier in the format `<name>-<version>`, for example: **eba-2.10, us-gaap-2020.0**. To view all the available taxonomy identifiers and versions, run the `list` command.
2. An **.altova_taxonomies** file downloaded from the Altova website.

Command line interface (Linux, macOS)

To uninstall a package, run:

```
sudo ./taxonomymanager uninstall FILTER...
```

Where **FILTER** means one of the following:

1. A taxonomy identifier in the format `<name>-<version>`, for example: **eba-2.10, us-gaap-2020.0**. To view all the available taxonomy identifiers and versions, run the `list` command.
2. An **.altova_taxonomies** file downloaded from the Altova website.
6.5 Command Line Interface

To call XBRL Taxonomy Manager at the command line, you need to know the path of the executable. By default, the XBRL Taxonomy Manager executable is installed at the following path:

<table>
<thead>
<tr>
<th>OS</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux*</td>
<td>/opt/Altova/StyleVisionServer2021/bin/taxonomymanager</td>
</tr>
<tr>
<td>macOS*</td>
<td>/usr/local/Altova/StyleVisionServer2021/bin/taxonomymanager</td>
</tr>
<tr>
<td>Windows</td>
<td>C:\ProgramData\Altova\SharedBetweenVersions\TaxonomyManager.exe</td>
</tr>
</tbody>
</table>

* The Linux and macOS paths are applicable only if you are running XBRL Taxonomy Manager on those operating systems in conjunction with Altova cross-platform server applications such as MapForce Server, StyleVision Server, or RaptorXML+XBRL Server.

By convention, this documentation omits the full path of the executable when describing a given command, and uses `<exec>` instead of the executable name, for example:

```
<exec> help
```

Where `<exec>` is the path or name of the executable.

Calling XBRL Taxonomy Manager from the installation directory

To call the executable without having to type the full path, change the current directory to the one below:

<table>
<thead>
<tr>
<th>OS</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>cd /opt/Altova/StyleVisionServer2021/bin</td>
</tr>
<tr>
<td>macOS</td>
<td>cd /usr/local/Altova/StyleVisionServer2021/bin</td>
</tr>
<tr>
<td>Windows</td>
<td>cd &quot;C:\ProgramData\Altova\SharedBetweenVersions&quot;</td>
</tr>
</tbody>
</table>

You can now run a command by calling the executable with a relative path, for example:

<table>
<thead>
<tr>
<th>OS</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>sudo ./taxonomymanager help</td>
</tr>
<tr>
<td>macOS</td>
<td>sudo ./taxonomymanager help</td>
</tr>
<tr>
<td>Windows</td>
<td>TaxonomyManager.exe help</td>
</tr>
</tbody>
</table>

Note: On Linux and macOS systems, the prefix `/` indicates that the executable is in the current directory. The prefix `sudo` indicates that the command must be run with root privileges.

Calling XBRL Taxonomy Manager from any directory

To call the executable from any directory, refer to it using the absolute path. Alternatively, if you want to call the program by typing just the executable name, you can edit the PATH environment variable of your operating environment.
system so that it includes the full path to the executable's directory. For ways to change the PATH environment variable, refer to the documentation of your operating system.

Notes:

- After changing the PATH environment variable, you may need to close the terminal window and open a new one, in order for the changes to take effect.
- On Linux and macOS, using `sudo` does not take into account the user's PATH.

Command line syntax
The general syntax for using the command line is as follows:

```
<exec> -h | --help | --version | <command> [options] [arguments]
```

In the listing above, the vertical bar `|` separates a set of mutually exclusive items. The square brackets `[]` indicate optional items. Essentially, you can type the executable path followed by either `--h`, `--help`, or `--version` options, or by a command. Each command may have options and arguments. The list of commands is described in the following sections.

6.5.1 help
This command provides contextual help about commands pertaining to XBRL Taxonomy Manager executable.

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

Where `[command]` is an optional argument which specifies any valid command name.

Remarks
You can also invoke help by typing a command followed by `--h` or `--help`, for example:

```
<exec> list --h
```

You can also invoke general help by typing `--h` or `--help` directly after the executable, for example:

```
<exec> --help
```

Example
The following command displays help about the `list` command:

```
Linux, macOS ./taxonomymanager help list
```
6.5.2 info

This command displays detailed information for each of the taxonomies supplied as argument. The detailed information includes the title, version, description, publisher, and dependency references.

Syntax
<exec> info [options] FILTER...

To specify multiple taxonomies, repeat FILTER as many times as necessary.

Arguments

<table>
<thead>
<tr>
<th>FILTER</th>
<th>List only taxonomies that contain this string in their name (for example, eba or us-gaap-2020.0). You can specify this argument multiple times. To view the full list of identifiers, run the list command.</th>
</tr>
</thead>
</table>

Options

|--help, --h Display help about this command at the command line.

