

Altova StyleVision Server 2021 ユーザーマニュアル

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.

公開日: 2015-2021

(C) 2015-2021 Altova GmbH

目次

1	はじめ	っこ	6
0	楼		Q
2	饭用比		0
2.1	FlowFor	ce ワークフロー内	9
2.2	スタンド	アロンのサーバーとして	10
3	Style	Vision サーバーセットアップ	11
3.1	プロセッ	サーコアとライセンス	12
3.2	Window	s でのセットアップ	13
	3.2.1	Windows へのインストール	13
	3.2.2	Windows でのライセンス	16
3.3	Linux で	゙゙のセットアップ	19
	3.3.1	Linux へのインストール	19
	3.3.2	Linux でのライセンス	22
	3.3.3	環境についてのメモ	23
3.4	Mac OS	X でのセットアップ	25
	3.4.1	macOS へのインストール	25
	3.4.2	macOS でのライセンス	27
	3.4.3	環境についてのメモ	28
3.5	追加セ	ットアップメモ	30

4 StyleVision Server コマンドライン

4.1	assignlicense (Windows \mathcal{OH})	33
4.2	exportresourcestrings	35
4.3	generate	37
4.4	help	40
4.5	licenseserver	41

31

4.6	pdfdata	43
4.7	setdeflang	45
4.8	setfopath	46
4.9	verifylicense (Windows $\mathcal{O}\mathcal{H}$)	48
4.10	version	49

5 StyleVision Server API

5.1	.NET イ	ンターフェイスについて	51
5.2	COM 1	ンターフェイスについて	52
5.3	Java 1	ンターフェイスについて	53
5.4	コードの)サンプル	54
	5.4.1	C++	54
	5.4.2	C#	55
	5.4.3	VBScript	57
	5.4.4	Visual Basic	58
	5.4.5	Java	59
5.5	APIレフ	ファレンス	61
	5.5.1	COM と .NET	61
	5.5.2	Java	68

XBRL タクソノミ マネジャー 6

72 6.1 6.2 インストール済みのタクソノミをビューする......80 6.3 6.4 6.5 6.5.1 6.5.2 6.5.3 6.5.4 6.5.5 6.5.6 6.5.7

50

6.5.8	uninstall	88
6.5.9	update	89
6.5.10	upgrade	90

インデックス

91

1 はじめに

Altova StyleVision Server は、<u>Altova StyleVisionの</u>内蔵の実行エンジの実装です。Altova の<u>FlowForce Server</u>®の のモジューリとして作動し、<u>スタイドアロンサーィー製品</u>¹⁰としても使用することができます。

StyleVision Server は、<u>FlowForce Server</u> (3) にデルイされた変更パッケージを変換します。これらの変換は、プログラムすることの できるタイムトリガー、ファイリルリガー、おけましモートトリガーなどをベースにして<u>FlowForce Server</u> により開始されます。更に、 StyleVision Server の機能は、<u>コマイドライン</u>(10)により呼び出すこともできます。

システム必要条件、インストール、とライセンス

StyleVision Server は以下のオペレーティングシステム上でサポートされています:

<u>Windows</u>

プラオフォーム更新済みのWindows 7 SP1、Windows 8、Windows 10

<u>Windows Server</u>

プラオフォーム更新済みのWindows Server 2008 R2 SP1 おけお以降

- ▼ Linux
 - CentOS 7 おより以降
 - RedHat 7 おより降
 - Debian 8 おお以降
 - Ubuntu 16.04 LTS おけお以降

次のライブラルコアプリケーションをインストール実行するためコ必要とされるライブラルです。下のッケージが使用中 Linux のマシンで 使用できない場合、yum(おけよ、適用できる場合、apt-get を) コマンドを実行してインストールしてください。

₩ - /Ň	CentOS, RedHat	Debian	Ubuntu
LicenseServer	krb5-libs	libgssapi-krb5-2	libgssapi-krb5-2
StyleVision Server	qt5-qtbase-gui, krb5- libs	libqt5gui5, libgssapi- krb5-2	libqt5gui5, libgssapi- krb5-2

メモ Altova チャート機能を使用する場合、チャートが正確に表示されるようこ少なくともフォントのI つがシステム上にインストールされているショントをしているショントをしているショントをしていたののfig ライブラルのfc-list コマンドを使用してくたさい。

▼ <u>macOS</u>

macOS 10.13 おけお以降

Style Vision Server は、32ゼナと64ゼナ Windows マントでご使用サビオます。インストールとライセンス引まする詳細は、 次を参照してけさい Windows、¹³ Linux、¹⁹ とmacOS²⁵ セナアプ。 メモ 記入可能な PDF の記入可能なパトが、macOS システム上で開かれ、PDF が不足している場合、Java 6 がマン上に インストールされて ないてとか原因の場合がおります。これが原因の場合、<u>https://support.apple.com/kb/dl1572?</u> <u>locale=en_US</u> から Java 6 をインストールすることができます。 Java 6 よ凡新しい いージョンがシステム上に既にインストール されている場合、Java 6 バージョンよりた古い いージョンをインストールした場合でも、システムのデフォルトバージョンとなる新しい バージョンの作動「影響を与えません。

最終更新日: 2021年02月24日

2 機能

StyleVision Server は、XML ファイルを出力 HTML、PDF、RTF、とDOCX ドキュメントにXSLT スタイルシートを使用して変換します。これらのXSLT スタイルシートは、スタイルシートデザイナーアプリケーション <u>Altova StyleVision</u> で作成された PXF から取得されます。

StyleVision Serverを以下の2つの方法で使用することができます:

- <u>Altova FlowForce ワークフロー</u>の一部とて使用。<u>Altova FlowForce</u>に関しての詳細は <u>Altova Web サイト</u>を参照してくたさい。
- コマボラインインターフェイス(CLI)を使用してアクセスするスタンドアロンサーバー製品¹⁹として使用。

XML 入力ファイルとPXF ファイルは 必要な出力ドキュメトを生成する Style Vision Server に やされます。

2.1 FlowForce ワークフロー内

FlowForce ジョブは、<u>Altova FlowForce Server</u>内で作成されます。FlowForce ジョブお以下を指定します:(i) StyleVision Server 変換の入力と出力。および(ii) 一日の特定の時間など特定の時間にジョブが実行されるためのトリガー。実行時にAltova FlowForce Server は、変換命令の変換を実行するStyleVision Server についします。

FlowForce ワークフロー内でのStyleVision Server の役割は、下の図に示されています。(FlowForce ジョブは、実行のさめに <u>Altova MapForce Server</u> に送られた<u>Altova MapForce</u> マンピングを作成することができるため、ワークフロー内でのMapForce Server の役割は表示されています)。



FlowForce ジョブによ呼び出すコま、コマイラインを使用して、StyleVision Server を呼び出すことができます。使用方法は、 StyleVision Server コマイライン⁽³¹⁾のセグションで説明されています。

2.2 スタンドアロンのサーバーとして

Style Vision Server をスタイアロン製品とて、Windows、Linux、およびmacOS システムにインストールすることができます。この バージョンでは、コマイドラインを使用してのみ機能が呼び出されます。使用方法は、<u>Style Vision Server コマイドライン</u>30のセクション て説明されています。

StyleVision サーバーセットアップ 3

このセグタンではStyleVision Server のセオアップのための手順について説明されて、ます。以下について説明されてます

- プロセッサーコアとライセンス¹² に関する情報
 Windows¹³ システム上のStyleVision Server のインストールとライセンス
- ・ Linux¹⁹システム上のStyleVision Server のインストールとライセンス
- macOS²⁵ システム上のStyleVision Server のインストールとライセンス •

3.1 プロセッサーコアとライセンス

Altova サード、製品へのライセス供与は(ロジカルはコアの数量ではなく製品マシントで使用することのできる物理的なプロセッサーの コアの数量をベースしています。例えば、デュアルコアプロセッサーイは、ケアドコアプロセッサー4コア、ヘキサコアはコアなどの2つのコアが存在 します。製品にライセンス供与されたコア数は、物理ませば仮想マシントに関わらず、サードーマシントで使用することのできるコアの数量と同 数、ませは、それ以上である必要があります。例えば、サードーが8コア持つ場合、8コアライセンを購入する必要があります。コアの数量を 満たすためにライセンスを組み合わせることもできます。ですから、2つの4コアライセンスを使用して、8コアライセンスの代わり18コアを達成する こともできます。

処理する容量は少ないが、CPUコアの大きなコンピューターサーバーを使用している場合、少ない数量のコアに割り当てられている仮想マ シンを作成し、その数量のライセンスを購入することもできます。このようなデプロイの場合、もちろん、コンピューターのすべてのコアを使用する場 合に上べこのようなデプロイの処理スピードは遅いです。

メモ 各 Altova サーバー製品ライセンスは、使用されていないライセンス供与能力が存在する場合でも、Altova サーバー製品がインストールされているマシンで、一度にコンのグライア・トマンンのためにつかり使用することができます。例えば、10 コアライセンスが6 CPU コアを持つマシンのために使用されている場合、残りのライセンス許与能力の4 コアは、他のグライア・トマシンのために同時 に使用することとできません。

単ースレイ実行

Altova サードを製品が単ースレドの実行を許可すると*単ースレドの実行を*使用できるようスないます。この場合、Altova サードを製品ライセンスようイセンスプール内で「コアのみ使用することができます。複数のコアを持つマシンムエの「コアライセンス」書い当てることができます。このような場合、その製品を作動するマシンム単一コアで作動します。(マルチコアでは使用可能な)複数スレンドを使用できな、ケッシ処理 「よこのような場合、その製品を作動するマシンム単一コアで作動します。

