

# **Altova StyleVision Server 2021**

**Benutzer- und Referenzhandbuch**

# **Altova StyleVision Server 2021**

## **Benutzer- und Referenzhandbuch**

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

---

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>5</b>
1.1	Funktionalität.....	7
1.1.1	Im FlowForce Workflow.....	7
1.1.2	Als eigenständiger Server.....	8
1.2	Einrichten von StyleVision Server.....	9
1.2.1	Prozesskerne und Lizenzen.....	9
1.2.2	Einrichten unter Windows.....	10
1.2.3	Einrichten unter Linux.....	15
1.2.4	Einrichten auf macOS.....	22
1.2.5	Zusätzliche Anmerkungen zur Konfiguration.....	27
1.3	StyleVision Server-Befehlszeile.....	29
1.3.1	assignlicense (nur Windows).....	30
1.3.2	exportresourcestrings.....	31
1.3.3	generate.....	33
1.3.4	help .....	36
1.3.5	licenseserver.....	37
1.3.6	pdfdata .....	38
1.3.7	setdeflang.....	40
1.3.8	setfopath.....	41
1.3.9	verifylicense (nur Windows).....	42
1.3.10	version .....	43
1.4	StyleVision Server API.....	45
1.4.1	Informationen zur .NET-Schnittstelle.....	45
1.4.2	Informationen zur COM-Schnittstelle.....	46
1.4.3	Informationen zur Java-Schnittstelle.....	46
1.4.4	Codebeispiele.....	47
1.4.5	API-Referenz.....	54

---

**Index**

**65**

# 1 Einführung

**Altova StyleVision Server** ist eine Implementierung des Built-in-Ausführungsprozessors von [Altova StyleVision](#). Das Produkt fungiert als Modul von [FlowForce Server](#) von Altova und ist auch als [eigenständiges Server-Produkt](#)<sup>8</sup> erhältlich.

StyleVision Server führt Transformationspakete aus, die auf einem [FlowForce Server](#) bereitgestellt wurden. Diese Transformationen werden auf Basis verschiedener programmierbarer Zeit-Trigger, Datei-Trigger oder Remote Trigger von [FlowForce Server](#) gestartet. Zusätzlich dazu können StyleVision Server-Funktionen [über die Befehlszeile](#)<sup>8</sup> aufgerufen werden.

## Systemanforderungen, Installation und Lizenzierung

StyleVision Server läuft auf den folgenden Betriebssystemen:

### ▼ Windows

Windows 7 SP1 mit Plattformupdate, Windows 8, Windows 10

### ▼ Windows Server

Windows Server 2008 R2 SP1 mit Plattformupdate oder höher

### ▼ Linux

- CentOS 7 oder neuer
- RedHat 7 oder neuer
- Debian 8 oder neuer
- Ubuntu 16.04 LTS oder neuer

Um die Applikation installieren und ausführen zu können, werden die folgenden Bibliotheken benötigt. Falls die unten angeführten Pakete auf Ihrem Linux-Rechner noch nicht zur Verfügung stehen, führen Sie die Befehl `yum` (oder ggf. `apt-get`) aus, um sie zu installieren.

Erforderlich für	CentOS, RedHat	Debian	Ubuntu
<b>LicenseServer</b>	krb5-libs	libgssapi-krb5-2	libgssapi-krb5-2
<b>StyleVision Server</b>	qt5-qtbase-gui, krb5-libs	libqt5gui5, libgssapi-krb5-2	libqt5gui5, libgssapi-krb5-2

**Anmerkung:** Wenn Sie beabsichtigen, die Diagrammfunktionalität von Altova zu verwenden, muss auf Ihrem System mindestens eine Schriftart installiert sein, damit die Diagramme korrekt dargestellt werden können. Installierte Schriftarten können Sie z.B. mit dem Befehl `fc-list` aus der [Fontconfig-Bibliothek](#) auflisten.

### ▶ macOS

macOS 10.13 oder neuer

StyleVision Server steht sowohl für 32-Bit als auch 64-Bit-Windows-Rechner zur Verfügung.

Nähere Informationen zur Installation und Lizenzierung finden Sie in den Abschnitten "Einrichten von..." für [Windows](#)<sup>10</sup>, [Linux](#)<sup>15</sup> und [macOS](#)<sup>22</sup>.

*Letzte Aktualisierung: 12.10.2020*

## 1.1 Funktionalität

StyleVision Server transformiert XML-Dateien mit Hilfe von XSLT-Stylesheets in HTML-, PDF-, RTF- und DOCX-Ausgabedokumente. Diese XSLT-Stylesheets sind in PXF-Dateien enthalten, die mit [Altova StyleVision](#), der Altova-Applikation zur Erstellung von Stylesheets, erstellt wurden.

StyleVision Server kann auf zwei Arten verwendet werden:

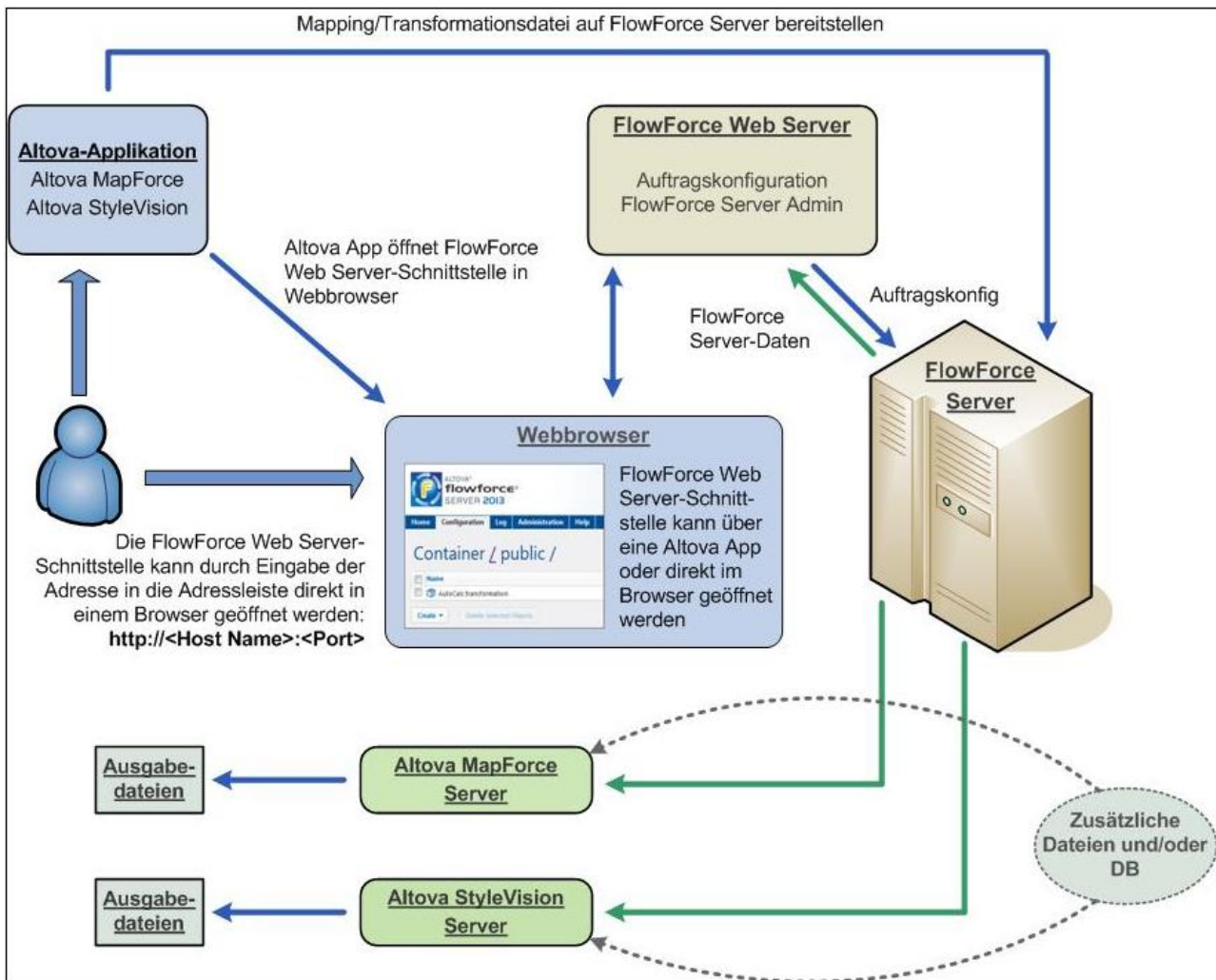
- als Teil des Altova [Altova FlowForce](#) Workflow. Nähere Informationen zu [Altova FlowForce](#) finden Sie auf der [Altova Website](#).
- als [eigenständiges Server-Produkt](#)<sup>8</sup>, das über die Befehlszeile (CLI) aufgerufen wird

Eine XML-Input-Datei und eine PXF-Datei werden an StyleVision Server übergeben, der das/die erforderliche(n) Ausgabedokument(e) erzeugt.

### 1.1.1 Im FlowForce Workflow

Ein FlowForce Server-Auftrag wird in [Altova FlowForce Server](#) erstellt. Im FlowForce Server-Auftrag ist Folgendes definiert: (i) die Inputs und Outputs einer StyleVision Server Transformation; und (ii) die Trigger, wann der Auftrag ausgeführt werden soll, z.B. jeden Tag zu einer bestimmten Zeit. Bei der Ausführung übergibt Altova FlowForce Server die Transformationsanweisungen an StyleVision Server, der dann die Transformation ausführt.

Im unten gezeigten Diagramm sehen Sie die Rolle von StyleVision Server im FlowForce Server Workflow. (Im Diagramm sehen Sie auch die Rolle von MapForce Server im Workflow, da FlowForce Server-Aufträge erstellt werden können, die [Altova MapForce](#) Mappings zur Ausführung an den [Altova MapForce Server](#) senden.)



StyleVision Server kann nicht nur über einen FlowForce Server-Auftrag, sondern auch über die Befehlszeile aufgerufen werden. Eine Beschreibung dazu finden Sie im Abschnitt [StyleVision Server-Befehlszeile](#) <sup>29</sup>.

## 1.1.2 Als eigenständiger Server

StyleVision Server kann auf Windows-, Linux- und macOS-Systemen als eigenständiges Produkt installiert werden. In dieser Version werden seine Funktionalitäten nur über die Befehlszeile aufgerufen. Eine Beschreibung dazu finden Sie im Abschnitt [StyleVision Server-Befehlszeile](#) <sup>29</sup>.



## 1.2 Einrichten von StyleVision Server

In diesem Abschnitt wird beschrieben, wie Sie StyleVision Server einrichten. Er enthält die folgenden Kapitel:

- Informationen zu [Prozessorkernen und Lizenzen](#) <sup>9</sup>
- Installation und Lizenzierung von StyleVision Server [auf Windows](#) <sup>10</sup>-Systemen
- Installation und Lizenzierung von StyleVision Server [auf Linux](#) <sup>15</sup>-Systemen
- Installation und Lizenzierung von StyleVision Server [auf macOS](#) <sup>22</sup>-Systemen

### 1.2.1 Prozessorkerne und Lizenzen

Die Lizenzierung von Altova Server-Produkten basiert auf der Anzahl der auf dem Produktrechner verfügbaren **physischen Prozessorkerne**. So hat z.B. ein Dual-Core-Prozessor zwei Prozessorkerne, ein Quad-Core-Prozessor hat vier Kerne, ein Hexa-Core-Prozessor hat sechs Kerne, usw. Die Anzahl der für ein Produkt lizenzierten Kerne muss größer oder gleich der Anzahl der auf diesem Serverrechner verfügbaren Kerne sein, unabhängig davon, ob es sich um einen physischen Rechner oder eine Virtual Machine handelt. Wenn ein Server z.B. acht Kerne hat, müssen Sie eine Lizenz für acht Kerne erwerben. Sie können Lizenzen auch kombinieren, um das Produkt für die entsprechende Anzahl von Kernen zu lizenzieren. So können z.B. anstelle einer Lizenz für 8 Kerne auch zwei Lizenzen für 4 Kerne für einen 8-Kern-Prozessor verwendet werden.

Wenn Sie einen Rechnerserver mit einer großen Anzahl von CPU-Kernen verwenden, aber nur geringe Datenmengen verarbeiten müssen, können Sie auch eine Virtual Machine erstellen, der eine geringere Anzahl an Kernen zugewiesen ist und eine Lizenz für diese Anzahl an Kernen erwerben. In diesem Fall ist die Verarbeitungsgeschwindigkeit natürlich geringer als bei Verwendung aller Kerne des Rechners.

**Anmerkung:** Jede Altova Server-Produktlizenz kann immer nur für einen Client-Rechner gleichzeitig verwendet werden, selbst wenn die Lizenzkapazität dieser Lizenz noch nicht ausgeschöpft ist. Wenn z.B. eine 10-Kern-Lizenz für einen Client-Rechner mit 6 CPU-Kernen verwendet wird, so können die verbleibenden Lizenzen für die restlichen 4 Kerne nicht gleichzeitig für einen anderen Client-Rechner verwendet werden.

#### Single-Thread-Ausführung

Wenn bei einem Altova-Server-Produkt eine Single-Thread-Ausführung möglich ist, so steht eine Option für die *Single-Thread-Ausführung* zur Verfügung. Wenn in solchen Fällen im Lizenzpool eine Altova Serverproduktlizenz für nur einen Prozessorkern verfügbar ist, können Sie einem Rechner mit mehreren Kernen diese Lizenz für einen Kern zuweisen. In diesem Fall führt der Rechner das Produkt an einem einzigen Kern aus. Dadurch verlangsamt sich die Verarbeitungsgeschwindigkeit, da kein Multi-Threading (welches bei mehreren Prozessorkernen möglich wäre) zur Verfügung steht. Das Produkt wird auf diesem Rechner im Single Thread-Modus ausgeführt.

Um einem Mehrkernrechner eine Lizenz für nur einen Kern zuzuweisen, aktivieren Sie in LicenseServer für das entsprechende Produkt das Kontrollkästchen *Limit to single thread execution*.

#### Schätzung der benötigten Prozessorkerne

Es gibt eine Reihe von externen Faktoren, die das Verarbeitungsvolumen und die Verarbeitungszeiten Ihres Servers beeinflussen (z.B. Hardware, CPU-Auslastung, Arbeitsspeicher für andere auf dem Server laufende Applikationen). Um die Leistung möglichst genau messen zu können, empfiehlt es sich, die Applikationen in

Ihrer Umgebung mit möglichst realistischen Datenvolumina und unter möglichst realistischen Bedingungen zu testen.

## 1.2.2 Einrichten unter Windows

In diesem Abschnitt werden die [Installation](#)<sup>10</sup> und [Lizenzierung](#)<sup>13</sup> von StyleVision Server auf Windows-Systemen beschrieben.

### [Installation unter Windows](#)<sup>10</sup>

- [Systemanforderungen](#)<sup>10</sup>
- [Installation von StyleVision Server](#)<sup>10</sup>
- [Altova LicenseServer](#)<sup>10</sup>
- [LicenseServer-Versionen](#)<sup>10</sup>
- [Testlizenz](#)<sup>10</sup>
- [Pfad des Applikationsordners](#)<sup>10</sup>

### [Lizenzierung unter Windows](#)<sup>13</sup>

- [Starten von ServiceController](#)<sup>13</sup>
- [Starten von LicenseServer](#)<sup>13</sup>
- [Registrieren von StyleVision Server](#)<sup>13</sup>
- [Zuweisen einer Lizenz](#)<sup>13</sup>

**Anmerkung:** Ab Version 2021 kann eine 32-Bit-Version von StyleVision Server nicht über eine 64-Bit-Version oder eine 64-Bit-Version nicht über eine 32-Bit-Version installiert werden. Sie müssen (i) die ältere Version entweder vor der Installation der neueren Version entfernen oder (ii) auf eine neuere Version, die dieselbe Bit-Version wie Ihre ältere Version hat, aktualisieren.

### 1.2.2.1 Installation unter Windows

Altova StyleVision Server ist für die Installation auf Windows-Systemen verfügbar. Im Folgenden finden Sie eine Anleitung zur Installation und Konfiguration.