Example
The following command displays detailed information about the eba-2.10 and us-gaap-2020.0 taxonomies:

| Linux, macOS | ./taxonomymanager info eba-2.10 us-gaap-2020.0 |
| Windows      | TaxonomyManager.exe info eba-2.10 us-gaap-2020.0 |

6.5.3 initialize

This command initializes XBRL Taxonomy Manager environment. It creates a cache directory where information about all taxonomies is stored. Initialization is performed automatically the first time when you install an Altova application that includes supports for XBRL Taxonomy Manager, so you don't need to run this command under normal circumstances. You typically need to run this command after executing the reset command.

Syntax
<exec> initialize [options]
The alias of this command is `init`.

**Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--help, --h</code></td>
<td>Display help about this command at the command line.</td>
</tr>
<tr>
<td><code>--silent, --s</code></td>
<td>Display only error messages. The default value is <code>false</code>.</td>
</tr>
<tr>
<td><code>--verbose, --v</code></td>
<td>Display more information during execution. The default value is <code>false</code>.</td>
</tr>
</tbody>
</table>

### 6.5.4 install

This command installs one or more taxonomies. Note the following:

- Installing or uninstalls a taxonomy from XBRL Taxonomy Manager takes effect for all users accounts on the same computer.
- Installing or uninstalls a taxonomy from XBRL Taxonomy Manager takes effect in all Altova XBRL-enabled applications installed on the same computer.
- If the current taxonomy has dependencies on other taxonomies, the dependent taxonomies are also installed (or uninstalled, as applicable).

**Syntax**

```
<exec> install [options] FILTER...
```

To specify multiple taxonomies to install, repeat `FILTER` as many times as necessary.

**Arguments**

<table>
<thead>
<tr>
<th>FILTER</th>
<th>Where <code>FILTER</code> means one of the following:</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--help, --h</code></td>
<td>Display help about this command at the command line.</td>
</tr>
<tr>
<td><code>--silent, --s</code></td>
<td>Display only error messages. The default value is <code>false</code>.</td>
</tr>
<tr>
<td><code>--verbose, --v</code></td>
<td>Display more information during execution. The default value is <code>false</code>.</td>
</tr>
</tbody>
</table>

1. A taxonomy identifier in the format `<name>-<version>`, for example: `eba-2.10, us-gaap-2020.0`. To view all the available taxonomy identifiers and versions, run the `list` command.
2. An `.altova_taxonomies` file downloaded from the Altova website.

You can also use abbreviated identifiers if they are unique, for example, `eba`. If you use an abbreviated identifier, this will install the latest available version of that taxonomy.
Example
The following command installs the latest **eba** (European Banking Authority) and **us-gaap** (US Generally Accepted Accounting Principles) taxonomies:

| Linux, macOS | ./taxonomymanager install eba us-gaap |
| Windows      | TaxonomyManager.exe install eba us-gaap |

6.5.5 list

Use this command to list taxonomies at the command line, in one of the following ways:

- list all available taxonomies
- list specific taxonomies
- list only installed taxonomies
- list only taxonomies that require upgrade.

Syntax

```
<exec> list [options] [FILTER...]
```

This command can be abbreviated with *ls*.

Arguments

| FILTER | List only taxonomies that contain this string in their name. You can specify this argument multiple times. |

Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--help, --h</td>
<td>Display help about this command at the command line.</td>
</tr>
<tr>
<td>--installed, --i</td>
<td>List only installed taxonomies. The default value is <strong>false</strong>.</td>
</tr>
<tr>
<td>--upgradeable, --u</td>
<td>List only taxonomies where patch upgrades are available. The default value is <strong>false</strong>.</td>
</tr>
</tbody>
</table>

Examples

To list all available taxonomies, run:

| Linux, macOS | ./taxonomymanager list |
| Windows      | TaxonomyManager.exe list |
To list only installed taxonomies, run:

<table>
<thead>
<tr>
<th>Platform</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux, macOS</td>
<td>./taxonomymanager list -i</td>
</tr>
<tr>
<td>Windows</td>
<td>TaxonomyManager.exe list -i</td>
</tr>
</tbody>
</table>

To list all taxonomies that contain either "eba" or "us-gaap" in their name, run:

<table>
<thead>
<tr>
<th>Platform</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux, macOS</td>
<td>./taxonomymanager list eba us-gaap</td>
</tr>
<tr>
<td>Windows</td>
<td>TaxonomyManager.exe list eba us-gaap</td>
</tr>
</tbody>
</table>

### 6.5.6 reset

This command removes all installed taxonomies and the cache directory.

**Warning:** This command deletes all installed taxonomies and their information.

After running this command, make sure to run the initialize command, in order to recreate the cache directory. Alternatively, run the reset command with the -i option.

Note that reset -i restores the original installation of the product, so it's recommended to run the update command as well, after performing a reset. Alternatively, run the reset command with the -i and -u options.

**Syntax**

```
<exec> reset [options]
```

**Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--help, --h</td>
<td>Display help about this command at the command line.</td>
</tr>
<tr>
<td>--init, --i</td>
<td>Initialize the XBRL Taxonomy Manager environment after reset. Valid values are true and false. The default value is false.</td>
</tr>
<tr>
<td>--silent, --s</td>
<td>Display only error messages. The default value is false.</td>
</tr>
<tr>
<td>--update, --u</td>
<td>Initialize and update the XBRL Taxonomy Manager environment after reset. Valid values are true and false. The default value is false.</td>
</tr>
<tr>
<td>--verbose, --v</td>
<td>Display additional information during execution. The default value is false.</td>
</tr>
</tbody>
</table>

**Examples**

To reset the XBRL Taxonomy Manager, run:
### 6.5.7 setdeflang

This command sets the language of XBRL Taxonomy Manager.