マルチコアマシノに単一コラティセンスを割り当てるコよその製品のためにLicenseServer内で「単一スレンド実行に制限する」チェックボックスを選択してくたさい。

必要なコアの見積もり

サーバーが扱うことのできるデータの量と処理回数を影響する多くの外部要素が存在します。(例えば、ハードウエア、CPU 上の現在のコード、サーバー上で作動する他のアプリケーションのメモルの割り当てなどか挙(応れます)。パフォーマンスを正確に測定するために、データの量、 条件、およびビジネスの環境にらん状態でアプリケーションをテストしてくたさい。

3.2 Windows でのセットアップ

このセグコンはWindows システムへのStyleVision Server のインストール 13とライセンス 16 について説明します。

Windows ヘクインストール

- システム必要条件
- インストールStyle Vision Server 13
- Altova LicenseServer
 ¹³
- LicenseServer 1-2-2-13
- · トライアルライセス¹³
- アプケーションフォルダーの場所¹³

Windows でのライセンス

- ServiceControllerの開始¹⁰
- LicenseServer の開始¹⁶
- StyleVision Serverの登録¹⁶
- ライセースの割り当て16
- メモ バージョン2021 以降から Style Vision Server の32 ビナバージョンは64 ビナバージョンにインストールすることはできません。 ん、おとは64 ビナバージョンを32 ビナバージョンにインストールすることはできません。(i) 新しい デジョンをインストールする前 「古い デジョンを削除する、おとは(ii) 古いインストールと同じビナバージョンを新規のデージョンに更新する必要があります。

3.2.1 Windows へのインストール

StyleVision Server はWindows システムへインストールすることができます。インストールとセナアップについては以下で説明されます。

- ▼ システム必要条件
 - ▼ <u>Windows</u> プラオフォーム更新済みのW indows 7 SP1、W indows 8、W indows 10
 - Windows Server

プラオフォーム更新済みのWindows Server 2008 R2 SP1 ませお以降

▼ StyleVision Serverのインストール

StyleVision Server のW indows システムへのインストールは以下の手順で行います:

- 個別のスタボアロンサー、一製品はStyleVision Serverと称されます。StyleVision Serverをインストールするこは、StyleVision Serverのインストーラーをダウレロードして実行してくたさい。スクレーンの手順に従ってくたさい。
- <u>FlowForce Server インストール シケージの一部とて</u>、<u>FlowForce Server</u> パケージの一部とて StyleVision Server をインストールするゴよ、FlowForce Server インストーラーをダウンロードして実行します。 StyleVision Server のインストールオプションを確認して、スクレーンの手順に従ってくたさい。

StyleVision Server と<u>FlowForce Server</u>のインストーラーはAltova ダウンロードセクー (<u>http://www.altova.com/ja/download.html</u>)で入手でき、必要な登録手続きともに製品をインストールすることができます。 インストール後、実行可能な Style Vision Server はデフォルトで以下で見つけることができます:

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

COM インターフェイスおよび、NET 環境を介して Style Vision Server を使用するために必要な登録はインストーラにより行われます。

Windows Server CONJAN-IL

W indows Server Core はGUI 機能の多くを使用しな 最小のW indows インストールです。StyleVision Server を W indows Server Core マンノコリアのようにインストールすることができます:

- Altova W eb サイトから Style Vision Server インストーラー実行可能ファイルをダンロードします。このファイルは Style Vision Server <version >. exe. とう名前です。サーバープラナトフォーム(32-bit おけは64-bit)に一致 する実行可能ファイルが選択されていることを確認してくたさい。
- 2. (Windows Server Core マンクロおい、標準のWindows マンクロコマト StyleVisionServer<version>.exe /u を実行します。これは.msi ファイルをインストーラー実行ファイルと同 じフォルダーで解凍します。
- 3. 解凍されてない.msi ファイルをWindows Server Core マンノニピーします。
- 4. Style Vision Server の前の、ージョンを更新している場合、次のステップを継続する前にStyle Vision Server を シャトダウムてくたさい。
- 5. コマド msiexec /i StyleVisionServer.msi を実行してインストールのために.msi ファイルを使用します。 Windows Server Core でのインストールにと開始されます。

.msi ファイルを同じ場所に保管します。インストールを後にアンインストール、修復、おけま変更する場合にこれらのファイルが必要になります。

インストールの戻り値をテストするコよ以下のスクリプトを実行します。戻りコードはSerrorlevels 環境関数の中にあります。0 の値を持つターシコードは成功を意味します。

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

戻りロードとインストールプロセルのログを持つサイレントインストールの場合、以下を実行します

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

インストールを変更する場合、以下を実行します:

msiexec /m StyleVisionServer.msi

インストールを修復する場合、以下を実行します:

msiexec /r StyleVisionServer.msi

StyleVision Server をアンインストールするコよ、以下を実行します:

msiexec /x StyleVisionServer.msi

StyleVision Server をサイレントにアンインストール、ログファイル内に詳細な結果をレポートするコよ、以下を実行します:

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

メモ Windows Server Core ではStyleVision Server のチャート と、ーコード 機能を使用することはできません。 タクノノミを インストーリオる コは タクノノミパッケージマネジャーをコマードラインを使用して実行してくたさい。 これを行う情報に関しては StyleVision Server マニュアルを参照してくたさい。

Altova LicenseServer

- Style Vision Server が作動するためコよ ネットワークのAltova License Server からライセンスを供与される必要が あります。
- StyleVision Server おさま<u>FlowForce Server</u>をWindows システムにインストールするさま、Altova LicenseServerをStyleVision Server おさま<u>FlowForce Server</u>をダウンロードしてインストールするオプンョンがあ ります。
- Altova LicenseServer が既にネットワークにインストールされている場合、新しいシンゴンのAltova LicenseServer が必要ない限り、再度インストール必要はありません。(次のポイント?, <u>LicenseServer のシージョンを参照してくたさい</u> (1))
- StyleVision Server おさた
 FlowForce Server のインストール中、適宜 Altova LicenseServer のインストールの オプションをチェックしてくたさい。
 - a. Altova LicenseServer かぜたんストールされていない場合、デフォルの設定を保持してくたさい。ウィザードが最新バージョンのAltova LicenseServer をウィザードを作動しているコンピューターにインストールします。
 - b. Altova LicenseServer かおごんストールされていなく、他のエピューターにAltova LicenseServer をんスト ールする場合、「使用中のコンピューターにAltova LicenseServer をインストールする」チェックボックスをクリ アして、「後で登録」を選択します。この場合、Altova LicenseServer のんストールと StyleVision Server の登録を個別に行う必要があります。
 - c. Altova LicenseServer が既立使用中のエビューターにインストールされているが、インストールウィザードにようた れている、・・ジョンより共低い場合は、デフォルトの設定を保持してくたさい、この場合、インストールウィザードが自動 的にLicenseServer バー・ジョンをダイアログボックスに表示されたパー・ジョンにアップグレードします。既存の登録およ びライセンス情報はアップグレードの後も保存されます。
 - d. Altova LicenseServer か既に使用中のエンピューター、おけ、 ネトワークにインストールされており、 ウィザードに 示される デジョンと同じの場合は、以下を行ってくたさい
 - i. 「使用中のエンピューターにAltova LicenseServer をインストールする」チェックボックスをクリックしてクリアします。
 - ii. 「この製品を登録する」からStyleVision Server を登録するAltova LicenseServer インスタンスを選択、おけよ「後で登録」を選択してくたさい。LicenseServer への関連を無視し、StyleVision Serverのインストールを継続する場合、「後で登録」を常に選択することができます。

Style Vision Server のAltova License Server への登録とライセス供与の詳細に関しては、セクション Windows てのライ セス¹⁵を参照してくたさい。

- LicenseServer バージョン
 - Altova サート製品はインストールされた Style Vision Server ハージョン ご適切な License Server の ージョン ま たは、License Server の最新の ージョン が必要です。
 - StyleVision Server の特定の デジョンゴ 適切な LicenseServer の デジョンか StyleVision Serverの インストー ル中表示されます。
 - LicenseServer の新しい・ジョンをインストールする前に、古い・ジョンはアンインストールされる必要があります。
 LicenseServer インストーラーは古い・ジョンを検出すると自動的に行います。
 - LicenseServer バージョンは下位互換性があります。StyleVision Server の全ての古し バージョンと作動します。
 - StyleVision Server の新しい、シジンをインストールする場合、そして、インストールされている LicenseServer バージョンが適切な LicenseServer バージョンより店に場合、Altova Web サイトから利用可能な最新バージョンをインストールます。
 - LicenseServerをアンインストールする際、古しい・ジョンのLicenseServerのすべての登録とライセンス情報はサーバーマシーンのデータベースに保存されます。このデータは新しい・ジョンがインストールされる際、自動的に新しい・ジョンにインポートされます。
 - 現在インストールされているLicenseServerの
 ・ジョン番号はLicenseServer構成ページ(全てのタブ)の下部に かます。

現在のバージョン: 3.6

▼ トライアルライセス

インストール中、30日間のStyleVision Server のトライアルライセンスをリケエストすることができます。リケエストを送信すると、登録した電子メールアドレストライアルライセンスが送信されます。