#### ▼ Systemanforderungen

##### ▼ Windows

Windows 7 SP1 mit Plattformupdate, Windows 8, Windows 10

##### ▼ Windows Server

Windows Server 2008 R2 SP1 mit Plattformupdate oder höher

#### ▼ Installation von StyleVision Server

StyleVision Server kann folgendermaßen auf Windows-Systemen installiert werden:

- als separates, eigenständiges Server-Produkt namens StyleVision Server. Um StyleVision Server zu installieren, laden Sie den StyleVision Server Installer herunter und installieren Sie das Programm. Befolgen Sie die Anweisungen auf dem Bildschirm.
- Als Teil des **Altova FlowForce Server Installationspakets** [[ FlowForce Server ist eine Applikation zur Planung und zeitlichen Festsetzung verschiedener Arten von Aufträgen für Altova-Server-Produkte. Daher bietet das FlowForce Server-Paket Optionen, Altova Server-Produkte (wie <%APPNAME%>) zusammen mit FlowForce Server zu installieren. Jedes Server-Produkt wird in einem separaten Programmordner installiert. ]]. Um StyleVision Server als Teil des **FlowForce Server**-Pakets zu installieren, laden Sie den FlowForce Server Installer herunter und installieren Sie das Programm. Befolgen Sie die Anweisungen auf dem Bildschirm und stellen Sie sicher, dass Sie die Option zur Installation von StyleVision Server aktiviert haben.

Beide Produkte, [FlowForce Server](#) und StyleVision Server, können vom Altova Download Center (<http://www.altova.com/dedownload.html>) heruntergeladen werden.

Nach der Installation befindet sich die ausführbare %APPNAME%>-Datei im folgenden Ordner:

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

#### ▼ Altova LicenseServer

- Damit StyleVision Server ausgeführt werden kann, muss das Programm auf einem Altova LicenseServer auf Ihrem Netzwerk registriert sein.
- Wenn Sie StyleVision Server oder **FlowForce Server** [[ Nähere Informationen zu FlowForce Server finden Sie unter dem vorherigen Punkt, 'Installation von <%APPNAME%>' ]]. auf Windows-Systemen installieren, steht eine Option zur Verfügung, mit der Altova LicenseServer zusammen mit StyleVision Server oder **FlowForce Server** [[ Nähere Informationen zu FlowForce Server finden Sie unter dem vorherigen Punkt, 'Installation von <%APPNAME%>' ]]. heruntergeladen und installiert werden kann.
- Wenn in Ihrem Netzwerk bereits ein Altova LicenseServer installiert ist, müssen Sie keinen weiteren installieren, es sei denn, es wird eine neuere Version von Altova LicenseServer benötigt (*Siehe nächster Punkt [LicenseServer-Versionen](#).*)
- Aktivieren oder deaktivieren Sie bei der Installation von StyleVision Server oder **FlowForce Server** [[ Nähere Informationen zu FlowForce Server finden Sie unter dem vorherigen Punkt, 'Installation von <%APPNAME%>' ]]. je nach Bedarf die Option zur Installation von Altova LicenseServer. Beachten Sie die folgenden Punkte:
  - a. Wenn Sie Altova LicenseServer noch nicht installiert haben, belassen Sie die Standardeinstellungen unverändert. Der Assistent installiert daraufhin auf dem Rechner, auf dem Sie den Assistenten gestartet haben, die neueste Version von Altova LicenseServer.
  - b. Wenn Sie Altova LicenseServer noch nicht installiert haben und Altova LicenseServer auf einem anderen Rechner installieren möchten, deaktivieren Sie die Option **Altova LicenseServer auf diesem Rechner installieren** und wählen Sie anschließend **Später registrieren**. In diesem Fall müssen Sie Altova LicenseServer separat installieren und StyleVision Server später registrieren.
  - c. Wenn Altova LicenseServer auf Ihrem Rechner bereits installiert wurde, aber eine niedrigere Versionsnummer als die im Installationsassistenten angegebene hat, belassen Sie die Standardeinstellungen unverändert. In diesem Fall aktualisiert der Installationsassistent Ihre LicenseServer Version automatisch auf die im Dialogfeld angegebene. Beachten Sie, dass die vorhandene Registrierung und die vorhandenen Lizenzierungsdaten nach dem Upgrade erhalten bleiben.
  - d. Wenn Altova LicenseServer bereits auf Ihrem Rechner oder in Ihrem Netzwerk installiert wurde und dieselbe Versionsnummer wie diejenige im Assistenten hat, gehen Sie folgendermaßen vor:

- i. Deaktivieren Sie das Kontrollkästchen **Altova LicenseServer auf diesem Rechner installieren**.
- ii. Wählen Sie unter **Dieses Produkt registrieren auf** die Altova LicenseServer-Instanz, auf der Sie StyleVision Server registrieren möchten oder wählen Sie **Später registrieren**. Beachten Sie, dass Sie immer die Möglichkeit haben, **Später registrieren** auszuwählen, wenn Sie die LicenseServer-Verknüpfungen ignorieren möchten und mit der Installation von StyleVision Server fortfahren möchten.

Nähere Informationen zum Registrieren und Lizenzieren von StyleVision Server auf dem Altova LicenseServer und zur Lizenzierung finden Sie im Abschnitt [Lizenzierung unter Windows](#)<sup>13</sup>.

#### ▼ LicenseServer-Versionen

- Altova Server-Produkte müssen entweder mit der zum Zeitpunkt der Release von StyleVision Server aktuellen Version von LicenseServer oder einer späteren Version von LicenseServer lizenziert werden.
- Welche Version von LicenseServer für eine bestimmte Version von StyleVision Server geeignet ist, wird während der Installation von StyleVision Server angezeigt. Sie können diese Version von LicenseServer zusammen mit StyleVision Server installieren oder Sie können LicenseServer separat installieren.
- Bevor Sie eine neuere Version von LicenseServer installieren, muss eine eventuell vorhandene ältere Version deinstalliert werden. Das Installationsprogramm von LicenseServer führt dies bei Vorhandensein einer älteren Version automatisch durch.
- LicenseServer-Versionen sind rückwärts kompatibel. Sie funktionieren auch mit älteren Versionen von StyleVision Server.
- Wenn Sie eine neue Version von StyleVision Server installieren und Ihre installierte Version von LicenseServer älter als der dafür vorgesehene LicenseServer ist, installieren Sie die neueste auf der Altova Website verfügbare Version.
- Bei der Deinstallation werden alle Registrierungs- und Lizenzierungsinformationen aus der älteren LicenseServer-Version in einer Datenbank auf Ihrem Server gespeichert. Diese Daten werden automatisch in die neuere Version importiert.
- Sie finden die Versionsnummer des aktuell installierten LicenseServer am unteren Rand der LicenseServer Konfigurationsseite (alle Register).

**Aktuelle Version: 3.5**

#### ▼ Testlizenz

Bei der Installation haben Sie die Option, eine 30-Tage-Testlizenz für StyleVision Server anzufordern. Nach Absendung der Anforderung erhalten Sie an die von Ihnen registrierte E-Mail-Adresse eine Testlizenz zugesandt.

#### ▼ Pfad des Applikationsordners

Die Applikation wird im folgenden Ordner installiert:

Windows 7, 8, 10	C:\Programme\Altova\
------------------	----------------------

32-Bit-Version auf 64-Bit OS

C:\Programme (x86)\Altova\

### 1.2.2.2 Lizenzierung unter Windows

StyleVision Server muss auf dem Altova LicenseServer lizenziert werden. Die Lizenzierung erfolgt in zwei Schritten:

1. **Registrieren von StyleVision Server** auf LicenseServer. Die Registrierung erfolgt über StyleVision Server.
2. **Zuweisen einer Lizenz** zu StyleVision Server über LicenseServer. Laden Sie die neueste Version von LicenseServer von der [Altova Website](#) herunter und installieren Sie diese auf Ihrem lokalen Rechner oder auf einem Rechner in Ihrem Netzwerk.

Im Folgenden sind diese Schritte unten kurz beschrieben. Ausführliche Informationen dazu finden Sie im [Benutzerhandbuch zu LicenseServer](#) auf der [Altova Website](#).

#### ▼ Starten von ServiceController

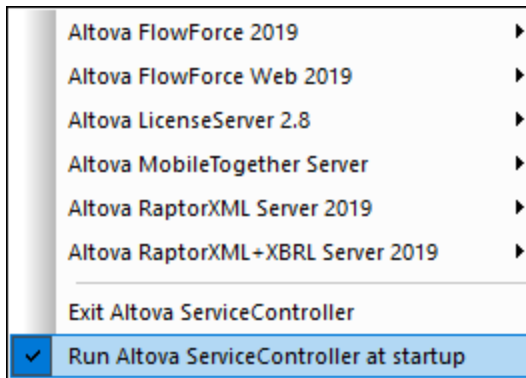
Der ServiceController wird gestartet, damit Sie damit LicenseServer starten können.

Der Altova ServiceController (in der Folge ServiceController genannt) ist eine Applikation, mit der Sie Altova-Dienste **auf Windows-Systemen** starten, beenden und konfigurieren können.

ServiceController wird mit Altova LicenseServer und *als Dienst installierten Altova Server-Produkten*(FlowForce Server, RaptorXML(+XBRL) Server und Mobile Together Server) installiert und kann durch Klicken auf **Start | Altova LicenseServer | Altova ServiceController** gestartet werden. (Dieser Befehl steht auch in den **Startmenüordnern** der *Altova Server-Produkten, die als Dienst installiert wurden*(FlowForce Server, RaptorXML(+XBRL) Server und Mobile Together Server), zur Verfügung). Nachdem der ServiceController gestartet wurde, kann er über die Task-Leiste aufgerufen werden (*Abbildung unten*).

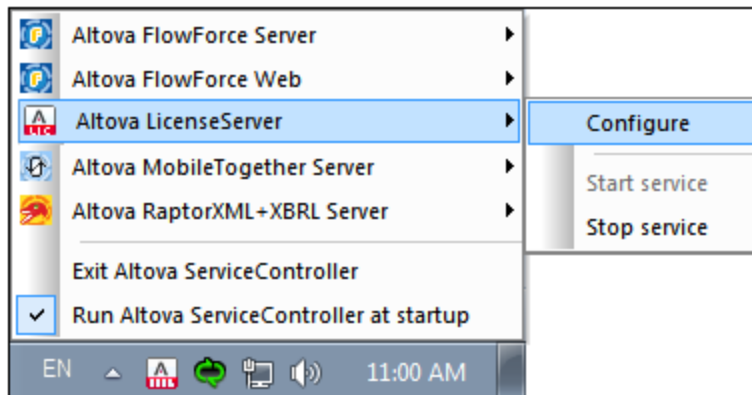


Damit der ServiceController automatisch ausgeführt wird, nachdem sich der Benutzer im System angemeldet hat, klicken Sie in der Task-Leiste auf das **ServiceController**-Symbol, um das **ServiceController**-Menü (*Abbildung unten*) aufzurufen und aktivieren Sie anschließend den Befehl **Run Altova ServiceController at Startup**. (Dieser Befehl ist standardmäßig aktiv). Um den ServiceController zu beenden, klicken Sie in der Task-Leiste auf das **ServiceController**-Symbol und klicken Sie im Menü, das daraufhin angezeigt wird (*Abbildung unten*) auf **Exit Altova ServiceController**.



#### ▼ Starten von LicenseServer

Um LicenseServer zu starten, klicken Sie in der Task-Leiste auf das Altova **ServiceController**-Symbol und platzieren Sie den Mauszeiger im angezeigten Menü über **Altova LicenseServer** (*Abbildung unten*) und wählen Sie im Untermenü von Altova LicenseServer den Befehl **Start Service**. Wenn LicenseServer bereits ausgeführt wird, ist die Option *Start Service* deaktiviert.



#### ▼ Registrieren von StyleVision Server

Um StyleVision Server über die Befehlszeilenschnittstelle zu registrieren, verwenden Sie den Befehl `licenseserver`:

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

Wenn z.B. `localhost` der Name des Servers ist, auf dem LicenseServer installiert ist:

```
styleVisionServer licenseserver localhost
```

Wenn StyleVision Server im Rahmen der [FlowForce Server](#) Installation installiert wurde, wird bei der Registrierung von FlowForce Server auch StyleVision Server automatisch auf LicenseServer registriert: (i) Starten Sie Altova FlowForce Web über den ServiceController (*siehe vorheriger Punkt*); (ii) Geben Sie Ihr Passwort für den Zugriff auf die Setup-Seite ein; (iii) Wählen Sie den LicenseServer-Namen oder die

Adresse aus und klicken Sie auf **Auf LicenseServer registrieren**. Nähere Informationen dazu finden Sie unter Registrieren von FlowForce Server.

Nachdem Sie das Produkt registriert haben, können Sie StyleVision Server über die Konfigurationsseite von LicenseServer über das Register Client Management eine Lizenz zuweisen.

#### ▼ Zuweisen einer Lizenz

Nachdem Sie StyleVision Server erfolgreich registriert haben, wird die Applikation auf der Konfigurationsseite von LicenseServer auf dem Register "Client Management" aufgelistet. Sie können nun zu LicenseServer wechseln und StyleVision Server eine Lizenz zuweisen.

Die Lizenzierung von Altova Server-Produkten basiert auf der Anzahl der auf dem Produktrechner verfügbaren Prozessorkerne. So hat z.B. ein Dual-Core-Prozessor zwei Prozessorkerne, ein Quad-Core-Prozessor hat vier Kerne, ein Hexa-Core-Prozessor hat sechs Kerne, usw. Die Anzahl der für ein Produkt lizenzierten Kerne muss größer oder gleich der Anzahl der auf diesem Serverrechner verfügbaren Kerne sein, unabhängig davon, ob es sich um einen physischen Rechner oder eine Virtual Machine handelt. Wenn ein Server z.B. acht Kerne hat (ein Octa-Core-Prozessor), müssen Sie mindestens eine Lizenz für acht Kerne erwerben. Sie können Lizenzen auch kombinieren, um das Produkt für die entsprechende Anzahl von Kernen zu lizenzieren. So können z.B. anstelle einer Lizenz für 8 Kerne auch zwei Lizenzen für 4 Kerne für einen 8-Kern-Prozessor verwendet werden.

Wenn Sie einen Server-Rechner mit einer großen Anzahl von CPU-Kernen verwenden, aber nur geringe Datenmengen verarbeiten müssen, können Sie auch eine Virtual Machine erstellen, der eine geringere Anzahl an Kernen zugewiesen ist und eine Lizenz für diese Anzahl an Kernen erwerben. In diesem Fall ist die Verarbeitungsgeschwindigkeit natürlich geringer als bei Verwendung aller Kerne des Rechners.

**Anmerkung:** Jede Altova Server-Produktlizenz kann immer nur für einen Client-Rechner, d.h. einen Rechner, auf dem das Altova Server-Produkt installiert ist, gleichzeitig verwendet werden, selbst wenn die Lizenzkapazität dieser Lizenz noch nicht ausgeschöpft ist. Wenn z.B. eine 10-Kern-Lizenz für einen Client-Rechner mit 6 CPU-Kernen verwendet wird, so können die verbleibenden Lizenzen für die restlichen 4 Kerne nicht gleichzeitig für einen anderen Client-Rechner verwendet werden.

## 1.2.3 Einrichten unter Linux

In diesem Abschnitt werden die [Installation](#)<sup>16</sup> und [Lizenzierung](#)<sup>19</sup> von StyleVision Server auf Linux-Systemen (Debian, Ubuntu, CentOS, RedHat) beschrieben.

### [Installation unter Linux](#)<sup>16</sup>

- [Systemanforderungen](#)<sup>16</sup>
- [Deinstallieren alter Versionen von Altova Server-Produkten](#)<sup>16</sup>
- [Herunterladen des Linux-Pakets](#)<sup>16</sup>
- [Installieren von StyleVision Server](#)<sup>16</sup>
- [Altova LicenseServer](#)<sup>16</sup>
- [LicenseServer-Versionen](#)<sup>16</sup>

### [Lizenzierung unter Linux](#)<sup>19</sup>

- [Starten von LicenseServer](#) <sup>19</sup>
- [Registrieren von StyleVision Server](#) <sup>19</sup>
- [Zuweisen einer Lizenz](#) <sup>19</sup>

### [Anmerkungen zur Umgebung](#) <sup>20</sup>

#### 1.2.3.1 Installation unter Linux

StyleVision Server steht für die Installation auf Linux-Systemen zur Verfügung. Im Folgenden finden Sie eine Beschreibung der Installation und Konfiguration des Produkts.

##### ▼ Systemanforderungen

###### ▼ *Linux*

- CentOS 7 oder neuer
- RedHat 7 oder neuer
- Debian 8 oder neuer
- Ubuntu 16.04 LTS oder neuer

Um die Applikation installieren und ausführen zu können, werden die folgenden Bibliotheken benötigt. Falls die unten angeführten Pakete auf Ihrem Linux-Rechner noch nicht zur Verfügung stehen, führen Sie die Befehl `yum` (oder ggf. `apt-get`) aus, um sie zu installieren.