**Syntax**

```
<exec> setdeflang language
```

Where `language` is a mandatory argument supplying the language code. The alias of this command is `sdl`.

**Arguments**

| `language` | The language to be set. Valid values are English (en), French (fr), German (de), Japanese (ja), and Spanish (es). |

**Options**

| `--help, --h` | Display help about this command at the command line. |

**Examples**

To set the language to Spanish, run:

<table>
<thead>
<tr>
<th>Platform</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux, macOS</td>
<td><code>./taxonomymanager setdeflang es</code></td>
</tr>
<tr>
<td>Windows</td>
<td><code>TaxonomyManager.exe setdeflang es</code></td>
</tr>
</tbody>
</table>

### 6.5.8 uninstall

This command uninstalls one or more taxonomies. By default, any taxonomies referenced by the current one are uninstalled as well. To uninstall just the current taxonomy and keep the referenced taxonomies, set the option `--k`.

**Syntax**

```
<exec> uninstall FILTER...
```

To specify multiple taxonomies, repeat `FILTER` as many times as necessary.
Arguments

<table>
<thead>
<tr>
<th>FILTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where FILTER means one of the following:</td>
</tr>
</tbody>
</table>

1. A taxonomy identifier in the format `<name>-<version>`, for example: `eba-2.10, us-gaap-2020.0`. To view all the available taxonomy identifiers and versions, run the list command.
2. An `.altova_taxonomies` file downloaded from the Altova website.

Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--help, --h</code></td>
<td>Display help about this command at the command line.</td>
</tr>
<tr>
<td><code>--keep-references, --k</code></td>
<td>If this option is set, then referenced taxonomies are not uninstalled. The default value is <code>false</code>.</td>
</tr>
<tr>
<td><code>--silent, --s</code></td>
<td>Display only error messages. The default value is <code>false</code>.</td>
</tr>
<tr>
<td><code>--verbose, --v</code></td>
<td>Display additional information during execution. The default value is <code>false</code>.</td>
</tr>
</tbody>
</table>

Example

The following command uninstalls the `eba-2.10` and `us-gaap-2020.0` taxonomies and their dependencies:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux, macOS</td>
<td><code>./taxonomymanager uninstall eba-2.10 us-gaap-2020.0</code></td>
</tr>
<tr>
<td>Windows</td>
<td><code>TaxonomyManager.exe uninstall eba-2.10 us-gaap-2020.0</code></td>
</tr>
</tbody>
</table>

6.5.9 update

This command queries the list of taxonomies available from the online storage and updates the local cache directory. The update of this information takes place implicitly and you shouldn't need to run this command unless you have performed a reset and initialize.

Syntax

```
<exec> update [options]
```

Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--help, --h</code></td>
<td>Display help about this command at the command line.</td>
</tr>
<tr>
<td><code>--silent, --s</code></td>
<td>Display only error messages. The default value is <code>false</code>.</td>
</tr>
<tr>
<td><code>--verbose, --v</code></td>
<td>Display additional information during execution. The default value is <code>false</code>.</td>
</tr>
</tbody>
</table>
Example
The following command updates the taxonomies information explicitly.

| Linux, macOS | ./taxonomymanager update |
| Windows      | TaxonomyManager.exe update |

6.5.10 upgrade

This command upgrades all eligible taxonomies to the latest available patch version. In other words, it performs only upgrades at patch level of a specific release. Running this command is meaningful only if there are upgradeable taxonomies available. You can identify upgradeable taxonomies by running the list -u command.

Syntax

```bash
<exec> upgrade [options]
```

Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--help, --h</code></td>
<td>Display help about this command at the command line.</td>
</tr>
<tr>
<td><code>--silent, --s</code></td>
<td>Display only error messages. The default value is <code>false</code>.</td>
</tr>
<tr>
<td><code>--verbose, --v</code></td>
<td>Display additional information during execution. The default value is <code>false</code>.</td>
</tr>
</tbody>
</table>
Index

C
Cores, 12

F
FlowForce Server,
and StyleVision Server, 9

I
Installation on Linux, 19
Installation on macOS, 26
Installation on Windows, 13

L
Licensing and processor cores, 12
Licensing on Linux, 22
Licensing on macOS, 28
Licensing on Windows, 15
Linux,
installation on, 19
licensing on, 22

M
macOS,
installation on, 26
licensing on, 28

P
Processor cores, 12

S
Setup,
on Linux, 19
on macOS, 26
on Windows, 13
Single-thread execution, 12
StyleVision Server,
features, 8
functionality, 8
in the FlowForce workflow, 9

W
Windows,
installation on, 13
licensing on, 15

X
XBRL Taxonomy Manager, 74