▼ アプケーション フォルダの場所

アプリケーションは以下のフォルダーにインストールされます

Windows 7、8、10	C:\Program Files\Altova\
64 ビナ OS 上の32 ビナバージョン	C:\Program Files (x86)\Altova\

3.2.2 Windows でのライセンス

StyleVision Server は作動するためにAltova LicenseServer にライセンスされている必要があります。ライセンス供与は以下の2 つのステップから構成されています:

- 1. StyleVision Server にLicenseServer を登録します。StyleVision Server から登録することができます。
- 2. StyleVision Server のライセンスを割り当てます。ローカルズンよけよ、ネトワーク上のマンノにインストールされる必要のある LicenseServer からライセンスを割り当てることができます。

実行するステップは以下に手短に説明されています。詳細は、Altova Web サイトのユーザーマニュアルを参照してくたさい。

▼ ServiceController の開始

Altova ServiceController はAltova LicenseServerを開始するためつど須です。

Altova ServiceController(略してServiceController)はWindowsシステム上でAltovaサビスを便利に開始、停止、 構成できるアプリケーションです。

ServiceController は Altova LicenseServer とおよび サービスとし てインストールされる Altova サーバー 製品 (FlowForce Server, RaptorXML(+XBRL) Server, and Mobile Together Server).スタート | Altova LicenseServer | Altova ServiceController をクトックして開始されます。(このコマンドは「開始」 メニューフォルダーでも使用することができます サービスとし てインストールされる Altova サーバー 製品(FlowForce Server, RaptorXML(+XBRL) Server, and Mobile Together Server)。)ServiceController が開始した後、システムトレイからアクセスすることができます。 (下部スクリーンショント)。



システムログイン時にServiceController の自動開始を指定するコスシステムレイのServiceController アイエをクトク して ServiceController メニューを表示します(下部スクノーンショナ)。スタートアップ時にAltova ServiceController を作動する(Run Altova ServiceController at Startup)コマンドに切り踏えます。(このコマンドはデフォリトで切り踏えられてい ます。)ServiceController を終了するコスシステムトレイのServiceController アイエンをクトックして、表示されるメニューから Altova ServiceController の終了(Exit Altova ServiceController)をクリックします(下部スクノーンショナ参照)。



▼ LicenseServer の開始

LicenseServer を開始するコンステムトレイの [ServiceController] アインをクリックします。メニューの [Altova LicenseServer] にポイントすると、(ア部スクリーンショント参照) がポップアップします。サブメニューから [Start Service] (サー ビスの開始) を選択します。LicenseServer か既に作動している場合、Start Service オプションは無効化されます。

۲	Altova FlowForce Server	
$(\mathbf{\hat{c}})$	Altova FlowForce Web	
A	Altova LicenseServer	Configure
Ø	Altova MobileTogether Server	Start service
<u>@</u>	Altova RaptorXML+XBRL Server	Stop service
	Exit Altova ServiceController	
~	Run Altova ServiceController at startup	
EI	N 🔺 🗛 之 🙀 🏟 11:00 AM	

▼ StyleVision Server の登録

StyleVision Server をコマイ・ラインインターフェイスから登録する場合、licenseserver コマイを使用します: StyleVisionServer licenseserver [options] ServerName-Or-IP-Address

例えば localhost がLicenseServer がインストールされているサードの名前の場合、: StyleVisionServer licenseserver localhost

StyleVision Server が<u>FlowForce Server</u>のインストールの一部としてインストールされる場合、 LicenseServer への FlowForce Server の登録の際、自動的に StyleVision Server も登録されます。

FlowForce Server ドキュメンテーションにFlowForce Server の登録の手順が説明されています。原則: (i) Altova FlowForce W eb をサービスとして ServiceController から開始します、前述のポイント参照; (ii) パワードを入力してセオ アップページにアクセスします; (iii) LicenseServer 名を選択ませは[LicenseServer により登録] をクリックします。 登録に成功した後、LicenseServer 構成ページのServer Management (サーバーの管理)タブへ移動して StyleVision Server ヘライセンを割り当てます。

▼ ライセスの割川当て

StyleVision Server の登録に成功した後、LicenseServer の構成ページのClient Management (クライアトの管理タブ)にリストされます。移動して StyleVision Server にライセンスの割り当てを行います。

Altova サードー製品へのライセスは製品マンで使用可能なプロセッサコアの数をベースしています。例えば、デュアルコアプロセッサはコアが2 つ、ケアド、コアプロセッサはコアが4 つ、ヘキサコアプロセッサはコアが6 つ等々。特定のサードーマシンとの製品 にライセスされたコアの数は、物理ませば仮想マンで、サードーで使用可能なコア数よ比多くませば同数である必要があります。例 えば、サードーが8 コズオクタルコアプロセッサの場合、少なくとも8-コアライセンを購入する必要があります。また、ライセスを合 計してコア数を満たすこともできます。2 つの4-コアライセンスは、8-コアライセンスの代わりにコオクタルコアサードーで使用できます。

大きいCPU コアを持つエピューターサーバーを使用し、少量を処理する場合、少ないロアを割り当てる仮想マンンを作成し、その数のライセンスを購入することもできます。このようなデプロイは、もちろん、サーバーの全ての利用可能なコアか利用されている場合によいべ、処理スピードが落ちます。

メモ 各 Altova サーバー製品のライセンスは、使用されていないライセンス容量があっても、1度に1つのクライアトイマンメンだ いたか使用することができません。例えば10-コアライセンスが6 CPU コアのクライアトトマンメン使用される場合、残りの 4 コアライセンスは他のマンンで同時に使用することができません。

3.3 Linux でのセットアップ

このセグションはLinux システム(Debian、Ubuntu、CentOS、RedHat) へのStyleVision Server の<u>インストール19 とライセン</u> ス⁽²²⁾ について説明します。

Linux へのインストール¹⁹

- システム必要条件¹⁹
- Altova サーバー製品の古し バージョンのアンインストール
- Linux パッケージのダウノロード¹⁹
- <u>StyleVision Serverのインストール¹⁹</u>
- Altova LicenseServer
 ¹⁹
- <u>LicenseServer</u> の ーション¹⁹
- トライアルライセス¹⁹

Linux でのライセンス 22

- <u>LicenseServer</u>の開始²²
- <u>StyleVision Server の登録</u>²²
- <u>ライセノスの割り当て</u>23

環境についてのメモ23

3.3.1 Linux へのインストール

StyleVision Server のLinux システムへのインストールは利用可能です。インストールとセナアップこしては以下で説明されます。

▼ システムの必要条件

- <u>Linux</u>
 - CentOS 7 おより降
 - RedHat 7 おけお以降
 - Debian 8 お北以降
 - Ubuntu 16.04 LTS おけお以降

次のライブラルはアプリケーションをインストール実行するオーカロ必要とされるライブラルです。下のパッケージが使用中 Linux の マンンで使用できない場合、yum(およ、適用できる場合、apt-get を)コマイを実行してインストールしてくけさい。

₩ - ,,,	CentOS, RedHat	Debian	Ubuntu
LicenseServer	krb5-libs	libgssapi-krb5-2	libgssapi-krb5-2
StyleVision Server	qt5-qtbase-gui, krb5-libs	libqt5gui5, libgssapi- krb5-2	libqt5gui5, libgssapi- krb5-2

メモ Altova チャート機能を使用する場合、チャートが正確に表示されるように少なくともフォントのI つがシステム上にインスト ールされている必要があります。インストールされているフォントをリストするには、例えば<u>Fontconfig ライブラルのfc-list</u>コ マドを使用してくたさい。 ▼ FlowForce Server 統合

StyleVision Server をFlowForce Server と共にインストーレする場合、FlowForce Server を最初にインストーレすること か奨励されます。それ以外の場合、StyleVision Server とFlowForce Server の両方をインストールナ後、以下のコマドを 実行してくたさい

cp /opt/Altova/StyleVisionServer2021/etc/*.tool /opt/Altova/FlowForceServer2021/to
ols

このコマドは.tool ファイルを Style Vision Server のetc デルクトリカら Flow Force Server /tools デルクトリニピーしま す。Flow Force Server は.tool ファイルを必要とこのファイルは Style Vision Server 実行可能ファイルへの ひを含みま す。Flow Force Server を Style Vision Server をインストールする前にインストールする場合このコマドを実行する必要にあり ません。

▼ Altova サードー製品の古し V デジョンアンインストールする方法

前のデジュをアンインストールする場合、以下の手順を踏んでたさい。Linux コマボラインインターフェイス(CLI)で、Altova サーチ製品がインストールされているか、以下のコマボで確認できます:

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

Style Vision Server がインストールされていない場合、以下の Style Vision Serverのインストールで説明されている手順を踏んでくたさい。

StyleVision Server か既にインストールされており、StyleVision Server の新しいデジョンをインストールたい場合、古い、ージョンを以下のコマイドでアンインストールてくたさい

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

Altova LicenseServer の古い、デジンをアンインストールする場合、以下のコマトで行ってくたさい

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

Debian とUbuntu システムでは、StyleVision Server がインストール後にインストール済みの製品としてリストする場合があります。この場合、purge コマイドを実行して StyleVision Server をリストからクリアします。 purge コマイドを 上記の remove コ マイ の*instead 代わりに*使用することもできます。