Erforderlich für	CentOS, RedHat	Debian	Ubuntu
<b>LicenseServer</b>	krb5-libs	libgssapi-krb5-2	libgssapi-krb5-2
<b>StyleVision Server</b>	qt5-qtbase-gui, krb5-libs	libqt5gui5, libgssapi-krb5-2	libqt5gui5, libgssapi-krb5-2

**Anmerkung:** Wenn Sie beabsichtigen, die Diagrammfunktionalität von Altova zu verwenden, muss auf Ihrem System mindestens eine Schriftart installiert sein, damit die Diagramme korrekt dargestellt werden können. Installierte Schriftarten können Sie z.B. mit dem Befehl `fc-list` aus der [Fontconfig-Bibliothek](#) auflisten.

##### ▼ FlowForce Server-Integration

Wenn Sie StyleVision Server zusammen mit FlowForce Server installieren, sollten Sie zuerst FlowForce Server installieren. Führen Sie andernfalls nach Installation von sowohl StyleVision Server als auch FlowForce Server den folgenden Befehl aus:

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

Dieser Befehl kopiert die `.tool`-Datei aus dem Verzeichnis `/etc` von StyleVision Server in das FlowForce Server `/tools`-Verzeichnis. Die Datei `.tool` wird von FlowForce Server benötigt; sie enthält den Pfad zur ausführbaren StyleVision Server Datei. Sie müssen diesen Befehl nicht ausführen, wenn Sie FlowForce Server vor StyleVision Server installieren.



#### ▼ Deinstallieren einer alten Version von Altova Server-Produkten

Wenn Sie eine Vorgängerversion deinstallieren müssen, gehen Sie folgendermaßen vor. Überprüfen Sie in der Linux-Befehlszeile (Command Line Interface = CLI), welche Altova Server-Produkte installiert sind. Verwenden Sie dazu den Befehl:

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

Falls StyleVision Server nicht installiert ist, beginnen Sie mit der Installation, wie dies weiter unten unter *Installieren von StyleVision Server* dokumentiert ist.

Falls StyleVision Server installiert ist und Sie eine neuere Version von StyleVision Server installieren möchten, deinstallieren Sie die alte Version mit dem Befehl:

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

Falls Sie eine neuere Version von Altova LicenseServer installieren möchten, deinstallieren Sie die alte Version mit dem Befehl:

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

#### ▼ Herunterladen des Linux-Pakets

Auf der [Altova Website](#) stehen StyleVision Server-Installationspakete für die folgenden Linux-Systeme zur Verfügung.

Distribution	Paketerweiterung
Debian	.deb
Ubuntu	.deb
CentOS	.rpm
RedHat	.rpm

Nachdem Sie das Linux-Paket von der [Altova Website](#) heruntergeladen haben, kopieren Sie das Paket in ein beliebiges Verzeichnis auf dem Linux-System. Da Sie zum Ausführen von StyleVision Server einen Altova LicenseServer benötigen, sollten Sie gleichzeitig mit StyleVision Server auch LicenseServer von der [Altova Website](#) herunterladen, anstatt LicenseServer erst später herunterzuladen.

#### ▼ Installieren von StyleVision Server

Wechseln Sie in einem Terminal-Fenster zu dem Verzeichnis, in das Sie das Linux-Paket kopiert haben. Wenn Sie es z.B. in ein Benutzerverzeichnis namens `MyAltova` (das sich z.B. im Verzeichnis `/home/User` befindet) kopiert haben, dann wechseln Sie folgendermaßen zu diesem Verzeichnis:

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

Installieren Sie StyleVision Server mit dem folgenden Befehl:

```
[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
```

Das StyleVision Server-Paket wird im folgenden Ordner installiert:

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

#### ▼ Altova LicenseServer

Damit ein Altova Server-Produkt - darunter auch StyleVision Server - ausgeführt werden kann, muss es auf einem Altova LicenseServer auf Ihrem Netzwerk registriert sein.

Auf Linux-Systemen muss Altova LicenseServer separat installiert werden. Laden Sie Altova LicenseServer von der [Altova Website](#) herunter und kopieren Sie das Paket in ein beliebiges Verzeichnis auf dem Linux-System. Installieren Sie es genau wie StyleVision Server (*siehe vorheriger Schritt*).

```
[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
```

Das LicenseServer-Paket wird im folgenden Ordner installiert:

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

Eine Anleitung, wie Sie StyleVision Server auf dem Altova LicenseServer registrieren und lizenzieren, finden Sie im Abschnitt [Lizenzierung unter Linux](#)<sup>19</sup>. Nähere Informationen dazu finden Sie außerdem in der [Dokumentation zu LicenseServer](#).

#### ▼ LicenseServer-Versionen

- Altova Server-Produkte müssen entweder mit der zum Zeitpunkt der Release von StyleVision Server aktuellen Version von LicenseServer oder einer späteren Version von LicenseServer lizenziert werden.
- Welche Version von LicenseServer für eine bestimmte Version von StyleVision Server geeignet ist, wird während der Installation von StyleVision Server angezeigt. Sie können diese Version von LicenseServer zusammen mit StyleVision Server installieren oder Sie können LicenseServer separat installieren.
- Bevor Sie eine neuere Version von LicenseServer installieren, muss eine eventuell vorhandene ältere Version deinstalliert werden. Das Installationsprogramm von LicenseServer führt dies bei Vorhandensein einer älteren Version automatisch durch.
- LicenseServer-Versionen sind rückwärts kompatibel. Sie funktionieren auch mit älteren Versionen von StyleVision Server.
- Wenn Sie eine neue Version von StyleVision Server installieren und Ihre installierte Version von LicenseServer älter als der dafür vorgesehene LicenseServer ist, installieren Sie die neueste auf der Altova Website verfügbare Version.
- Bei der Deinstallation werden alle Registrierungs- und Lizenzierungsinformationen aus der älteren

LicenseServer-Version in einer Datenbank auf Ihrem Server gespeichert. Diese Daten werden automatisch in die neuere Version importiert.

- Sie finden die Versionsnummer des aktuell installierten LicenseServer am unteren Rand der LicenseServer Konfigurationsseite (alle Register).

**Aktuelle Version: 3.5**

### 1.2.3.2 Lizenzierung unter Linux

StyleVision Server muss auf dem Altova LicenseServer lizenziert werden. Die Lizenzierung erfolgt in zwei Schritten:

1. **Registrieren von StyleVision Server** auf LicenseServer. Die Registrierung erfolgt über StyleVision Server.
2. **Zuweisen einer Lizenz** zu StyleVision Server über LicenseServer. Laden Sie die neueste Version von LicenseServer von der [Altova Website](#) herunter und installieren Sie diese auf Ihrem lokalen Rechner oder auf einem Rechner in Ihrem Netzwerk.

Im Folgenden sind diese Schritte unten kurz beschrieben. Ausführliche Informationen dazu finden Sie im [Benutzerhandbuch zu LicenseServer](#) auf der [Altova Website](#).

#### ▼ Starten von LicenseServer

Damit StyleVision Server auf dem LicenseServer korrekt registriert und lizenziert werden kann, muss LicenseServer als Daemon im Netzwerk ausgeführt werden. Starten Sie LicenseServer mit dem folgenden Befehl als Daemon:

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

Um LicenseServer zu beenden, ersetzen Sie `start` in den obigen Befehlen durch `stop`. Beispiel:

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

#### ▼ Registrieren von StyleVision Server

Um StyleVision Server über die Befehlszeilenschnittstelle zu registrieren, verwenden Sie den Befehl `licenseserver`:

```
sudo /opt/Altova/StyleVisionServer2021/bin/stylevisionserver licenseserver
[options] ServerName-Or-IP-Address
```

Wenn z.B. `localhost` der Name des Servers ist, auf dem LicenseServer installiert ist:

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

Im obigen Befehl ist `localhost` der Name des Servers, auf dem LicenseServer installiert ist. Beachten Sie außerdem, dass der Pfad der ausführbaren StyleVision Server Datei folgendermaßen lautet:

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

Nachdem Sie StyleVision Server erfolgreich registriert haben, gehen Sie zum Register "Client Management" der LicenseServer-Konfigurationsseite, um StyleVision Server eine Lizenz zuzuweisen.

#### ▼ Zuweisen einer Lizenz

Nachdem Sie StyleVision Server erfolgreich registriert haben, wird das Produkt auf der Konfigurationsseite von LicenseServer auf dem Register Client Management aufgelistet. Sie können LicenseServer nun aufrufen und StyleVision Server eine Lizenz zuweisen.

Die Lizenzierung von Altova Server-Produkten basiert auf der Anzahl der auf dem Produktrechner verfügbaren Prozessorkerne. So hat z.B. ein Dual-Core-Prozessor zwei Prozessorkerne, ein Quad-Core-Prozessor hat vier Kerne, ein Hexa-Core-Prozessor hat sechs Kerne, usw. Die Anzahl der für ein Produkt lizenzierten Kerne muss größer oder gleich der Anzahl der auf diesem Serverrechner verfügbaren Kerne sein, unabhängig davon, ob es sich um einen physischen Rechner oder eine Virtual Machine handelt. Wenn ein Server z.B. acht Kerne hat (ein Octa-Core-Prozessor), müssen Sie mindestens eine Lizenz für acht Kerne erwerben. Sie können Lizenzen auch kombinieren, um das Produkt für die entsprechende Anzahl von Kernen zu lizenzieren. So können z.B. anstelle einer Lizenz für 8 Kerne auch zwei Lizenzen für 4 Kerne für einen 8-Kern-Prozessor verwendet werden.

Wenn Sie einen Server-Rechner mit einer großen Anzahl von CPU-Kernen verwenden, aber nur geringe Datenmengen verarbeiten müssen, können Sie auch eine Virtual Machine erstellen, der eine geringere Anzahl an Kernen zugewiesen ist und eine Lizenz für diese Anzahl an Kernen erwerben. In diesem Fall ist die Verarbeitungsgeschwindigkeit natürlich geringer als bei Verwendung aller Kerne des Rechners.

**Anmerkung:** Jede Altova Server-Produktlizenz kann immer nur für einen Client-Rechner, d.h. einen Rechner, auf dem das Altova Server-Produkt installiert ist, gleichzeitig verwendet werden, selbst wenn die Lizenzkapazität dieser Lizenz noch nicht ausgeschöpft ist. Wenn z.B. eine 10-Kern-Lizenz für einen Client-Rechner mit 6 CPU-Kernen verwendet wird, so können die verbleibenden Lizenzen für die restlichen 4 Kerne nicht gleichzeitig für einen anderen Client-Rechner verwendet werden.

### 1.2.3.3 Anmerkungen zur Umgebung

#### Ordner

Im Folgenden finden Sie eine Liste wichtiger Ordner in Ihrer StyleVision Server-Installation.

##### ☐ Root-Ordner der Installation

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

#### ▣ Lizenzdateien

```
/var/opt/Altova/StyleVisionServer
```

#### ▣ Umgebungseinstellungen

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

Die Datei für die Umgebungseinstellungen (normalerweise mit dem Namen `jdbc.sh`) wird beim Systemstart ausgeführt. Die Definitionen darin müssen entsprechend Ihrer jeweiligen Umgebung konfiguriert sein. Der obige Pfad dient nur als allgemeines Beispiel.

**Anmerkung:** Die Datei für die Umgebungseinstellungen definiert die Variablen für **alle Benutzer** im System, seien Sie daher vorsichtig, wenn Sie die Einstellungen ändern. Wenn Sie z.B. einen Klassenpfad in dieser Datei ändern, wird diese Änderung systemweit wirksam. Wenn Sie nur Änderungen für StyleVision Server vornehmen möchten, sollten Sie eventuell eine Unit-Datei verwenden (Eine Beschreibung dazu finden Sie weiter unten unter *JDBC-Verbindungen*).

## Datenbankverbindungen

Unter Linux werden die folgenden Datenbankverbindungen unterstützt:

- JDBC - Sie können JDBC für alle unterstützten Datenbanken mit Ausnahme von Microsoft Access verwenden
- Native Verbindungen - stehen derzeit für SQLite- und PostgreSQL-Datenbanken zur Verfügung

Beachten Sie die folgenden Punkte, wenn Sie JDBC verwenden:

- Das Java Runtime Environment oder SDK muss installiert sein
- Die JDBC-Treiber für die Zieldatenbank müssen installiert sein
- Die folgenden Umgebungsvariablen müssen für Ihre Umgebung richtig konfiguriert werden:
  - `CLASSPATH`: Um die jar-Dateien für die Verbindung zur JDBC-Datenbank zu finden. Die jar-Dateien, können entweder (i) in ein ausführbares Skript (wie `jdbc.sh`), das bei Systemstart ausgeführt wird, eingegeben werden oder (ii) in eine Unit-Datei, die ausgeführt wird, wenn StyleVision Server als Dienst gestartet wird. Die Verwendung einer Unit-Datei zur Definition der jar-Dateien hat den Vorteil, dass die für die JDBC-Verbindungen von StyleVision Server benötigten jar-Dateien gefunden werden, ohne dass Sie die bestehende Systemkonfiguration ändern müssen. Unten sehen Sie den Inhalt einer Unit-Datei.
  - `PATH`: Um die JRE zu finden. Nicht unbedingt erforderlich, hängt von der Installation ab
  - `JAVA_HOME`: Falls nötig - abhängig von der Installation.

## Liste wichtiger Dateien

Das folgende Shell-Skript (oder Unit-Datei) wird in den Ordner `/opt/Altova/StyleVisionServer/etc` kopiert, damit bereits vorhandene Konfigurationsdateien nicht überschrieben werden. Siehe auch Abschnitt *JDBC-Verbindungen* weiter oben. Nehmen Sie die nötigen Änderungen vor, wie erforderlich. Die blau markierten Teile sind umgebungsspezifisch und müssen an Ihre individuelle Umgebung angepasst werden.

#### ▣ Shell-Skript (Unit-Datei)

```
#!/bin/sh - environment -
```

```

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

```

## 1.2.4 Einrichten auf macOS

In diesem Abschnitt werden die [Installation](#)<sup>22</sup> und [Lizenzierung](#)<sup>25</sup> von StyleVision Server auf macOS-Systemen beschrieben.

### [Installation unter macOS](#)<sup>22</sup>

- [Systemanforderungen](#)<sup>22</sup>
- [Deinstallieren alter Versionen von Altova Server-Produkten](#)<sup>22</sup>
- [Herunterladen des macOS-Pakets](#)<sup>22</sup>
- [Installieren von StyleVision Server](#)<sup>22</sup>
- [Altova LicenseServer](#)<sup>22</sup>
- [LicenseServer-Versionen](#)<sup>22</sup>

### [Lizenzierung unter macOS](#)<sup>25</sup>

- [Starten von LicenseServer](#)<sup>25</sup>
- [Registrieren von StyleVision Server](#)<sup>25</sup>
- [Zuweisen einer Lizenz](#)<sup>25</sup>

### [Anmerkungen zur Umgebung](#)<sup>26</sup>

#### 1.2.4.1 Installation auf macOS

StyleVision Server steht für die Installation auf macOS-Systemen zur Verfügung. Im Folgenden finden Sie eine Beschreibung der Installation und Konfiguration.

##### ▼ Systemanforderungen

###### ▶ *macOS*

macOS 10.13 oder neuer

#### ▼ FlowForce Server-Integration

Wenn Sie StyleVision Server zusammen mit FlowForce Server installieren, sollten Sie zuerst FlowForce Server installieren. Falls Sie APPNAME%> vor FlowForce Server installieren, führen Sie nach Installation von sowohl StyleVision Server als auch FlowForce Server den folgenden Befehl aus:

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

Dieser Befehl kopiert die **.tool**-Datei aus dem Verzeichnis **/etc** von StyleVision Server in das FlowForce Server **/tools**-Verzeichnis. Die Datei **.tool** wird von FlowForce Server benötigt; sie enthält den Pfad zur ausführbaren StyleVision Server Datei. Sie müssen diesen Befehl nicht ausführen, wenn Sie FlowForce Server vor StyleVision Server installieren.