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

▼ Linux パッケージのダウノロード

以下のStyleVision Server のLinux システムへの シケージは Altova Web サイト で使用可能です。

配布	パッケージ拡張子
Debian	.deb
Ubuntu	.deb
CentOS	.rpm

配布	パッケージ拡張子
RedHat	.rpm

Linux パッケージのダウレロード後、Linux システムご直接コピーしてくたさい。 Style Vision Server を作動するけなは Altova License Server か必要なけな、 <u>Altova Web サイト</u>から Style Vision Server をダウレロードと同時に License Server を ダウレロードと同時に License Server を

▼ StyleVision Server のインストール

ターミナルウイボウで、Linux パッケージをコピーしたい デルクトリン 切替えてください。例えば、MyAltova と称されるユーザーディ レカーリニピーしたとします、(例えば/home/User ディンカーリン存在するとします)、ディレカーを以下のようこスイッチします: cd /home/User/MyAltova

以下のコマイを使用して Style Vision Server インストールする

[Debian]:	sudo dpkginstall stylevisionserver-2021-debian.deb
[Ubuntu]:	sudo dpkginstall stylevisionserver-2021-ubuntu.deb
[CentOS]:	<pre>sudo rpm -ivh stylevisionserver-2021-1.x86_64.rpm</pre>
[RedHat]:	<pre>sudo rpm -ivh stylevisionserver-2021-1.x86_64.rpm</pre>

StyleVision Server パッケージはフォルダーにインストールされます: /opt/Altova/StyleVisionServer2021

Altova LicenseServer

StyleVision Serverを含むAltova サード製品を作動するコよ、サード製品はネオワークでAltova LicenseServer を介して、ライセンを与えられなければなりません。

```
Linux システムでは、Altova LicenseServer は個別にインストールされる必要があります。 Altova Web サイト から
LicenseServer をダウムロードして、パッケージをLinux システムのディレクトリにコピーします。 StyleVision Server 同様インス
トールします、前のステップ参照)。
```

[Debian]: sudo dpkg --install licenseserver-3.6-debian.deb
[Ubuntu]: sudo dpkg --install licenseserver-3.6-ubuntu.deb
[CentOS]: sudo rpm -ivh licenseserver-3.6-1.x86_64.rpm
[RedHat]: sudo rpm -ivh licenseserver-3.6-1.x86_64.rpm

LicenseServer パッケージは以下にインストールされます:

/opt/Altova/LicenseServer

StyleVision Server をAltova LicenseServer で登録して、ライセスを与えることに関しての詳細は、セグヨン Linux でのラ イセス²²を参照してくたさい。

▼ LicenseServer バージョン

- Altova サーバー製品はインストールされた Style Vision Server バージョンに適切な License Server の デジョン、ま けよ License Server の最新の デジョンが必要です。
- StyleVision Server の特定の デジョンゴ適切な LicenseServer の デジョンが StyleVision Serverの インストー ル中表示されます。
- LicenseServer の新しい・ジョンをインストールする前に、古い・ジョンはアンインストールされる必要があります。
 LicenseServer インストーラーは古い・ジョンを検出すると自動的に行います。
- LicenseServer バージョンは下位互換性があります。StyleVision Server の全ての古し バージョンと作動します。

- StyleVision Server の新しい・ジョンをインストールする場合、そして、インストールされている LicenseServer バージョンが適切な LicenseServer バージョンより店に場合、Altova Web サイトから利用可能な最新バージョンをインストールます。
- LicenseServer をアンインストールする際、古いレージョンのLicenseServer のすべての登録とライセンス情報はサーバーマシーンのデータベースに保存されます。このデータは新しいレージョンがインストールされる際、自動的に新しいレージョンにインポートされます。
- 現在インストールされているLicenseServerの、デジン番号はLicenseServer構成ページ(全てのタブ)の下部に あます。

現在のバージョン: 3.6

3.3.2 Linux でのライセンス

StyleVision Server は作動するためにAltova LicenseServer にライセンスされている必要があります。ライセンス供与は以下の2つのステップから構成されています:

- 1. StyleVision Server にLicenseServer を登録します。StyleVision Server から登録することができます。
- 2. Style Vision Server のライセンスを割り当てます。ローカルズンオゴよ、ホトワーク上のマンノにインストールされる必要のある License Server からライセンスを割り当てることができます。

実行するステップは以下に手短に説明されています。詳細は、Altova Webサイトのユーザーマニュアルを参照してくたさい。

▼ LicenseServer の開始

StyleVision Server をLicenseServer に正しく登録しライセンス与えるかりゴよ LicenseServer はネトワークのデーモンとして作動していなければなりません。以下のコマイで、LicenseServer をデーモンとて開始してくたさい

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

LicenseServer を停止する必要がある場合、上記のコマイのstart をstop と置換えてけ違い。例えば: sudo /etc/init.d/licenseserver stop

- ▼ StyleVision Serverの登録
 - コマドラインインターフェイスから Style Vision Server を登録する場合、licenseserver コマドを使用します: sudo /opt/Altova/Style VisionServer2021/bin/style visionserver licenseserver [options] ServerName-Or-IP-Address

例えば localhost がLicenseServer のインストールされたサードの名前である場合: sudo /opt/Altova/StyleVisionServer2021/bin/stylevisionserver licenseserver localhost

上記のコマイで、localhost がLicenseServer のインストールされたサーバーの名前です。StyleVision Server 実 行可能の場所を確認してたさい。 /opt/Altova/StyleVisionServer2021/bin/

登録に成功した後 License Server の構成ページのケライアント管理タブに移動し Style Vision Server にライセンを割り当てます。

▼ ライセノスの割り当て

登録に成功した後 LicenseServer の構成ページのケライアト管理タブに移動し StyleVision Server にライセンを割り当てます。

Altova サード、製品へのライセスは製品マンで使用可能なプロセッサコアの数をベースしています。例えば、デュアルコアプロセッサはコアが2 つ、ケアド コアプロセッサはコアが4 つ、ヘキサコアプロセッサはコアが6 つ等々。特定のサードーマシンとの製品 にライセスされたコアの数は、物理ませば仮想マンで、サードーで使用可能なコア数よ比多くませば同数である必要があります。例 えば、サードーが8 コア、オクタルコアプロセッサの場合、少なくとも8-コアライセンを購入する必要があります。また、ライセスを合 計してコア数を満たすこともできます。2 つの4-コアライセンスは、8-コアライセンスの代わりにコオクタルコアサードーで使用できます。

大きいCPU コアを持つエピューターサー、、そ使用し、少量を処理する場合、少ないコアを割り当てる仮想マシンを作成し、その数のライセンスを購入することもできます。このようなデプロイは、もちろん、サーバーの全ての利用可能なコアが利用されている場合によ比べ、処理スピードが落ちます。

メモ 各 Altova サーバー製品のライセンスは、使用されていないライセンス容量があっても、1度に1つのクライアトイマンスナビ したか使用することができません。例えば10-コアライセンスが6 CPU コアのクライアトトマンス さ使用される場合、残りの 4 コアライセンスは他のマンンで同時に使用することができません。

3.3.3 環境についてのメモ

フォルダー

以下が StyleVision Server セナアップの重要なフォルダーのノストです。

□ <u>インスト</u>−ルル+

/opt/Altova/StyleVisionServer2021/

□ <u>ライセンスファイル</u>

/var/opt/Altova/StyleVisionServer

□ <u>環境設定</u>

/etc/profile.d/jdbc.sh (通常 jdbc.sh ど称される)環境設定ファイルドンステムの起動時に実行されます。内部の定義は、使用する特定の環境に指定 される必要があます。上の、マのサンプルは一般的なガイドとして使用してください。

メモ 環境設定ファイルおシステム上のすべてのユーザーの変数を設定するため、設定する際は注意してくたさい。例えば、このファイル内でクラス・ひを変更すると、変更はシステム全体に適用されます。StyleVision Server ためこのみ変更を加える場合、ユニットファイルの使用を考慮してくたさい(JDBC 接続のセクションで説明されています)。

データベースへの接続

Linux 上では、以下のデータベース接続かりポートされています:

- JDBC Microsoft Access 以外のサポートされているすべてのデータベースにJDBC を使用することができます。
- ネイティブな接続 SQLite とPostgreSQL データベースのために使用することができます。
- •

JDBC を使用する場合、以下の点に注意してくたさい

- Java Runtime Envioronment おはSDK がんストールされる必要があります。
- ターゲオデータベースのためのJDBC-接続がインストールされる必要があります。
- 次の環境変数が環境のために正しく設定される必要がみます:
 - o CLASSPATH: JDBC データベースへ接続する.jar ファイルを検索するココお以下を行います; .jar ファイルを以下として入 カすることができます(i) システムの起動時に実行される(jdbc.sh などの) 実行可能スクレプト、まけよ(ii) StyleVision Server かサービスとして開始される際に実行されるユニットファイル。.jar ファイルを指定するけっかユニットファ イルの使用は、StyleVision Server のJDBC 接続のけっかこの要なファイルが既存のシステムの構成を変更することが使 用できる利点があります。ユニットファイルは下にコストされています。
 - O PATH: JRE を検索します。しかし、インストールによっては必要のよい場合があります。
 - o JAVA_HOME: **ハノストールコ応じて必要な場合設定します**。

重要なファイルのノスト

以下のシェルスクリプト(おけよ ユニ・トファイル) がフォルダー /opt/Altova/StyleVisionServer/etc にこピーされ、既存の構成ファイルを上書きします。必要になじて必要な変更を加えます。 おこ、上のセグション JDBC 接続を参照してください。 青で イライトされた環境固有であり、使用中の環境に合うよ 調整されなければないません