#### ▼ Deinstallieren alter Versionen von Altova Server-Produkten

Klicken Sie im Finder im Applikationsordner mit der rechten Maustaste auf das StyleVision Server-Symbol und wählen Sie den Befehl **Move to Trash**. Die Applikation wird daraufhin in den Papierkorb verschoben. Sie müssen die Applikation jetzt noch aus dem Ordner `usr` entfernen. Verwenden Sie dazu den folgenden Befehl:

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

Wenn Sie eine ältere Version von Altova LicenseServer deinstallieren müssen, stoppen Sie den Dienst zuerst mit dem folgenden Befehl:

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

Um zu überprüfen, ob der Dienst gestoppt wurde, öffnen Sie den Activity Monitor im Finder und vergewissern Sie sich, dass sich LicenseServer nicht auf der Liste befindet. Fahren Sie anschließend mit der Deinstallation fort, wie oben für StyleVision Server beschrieben.

#### ▼ Herunterladen der Disk Image-Datei

Laden Sie die Disk Image (.dmg)-Datei von der Altova-Website herunter (<http://www.altova.com/de/download.html>).

#### ▼ Installieren von StyleVision Server

Klicken Sie auf die heruntergeladene Disk Image (.dmg)-Datei, um sie zu öffnen. Dadurch wird der StyleVision Server-Installer als neues virtuelles Laufwerk auf Ihrem Computer angezeigt. Doppelklicken Sie auf diesem neuen virtuellen Laufwerk auf das Installer-Paket (.pkg). Befolgen Sie die selbsterklärenden Anweisungen des Installationsassistenten, in dem Sie auch die Lizenzvereinbarung akzeptieren müssen, bevor Sie mit der Installation fortfahren können. Um das Laufwerk nach der Installation wieder auszuwerfen, klicken Sie auf die rechte Maustaste und wählen Sie die Befehl **Auswerfen**.

Das StyleVision Server-Paket wird im folgenden Ordner installiert:

```
/usr/local/Altova/StyleVisionServer2021 (Applikationsbinärdateien)  
/var/Altova/StyleVisionServer (Datendateien: Datenbank und Logs)
```

### ▼ Altova LicenseServer

Damit ein Altova Server-Produkt - darunter auch StyleVision Server - ausgeführt werden kann, muss es auf einem Altova LicenseServer auf Ihrem Netzwerk lizenziert sein.

Das Altova LicenseServer-Installationspaket finden Sie auf dem virtuellen Laufwerk, das Sie im vorhergehenden Schritt angelegt haben. Um Altova LicenseServer zu installieren, doppelklicken Sie auf das Installer-Paket auf dem virtuellen Laufwerk und befolgen Sie die Anweisungen auf dem Bildschirm. Damit die Installation fortgesetzt wird, müssen Sie die Lizenzvereinbarung akzeptieren.

Altova LicenseServer kann auch separat von der Altova Website heruntergeladen und installiert werden (<http://www.altova.com/de/download.html>).

Das LicenseServer-Paket wird im folgenden Ordner installiert:

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

Informationen zum Registrieren von StyleVision Server auf dem Altova LicenseServer und zum Lizenzieren des Produkts finden Sie im Abschnitt [Lizenzierung unter macOS](#) <sup>25</sup>.

### ▼ LicenseServer-Versionen

- Altova Server-Produkte müssen entweder mit der zum Zeitpunkt der Release von StyleVision Server aktuellen Version von LicenseServer oder einer späteren Version von LicenseServer lizenziert werden.
- Welche Version von LicenseServer für eine bestimmte Version von StyleVision Server geeignet ist, wird während der Installation von StyleVision Server angezeigt. Sie können diese Version von LicenseServer zusammen mit StyleVision Server installieren oder Sie können LicenseServer separat installieren.
- Bevor Sie eine neuere Version von LicenseServer installieren, muss eine eventuell vorhandene ältere Version deinstalliert werden. Das Installationsprogramm von LicenseServer führt dies bei Vorhandensein einer älteren Version automatisch durch.
- LicenseServer-Versionen sind rückwärts kompatibel. Sie funktionieren auch mit älteren Versionen von StyleVision Server.
- Wenn Sie eine neue Version von StyleVision Server installieren und Ihre installierte Version von LicenseServer älter als der dafür vorgesehene LicenseServer ist, installieren Sie die neueste auf der Altova Website verfügbare Version.
- Bei der Deinstallation werden alle Registrierungs- und Lizenzierungsinformationen aus der älteren LicenseServer-Version in einer Datenbank auf Ihrem Server gespeichert. Diese Daten werden automatisch in die neuere Version importiert.
- Sie finden die Versionsnummer des aktuell installierten LicenseServer am unteren Rand der LicenseServer Konfigurationsseite (alle Register).

**Aktuelle Version: 3.5**



## 1.2.4.2 Lizenzierung auf macOS

StyleVision Server muss auf dem Altova LicenseServer lizenziert werden. Die Lizenzierung erfolgt in zwei Schritten:

1. **Registrieren von StyleVision Server** auf LicenseServer. Die Registrierung erfolgt über StyleVision Server.
2. **Zuweisen einer Lizenz** zu StyleVision Server über LicenseServer. Laden Sie die neueste Version von LicenseServer von der [Altova Website](#) herunter und installieren Sie diese auf Ihrem lokalen Rechner oder auf einem Rechner in Ihrem Netzwerk.

Im Folgenden sind diese Schritte unten kurz beschrieben. Ausführliche Informationen dazu finden Sie im [Benutzerhandbuch zu LicenseServer](#) auf der [Altova Website](#).

### ▼ Starten von LicenseServer

Damit StyleVision Server auf dem LicenseServer korrekt registriert und lizenziert werden kann, muss LicenseServer als Daemon im Netzwerk ausgeführt werden. Starten Sie LicenseServer mit dem folgenden Befehl als Daemon:

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

Um LicenseServer zu beenden, ersetzen Sie `load` im obigen Befehl durch `unload`. Beispiel:

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

### ▼ Registrieren von StyleVision Server

Um StyleVision Server über die Befehlszeilenschnittstelle zu registrieren, verwenden Sie den Befehl `licenseserver`:

```
sudo /opt/Altova/StyleVisionServer2021/bin/stylevisionserver licenseserver  
[options] ServerName-Oder-IP-Adresse
```

Wenn z.B. `localhost` der Name des Servers ist, auf dem LicenseServer installiert ist:

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

Im obigen Befehl ist `localhost` der Name des Servers, auf dem LicenseServer installiert ist. Beachten Sie außerdem, dass der Pfad der ausführbaren StyleVision Server Datei folgendermaßen lautet:

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

Nachdem Sie StyleVision Server erfolgreich registriert haben, können Sie dem Produkt nun über das Register "Client Management" der LicenseServer-Konfigurationsseite eine Lizenz zuweisen.

### ▼ Zuweisen einer Lizenz

Nachdem Sie StyleVision Server erfolgreich registriert haben, wird das Produkt auf der Konfigurationsseite von LicenseServer auf dem Register "Client Management" aufgelistet. Sie können LicenseServer nun

aufrufen und StyleVision Server eine Lizenz zuweisen.

Die Lizenzierung von Altova Server-Produkten basiert auf der Anzahl der auf dem Produktrechner verfügbaren Prozessorkerne. So hat z.B. ein Dual-Core-Prozessor zwei Prozessorkerne, ein Quad-Core-Prozessor hat vier Kerne, ein Hexa-Core-Prozessor hat sechs Kerne, usw. Die Anzahl der für ein Produkt lizenzierten Kerne muss größer oder gleich der Anzahl der auf diesem Serverrechner verfügbaren Kerne sein, unabhängig davon, ob es sich um einen physischen Rechner oder eine Virtual Machine handelt. Wenn ein Server z.B. acht Kerne hat (ein Octa-Core-Prozessor), müssen Sie mindestens eine Lizenz für acht Kerne erwerben. Sie können Lizenzen auch kombinieren, um das Produkt für die entsprechende Anzahl von Kernen zu lizenzieren. So können z.B. anstelle einer Lizenz für 8 Kerne auch zwei Lizenzen für 4 Kerne für einen 8-Kern-Prozessor verwendet werden.

Wenn Sie einen Server-Rechner mit einer großen Anzahl von CPU-Kernen verwenden, aber nur geringe Datenmengen verarbeiten müssen, können Sie auch eine Virtual Machine erstellen, der eine geringere Anzahl an Kernen zugewiesen ist und eine Lizenz für diese Anzahl an Kernen erwerben. In diesem Fall ist die Verarbeitungsgeschwindigkeit natürlich geringer als bei Verwendung aller Kerne des Rechners.

**Anmerkung:** Jede Altova Server-Produktlizenz kann immer nur für einen Client-Rechner, d.h. einen Rechner, auf dem das Altova Server-Produkt installiert ist, gleichzeitig verwendet werden, selbst wenn die Lizenzkapazität dieser Lizenz noch nicht ausgeschöpft ist. Wenn z.B. eine 10-Kern-Lizenz für einen Client-Rechner mit 6 CPU-Kernen verwendet wird, so können die verbleibenden Lizenzen für die restlichen 4 Kerne nicht gleichzeitig für einen anderen Client-Rechner verwendet werden.

### 1.2.4.3 Anmerkungen zur Umgebung

#### Ordner

Im Folgenden finden Sie eine Liste wichtiger Ordner in Ihrer StyleVision Server-Installation.

☐ Root-Ordner der Installation

`/usr/local/Altova/StyleVisionServer2021/`

☐ Lizenzdateien

`/var/Altova/StyleVisionServer`

☐ Umgebungseinstellungen

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

Die Datei für die Umgebungseinstellungen muss entsprechend Ihrer jeweiligen Umgebung konfiguriert sein. Der obige Pfad dient nur als allgemeines Beispiel.

**Anmerkung:** Diese Umgebungsvariablen sind nur für den StyleVision Server Prozess definiert und haben keine Auswirkung auf andere Benutzer.

## Datenbankverbindungen

Auf MacOS werden die folgenden Datenbankverbindungen unterstützt:

- JDBC - Sie können JDBC für alle unterstützten Datenbanken mit Ausnahme von Microsoft Access verwenden
- Native Verbindungen - stehen derzeit für SQLite- und PostgreSQL-Datenbanken zur Verfügung

Beachten Sie die folgenden Punkte, wenn Sie JDBC verwenden:

- Das Java Runtime Environment oder SDK muss installiert sein
- Die JDBC-Treiber für die Zieldatenbank müssen installiert sein
- Die folgenden Umgebungsvariablen müssen für Ihre Umgebung richtig konfiguriert werden:
  - `CLASSPATH`: Um die jar-Dateien zu finden. Der Class Path wird in der Datei `plist` definiert.
  - `PATH`: Um die JRE zu finden. Nicht unbedingt erforderlich, hängt von der Installation ab
  - `JAVA_HOME`: Falls nötig - abhängig von der Installation.

## Java 6 für ausfüllbare PDF-Formulare

Wenn die ausfüllbaren Teile eines ausfüllbaren PDF-Formulars fehlen, wenn das PDF-Dokument auf einem MacOS-System geöffnet wird, liegt dies wahrscheinlich daran, dass Java 6 auf dem Rechner nicht installiert ist. Falls dies der Fall ist, können Sie Java 6 von [https://support.apple.com/kb/dl1572?locale=en\\_US](https://support.apple.com/kb/dl1572?locale=en_US) installieren. Wenn bereits eine neuere Version als Java 6 installiert ist, hat die Installation der älteren Java 6-Version keine Auswirkung auf das Funktionieren der neueren Version, die die Standardversion des Systems bleibt.

## 1.2.5 Zusätzliche Anmerkungen zur Konfiguration

Um die im Lieferumfang von StyleVision Server im Unterordner `etc\examples` des Applikationsordners inkludierten Beispiele ausführen zu können, muss die StyleVision Server DLL auf dem System korrekt registriert sein. Oft tritt dann ein Registrierungsfehler auf, wenn Sie im Laufe der Zeit auf demselben Rechner verschiedene Bit-Versionen von StyleVision Server (32-Bit und 64-Bit) installiert haben.

Um die StyleVision Server DLL (entweder die 32-Bit- oder die 64-Bit-Version) auf Windows-Rechnern korrekt zu registrieren, gehen Sie folgendermaßen vor:

1. Öffnen Sie im Administrator-Modus eine Eingabeaufforderung.
2. Wechseln Sie zu dem Ordner, in dem sich die DLL befindet. Der Befehl dazu lautet: `cd C:\Program Files\Altova\StyleVisionServer2021\bin`
3. Führen Sie den folgenden Befehl aus, um die DLL (entweder 32-Bit oder 64-Bit) zu registrieren:  
`regsvr32 styleVisionServer.dll`
4. Vergewissern Sie sich, dass eine Meldung angezeigt wird, dass die Registrierung erfolgreich war.
5. Öffnen Sie Visual Studio.
6. Laden Sie das Projekt mit Hilfe von `styleVisionServerAPI_sample.sln`.
7. Bestätigen Sie, dass Ihre `program.cs`-Datei gültige Pfade enthält.
8. Starten Sie das Projekt mit Hilfe von **Strg+F5**

**Anmerkung:** Der Pfad zum Applikationsordner unter Windows ist normalerweise: `C:\Program Files\Altova\StyleVisionServer2021`.

## 1.3 StyleVision Server-Befehlszeile

### Standardpfad der ausführbaren StyleVision Server-Datei

Im Folgenden finden Sie die jeweiligen Standardpfade der ausführbaren StyleVision Server-Datei:

<i>Linux</i>	<code>/opt/Altova/StyleVisionServer2021/bin/<b>stylevisionserver</b></code>
<i>Mac</i>	<code>/usr/local/Altova/StyleVisionServer2021/bin/<b>stylevisionserver</b></code>
<i>Windows</i>	<code>&lt;ProgramFilesFolder&gt;\Altova\StyleVisionServer2021\bin\<b>StyleVisionServer.exe</b></code>

### Verwendung und Liste der CLI-Befehle

Die Befehlszeilensyntax lautet:

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

- `--help` (Kurzform `--h`) Zeigt den Hilfetext zum jeweiligen Befehl an. Wenn kein Befehl angegeben ist, werden alle Befehle der ausführbaren Datei mit jeweils einer kurzen Beschreibung des Befehls aufgelistet.
- `--version` Zeigt die Versionsnummer von StyleVision Server an.
- `<command>` ist der auszuführende Befehl. Die Befehle sind in den Unterabschnitten dieses Abschnitts beschrieben (*siehe Liste unten*).
- `[options]` sind die Optionen eines Befehls. Diese werden mit ihren jeweiligen Befehlen aufgelistet und beschrieben.
- `[arguments]` sind die Argumente eines Befehls. Diese werden mit ihren jeweiligen Befehlen aufgelistet und beschrieben.

#### ▼ Groß- und Kleinschreibung und Schrägstriche in der Befehlszeile

`stylevisionserver` unter *Windows*

`stylevisionserver` unter *Windows und Unix (Linux, Mac)*

\* Beachten Sie, dass klein geschriebene Befehle (`stylevisionserver`) auf allen Plattformen (Windows, Linux und Mac) funktionieren, während großgeschriebene Befehle (`StyleVisionServer`) nur unter Windows und Mac ausgeführt werden.

\*Verwenden Sie auf Linux und Mac-Systemen Schrägstriche und auf Windows-Systemen umgekehrte Schrägstriche.

### CLI-Befehle

Im Folgenden sind die gültigen CLI-Befehle von StyleVision Server aufgelistet, die in den Unterabschnitten dieses Abschnitts näher beschrieben werden.

- [assignlicense](#)<sup>30</sup>: Lädt eine Lizenz in LicenseServer hoch und weist sie StyleVision Server auf diesem Rechner zu.
- [exportresourcestrings](#)<sup>31</sup>: Exportiert alle Applikationsressourcenstrings in eine XML-Datei.

- [generate](#)<sup>33</sup>: Generiert ein oder mehrere Dokumente anhand einer XML-Input-Datei und eines XSLT-Stylesheet in der PXF-Input-Datei.
- [help](#)<sup>36</sup>: Zeigt die Syntax und andere Informationen zu dem im Argument angegebenen Befehl - oder wenn kein Befehl angegeben ist, zu allen Befehlen - an.
- [licenseserver](#)<sup>37</sup>: Registriert StyleVision Server auf einem LicenseServer im lokalen Netzwerk.
- [pdfdata](#)<sup>38</sup>: Generiert in einer FDF- oder XML-Datei anhand einer PDF-Datei Formulardaten.
- [setdeflang](#)<sup>40</sup>: Definiert die Standardsprache von StyleVision Server.
- [setfopath](#)<sup>41</sup>: Wählt einen alternativen FO-Prozessor für die anschließende PDF-Generierung aus.
- [verifylicense](#)<sup>42</sup>: Überprüft, ob die aktuelle Version von StyleVision Server lizenziert ist. Überprüft optional, ob sie mit einem bestimmten Lizenzschlüssel lizenziert ist.
- [version](#)<sup>43</sup>: Zeigt die Versionsnummer von StyleVision Server an.