■ シェルスクレプト (ユニットファイル)

```
#- jdbc - environment -
export PATH=/usr/local/jdk1.7.0_17/bin:/usr/lib64/qt-
3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/qa/bin
export JAVA_HOME=/usr/local/jdk1.7.0_17
export
CLASSPATH=/usr/local/jdbc/oracle/ojdbc6.jar:/usr/local/jdbc/oracle/xdb.jar:/usr/local
/jdbc/oracle/xmlparserv2.jar:/usr/local/jdbc/postgre/postgresql-9.0-
801.jdbc4.jar:/usr/local/jdbc/mssql/sqljdbc4.jar:/usr/local/jdbc/iseries/lib/jt400.ja
r:/usr/local/jdbc/mysql/mysql-connector-java-5.1.16-
bin.jar:/usr/local/jdbc/sqlite/sqlitejdbc-
v056.jar:/usr/local/jdbc/Informix_JDBC_Driver/lib/ifxjdbc.jar:/usr/local/jdbc/sybase/
jconn7/jconn4.jar:/usr/local/jdbc/db2/db2jcc.jar:/usr/local/jdbc/db2/db2jcc_license_c
u.jar:./:
```

3.4 Mac OS X でのセットアップ

このセクコンは Style Vision Server のmacOS システムへのインストール25 とライセス27 について説明します。

macOS へのインストール25

- システム必要条件²⁵
- 古いデジョンのAltova サード製品のアンインストール
- macOS パッケージのダウノロード²⁵
- StyleVision Serverのインストール²⁵
- Altova LicenseServer²⁵
- LicenseServer $\mathcal{O} \leftarrow \mathcal{O}^{25}$
- · トライアルライセス25

macOS でのライセンス27

- LicenseServer の開始⁽²⁷⁾
- <u>StyleVision Serverの</u>登録²⁷
- ライセノスの割り当て²⁷

環境についてのメモ 28

3.4.1 macOS へのインストール

StyleVision Server のmacOS へのインストールは利用可能です。インストールとセナアップについては以下で説明されます。

▼ システム必要条件

▼ <u>macOS</u>

macOS 10.13 おより降

▼ FlowForce Server 統合

StyleVision Server をFlowForce Server と共にインストールする場合、FlowForce Server を最初にインストールすること か奨励されます。StyleVision Server がFlowForce Server の前にインストールされている場合、StyleVision Server と FlowForce Server を両方インストールた後に、次のコマイを実行します。

cp /usr/local/Altova/StyleVisionServer2021/etc/*.tool /usr/local/Altova/FlowForceS
erver2021/tools

このコマイは.tool ファイルを Style Vision Server のetc デルクトリから Flow Force Server /tools デルクトリニピーします。 Flow Force Server はtool ファイルを必要と このファイルは Style Vision Server 実行可能への ひを含みます。 Flow Force Server を Style Vision Server をインストールする前にインストールする場合このコマイを実行する必要はありません。

▼ Altova サーバー製品の古いビデョンのアンインストール

アプリケーションターミナルで、%APPNAME%>アイエンを右クリックし、「ごみ箱へ移動」を選択します。アプリケーションはごみ箱に移動されます。しかし、 usr フォルダーからアプリケーションを削除しなければなりません。このためには以下のエマイを使用します:

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

Altova LicenseServerの古い、ージョンをアンインストールする場合、サービスとしての作動を停止しなければなりません。このため にはおい下のコマンドを使用します。

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

サービスが停止されたか確認するコよアクティビティモニターターミナルを開き、LicenseServer がノストレナルにとを確認します。 StyleVision Server の説明と同じ手順でアンイレストールします。

▼ ディスクイメージファイルのダウノロード

Altova Web サイトからディスクイメージファイル(.dmg)をダウンロードします(http://www.altova.com/ja/download.html)。

▼ StyleVision Server のインストール

ダウンロードされナディスクイメージ(.dmg)をクリックして開きます。これによりStyleVision Server インストーラーが使用中のコン ビューター上で新規の仮想ドライブとして表示されます。新規の仮想ドライブ上で、インストーラーパッケージ(.pkg)をダブルクリックしま す。インストーラーウィザードに従します。これらは、続行するオーカロンと要なライセンス使用許諾書への同意が必要なステップを含むすぐ に3理解することのできるステップです。インストール後、ドライブを取り出すコは、右クリックして取り出しを選択します。

StyleVision Server パッケージはフォルダー内にインストールされます /usr/local/Altova/StyleVisionServer2021(アプルケーション・イナル /var/Altova/StyleVisionServer(データファイル・データベースとログ)

Altova LicenseServer

StyleVision Serverを含むAltova サード製品を作動するコよ サーバー製品はネオワークでAltova LicenseServer を介して、ライセンズを与えられなければよりません。

macOS システムでは、Altova LicenseServer は個別にインストールされる必要があります。Altova Web サイト から Altova LicenseServer をダウムロードして、インストーラー・シケージをダブルクトック、インストールを開始します。インストールを続行するためコよ、使用許諾契約書に合意する必要があります。

Altova LicenseServer をAltova Web サイトからダウレロードし個別にインストールすることもできます (http://www.altova.com/ja/download.html)。

LicenseServer パッケージお以下のフォルダーにインストールされます: /usr/local/Altova/LicenseServer

StyleVision Server をAltova LicenseServer に登録し、ライセンを供与するコはmacOS でのライセンス²²⁰のセクションを参照してくたさい。

- LicenseServer バージョン
 - Altova サーバー製品はインストールされた Style Vision Server バージョンに適切な License Server のデジョン、ま はよ License Server の最新の、デジョンが必要です。
 - StyleVision Server の特定の デジョンゴ適切な LicenseServer の デジョンが StyleVision Serverの インストー ル中表示されます。
 - LicenseServer の新しいデションをインストールする前に、古いデジョンオアンインストールされる必要があます。 LicenseServer インストーラーは古いデジョンを検出すると自動的に行います。

- LicenseServer バージョンは下位互換性があります。StyleVision Server の全ての古し バージョンと作動します。
- StyleVision Server の新しい・ジョンをインストールする場合、そして、インストールされている LicenseServer バージョンが適切な LicenseServer バージョンより古い場合、Altova Web サイトから利用可能な最新バージョンをインストールします。
- LicenseServerをアンインストールする際、古しい・・ジョンのLicenseServerのすべての登録とライセンス情報はサーバーマシーンのデータベースに保存されます。このデータは新しい、・・ジョンがインストールされる際、自動的に新しい、・・ジョンにインポートされます。
- 現在インストールされているLicenseServerの、デジン番号はLicenseServer構成ページ(全てのタブ)の下部に あます。

現在のバージョン: 3.6

3.4.2 macOS でのライセンス

StyleVision Server は作動するためにAltova LicenseServer にライセンスされている必要があります。ライセンス供与は以下の2つのステップから構成されています:

- 1. StyleVision Server にLicenseServer を登録します。StyleVision Server から登録することができます。
- 2. Style Vision Server のライセンスを割り当てます。ローカルズンさせま、ネトワーク上のマンメニインストールされる必要のある License Server からライセンスを割り当てることができます。

実行するステップは以下に手短に説明されています。詳細は、Altova Web サイトのユーザーマニュアルを参照してくたさい。

▼ LicenseServer の開始

StyleVision Server をLicenseServer に正しく登録しライセス与えなかけました。 して作動してなければいません。以下のコマボで、LicenseServer をデーモンとて開始してたさい。 sudo launchctl load /Library/LaunchDaemons/com.altova.LicenseServer.plist

LicenseServerを停止する必要がある場合、上記コマイのloadをunloadと置換えてなだい。 sudo launchctl unload /Library/LaunchDaemons/com.altova.LicenseServer.plist

▼ StyleVision Server の登録

コマイラインインターフェイスから Style Vision Server を登録する場合、licenseserver コマイを使用します: sudo /usr/local/Altova/StyleVisionServer2021/bin/StyleVisionServer licenseserver [options] ServerName-Or-IP-Address

例えば、localhost がLicenseServer のインストールされたサーバーの名前である場合: sudo /usr/local/Altova/StyleVisionServer2021/bin/StyleVisionServer licenseserver localhost

上記のコマイで、localhost がLicenseServer のインストールされたサーバーの名前です。StyleVision Server 実 行可能の場所を確認してたさい /usr/local/Altova/StyleVisionServer2021/bin/

登録に成功した後、LicenseServer 構成ページのカライアナト管理タブに移動して、StyleVision Server ヘライセンスを割り当てます。

▼ ライセスの割当て

StyleVision Server の登録に成功した後、LicenseServer の構成ページのカライアナト管理タブロノストされます。移動して StyleVision Server にライセンを割り当てます。

Altova サーバー製品へのライセンスは製品マシンで使用可能なプロセッサコアの数をベースコンロ、ます。例えば、デュアルコアプロ セッサはコアが2 つ、ケアド、コアプロセッサはコアが4 つ、ヘキサコアプロセッサはコアが6 つ等々。特定のサーバーマシン上の製品 にライセンスされたコアの数は、物理ませば仮想マシンで、サーバーで使用可能なコア数より比多くませば同数である必要があります。例 えば、サーバーが8 コバオクタルコアプロセッサの場合、少なくと8.8-コアライセンズを購入する必要があります。また、ライセンズを合 計してコア数を満たすこともできます。2 つの4-コアライセンスは、8-コアライセンズの代わりにコケサーバーで使用できます。