### 1.3.1 assignlicense (nur Windows)

#### Syntax und Beschreibung

Der Befehl `assignlicense` steht nur unter Windows (nicht aber auf Linux- oder Mac-Systemen) zur Verfügung. Mit diesem Befehl wird eine Lizenzdatei auf den Altova LicenseServer, auf dem StyleVision Server registriert ist (siehe Befehl `licenseserver`) hochgeladen und die Lizenz wird StyleVision Server zugewiesen. Der Befehl erhält den Pfad einer Lizenzdatei als Argument. Außerdem können Sie mit dem Befehl die Gültigkeit einer Lizenz überprüfen.

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

- Das Argument `FILE` erhält den Pfad der Lizenzdatei.
- Mit der Option `--test-only` wird die Lizenzdatei auf LicenseServer hochgeladen und auf ihre Gültigkeit überprüft. Sie wird jedoch StyleVision Server nicht zugewiesen.

Nähere Informationen zur Lizenzierung finden Sie in der Dokumentation zu LicenseServer (<https://www.altova.com/manual/de/AltovaLicenseServer/>).

#### ▼ Groß- und Kleinschreibung und Schrägstriche in der Befehlszeile

`stylevisionserver` unter Windows

`stylevisionserver` unter Windows und Unix (Linux, Mac)

\* Beachten Sie, dass klein geschriebene Befehle (`stylevisionserver`) auf allen Plattformen (Windows, Linux und Mac) funktionieren, während großgeschriebene Befehle (`StyleVisionServer`) nur unter Windows und Mac ausgeführt werden.

\*Verwenden Sie auf Linux und Mac-Systemen Schrägstriche und auf Windows-Systemen umgekehrte Schrägstriche.

#### ▼ Umgekehrte Schrägstriche, Leerzeichen und Sonderzeichen auf Windows-Systemen

Auf Windows-Systemen: Verwenden Sie bei Vorhandensein von Leerzeichen oder Sonderzeichen in Strings (z.B. in Datei- oder Ordnernamen oder Firmen-, Personen- oder Produktnamen)

Anführungszeichen: z.B., `"Meine Datei"`. Beachten Sie jedoch, dass ein von einem doppelten Anführungszeichen gefolger umgekehrter Schrägstrich (z.B: `"C:\Mein Verzeichnis\"`) eventuell nicht korrekt gelesen wird, da der umgekehrte Schrägstrich auch den Beginn einer Escape-Sequenz markiert und die Escape-Sequenz `\` für ein doppeltes Anführungszeichen steht. Wenn Sie diese Zeichensequenz mit einem Escape versehen wollen, verwenden Sie einen vorangestellten umgekehrten Schrägstrich, wie den folgenden: `\\`. Einfacher ausgedrückt: Wenn Sie einen Dateipfad, der Leerzeichen oder einen

umgekehrten Schrägstrich am Ende enthält, schreiben müssen, so schreiben Sie diesen folgendermaßen: `"C:\Mein Verzeichnis\\"`.

## Beispiele

Beispiele für den Befehl `assignlicense`:

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

- Mit dem ersten Befehl oben wird die angegebene Lizenzdatei auf LicenseServer hochgeladen und StyleVision Server zugewiesen.
- Mit dem letzten Befehl wird die angegebene Lizenz auf LicenseServer hochgeladen und auf ihre Gültigkeit überprüft, ohne sie StyleVision Server zuzuweisen.

## Optionen

Optionen werden in ihrer kurzen Form (falls verfügbar) und in ihrer langen Form aufgelistet. Für die kurze und die lange Form können ein oder zwei Bindestriche verwendet werden. Eine Option kann, muss aber keinen Wert erhalten. Eine Option, die einen Wert erhält, wird folgendermaßen geschrieben: `--option=wert`. Werte können außer in zwei Fällen ohne Anführungszeichen definiert werden: (i) wenn der Wertestring Leerzeichen enthält oder (ii) wenn in der Beschreibung der Option explizit erwähnt ist, dass Anführungszeichen zwingend erforderlich sind. Wenn eine Option einen Booleschen Wert erhält und kein Wert definiert ist, so ist der Standardwert der Option `TRUE`. Mit Hilfe der Option `--h, --help` können Sie Informationen über den Befehl anzeigen.

### ▼ test-only [t]

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

Die Werte sind `true|false`. Bei `true` wird die Lizenzdatei auf LicenseServer hochgeladen und auf ihre Gültigkeit überprüft, aber nicht zugewiesen.

## 1.3.2 exportresourcestrings

### Syntax und Beschreibung

Der Befehl `exportresourcestrings` gibt eine XML-Datei aus, die die Ressourcenstrings der StyleVision Server-Applikation in der definierten Sprache enthält. Als Exportsprachen stehen Englisch (`en`), Deutsch (`de`), Spanisch (`es`), Französisch (`fr`) und Japanisch (`ja`) zur Verfügung.

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

- Das Argument `LanguageCode` gibt die Sprache der Ressourcenstrings in der XML-Ausgabedatei an; dies ist die *Exportsprache*. Derzeit unterstützte Exportsprachen sind (mit den Sprachcodes in Klammern): Englisch (`en`), Deutsch (`de`), Spanisch (`es`), Französisch (`fr`) und Japanisch (`ja`).
- Das Argument `XMLOutputFile` definiert den Namen und Pfad der XML-Ausgabedatei.

Eine Anleitung zum Erstellen von Lokalisierungen finden Sie weiter unten.

▼ Groß- und Kleinschreibung und Schrägstriche in der Befehlszeile

`styleVisionServer` unter Windows

`stylevisionserver` unter Windows und Unix (Linux, Mac)

\* Beachten Sie, dass klein geschriebene Befehle (`stylevisionserver`) auf allen Plattformen (Windows, Linux und Mac) funktionieren, während groß geschriebene Befehle (`StyleVisionServer`) nur unter Windows und Mac ausgeführt werden.

\*Verwenden Sie auf Linux und Mac-Systemen Schrägstriche und auf Windows-Systemen umgekehrte Schrägstriche.

▼ Umgekehrte Schrägstriche, Leerzeichen und Sonderzeichen auf Windows-Systemen

Auf Windows-Systemen: Verwenden Sie bei Vorhandensein von Leerzeichen oder Sonderzeichen in Strings (z.B. in Datei- oder Ordnernamen oder Firmen-, Personen- oder Produktnamen)

Anführungszeichen: z.B., "`Meine Datei`". Beachten Sie jedoch, dass ein von einem doppelten Anführungszeichen gefolgt umgekehrter Schrägstrich (z.B.: "`C:\Mein Verzeichnis\`") eventuell nicht korrekt gelesen wird, da der umgekehrte Schrägstrich auch den Beginn einer Escape-Sequenz markiert und die Escape-Sequenz "`\`" für ein doppeltes Anführungszeichen steht. Wenn Sie diese Zeichensequenz mit einem Escape versehen wollen, verwenden Sie einen vorangestellten umgekehrten Schrägstrich, wie den folgenden: "`\\`". Einfacher ausgedrückt: Wenn Sie einen Dateipfad, der Leerzeichen oder einen umgekehrten Schrägstrich am Ende enthält, schreiben müssen, so schreiben Sie diesen folgendermaßen: "`C:\Mein Verzeichnis\\`".

## Beispiele

Beispiele für den Befehl `exportresourcestrings`:

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

- Mit dem Befehl oben wird unter `c:\` eine Datei namens `Strings.xml` erstellt, die alle Ressourcenstrings der StyleVision Server Applikation in englischer Sprache enthält.

## Erstellen lokalisierter Versionen von StyleVision Server

Sie können für jede Sprache Ihrer Wahl eine lokalisierte Version von StyleVision Server erstellen. Im Ordner `c:\Programme (x86)\Altova\StyleVisionServer2021\bin` stehen fünf lokalisierte Versionen (Englisch, Deutsch, Spanisch, Französisch und Japanisch) bereits zur Verfügung. Eine Lokalisierung für diese Sprache ist daher nicht mehr notwendig.

Folgendermaßen können Sie eine lokalisierte Version erstellen:

1. Generieren Sie mit Hilfe des Befehls `exportresourcestrings` (siehe *Befehlssyntax oben*) eine XML-Datei, die die Ressourcenstrings enthält. Die Ressourcenstrings in dieser XML-Datei sind in einer der fünf unterstützten Sprachen: je nachdem, welches `LanguageCode`-Argument mit dem Befehl verwendet wird, in Englisch (`en`), Deutsch (`de`), Spanisch (`es`), Französisch (`fr`) oder Japanisch (`ja`).
2. Übersetzen Sie die Ressourcenstrings aus einer der fünf unterstützten Sprachen in die Zielsprache. Die Ressourcenstrings bilden den Inhalt der `<string>` Elemente in der XML-Datei. Übersetzen Sie keine Variablen in geschweiften Klammern wie z.B. `{option}` oder `{product}`.
3. Wenden Sie sich an den [Altova Support](#), um anhand Ihrer übersetzten XML-Datei eine lokalisierte StyleVision Server DLL-Datei zu generieren.



4. Nachdem Sie Ihre lokalisierte DLL-Datei vom [Altova Support](#) erhalten haben, speichern Sie diese unter `C:\Programme (x86)\Altova\StyleVisionServer2021\bin`. Ihre DLL-Datei wird einen Namen in der Form `StyleVisionServer2021_1c.dll` haben. Der `1c` Teil des Namens enthält den Sprachencode. So steht z.B. in `StyleVisionServer2021_de.dll` der Teil `de` für Deutsch.
5. Führen Sie den Befehl `setdeflang` aus, um Ihre lokalisierte DLL als die zu verwendende StyleVision Server Applikation zu definieren. Verwenden Sie den Sprachencode, der Teil des DLL-Namens ist, als Argument des Befehls `setdeflang`.

**Anmerkung:** Altova StyleVision Server ist mit Unterstützung für fünf Sprachen erhältlich: Englisch, Deutsch, Spanisch, Französisch und Japanisch. Sie müssen daher keine lokalisierte Version dieser Sprachen erstellen. Um eine dieser Sprachen als Standardsprache festzulegen, verwenden Sie den StyleVision Server Befehl `setdeflang`.

### 1.3.3 generate

#### Syntax und Beschreibung

Der Befehl `generate` generiert durch Transformieren der XML-Input-Datei mit Hilfe des/der in der PXF-Input-Datei enthaltenen XSLT-Dokuments/Dokumente eine oder mehrere Ausgabedateien (HTML, PDF, RTF und/oder DOCX).

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

- Die Option `--inputxml` ist obligatorisch und gibt den Pfad der XML-Datei an.
- Das Argument *InputPXF* definiert den Pfad der PXF-Datei, die das/die zur Generierung des/der Ausgabedokuments/Ausgabedokumente verwendete(n) XSLT-Dokument(e) enthält. PXF-Dateien wird mit der [Altova-Applikation StyleVision](#) generiert.
- Jedes Ausgabeformat wird durch Definition einer Option für diese Ausgabe (*siehe Liste der Optionen unten*) generiert. Der Wert jeder Option ist ein Pfad, der definiert, wo die Ausgabe generiert werden soll.

**Anmerkung:** StyleVision Server verwendet [Apache FOP](#), den FO-Prozessor des Apache Project zur Generierung von PDF-Dateien anhand von FO. Apache FOP wird zusammen mit StyleVision Server unter dem folgenden Pfad installiert: Auf Windows-Systemen, `ProgramData\Altova\SharedBetweenVersions`; auf Linux- und macOS-Systemen in einem Ordner unterhalb des Ordners `StyleVisionServer2021`. Beachten Sie, dass für Apache FOP **Java Runtime Environment 1.6 oder höher** auf dem StyleVision Server-Rechner installiert sein muss. Installieren Sie für die 32-Bit-Version von StyleVision Server die Java-32-Bit-Version und für die 64-Bit-Version von StyleVision Server die Java-64-Bit-Version.

#### ▼ Groß- und Kleinschreibung und Schrägstriche in der Befehlszeile

`stylevisionserver` unter Windows

`stylevisionserver` unter Windows und Unix (Linux, Mac)

\* Beachten Sie, dass klein geschriebene Befehle (`stylevisionserver`) auf allen Plattformen (Windows, Linux und Mac) funktionieren, während großgeschriebene Befehle (`StyleVisionServer`) nur unter Windows und Mac ausgeführt werden.

\*Verwenden Sie auf Linux und Mac-Systemen Schrägstriche und auf Windows-Systemen umgekehrte Schrägstriche.

#### ▼ Umgekehrte Schrägstriche, Leerzeichen und Sonderzeichen auf Windows-Systemen

Auf Windows-Systemen: Verwenden Sie bei Vorhandensein von Leerzeichen oder Sonderzeichen in Strings (z.B. in Datei- oder Ordernamen oder Firmen-, Personen- oder Produktnamen) Anführungszeichen: z.B., "Meine Datei". Beachten Sie jedoch, dass ein von einem doppelten Anführungszeichen gefolgt umgekehrter Schrägstrich (z.B: "C:\Mein Verzeichnis\") eventuell nicht korrekt gelesen wird, da der umgekehrte Schrägstrich auch den Beginn einer Escape-Sequenz markiert und die Escape-Sequenz \" für ein doppeltes Anführungszeichen steht. Wenn Sie diese Zeichensequenz mit einem Escape versehen wollen, verwenden Sie einen vorangestellten umgekehrten Schrägstrich, wie den folgenden: \\". Einfacher ausgedrückt: Wenn Sie einen Dateipfad, der Leerzeichen oder einen umgekehrten Schrägstrich am Ende enthält, schreiben müssen, so schreiben Sie diesen folgendermaßen: "C:\Mein Verzeichnis\\".

## Beispiele

Beispiele für den Befehl `generate`:

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

- Der obige Befehl enthält die obligatorische Option `--inputxml`, das Argument `InputPXF` (`Test.pxf`) sowie mindestens eine Option zur Ausgabeerstellung (`--html` in allen Beispielen oben).
- Die verwendete XML-Input-Datei kann sich innerhalb der PXF-Datei (*siehe erstes und drittes Beispiel oben*) befinden oder eine externe XML-Datei sein (die sich außerhalb der PXF-Datei befindet; *siehe erstes und viertes Beispiel oben*).
- Der `--inputxml` Switch wird ignoriert, wenn die Hauptschemaquelle eine Datenbank oder DB-XML ist, muss aber aus syntaktischen Gründen vorhanden sein und Sie sollten z.B. `--inputxml=database` verwenden.
- Wenn die Option `--html` zur Erstellung des Output wie im Beispiel oben einen relativen Pfad erhält, so wird die Output-Datei unter einem Pfad relativ zum Ordner, in dem sich die PXF-Datei befindet, erstellt.

## Optionen

### ▼ inputxml [xml]

`--xml, --inputxml = PathToXMLFile`

Diese Option ist obligatorisch. Sie definiert den Pfad zu der zu verarbeitenden XML-Datei. Die XML-Datei kann sich innerhalb oder außerhalb der PXF-Datei befinden. Um als Ziel XML-Dateien innerhalb einer PXF-Datei zu definieren, verwenden Sie den Lokator `|zip` (siehe markierter Teil in den Beispielen unten). Die Option `--inputxml` wird ignoriert, wenn es sich bei der Hauptschemaquelle der Input-PXF-Datei um eine Datenbank oder eine DB-XML-Quelldatei handelt.

### ▼ dbwhere [dbw]

`--dbw, --dbwhere = WHEREClause`

Eine SQL-WHERE-Klausel, die festlegt, welche Zeilen einer DB-XML-Quelldatei verarbeitet werden sollen.

## ▼ param [p]

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

Weist einem in der PXF-Datei definierten Parameter einen Wert zu.

Form: `--param=$ParamName:ParamValue`. Die Option `--param` muss vor jedem Parameter verwendet werden. Verwenden Sie Anführungszeichen, wenn `ParamName` oder `ParamValue` ein Leerzeichen enthält.

Beispiel: `--p=$company:"Nanonull Inc"`.

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

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

Erhält den Wert `true` oder `false`. Bei Auswahl von `true` können Ausgabedokumente nur in dem Ordner erstellt werden, in dem die Hauptausgabedatei (HTML, PDF, RTF, DOCX, FO) erstellt wird. Dadurch werden andere Ordner, falls nötig, geschützt. Die Standardeinstellung ist `false`.

## ▼ outhtml [html]

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

Pfad zur zu generierenden HTML-Datei

## ▼ outpdf [pdf]

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

Pfad zur zu generierenden PDF-Datei

## ▼ outrtf [rtf]

```
--rtf, --outrtf = FilePath
```

Pfad zur zu generierenden RTF-Datei

## ▼ outdocx [docx]

```
--docx, --outdocx = FilePath
```

Pfad zur zu generierenden DOCX-Datei

## ▼ outfo [fo]

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

Pfad zur zu generierenden FO-Datei

## ▼ generate-html-output-as-mime

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

Die Werte sind `true|false`. Wenn die Option nicht definiert ist, ist der Standardwert `false`. Wenn die Option ohne Wert definiert ist, so ist er `true`. Die HTML-Ausgabe wird als Mime-Stream generiert.

## ▼ verbose [v]

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

Die Werte sind `true|false`. Aktiviert bzw. deaktiviert Anzeige aller Meldungen. Der Standardwert ist `false`, wenn die Option nicht angegeben wird, `true`, wenn die Option ohne einen Wert angegeben wird.

## ▼ lang [l]

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

Die zur Anzeige der Meldung verwendete Sprache.

Mit Hilfe der Option `--h, --help` können Sie Informationen über den Befehl anzeigen.

Optionen werden in ihrer kurzen Form (falls verfügbar) und in ihrer langen Form aufgelistet. Für die kurze und die lange Form können ein oder zwei Bindestriche verwendet werden. Eine Option kann, muss aber keinen Wert erhalten. Eine Option, die einen Wert erhält, wird folgendermaßen geschrieben: `--option=wert`. Werte können außer in zwei Fällen ohne Anführungszeichen definiert werden: (i) wenn der Wertestring Leerzeichen enthält oder (ii) wenn in der Beschreibung der Option explizit erwähnt ist, dass Anführungszeichen zwingend erforderlich sind. Wenn eine Option einen Booleschen Wert erhält und kein Wert definiert ist, so ist der Standardwert der Option `TRUE`. Mit Hilfe der Option `--h, --help` können Sie Informationen über den Befehl anzeigen.

## 1.3.4 help

### Syntax und Beschreibung

Der Befehl `help` hat ein einziges Argument (`Command`): den Namen des Befehls, zu dem die Hilfe benötigt wird. Er zeigt die korrekte Syntax des Befehls, seine Optionen sowie andere relevante Informationen an. Wenn das Argument `Command` nicht angegeben wird, werden alle Befehle der ausführbaren Datei aufgelistet, wobei zu jedem eine kurze Textbeschreibung angezeigt wird.

```
stylevisionserver help Command
```

#### ▼ Groß- und Kleinschreibung und Schrägstriche in der Befehlszeile

`stylevisionserver` unter Windows

`stylevisionserver` unter Windows und Unix (Linux, Mac)

\* Beachten Sie, dass klein geschriebene Befehle (`stylevisionserver`) auf allen Plattformen (Windows, Linux und Mac) funktionieren, während großgeschriebene Befehle (`StyleVisionServer`) nur unter Windows und Mac ausgeführt werden.

\*Verwenden Sie auf Linux und Mac-Systemen Schrägstriche und auf Windows-Systemen umgekehrte Schrägstriche.

### Beispiel

Beispiel für den Befehl `help`, um Informationen über den Befehl `licenseserver` anzuzeigen:

```
stylevisionserver help licenseserver
```

### Die Option `--help`

Die Hilfe zu einem Befehl kann auch über die Option `--help` im Anschluss an diesen Befehl aufgerufen werden. Mit den beiden unten stehenden Befehlen erhalten Sie dasselbe Ergebnis:

```
stylevisionserver licenseserver --help
```

Im obigen Befehl wird die Option `--help` des Befehls `licenseserver` verwendet.

```
stylevisionserver help licenseserver
```

Der Befehl `help` erhält `licenseserver` als Argument.

In beiden Fällen wird die Hilfe zum Befehl `licenseserver` angezeigt.

## 1.3.5 licenseserver

### Syntax und Beschreibung

Bei Ausführung des Befehls `licenseserver` wird StyleVision Server auf dem durch das Argument `Server-Or-IP-Address` definierten LicenseServer registriert. Damit der Befehl `licenseserver` erfolgreich ausgeführt werden kann, müssen sich die beiden Server (StyleVision Server und LicenseServer) im selben Netzwerk befinden und LicenseServer muss ausgeführt werden. Außerdem benötigen Sie zum Registrieren von StyleVision Server auf dem LicenseServer Administratorrechte.

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

- Das Argument `Server-Or-IP-Address` erhält den Namen oder die IP-Adresse des LicenseServer-Rechners.
- Um eine Evaluierungslizenz für StyleVision Server anzufordern, geben Sie die entsprechenden Werte für die Optionen `--company`, `--email` und `--name` an. Um eine Evaluierungslizenz für ein zusätzliches Altova-Produkt (d.h. zusätzlich zu StyleVision Server) anzufordern, fügen Sie die Option `--request-eval-license` mit einem Wert, der dem Name dieses Produkts entspricht, hinzu. *Nähere Informationen dazu finden Sie in der Liste weiter unten.*

Sobald StyleVision Server erfolgreich auf dem LicenseServer registriert wurde, erhalten Sie eine entsprechende Meldung. Darin wird auch die URL des LicenseServer angezeigt. Sie können nun zu LicenseServer wechseln und StyleVision Server eine Lizenz zuweisen. Nähere Informationen zur Lizenzierung finden Sie in der Dokumentation zu LicenseServer (<https://www.altova.com/manual/de/AltovaLicenseServer/>).

#### ▼ Groß- und Kleinschreibung und Schrägstriche in der Befehlszeile

`stylevisionServer` unter Windows

`stylevisionserver` unter Windows und Unix (Linux, Mac)

\* Beachten Sie, dass klein geschriebene Befehle (`stylevisionserver`) auf allen Plattformen (Windows, Linux und Mac) funktionieren, während großgeschriebene Befehle (`StylevisionServer`) nur unter Windows und Mac ausgeführt werden.

\*Verwenden Sie auf Linux und Mac-Systemen Schrägstriche und auf Windows-Systemen umgekehrte Schrägstriche.

#### ▼ Umgekehrte Schrägstriche, Leerzeichen und Sonderzeichen auf Windows-Systemen

Auf Windows-Systemen: Verwenden Sie bei Vorhandensein von Leerzeichen oder Sonderzeichen in Strings (z.B. in Datei- oder Ordernamen oder Firmen-, Personen- oder Produktnamen)

Anführungszeichen: z.B., `"Meine Datei"`. Beachten Sie jedoch, dass ein von einem doppelten Anführungszeichen gefolgt umgekehrter Schrägstrich (z.B.: `"C:\Mein Verzeichnis\"`) eventuell nicht korrekt gelesen wird, da der umgekehrte Schrägstrich auch den Beginn einer Escape-Sequenz markiert und die Escape-Sequenz `\"` für ein doppeltes Anführungszeichen steht. Wenn Sie diese Zeichensequenz

mit einem Escape versehen wollen, verwenden Sie einen vorangestellten umgekehrten Schrägstrich, wie den folgenden: `\\`. Einfacher ausgedrückt: Wenn Sie einen Dateipfad, der Leerzeichen oder einen umgekehrten Schrägstrich am Ende enthält, schreiben müssen, so schreiben Sie diesen folgendermaßen: `"C:\Mein Verzeichnis\\"`.

## Beispiele

Beispiele für den Befehl `licenseserver`:

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

Die Befehle oben definieren den Rechner namens `DOC.altova.com` und den Rechner des Benutzers (`localhost` bzw. `127.0.0.1`) als den Rechner, auf dem Altova LicenseServer ausgeführt wird. In jedem dieser Fälle wird StyleVision Server auf dem LicenseServer auf dem angegebenen Rechner registriert. Mit dem letzten Befehl wird zum Ausführen des Befehls die ausführbare Server-Datei aufgerufen.

## Optionen

Optionen werden in ihrer kurzen Form (falls verfügbar) und in ihrer langen Form aufgelistet. Für die kurze und die lange Form können ein oder zwei Bindestriche verwendet werden. Eine Option kann, muss aber keinen Wert erhalten. Eine Option, die einen Wert erhält, wird folgendermaßen geschrieben: `--option=wert`. Werte können außer in zwei Fällen ohne Anführungszeichen definiert werden: (i) wenn der Wertestring Leerzeichen enthält oder (ii) wenn in der Beschreibung der Option explizit erwähnt ist, dass Anführungszeichen zwingend erforderlich sind. Wenn eine Option einen Booleschen Wert erhält und kein Wert definiert ist, so ist der Standardwert der Option `TRUE`. Mit Hilfe der Option `--h, --help` können Sie Informationen über den Befehl anzeigen.

### ▼ json [j]

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

Die Werte sind `true|false`. Bei `true` wird das Ergebnis des Registrierungsversuchs als JSON-Objekt ausgegeben, das von Rechnern geparkt werden kann.

## 1.3.6 pdfdata

### Syntax und Beschreibung

Der Befehl `pdfdata` generiert anhand der als *InputPDF*-Argument bereitgestellten PDF-Datei eine FDF- oder XML-Datei.

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

- Das Argument *InputPDF* definiert den Pfad zu der PDF-Datei, anhand welcher die FDF- oder XML-Ausgabedatei generiert wird. Wenn die PDF-Datei keine Formulardaten enthält, so enthält auch die generierte XML-Datei keine Formulardaten.

- Mit Hilfe der Option `--outfdf` können Sie den Pfad der generierten FDF-Datei bzw. mit der Option `--outxml` den Pfad der generierten XML-Datei definieren

Nähere Informationen zum Erstellen von FDF-Dateien und ausfüllbaren PDF-Formularen finden Sie in der Dokumentation zu [Altova StyleVision \(Enterprise Edition\)](#).

#### ▼ Groß- und Kleinschreibung und Schrägstriche in der Befehlszeile

`stylevisionserver` unter Windows

`stylevisionserver` unter Windows und Unix (Linux, Mac)

\* Beachten Sie, dass klein geschriebene Befehle (`stylevisionserver`) auf allen Plattformen (Windows, Linux und Mac) funktionieren, während großgeschriebene Befehle (`StyleVisionServer`) nur unter Windows und Mac ausgeführt werden.

\*Verwenden Sie auf Linux und Mac-Systemen Schrägstriche und auf Windows-Systemen umgekehrte Schrägstriche.

#### ▼ Umgekehrte Schrägstriche, Leerzeichen und Sonderzeichen auf Windows-Systemen

Auf Windows-Systemen: Verwenden Sie bei Vorhandensein von Leerzeichen oder Sonderzeichen in Strings (z.B. in Datei- oder Ordnernamen oder Firmen-, Personen- oder Produktnamen)

Anführungszeichen: z.B., "Meine Datei". Beachten Sie jedoch, dass ein von einem doppelten Anführungszeichen gefolger umgekehrter Schrägstrich (z.B: "C:\Mein Verzeichnis\" ) eventuell nicht korrekt gelesen wird, da der umgekehrte Schrägstrich auch den Beginn einer Escape-Sequenz markiert und die Escape-Sequenz `\\"` für ein doppeltes Anführungszeichen steht. Wenn Sie diese Zeichensequenz mit einem Escape versehen wollen, verwenden Sie einen vorangestellten umgekehrten Schrägstrich, wie den folgenden: `\\\"`. Einfacher ausgedrückt: Wenn Sie einen Dateipfad, der Leerzeichen oder einen umgekehrten Schrägstrich am Ende enthält, schreiben müssen, so schreiben Sie diesen folgendermaßen: `"C:\Mein Verzeichnis\\"`.

## Beispiele

Beispiele für den Befehl `pdfdata`:

```
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
```

Mit den obigen Beispielen wird anhand derselben PDF-Input-Datei eine FDF-Datei bzw. eine XML-Datei erstellt.

## Optionen

#### ▼ outfdf

`--outfdf = FilePath`

Der Pfad zur generierten FDF-Datei.

#### ▼ outxml

`--outxml = FilePath`

Der Pfad zur generierten XML-Datei.

Mit Hilfe der Option `--h, --help` können Sie Informationen über den Befehl anzeigen.

Optionen werden in ihrer kurzen Form (falls verfügbar) und in ihrer langen Form aufgelistet. Für die kurze und die lange Form können ein oder zwei Bindestriche verwendet werden. Eine Option kann, muss aber keinen Wert erhalten. Eine Option, die einen Wert erhält, wird folgendermaßen geschrieben: `--option=wert`. Werte können außer in zwei Fällen ohne Anführungszeichen definiert werden: (i) wenn der Wertestring Leerzeichen enthält oder (ii) wenn in der Beschreibung der Option explizit erwähnt ist, dass Anführungszeichen zwingend erforderlich sind. Wenn eine Option einen Booleschen Wert erhält und kein Wert definiert ist, so ist der Standardwert der Option `TRUE`. Mit Hilfe der Option `--h, --help` können Sie Informationen über den Befehl anzeigen.

## 1.3.7 setdeflang

### Syntax und Beschreibung

Der Befehl `setdeflang` (Kurzform ist `sd1`) definiert die Standardsprache von StyleVision Server. Verfügbare Sprachen sind Englisch (`en`), Deutsch (`de`), Spanisch (`es`), Französisch (`fr`) und Japanisch (`ja`). Der Befehl erhält ein obligatorisches Argument *LanguageCode*.

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

- Das Argument *LanguageCode* definiert die Standardsprache von StyleVision Server. Die entsprechenden Werte sind: `en`, `de`, `es`, `fr`, `ja`.
- Mit Hilfe der Option `--h, --help` können Sie Informationen über den Befehl anzeigen.

#### ▼ Groß- und Kleinschreibung und Schrägstriche in der Befehlszeile

```
stylevisionserver unter Windows
```

```
stylevisionserver unter Windows und Unix (Linux, Mac)
```

\* Beachten Sie, dass klein geschriebene Befehle (`stylevisionserver`) auf allen Plattformen (Windows, Linux und Mac) funktionieren, während großgeschriebene Befehle (`StyleVisionServer`) nur unter Windows und Mac ausgeführt werden.

\*Verwenden Sie auf Linux und Mac-Systemen Schrägstriche und auf Windows-Systemen umgekehrte Schrägstriche.

### Beispiele

Beispiele für den Befehl `setdeflang` (`sd1`):

```
stylevisionserver sd1 en
stylevisionserver setdeflang es
```

- Mit dem ersten Befehl wird als Standardsprache von StyleVision Server Englisch definiert.
- Mit dem zweiten Befehl wird als Standardsprache von StyleVision Server Spanisch definiert.



## Optionen

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

### 1.3.8 setfopath

#### Syntax und Beschreibung

Der Befehl `setfopath` (Kurzform ist `sfp`) definiert den Pfad zu einem anderen Apache-FOP-Prozessor als dem im StyleVision Server-Paket inkludierten.

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

- Standardmäßig wird zum Verarbeiten von FO-Dokumenten und Generieren von PDF-Ausgabedokumenten der in StyleVision Server inkludierte FOP-Prozessor verwendet. Wenn eine andere Apache FOP-Prozessorinstanz als die mit StyleVision Server installierte verwendet werden soll, verwenden Sie den Befehl `setfopath`, wobei mit dem Argument `Path` der Pfad zum gewünschten FO-Prozessor angegeben wird.
- Nachdem mit dem Befehl `setfopath` ein anderer FO-Prozessor definiert wurde, wird bei Verwendung nachfolgender `generate` <sup>33</sup> Befehle dieser Prozessor für die PDF-Generierung verwendet. Um den Prozessor wieder zu wechseln, verwenden Sie wieder den Befehl `setfopath`. Um zurück zum FOP-Prozessor von StyleVision Server zu wechseln, navigieren Sie zum FOP-Ordner auf Ihrem System und verwenden Sie diesen Pfad als Argument von `setfopath`.
- Auf Windows-Systemen befindet sich der mit StyleVision Server installierte Ordner FOP unter `ProgramData\Altova\SharedBetweenVersions`; auf Linux- und macOS-Systemen befindet er sich in einem untergeordnete Ordner des Ordners `StyleVisionServer2021`.

Nähere Informationen zu FDF-Dateien und der Erstellung von ausfüllbaren PDF-Formularen finden Sie in der Dokumentation zu [Altova StyleVision \(Enterprise Edition\)](#).