大きいCPU コアを持つエピューターサーバーを使用し、少量を処理する場合、少ないコアを割り当てる仮想マンンを作成し、その数のライセンスを購入することもできます。このようなデプロイは、もちろん、サーバーの全ての利用可能なコアが利用されている場合によ比べ、処理スピードが落ちます。

メモ 各 Altova サーバー製品のライセンスは、使用されていないライセンス容量があっても、1度に1つのクライアトイマンスナビ したか使用することができません。例えば10-コアライセンスが6 CPU コアのクライアトトマンス さ使用される場合、残りの 4 コアライセンスは他のマンンで同時に使用することができません。

3.4.3 環境についてのメモ

フォルダー

以下がStyleVision Server セオアップの重要なフォルダーのリストです。

= <u>1~2+-1~1~+</u>

/usr/local/Altova/StyleVisionServer2021/

∃ <u>ライセンスファイル</u>

/var/Altova/StyleVisionServer

□ <u>環境設定</u>

/Library/LaunchDaemons/com.altova.StyleVisionServer.plist 環境設定ファイルは特定の環境に応じて定義してください。上記の例ののは一般的なガイドとして使用してくたさい。 メモ これらの環境変数は、StyleVision Serverとして定義されたファイルに存在しなくてはなりません。デフォルトではこのファ イルは、次の通りです:

データベースへの接続

MacOS 上では、以下のデータベース接続かりポートされています:

- JDBC Microsoft Access 以外のサポートされているすべてのデータベースにJDBC を使用することができます。
- ネイティブは接続 SQLite とPostgreSQL データベースのために使用することができます。
- •

JDBC を使用する場合、以下の点に注意してくたさい

- Java Runtime Envioronment おけまSDK がインストールされる必要が決ます。
- ターゲオ データベースのためのJDBC-接続がインストールされる必要があります。
- 次の環境変数が環境のために正しく設定される必要がみます:
 - o CLASSPATH: JDBC データベースへ接続する.jar ファイルを検索するコお以下を行します; .jar ファイルを以下として入 カすることができます(i)システムの起動時に実行される(jdbc.sh などの)実行可能スタリプト、おけよ(ii) StyleVision Server かサービスとして開始される際に実行されるユニットファイル。.jar ファイルを指定するナダのユニットファ イルの使用は、StyleVision Server のJDBC 接続のナックロンジ要なファイルが既存のシステムの構成を変更することよく使 用できる利点があります。ユニットファイルは下にリストされています。
 - O PATH: JRE を検索します。しかし、インストールによっては必要のない場合があります。
 - o JAVA HOME: インストールコ応じて必要な場合設定します。

記入可能なPDF 書式のための Java 6

Mac OS システム上でPDF か開かれる際に、記入可能なの記入可能なPDF 部分が不足している場合、最も可能性のある原因とて Java 6 がマシイニインストールされていたい 場合か考えられます。この場合、<u>https://support.apple.com/kb/dl1572?</u> <u>locale=en_US</u> から Java 6 をインストールすることができます。 Java 6 よ児最新の、デジョンがインストールされている場合、 Java 6 バ ージョンよ児店に インストールは、システムのデフォルトの、デジョンである新しし、バージョンとの作業に影響を及りましません。

3.5 追加セットアップメモ

アプリケーションフォルダーのetc\examples ザブフォルダー内のStyleVision Server とッケージされているWindows サンプルを実行する」コよ、StyleVision Server DLL がシステムと共に正確に登録されている必要があります。登録エラーは通常、オーバータイムまは、単一のマシン上に異なるビナル・デジョンのStyleVision Server (32 ビット おけよ64 ビット) がインストールされている場合通常発生します。

StyleVision Server DLL (32 ビナ おは64 ビナ)をWindows マンノこ正確に登録するはよ 以下を行います:

- 1. 管理者モードでコマイプロンプトを開きます。
- 2. DLL がロケートされているフォルダーンゴル替えます。これを行うコマンドは以下の通りです: cd C:\Program Files\Altova\StyleVisionServer2021\bin
- 3. DLL (32ゼナ おは64ゼナ)を登録するために以下のコマドを実行します: regsvr32 StyleVisionServer.dll
- 4. 登録の成功を通知するポップアップが表示されていることを確認してくたさい。
- 5. Visual Studio を開きます。
- 6. StyleVisionServerAPI_Sample.sln を使用してプロジェクトを開きます。
- 7. Program.cs ファイルコは有効ないなたか含まれていることを確認します。
- 8. Ctrl+F5 を使用してプロジェクトを実行します。
- メモ Windows システム上のアプリケーションフォルダーへの 対議常以下の通りです: C:\Program Files\Altova\StyleVisionServer2021.

4 StyleVision Server コマンドライン

StyleVision Server 実行可能ファイルのデフォルトのロケーション

StyleVision Server 実行可能ファイルのデフォルトのロケーションは以下の通りです:

Linux /opt/Altova/StyleVisionServer2021/bin/stylevisionserver

Mac /usr/local/Altova/StyleVisionServer2021/bin/stylevisionserver

Windows <*ProgramFilesFolder*>\Altova\StyleVisionServer2021\bin**StyleVisionServer.exe**

CLI コマドの使用方法とスト

コマボライン構文は以下の通りです

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

- --help(短縮形 --h)は与えられたコマドのヘルプラキストを表示します。コマドに名前がは場合、実行可能ファイルの全てのコマドがコマドの短い説明と共にしていたかます。
- --version はStyleVision Server の、ージョン番号を表示します。
- <command>は実行されるコマメドです。このセクションのサブセクション内でコマメドについて説明されています(のリストを参照してください)。
- [options] はコマドのオプタンです。これらのコマドはしひされ、対応するコマドと共に説明されています。
- [arguments] はマイの引数です。これらのコマイはノストされ、対応するコマイと共に説明されています。

▼ コマボライン上の大文字と小文字の区別とスラッシュ

W indows 上のStyleVisionServer W indows とUnix (Linux, Mac) 上の stylevisionserver

* 小文字 (stylevisionserver) はすべてのプラナトフォーム(Windows、Linux、および Mac) で使用することができますが が、小文字(StyleVisionServer) はWindows とMac のみで使用することができます。 * Linux とMac 上ではスラッシュを使用し、Windows 上では ジックスラッシュを使用します。

CLI コマンド

使用可能なコマドは下にリストされており、このセグタンのサブセグタンで説明されています。

- <u>assignlicense</u>³³: LicenseServer にライセンスをアップロードし、マシン上のStyleVision Server に割り当てます。
- <u>exportresourcestrings</u>⁽³³⁾: XML ファイルに全てのアプリケーションリノース文字列をエクスポートします。
- generate³³:入力 PXF ファイル内の入力 XML ファイルとXSLT スタイルシートから1 つまけ」複数のドキュメトを生成します。
- <u>help</u> (40):構文と引数内に与えられたコマンドに関する他の精報を表示します。(引数が提供されていない場合全てのコマンドに 関する情報を提供します)。
- <u>licenseserver</u>⁴¹: ローカルネトワーク上のLicenseServer にStyleVision Server を登録します。
- pdfdata⁴³: FDF おけはXML ファイルにPDF ファイルからデーダを生成します。
- <u>setdeflang</u>⁴⁵: StyleVision Server はのデフォルト言語を設定します。
- setfopath⁴⁶: PDF 生成のための代替 FO プロセッサーを選択します。
- <u>verifylicense</u>⁴³:現在のStyleVision Server かうイセンスされているかをチェックします。任意で、与えられたライセンスキーを使用してライセンスが供与されているかをチェックします。

• version⁴⁹: Style Vision Server の ージン番号を表示します。

4.1 assignlicense (Windows のみ)

構文と説明

assignlicense コマドはWindows 上のみで使用することができます(Linux おされるこ システム上では使用することができません)。StyleVision Server が登録されている Altova LicenseServer にライセンスファイルをアップロードし、StyleVision Server にライセンスを割り当てます(licenseserver コマドを参照してくたさい)。ライセンスの以を引数として取ります。コマドによりライセンスの有効性をテストすることができます。

stylevisionserver assignlicense [options] FILE

- FILE 引数はライセンスファイルの やを取ります。
- --test-only オプタンはLicenseServer にライセスをアップロードし、ライセスを検証しますが、StyleVision Server にライセスを割り当てません。

ライセンスご関する詳細は LicenseServer ドキュメトを確認してけさい (https://www.altova.com/manual/ja/AltovaLicenseServer/)。

▼ コマボライン上の大文字と小文字の区別とスラッシュ

W indows 上のStyleVisionServer W indows とUnix (Linux, Mac) 上の stylevisionserver

* 小文字 (stylevisionserver) はすべてのプラナフォーム(Windows、Linux、およびMac) で使用することができますが、小文字(StyleVisionServer) はWindows とMac のみで使用することができます。 * Linux とMac 上ではスラッシュを使用し、Windows 上では ジックスラッシュを使用します。