#### ▼ Groß- und Kleinschreibung und Schrägstriche in der Befehlszeile

`stylevisionserver` unter Windows

`stylevisionserver` unter Windows und Unix (Linux, Mac)

\* Beachten Sie, dass klein geschriebene Befehle (`stylevisionserver`) auf allen Plattformen (Windows, Linux und Mac) funktionieren, während großgeschriebene Befehle (`StyleVisionServer`) nur unter Windows und Mac ausgeführt werden.

\*Verwenden Sie auf Linux und Mac-Systemen Schrägstriche und auf Windows-Systemen umgekehrte Schrägstriche.

#### ▼ Umgekehrte Schrägstriche, Leerzeichen und Sonderzeichen auf Windows-Systemen

Auf Windows-Systemen: Verwenden Sie bei Vorhandensein von Leerzeichen oder Sonderzeichen in Strings (z.B. in Datei- oder Ordernamen oder Firmen-, Personen- oder Produktnamen)

Anführungszeichen: z.B., `"Meine Datei"`. Beachten Sie jedoch, dass ein von einem doppelten Anführungszeichen gefolgt umgekehrter Schrägstrich (z.B: `"C:\Mein Verzeichnis\"`) eventuell nicht korrekt gelesen wird, da der umgekehrte Schrägstrich auch den Beginn einer Escape-Sequenz markiert und die Escape-Sequenz `\"` für ein doppeltes Anführungszeichen steht. Wenn Sie diese Zeichensequenz mit einem Escape versehen wollen, verwenden Sie einen vorangestellten umgekehrten Schrägstrich, wie den folgenden: `\\\"`. Einfacher ausgedrückt: Wenn Sie einen Dateipfad, der Leerzeichen oder einen

umgekehrten Schrägstrich am Ende enthält, schreiben müssen, so schreiben Sie diesen folgendermaßen: `"C:\Mein Verzeichnis\"`.

## Beispiele

Nach Ausführung des Befehls `setfopath` können Sie mit Hilfe des Befehls `generate`<sup>33</sup> und unter Verwendung des soeben definierten FO-Prozessors eine PDF-Datei generieren.

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

Die oben aufgelisteten Befehle führen folgende Schritte aus:

1. Der Befehl `setfopath` definiert, dass der FOP-Prozessor unter dem Pfad `C:\FOP\FOP.bat` verwendet werden soll, um die PDF-Datei zu generieren, die mit dem nachfolgenden Befehl generiert wird.
2. Der Befehl `generate` generiert anhand der definierten XML-Input-Datei eine PDF-Datei und verwendet in der PDF-Datei enthaltene Transformationsdateien. Zum Generieren der PDF-Datei wird der im vorherigen Befehl definierte FOP-Prozessor verwendet.

## Optionen

Mit Hilfe der Option `--h, --help` können Sie Informationen über den Befehl anzeigen.

Optionen werden in ihrer kurzen Form (falls verfügbar) und in ihrer langen Form aufgelistet. Für die kurze und die lange Form können ein oder zwei Bindestriche verwendet werden. Eine Option kann, muss aber keinen Wert erhalten. Eine Option, die einen Wert erhält, wird folgendermaßen geschrieben: `--option=wert`. Werte können außer in zwei Fällen ohne Anführungszeichen definiert werden: (i) wenn der Wertestring Leerzeichen enthält oder (ii) wenn in der Beschreibung der Option explizit erwähnt ist, dass Anführungszeichen zwingend erforderlich sind. Wenn eine Option einen Booleschen Wert erhält und kein Wert definiert ist, so ist der Standardwert der Option `TRUE`. Mit Hilfe der Option `--h, --help` können Sie Informationen über den Befehl anzeigen.

### 1.3.9 verifylicense (nur Windows)

#### Syntax und Beschreibung

Mit dem Befehl `verifylicense` wird überprüft, ob das aktuelle Produkt lizenziert ist. Zusätzlich können Sie mit der Option `--license-key` überprüfen, ob dem Produkt bereits ein bestimmter Lizenzschlüssel zugewiesen wurde. Dieser Befehl wird nur auf Windows-Systemen, nicht aber unter Linux oder Max unterstützt.

```
stylevisionserver verifylicense [options]
```

- Um zu überprüfen, ob StyleVision Server eine bestimmte Lizenz zugewiesen ist, geben Sie den Lizenzschlüssel als Wert der Option `--license-key` an.

Nähere Informationen zur Lizenzierung finden Sie in der Dokumentation zu LicenseServer (<https://www.altova.com/manual/de/AltovaLicenseServer/>).

▼ Groß- und Kleinschreibung und Schrägstriche in der Befehlszeile

`styleVisionServer` unter Windows

`stylevisionserver` unter Windows und Unix (Linux, Mac)

\* Beachten Sie, dass klein geschriebene Befehle (`stylevisionserver`) auf allen Plattformen (Windows, Linux und Mac) funktionieren, während groß geschriebene Befehle (`StyleVisionServer`) nur unter Windows und Mac ausgeführt werden.

\*Verwenden Sie auf Linux und Mac-Systemen Schrägstriche und auf Windows-Systemen umgekehrte Schrägstriche.

## Beispiele

Beispiele für den Befehl `verifylicense`:

```
stylevisionserver verifylicense
```

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

- Mit dem ersten Befehl wird überprüft, ob StyleVision Server lizenziert ist.
- Mit dem zweiten Befehl wird überprüft, ob StyleVision Server mit dem in der Option `--license-key` definierten Lizenzschlüssel lizenziert ist.

## Optionen

Optionen werden in ihrer kurzen Form (falls verfügbar) und in ihrer langen Form aufgelistet. Für die kurze und die lange Form können ein oder zwei Bindestriche verwendet werden. Eine Option kann, muss aber keinen Wert erhalten. Eine Option, die einen Wert erhält, wird folgendermaßen geschrieben: `--option=wert`. Werte können außer in zwei Fällen ohne Anführungszeichen definiert werden: (i) wenn der Wertestring Leerzeichen enthält oder (ii) wenn in der Beschreibung der Option explizit erwähnt ist, dass Anführungszeichen zwingend erforderlich sind. Wenn eine Option einen Booleschen Wert erhält und kein Wert definiert ist, so ist der Standardwert der Option `TRUE`. Mit Hilfe der Option `--h, --help` können Sie Informationen über den Befehl anzeigen.

▼ `license-key` [!]

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

Überprüft, ob StyleVision Server mit dem als Wert dieser Option definierten Lizenzschlüssel lizenziert ist.

## 1.3.10 version

### Syntax und Beschreibung

Mit dem Befehl `version` wird die Versionsnummer von StyleVision Server angezeigt.

```
stylevisionserver version
```

## ▼ Groß- und Kleinschreibung und Schrägstriche in der Befehlszeile

`styleVisionServer` unter Windows

`stylevisionserver` unter Windows und Unix (Linux, Mac)

\* Beachten Sie, dass klein geschriebene Befehle (`stylevisionserver`) auf allen Plattformen (Windows, Linux und Mac) funktionieren, während großgeschriebene Befehle (`StyleVisionServer`) nur unter Windows und Mac ausgeführt werden.

\*Verwenden Sie auf Linux und Mac-Systemen Schrägstriche und auf Windows-Systemen umgekehrte Schrägstriche.

## Beispiel

Beispiel für den Befehl `version`:

```
stylevisionserver version
```

## 1.4 StyleVision Server API

StyleVision Server enthält eine API (Application Programming Interface), die Sie von Ihrem .NET-, COM- oder Java-basierten Code aus programmatisch aufrufen können.

Dieser Abschnitt ist in die folgenden Unterabschnitte gegliedert:

- [Informationen zur .NET-Schnittstelle](#) <sup>45</sup>
- [Informationen zur COM-Schnittstelle](#) <sup>46</sup>
- [Informationen zur Java-Schnittstelle](#) <sup>46</sup>
- [Codebeispiele](#) <sup>47</sup>
- [API-Referenz](#) <sup>54</sup>

### 1.4.1 Informationen zur .NET-Schnittstelle

Die .NET-Schnittstelle bildet einen Wrapper rund um die COM-Schnittstelle. Sie dient als primäre, von Altova signierte Interop Assembly und verwendet den Namespace `Altova.StyleVisionServer`.

StyleVision Server wird bei der Installation automatisch als COM-Serverobjekt registriert und muss daher nicht manuell registriert werden. Wenn Sie einen Zugriffsfehler erhalten, öffnen Sie die Komponentendienste und räumen Sie dem Konto, auf dem der Applikationspool, der StyleVision Server enthält, ausgeführt wird, Zugriffsrechte ein.

Um StyleVision Server in Ihrem .NET-Projekt verwenden zu können, fügen Sie eine Referenz zur `Altova.StyleVisionServer.dll` Datei hinzu (siehe Anleitung unten). Die Datei `Altova.StyleVisionServer.dll` befindet sich im Ordner `bin` des StyleVision Server Installationsordners. Diese `.dll`-Datei wird bei der Installation von StyleVision Server automatisch zum GAC (Global Assembly Cache (GAC)) hinzugefügt (der GAC befindet sich normalerweise im Ordner `C:\WINDOWS\assembly`).

Sobald StyleVision Server als COM-Serverobjekt registriert wurde und die Datei `Altova.StyleVisionServer.dll` der .NET-Schnittstelle zur Verfügung steht, stehen die StyleVision Server AP-Funktionen in Ihrem .NET-Projekt zur Verfügung.

#### So fügen Sie in einem Visual Studio .NET-Projekt eine Referenz zur StyleVision Server-DLL hinzu

1. Klicken Sie, während das .NET-Projekt in Visual Studio geöffnet ist, auf **Project | Add Reference**. Daraufhin wird das Dialogfeld "Add Reference" geöffnet.
2. Suchen Sie auf dem Register "Browse" nach dem Ordner `<StyleVisionServer application folder>/bin`, wählen Sie `Altova.StyleVisionServer.dll` aus und klicken Sie auf **OK**.

Sie können die Struktur der `Altova.StyleVisionServer` Assembly mit dem Visual Studio Object Browser anzeigen (um den Object Browser aufzurufen, klicken Sie im Menü **View** auf **Object Browser**).

## 1.4.2 Informationen zur COM-Schnittstelle

StyleVision Server wird bei der Installation automatisch als COM-Serverobjekt registriert. Um zu überprüfen, ob die Registrierung erfolgreich war, öffnen Sie den Registry Editor (z.B. durch Eingabe des Befehls `regedit.exe` in der Befehlszeile). Wenn die Registrierung erfolgreich war, enthält die Registry die Klassen `StyleVision.Server`. Diese beiden Klassen befinden sich normalerweise unter `HKEY_LOCAL_MACHINE\SOFTWARE\Classes`.

Sobald das COM-Serverobjekt registriert ist, können Sie es über Applikationen und Programmiersprachen mit Programmierunterstützung für COM-Aufrufe aufrufen. Wenn Sie den Pfad des StyleVision Server Installationspakets ändern möchten, sollten Sie StyleVision Server am besten deinstallieren und anschließend im gewünschten Ordner installieren. Auf diese Art wird die nötige Deregistrierung und Registrierung bei der Installation durchgeführt.

## 1.4.3 Informationen zur Java-Schnittstelle

Um die StyleVision Server API über Java-Code aufrufen zu können, fügen Sie die folgenden Referenzen zur `.classpath`-Datei Ihres Java-Projekts hinzu.

<code>StyleVisionServer.jar</code>	Die Bibliothek, die mit StyleVision Server kommuniziert
<code>StyleVisionServer_JavaDoc.zip</code>	Die Dokumentation zur StyleVision Server API

Beide Dateien befinden sich im Ordner `bin` des StyleVision Server Installationsordners. Sie können die Dateien entweder über ihren Originalordner referenzieren oder sie in einen anderen Ordner kopieren, falls Ihr Projekt dies erfordert.

### Hinzufügen der Bibliotheksreferenzen in Eclipse

Sie können die `classpath`-Referenzen in Eclipse durch Bearbeiten der Eigenschaften des Java-Projekts hinzufügen. Die Beispielanleitung unten gilt für Eclipse 4.4.

1. Klicken Sie, während das Projekt in Eclipse geöffnet ist, auf das Menü **Project** und anschließend auf **Properties**.
2. Klicken Sie auf dem Register "Libraries" auf **Add External JARs** und navigieren Sie zur Datei `StyleVisionServer.jar` im StyleVision Server Installationsordner.
3. Erweitern Sie unter *JARs and class folders on the build path* den Eintrag `StyleVisionServer.jar` und doppelklicken Sie anschließend auf den Eintrag `Javadoc location: (None)`.
4. Stellen Sie sicher, dass die Optionen *Javadoc in archive* und *External file* aktiviert sind und navigieren Sie anschließend zur Datei `StyleVisionServer_JavaDoc.zip` im StyleVision Server Installationsordner.
5. Klicken Sie auf **OK**. Die Referenz zur StyleVision Server Bibliothek und zum Javadoc-Archiv wird zur `.classpath`-Datei des Projekts hinzugefügt.

## 1.4.4 Codebeispiele

Dieser Abschnitt enthält Beispiele für die folgenden Programmiersprachen:

- [C++](#)<sup>48</sup>
- [C#](#)<sup>47</sup>
- [VBScript](#)<sup>51</sup>
- [Visual Basic](#)<sup>53</sup>
- [Java](#)<sup>50</sup>

### 1.4.4.1 C#

Im Beispiel unten wird gezeigt, wie Sie über C#-Code anhand einer PXF-Datei und einer XML-Input-Datei eine RTF-Output-Datei generieren. Stellen Sie sicher, dass StyleVision Server installiert und lizenziert ist und als COM-Serverobjekt zur Verfügung steht. Die Registrierung als COM-Serverobjekt erfolgt normalerweise während der Installation von StyleVision Server. Eine Anleitung, wie Sie überprüfen, ob die Registrierung erfolgreich war, finden Sie unter [Informationen zur COM-Schnittstelle](#)<sup>46</sup>.

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

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

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

        //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);
    }
    catch (System.Runtime.InteropServices.COMException ex)
    {
        // some general error like an invalid license happened
        System.Console.WriteLine("Internal Error - " + ex.Message);
    }
}
}
}
}
}

```

### 1.4.4.2 C++

Im Beispiel unten wird gezeigt, wie Sie über C++-Code anhand einer PXF-Datei und einer XML-Input-Datei eine RTF-Output-Datei generieren. Stellen Sie sicher, dass StyleVision Server installiert und lizenziert ist und als COM-Serverobjekt zur Verfügung steht. Die Registrierung als COM-Serverobjekt erfolgt normalerweise während der Installation von StyleVision Server. Eine Anleitung, wie Sie überprüfen, ob die Registrierung erfolgreich war, finden Sie unter [Informationen zur COM-Schnittstelle](#)<sup>46</sup>.

```

// StyleVisionServerAPI_Sample.cpp : Defines the entry point for the console application.
//
#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[])
{
    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
        // pSVS->InputXML = "ExpReport.pxf|zip\\ExpReport.xml";
        // Easiest way is to refer to the file as being embedded in the
transformation file
        pSVS->InputXML = "altova://packagedfile/ExpReport.xml";

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

        //Prepare the parameters, if your design uses parameters
        //pSVS->AddParameter( "testparam1", "value 1" );

        //Run the transformation; the output will be stored at C:\\temp\\ExpReport.rtf
        // NOTE Please adapt the path to the input file in order to run the sample
        if (pSVS->Generate("ExpReport.pxf"))
        {
            std::cout << pSVS->LastExecutionMessage << std::endl;
            std::cout << "Success - finished execution" << std::endl;
        }
    }
}

```

```

        else
            std::cout << pSVS->LastExecutionMessage << std::endl;
    }
    catch ( _com_error& err )
    {
        BSTR bstrMessage;
        (err).ErrorInfo()->GetDescription( &bstrMessage );
        std::cout << "Exception occurred: " <<
        _com_util::ConvertBSTRToString( bstrMessage ) << std::endl;
    }

    CoUninitialize();
    return 0;
}

```

### 1.4.4.3 Java

Im Beispiel unten wird gezeigt, wie Sie über Java-Code anhand einer PXF-Datei und einer XML-Input-Datei eine RTF-Output-Datei generieren. Stellen Sie sicher, dass StyleVision Server installiert und lizenziert ist und als Serverobjekt zur Verfügung steht. Die Registrierung als Serverobjekt erfolgt normalerweise während der Installation von StyleVision Server. Eine Anleitung, wie Sie überprüfen, ob die Registrierung erfolgreich war, finden Sie unter [Informationen zur Java-Schnittstelle](#)<sup>46</sup>.

```

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

            //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
            (same dir with the StyleVisionServer.jar)
            //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
// NOTE Please adapt the path to the input file in order to run the sample
if ( objSVS.generate( "C:\\Program Files (x86)\\Altova\\StyleVisionServer2021\\
etc\\Examples\\ExpReport.pxf" ) )
    System.out.println( "Success - finished execution" );
else
    System.out.println( objSVS.getLastExecutionMessage() );
}
catch ( Exception e)
{
    e.printStackTrace();
}
}
}

```

#### 1.4.4.4 VBScript

Im Beispiel unten wird gezeigt, wie Sie über VB Script-Code anhand einer PXF-Datei und einer XML-Input-Datei eine RTF-Output-Datei generieren. Stellen Sie sicher, dass StyleVision Server installiert und lizenziert ist und als COM-Serverobjekt zur Verfügung steht. Die Registrierung als COM-Serverobjekt erfolgt normalerweise während der Installation von StyleVision Server. Eine Anleitung, wie Sie überprüfen, ob die Registrierung erfolgreich war, finden Sie unter [Informationen zur COM-Schnittstelle](#)<sup>46</sup>.

##### Option Explicit

```

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

```

```
    If Err.Number <> 0 then
        WScript.Echo("You are running in a 64-bit process but StyleVision Server COM-
API 64-bit seems not to be installed on your system.")
        WScript.Echo("If you have installed 32-bit StyleVision Server consider calling
your script from the 32-bit console 'C:\Windows\SysWOW64\cmd.exe.'")
        WScript.Quit -1
    End If
End If
On Error Goto 0          ' re-enable default error promotion

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

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

' The Generate method will return 'True' if generation was successful otherwise 'False'.
' In the case of fundamental errors like termination of the server process a COM error will
be raised which
' can be handled using the VBScript Err object.
On Error Resume Next ' ignore any COM errors avoiding uncontrolled script termination
Err.Clear

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

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

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

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

' Run the transformation. The PXF file path can be relative to the working folder or
absolute.
WScript.Echo("Generating output from ExpReport.pxf...")
```

```

If ( objSVS.Generate( "ExpReport.pxf" ) ) Then
    WScript.Echo( objSVS.LastExecutionMessage )
    WScript.Echo( "Success - finished execution" )
Else
    WScript.Echo( objSVS.LastExecutionMessage )
End If

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

On Error Goto 0          ' re-enable default error promotion

```

### 1.4.4.5 Visual Basic

Im Beispiel unten wird gezeigt, wie Sie über C++-Code anhand einer PXF-Datei und einer XML-Input-Datei eine RTF-Output-Datei generieren. Stellen Sie sicher, dass StyleVision Server installiert und lizenziert ist und als COM-Serverobjekt zur Verfügung steht. Die Registrierung als COM-Serverobjekt erfolgt normalerweise während der Installation von StyleVision Server. Eine Anleitung, wie Sie überprüfen, ob die Registrierung erfolgreich war, finden Sie unter [Informationen zur COM-Schnittstelle](#)<sup>46</sup>.

```
Option Explicit On
```

```
Module Program
```

```
    Sub Main()
```

```
        Try
```

```
            'Create a StyleVision Server object
```

```
            Dim objSVS As Altova.StyleVisionServer.Server = New
```

```
Altova.StyleVisionServer.Server
```

```
            'Set a working directory - used for output and for intermediate files
```

```
            'objSVS.WorkingDirectory = "C:\Program Files (x86)
```

```
\Altova\MapForceServer2020\etc\Examples"
```

```
            objSVS.WorkingDirectory = "..\..\.."
```

```
            'Default path to the StyleVision Server executable is the installation path
            (same dir with the StyleVisionServer.dll)
```

```
            'In case you moved the binaries on the disk, you need to explicitly set the
            path to the .exe file
```

```
            'objSVS.ServerPath = "C:\Program Files (x86)
```

```
\Altova\StyleVisionServer2020\bin\StyleVisionServer.exe"
```

```
            'objSVS.ServerPath = "C:\Program
```

```
Files\Altova\StyleVisionServer2020\bin\StyleVisionServer.exe"
```

```
            'Prepare the name of the working XML
```

```

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

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

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

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

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

End Sub

End Module

```

## 1.4.5 API-Referenz

Dieser Abschnitt ist eine Benutzerreferenz für die StyleVision Server API.

- [COM und .NET](#)<sup>54</sup>
- [Java](#)<sup>61</sup>

### 1.4.5.1 COM und .NET

Die StyleVisionServer API stellt die **Schnittstelle IServer**, zur Verfügung, über die Sie eine neue StyleVision Server-Objektinstanz erstellen und Zugriff auf StyleVision Server erhalten.

Die Schnittstelle IServer hat die folgenden Methoden und Parameter.

## Methoden

### ▼ AddParameter

Weist einem in der PXF-Datei definierten Parameter einen Wert zu.

#### ▣ C#

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

#### ▣ C++

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

#### ▣ VB

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

### ▼ ClearParameterList

Löscht die Liste der Parameter.

#### ▣ C#

```
void ClearParameterList()
```

#### ▣ C++

```
HRESULT ClearParameterList();
```

#### ▣ VB

```
Sub ClearParameterList()
```

### ▼ Generate

Generiert anhand der mit `TransfPath` definierten PXF-Datei eine oder mehrere Ausgabedateien (HTML, PDF, RTF, und/oder DOCX). Der Befehl transformiert die XML-Input-Datei (in der PXF-Datei der XML-Arbeitsdatei) anhand des in der PXF-Datei enthaltenen XSLT-Dokuments. Gibt bei Erfolg `TRUE`, andernfalls `FALSE` zurück.

#### ▣ C#

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

#### ▣ C++

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

#### ▣ VB

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

## Eigenschaften

### ▼ APIMajorVersion

Ruft die Hauptversion der StyleVision Server API ab. Diese unterscheidet sich eventuell von der Produktversion, wenn die API mit einem anderen Server verbunden ist.

#### ▣ C#

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

#### ▣ C++

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

#### ▣ VB

```
ReadOnly Property APIMajorVersion As Integer
```

### ▼ APIMinorVersion

Ruft die Nebenversion der StyleVision Server API ab. Diese unterscheidet sich eventuell von der Produktversion, wenn die API mit einem anderen Server verbunden ist.

#### ▣ C#

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

#### ▣ C++

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

#### ▣ VB

```
ReadOnly Property APIMinorVersion As Integer
```

### ▼ APIServicePackVersion

Ruft die Service Pack-Version der StyleVision Server API ab. Diese unterscheidet sich eventuell von der Produktversion, wenn die API mit einem anderen Server verbunden ist.

#### ▣ C#

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

#### ▣ C++

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

#### ▣ VB

```
ReadOnly Property APIServicePackVersion As Integer
```

### ▼ InputXML

Definiert den Namen und Pfad der zu verarbeitenden XML-Datei (der XML-Arbeitsdatei in der PXF-Datei).



## [-] C#

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

## [-] C++

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

## [-] VB

```
Property InputXML As String
```

## ▼ Is64Bit

Gibt TRUE zurück, wenn der StyleVision Server-Prozessor eine ausführbare 64-Bit-Datei ist.

## [-] C#

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

## [-] C++

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

## [-] VB

```
ReadOnly Property Is64Bit As Boolean
```

## ▼ LastExecutionMessage

Ruft die beim letzten Generate-Befehl erhaltene Meldung auf.

## [-] C#

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

## [-] C++

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

## [-] VB

```
ReadOnly Property LastExecutionMessage As String
```

## ▼ MajorVersion

Ruft die Hauptversion von StyleVision Server ab.

## [-] C#

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

## [-] C++

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

## [-] VB

```
ReadOnly Property MajorVersion As Integer
```

## ▼ MinorVersion

Ruft die Nebenversion von StyleVision Server ab.

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

## ▼ OutputDOCX

Definiert den Namen und Pfad der DOCX-Ausgabedatei.

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

## ▼ OutputFO

Definiert den Namen und Pfad der FO-Ausgabedatei.

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

## ▼ OutputHTML

Definiert den Namen und Pfad der HTML-Ausgabedatei.

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

### ▼ OutputPDF

Definiert den Namen und Pfad der PDF-Ausgabedatei.

#### ▣ C#

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

#### ▣ C++

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

#### ▣ VB

```
Property OutputPDF As String
```

### ▼ OutputRTF

Definiert den Namen und Pfad der RTF-Ausgabedatei.

#### ▣ C#

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

#### ▣ C++

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

#### ▣ VB

```
Property OutputRTF As String
```

### ▼ ProductName

Ruft den Namen des Produkts ab: "StyleVision Server"

#### ▣ C#

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

#### ▣ C++

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

#### ▣ VB

```
ReadOnly Property ProductName As String
```

### ▼ ProductNameAndVersion

Ruft den vollständigen Namen des Produkts einschließlich Versionsnummer ab: "StyleVision Server 2014r2 sp1 (x64)".

#### ▣ C#

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

#### ▣ C++

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

## ▣ VB

```
ReadOnly Property ProductNameAndVersion As String
```

## ▼ ServerPath

Ruft den Pfad zur ausführbaren StyleVision Server Datei ab oder definiert ihn.

## ▣ C#

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

## ▣ C++

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

## ▣ VB

```
Property ServerPath As String
```

## ▼ ServicePackVersion

Ruft die Service Pack-Version von StyleVision Server ab (z.B.: **1** für Altova StyleVision Server 2014 r2 sp1 (x64).)

## ▣ C#

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

## ▣ C++

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

## ▣ VB

```
ReadOnly Property ServicePackVersion As Integer
```

## ▼ WhereClause

Definiert eine SQL WHERE-Klausel, die festlegt, welche Zeilen einer DB-XML-Schemaquelle verarbeitet werden sollen.

## ▣ C#

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

## ▣ C++

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

## ▣ VB

```
Property WhereClause As String
```

## ▼ WorkingDirectory

Ruft das aktuelle Verzeichnis für laufende Aufträge ab oder definiert dieses. Relative Pfade werden anhand der Arbeitsverzeichnisses ausgewertet.

#### ▣ C#

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

#### ▣ C++

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

#### ▣ VB

```
Property WorkingDirectory As String
```

## 1.4.5.2 Java

Das Paket `com.altova.stylevisionserver` besteht aus den folgenden Klassen:

- `public class StyleVisionServer` (*siehe Beschreibung unten*)
- `public class StyleVisionServerException` extends [Exception](#)

### Die Klasse `styleVisionServer`

Die Klasse `StyleVisionServer` erstellt eine neue StyleVision Server Objektinstanz und bietet Zugriff auf StyleVision Server. Im Folgenden finden Sie eine Beschreibung der Methoden der Schnittstelle `StyleVisionServer`.

### Methoden der Klasse `styleVisionServer`

Im Folgenden finden Sie eine alphabetisch geordnete Liste der Methoden der Klasse `StyleVisionServer`.

#### ▼ `addParameter`

```
public void addParameter(String name, String value)
```

Fügt den Namen und Wert eines neuen Parameters hinzu. Jeder Parameter und sein Wert werden in einem separaten Methodenaufruf definiert. Parameter müssen im XSLT-Dokument deklariert sein.

#### Parameter:

`name`: Enthält den Namen des Parameters als String.

`value`: Enthält den Wert des Parameters als String.

#### ▼ `clearParameterList`

```
public void clearParameterList()
```

Löscht die Liste der Parameter.

#### ▼ `generate`

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

Verarbeitet die in `transfPath` definierte PXF-Datei. Löst `StyleVisionServerException` aus.

Parameter:

`transfPath`: eine absolute URL, die den Pfad der PXF-Datei angibt.

Rückgabe:

`true()` bei erfolgreicher Ausführung

`false()` wenn die Ausführung fehlschlägt

Verwenden Sie bei einem Fehler `getLastExecutionMessage()`

▼ `getAPIMajorVersion`

```
public int getAPIMajorVersion()
```

Ruft die Hauptversion der StyleVision Server API ab. Sie unterscheidet sich eventuell von der Produktversion, wenn die API mit einem anderen Server verbunden ist.

▼ `getAPIMinorVersion`

```
public int getAPIMinorVersion()
```

Ruft die Nebenversion der StyleVision Server API ab. Sie unterscheidet sich eventuell von der Produktversion, wenn die API mit einem anderen Server verbunden ist.

▼ `getAPIServicePackVersion`

```
public int getAPIServicePackVersion()
```

Ruft die Service Pack-Version der StyleVision Server API ab. Sie unterscheidet sich eventuell von der Produktversion, wenn die API mit einem anderen Server verbunden ist.

▼ `getLastExecutionMessage`

```
public String getLastExecutionMessage()
```

Ruft die beim letzten generate-Befehl erhaltene Meldung auf.

▼ `getMajorVersion`

```
public int getMajorVersion()
```

Ruft die Hauptversion der Applikation ab.

▼ `getMinorVersion`

```
public int getMinorVersion()
```

Ruft die Nebenversion der Applikation ab.

▼ `getProductName`

```
public String getProductName()
```

Ruft den Produktnamen ab.

▼ `getProductNameAndVersion`

```
public String getProductNameAndVersion()
```

Ruft den vollständigen Namen und die Versionsnummer des Produkts ab.

▼ `getServicePackVersion`

```
public int getServicePackVersion()
```

Ruft die Service Pack-Version von StyleVision Server ab.

▼ is64bit

```
public boolean is64bit()
```

Überprüft, ob die ausführbare Datei eine 64-Bit-Datei ist.

Rückgabe:

true() für StyleVision Server (x64), andernfalls false().

▼ setInputXML

```
public void setInputXML(String path)
```

Definiert die zu verarbeitende XML-Datei. Dabei muss es sich um den Pfad der in der PXF-Datei definierten XML-Arbeitsdatei handeln.

Parameter:

path: Enthält den Pfad der XML-Arbeitsdatei in der PXF-Datei.

▼ setOutputDOCX

```
public void setOutputDOCX(String path)
```

Definiert den Namen und Pfad der zu generierenden DOCX-Datei.

Parameter:

path: Der Name und Pfad der zu generierenden DOCX-Datei.

▼ setOutputFO

```
public void setOutputFO(String path)
```

Definiert den Namen und Pfad der zu generierenden FO-Datei.

Parameter:

path: Der Name und Pfad der zu generierenden FO-Datei

▼ setOutputHTML

```
public void setOutputHTML(String path)
```

Definiert den Namen und Pfad der zu generierenden HTML-Datei.

Parameter:

path: Der Name und Pfad der zu generierenden HTML-Datei.

▼ setOutputPDF

```
public void setOutputPDF(String path)
```

Definiert den Namen und Pfad der zu generierenden PDF-Datei.

Parameter:

path: Der Name und Pfad der zu generierenden PDF-Datei.

▼ setOutputRTF

```
public void setOutputRTF(String path)
```

Definiert den Namen und Pfad der zu generierenden RTF-Datei.

Parameter:

path: Der Name und Pfad der zu generierenden RTF-Datei.

▼ setServerPath

```
public void ssetServerPath(String serverFile)
```

Definiert den Pfad der ausführbaren StyleVisionServer Datei.

Parameter:

serverFile: Der Pfad der ausführbaren StyleVisionServer Datei.

▼ setWhereClause

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

Definiert eine SQL WHERE-Klausel, mit der festgelegt wird, welche Zeilen einer DB-XML-Schemaquelle verarbeitet werden sollen.

Parameter:

whereClause: Die SQL where-Klausel, die angibt, welche Zeilen einer DB-XML-Schemaquelle verarbeitet werden sollen.

▼ setWorkingDirectory

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

Definiert ein Standardverzeichnis. Relative Pfade werden relativ zum Verzeichnis aufgelöst.

Parameter:

workingDirectory: Der Pfad des Standard (Arbeits)-Verzeichnisses



# Index

## C

**Cores, 9**

## E

**Einrichten,**

- auf macOS, 22
- unter Linux, 15
- unter Windows, 10

## F

**FlowForce Server,**

- und StyleVision Server, 7

## I

**Installation auf macOS, 22**

**Installation unter Linux, 16**

**Installation unter Windows, 10**

## L

**Licensing and processor cores, 9**

**Linux,**

- Installation unter, 16

**Lizenzierung auf macOS, 25**

**Lizenzierung unter Windows, 13, 19**

## M

**macOS,**

- Installation, 22

Lizenzierung, 25

## P

**Processor cores, 9**

## S

**Single-thread execution, 9**

**StyleVision Server,**

- Features, 7
- Funktionalität, 7
- im FlowForce Workflow, 7

## W

**Windows,**

- Installation, 10
- Lizenzieren unter, 13, 19