▼ Windows システム上での ジクスラッシュ、スペース および特別文字

W indows システム (例えば ファイル おけよ フォルダー名、企業名、おけよ 個人、おけよ 製品名内で スペース おけば特別文 字かめ文字列内で発生すると引用符を使用してくたさい 例えば "My File"。しかしなから、二重引用符の後の シクスラシュ (例えば、 "C:\My directory\") は正確に読み取られない場合かあります。これは シクスラッシュ文字がエスケーデンーケンスの 開始を示すためであり、エスケーデンーケンス ¥ は二重引用符文字を示すためです。この文字のシーケンスをエスケープする場合は 以下のようご前につ シクスラッシュを次のようご使用します: \\"。要約すると スペースおけお シクスラッシュを含むファイル ひを作成 する場合、以下のような名前を与えます: "C:\My Directory\"。

例

assignlicense コマドのサノプル:

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

- 上記の最初のコマドはLicenseServer に指定されたライセンスをアップロードし、StyleVision Server に割り当てます。
- 最後のコマドはLicenseServer は指定されたライセンスアップロードし、StyleVision Server に割り当てることなく検証します。

オプション

(使用できる場合)オプションは短いおよび長い書式でリストされています。短いおよび長い書式のためコークおけまつのダッシュを使用すること ができます。オプションは植を取る、おけよ、取らない場合かあります。値をとる場合、以下のようしています: --option=value。値は引 用符無して2つの場合で指定することができます: (i) 値文字列にスペースが含まれる場合。おけよ (ii) 引用符が必要とするオプションの説 明内で明示的に指示されている場合。オプションカジール値の値を取り、ブールの値が指定されていない場合、オプションのデフォルトの値は TRUE です。--h, --help オプノコンを使用してコマンドに関する情報を表示します。

- test-only [t]
 - --t, --test-only = true | false

値はtrue false です。true の場合、License Server にライセンスアイルがアップロードされ、検証されますが、割り当てられることはありません。

4.2 exportresourcestrings

構文と説明

exportresourcestrings コマンドは特定の言語内のStyleVision Server アプリケーションのリノース文字列を含むXML ファイルを出力します。使用することのできる言語は英語(en)、ドイン語(de)、スペイン語(es)、フランス語(fr)、と日本語(ja)です。

stylevisionserver exportresourcestrings [options] LanguageCode XMLOutputFile

- LanguageCode 引数は出力 XML ファイル内のノース言語の言語を与えます。これはエクスポート言語です。許可されて いるエクスポート言語は以下の通りです(それぞれの言語コードがかって囲まれています):英語(en)、ドイン語(de)、スペイン語(es)、フランス語(fr)、と日本語(ja)。
- XMLOutputFile 引数は出力 XML ファイルの ひと名前を指定します

ローカルビーションの作成方法は以下のとおりです。

▼ コマボライン上の大文字と小文字の区別とスラッシュ

W indows 上のStyleVisionServer W indows とUnix (Linux, Mac) 上の stylevisionserver

* 小文字 (stylevisionserver) はすべてのプラナフォーム(Windows、Linux、および Mac) で使用することができますが、小文字(StyleVisionServer) はWindows とMac のみで使用することができます。 * Linux とMac 上ではスラッシュを使用し、Windows 上では ジックスラッシュを使用します。

▼ Windows システム上での ックスラッシュ、スペース および特別文字

W indows システム (例えば ファイル おけよ フォルダー名、企業名、おけよ 個人、おけよ 製品名内 でスペース おけお特別文 字かめ文字列内で発生すると引用符を使用してくたさい 例えば "My File"。しかしなから、二重引用符の後の シクスラジュ (例えば、 "C:\My directory\") は正確に読み取られない場合がおります。これは シクスラジュ文字がエスケープシーケンスの 開始を示すためであり、エスケープシーケンス ¥ は二重引用符文字を示すためです。この文字のシーケンスをエスケープする場合は 以下のようご前に ジクスラジュを次のようご使用します: \\"。要約すると スペースおけま ジクスラジュを含むファイル やを作成 する場合、以下のような名前を与えます: "C:\My Directory\"。

例

exportresourcestrings コマドのサンプル

stylevisionserver exportresourcestrings de c:\Strings.xml

• 上ののコマイ はStyle Vision Server のドイン語リノース文字列を含む Strings.xml と呼ばれるファイルを c: \ に作成 します。

StyleVision Server のローカライズされたノ デジョンの作成

選択された言語で Style Vision Server のコーカライズされた・デションを作成することができます。5つのコーカライズされた・デション(英語、ドイン語、スペイン語、フランス語、および日本語) は C: \ Program Files (x86) \Altova \ Style Vision Server 2021 \ bin フォルダー内で使用することができるため、作成する必要はありません。

ローカライズされたノージョンを以下のように作成します

1. export resourcestrings コマドを使用してリノース文字列を含むXML ファイルを作成します(上記のコマド構文を 参照してくたさし)。このXML ファイル内のリノース文字列は、サポトされる以下の5つの言語のひとつです: コマドと共に使用 される Language Code 引数に従い英語 (en)、ドイン語 (de)、スペイン語 (es)、フランス語 (fr)、おけよ日本語 (ja)。

- 2. 5つのサポトされる言語の1つからのハノース文字列からターゲホ言語に翻訳します。リノース文字列はXML ファイル内の <string> 要素のエレテンツです。{option} おは{product} などの、中かつ内の変数は翻訳したいでください。
- 3. ローカライズされた Style Vision Server DLL ファイルを翻訳された XML ファイルから生成するために、Altova サポートに連絡してくたさい。
- 4. <u>Altova サポト</u> からローカライズされたDLLファイルを受信すると DLL を C:\Program Files (x86)\Altova\<% APPFOLDER\bin フォルダー内に保存します。DLL ファイルはフォーム StyleVisionServer2021_1c.dll の名前を 持ちます。名前の_1c 部分には 言語コードが含まれています。例えば、StyleVisionServer2021_de.dll では de の部分は ドイン語 (Deutsch) のすめの言語コードです。
- 5. setdeflang コマドを実行し、使用する Style Vision Server アプリケーションと同様コーカライズされた DLL ファイルを 設定します。setdeflang コマドの引数のために、 DLL 名の一部である言語コードを使用してくたさい。
- メモ Altova StyleVision Server はつの言語のナダのサポートを提供しています:英語、ドイツ語、スペイン語、フランス語、および日本語。これらの言語のユーカライズされナットジョンを作成する必要はありません。これらの言語をデフォリトの言語として設定するこは、StyleVision Server のsetdeflang コマイドを使用します。
4.3 generate

構文と説明

PXF ファイル内に含まれるXSLTドキュメトを使用して入力ファイルを生成し、generate コマド(省略名 gen)はつまけま複数の出力ファイル(HTML、PDF、RTF、およびがまけまDOCX)を生成します。

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

- --inputxml オプションは必須です。XML ファイルへの やを提供します。
- InputPXF 引数は出力ドキュメトを生成するために使用されるXSLTドキュメトを含むPXF ファイルへの やを指定します。PXF ファイルはAltovaのStyle Vision アプリケーションを使用して作成されます。
- 各出力フォーマナはその出力のナダのオプションを指定することにお生成されます(ア/こ)ストされるオプションを参照してくたさし)。各オプションの値は出力の生成先へのなです。
- メモ StyleVision Server は、FO からPDF ファイルを生成するかのApache ProjectのFO プロセッサーである Apache FOP を使用します。Apache FOP はStyleVision Server と共に以下の場所にインネールされて、ます。Windows シ ステム上では、ProgramData\Altova\SharedBetweenVersions内にインネールされて、ます。Linux とmacOS システム上では、StyleVisionServer2021フォルダーの子孫フォルダー内にインネールされて、ます。Apache FOP は Java Runtime Environment 1.6 おけは以降がStyleVision Server マンン上にインネールされて、ることを必要と ます。32 ビット StyleVision Server のかがはよ 32 ビット Java をインネールしてくたさい。64 ビット StyleVision Server のかりは、64 ビット Java をインネールてくたさい。
- ▼ コマボライン上の大文字と小文字の区別とスラッシュ

W indows 上のStyleVisionServer W indows とUnix (Linux, Mac) 上の stylevisionserver

* 小文字(stylevisionserver)はすべてのプラナフォーム(Windows、Linux、およびMac)で使用することができますが、小文字(StyleVisionServer)はWindowsとMacのみで使用することができます。 * LinuxとMac 上ではスラッシュを使用し、Windows 上では、シックスラッシュを使用します。

▼ Windows システム上での シクスラッシュ、スペース および特別文字

W indows システム (例えば ファイル およ フォルダー名、企業名、およ 個人、およ 製品名内 でスペース およお 別文 字か 文字列内で発生すると引用符を使用してくた (例えば "My File"。しかしなから、二重引用符の後の ジクスラジュ (例えば "C:\My directory\") は正確に読み取られな 場合かがます。これは ジクスラジュ文字がエスケープノーケンスの 開始を示すためであり、エスケープノーケンス ¥ は二重引用符文字を示すためです。この文字のノーケンスをエスケープする場合は 以下のようご前に ジクスラジュを次のようご使用します: \\"。要約すると スペースおよお ジクスラジュを含むファイル やを作成 する場合、以下のような名前を与えます: "C:\My Directory\"。

サンプル

generate コマイのサンプル:

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

- 上記のコマド におり通の--inputxml オプタン、InputPXF 引数(Test.pxf)、およびの出力作成のオプタン(上記のサンプルでは--html)の最小値が含まれています。
- 入力 XML ファイルはPXF ファイル内で見つけることができます(上記の2番目と3番目のサンプルを参照してくたさい)。まけは 外部 XML ファイルであることもできます(ファイルはPXF ファイルの外部に存在します。上記の最初および4番目のサンプルを 参照してくたさい)。
- --inputxml スイッチはンインスキーマノースがDB おけまDB-XML の場合無視されますが、構文上の理由から、 -inputxml=database などを使用することが奨励されます。
- 出力作成オプション -- html は上のサンプルで示されている通い相対的ない ひを取ります。このため、出力ファイルのコケーションは PXF ファイルであるフォルダーニオして相対的にないます。

オプション

inputxml [xml]

--xml, --inputxml = PathToXMLFile

このオプションは必須です。XML ファイルへの やを指定します。XML ファイルはPXF ファイルの内部、おけよ 外部に存在します。 PXF ファイルの内部で XML ファイルをターゲルする ゴよ | zip ロケーターを使用します (上のサンプル内の イライトされた箇所を 参照してくたさい)。--inputxml オプションは入力 PXF のメインスキーマノースが DB おけ a DB-XML の場合無視されます。

dbwhere [dbw]

<mark>--dbw, --dbwhere =</mark> WHEREClause DB-XML ソースの行か処理されるかを決定する SQL WHERE 句。

🔻 param [p]

--p, --param = <mark>\$</mark>ParamName:ParamValue

PXF ファイル内で定義される/ ラメーターに値を提供します。--param スイッチルな/ ラメーターの前で使用される必要があります。 ParamName おけま ParamValue にスペースが含まれる場合引用符を使用します。例: --p=\$company:"Nanonull Inc"

prohibit-output-outside-target-folder

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

値はtrue | false です。true の場合、メイン出力ファイル(HTML, PDF, RTF, DOCX, FO) が作成されるフォルダー内以外 出の出力の作成を許可しません。これにより、必要な場合他のフォルダーのすめの保護が提供されます。デフォルトは false です。

outhtml [html]

<mark>--html, --outhtml =</mark> <mark>FilePath</mark> 生成するHTML ファイルへの ぷ。

outpdf [pdf]

<mark>--pdf, --outpdf =</mark> FilePath 生成するPDF ファイルへの ぷ。

outrtf [rtf]

<mark>--rtf, --outrtf =</mark> FilePath 生成するRTFファイルへの ぷ。

outdocx [docx]

<mark>--docx, --outdocx =</mark> FilePath 生成するDOCX ファイルへのぷ。 outfo [fo]

<mark>--fo, --outfo =</mark> FilePath 生成するFO ファイルへの ぷ。

▼ generate-html-output-as-mime

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

値はtrue false です。このオプタンか指定されていない場合、デフォルトはfalse です。値が提供されていたい場合は true です。true の場合、HTML 出力はmime ストリームと出力されます。

verbose [v]

--v, --verbose = true|false

値はtrue false です。すべてのメッセージの表示をオンとオフィゴル替えます。このオプションか指定されていない場合、デフォルトはfalse です。値が提供されていない場合はtrueです。

lang [l]

<mark>--1, --lang =</mark> en|de|es|fr|ja メッセーンを表示するために使用されている言語。

--h, --help オプションを使用してコマンドに関する情報を表示します。

(使用できる場合)オプションお豆におよび長に書式でノストされて、ます。短いおよび長い書式のオーガコ つまけ おこのが シュを使用すること ができます。オプションは植を取る、まけよ、取らは、場合かあります。値をとる場合、以下のよう ことがます: --option=value。値は月 用符無して2つの場合で指定することができます: (i)値文字列にてペースが含まれる場合。まけよ、(ii)引用符が必要とするオプションの説 明内で明示的に指示されている場合。オプションがブール値の値を取り、ブールの値が指定されていない場合、オプションのデフォルトの値は TRUE です。 --h, --help オプションを使用してコマンドに関する情報を表示します。

4.4 help

構文と説明

```
help コマイは、ヘルプを必要とするコマイの名前である単一の引数(Command)を取ります。コマイの構文、オプション、および他の関連する情報を表示します。Command引数が指定されていない場合、説明と共に実行可能ファイルのすべてのコマイがリストされます。
```

stylevisionserver help Command

▼ コマンドライン上の大文字と小文字の区別とスラッシュ

W indows 上のStyleVisionServer W indows とUnix (Linux, Mac) 上の stylevisionserver

* 小文字 (stylevisionserver) はすべてのプラナフォーム(Windows、Linux、および Mac) で使用することができますが、小文字(StyleVisionServer) はWindows とMac のみで使用することができます。 * Linux とMac 上ではスラッシュを使用し、Windows 上では ジックスラッシュを使用します。

例

help コマドのサノプルはlicenserver コマドに関する情報を表示します:

stylevisionserver help licenseserver

-help オプション

へにプ情報が必要とされるコマイドの--help オプションを使用してへいプ情報を使用することもできます。 下の2つのコマイドは同じ結果を生成します:

stylevisionserver licenseserver --help

上記のコマドはlicenseserver コマドの--help オプションを使用します。

stylevisionserver help licenseserver

help コマイは別数とてlicenseserverを取ります。

両方のコマドはlicenseserver コマドに関するへいプ情報を表示します。

4.5 licenseserver

構文と説明

1icenseserver コマドは Style Vision Server を Server-Or-IP-Address 引数にお指定されている Altova License Server を使用して登録します。 licenseserver コマド実行されるは、 2つのサード(Style Vision Server と License Server)が同じやトワーク上で作動しており License Server が作動している必要があります。 Style Vision Server を License Server と登録する管理特権が必要についます。

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

• Server-Or-IP-Address 引数は LicenseServer マシンの名前、おは、IP アドレスを取ります。

StyleVision Server がLicenseServer へ登録されるとこの件に関連するシッセージを受け取ります。シッセージはLicenseServer のURL も表示します。LicenseServer に移動して、StyleVision Server にライセンを割り当てることもできます。ライセンスに関する詳細は、LicenseServer ドキュメントを確認してくたさい(<u>https://www.altova.com/manual/ja/AltovaLicenseServer/</u>)。

▼ コマボライン上の大文字と小文字の区別とスラッシュ

W indows 上のStyleVisionServer W indows とUnix (Linux, Mac) 上の stylevisionserver

* 小文字 (stylevisionserver) はすべてのプラナフォーム(Windows、Linux、および Mac) で使用することができますが、小文字(StyleVisionServer) はWindows とMac のみで使用することができます。 * Linux とMac 上ではスラッシュを使用し、Windows 上では ジックスラッシュを使用します。

▼ Windows システム上での シクスラッシュ、スペース および特別文字

W indows システム (例えば ファイル おけよ フォルダー名、企業名、おけよ 個人、おけよ 製品名内でスペース おけは特別文 字かり文字列内で発生すると引用符を使用してくたさい 例えば "My File"。しかしなから、二重引用符の後の シクスラシュ (例えば、 "C:\My directory\") は正確に読み取られない場合がおります。これは シクスラシュ文字がエスケープノーケンスの 開始を示すためであり、エスケープノーケンス ¥ は二重引用符文字を示すためです。この文字のシーケンスをエスケープする場合は 以下のようご前に ジクスラシュを次のようご使用します: \\"。要約すると スペースおけま ジクスラッシュを含むファイル ジを作成 する場合、以下のような名前を与えます: "C:\My Directory\"。

例

licenseserver コマドのサンプル

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

上記のコマドは、それそれ DOC.altova.com とう名前のマシンとユーザーのマシン(localhost と127.0.0.1)を Altova License Server を作動するマシンとして指定します。双方の場合、指定されているマシン上の License Server を使用してコマドは Style Vision Server を登録します。最後のコマドは、コマドを実行するためにサーバー実行可能ファイルを呼び出します。

オプション

(使用できる場合)オプションお短いおよび長い書式でリストされています。短いおよび長い書式のナータコンフますよりつびッシュを使用することができます。オプションは植を取る、おけよ、取らない場合かあります。値をとる場合、以下のようゴンドます:--option=value。値は月 用符無して2つの場合で指定することができます:(i)値文字列にスペースが含まれる場合。まけよ、(ii)引用符が必要とするオプションの説明内で明示的に指示されている場合。オプションのブール値の値を取り、ブールの値が指定されていない場合、オプションのデフォルトの値は TRUE です。--h, --help オプションを使用してコマンドに関する情報を表示します。

🔻 json [j]

--j, --json = true false 値はtrue false です。true の場合、マシン解読可能な JSON オブジェクトとて登録の結果を出力します。

4.6 pdfdata

構文と説明

pdfdata コマドは InputPDF 引数とて提出されたPDF ファイルからFDF ファイルませはXML ファイルを生成します。

stylevisionserver pdfdata [options] InputPDF

- InputPDF 引数は出力 FDF おさまXML ファイルが生成される PDF ファイルへの やを指定します。 PDF ファイルにフォーム データが存在しない場合、生成されたファイルコまフォームデータが含まれません。
- --outfdf オプションを使用して、生成されたFDF ファイル、おけよ、生成されたXMLファイルのロケーションを指定する-outxml オプションを指定します。

FDF ファイルと記入可能な PDF フォームのデザインの詳細に関しては、Altova Style Vision (Enterprise Edition) ドキュメントを参照してくたさい。

▼ コマボライン上の大文字と小文字の区別とスラッシュ

W indows 上のStyleVisionServer W indows とUnix (Linux, Mac) 上の stylevisionserver

*小文字(stylevisionserver)はすべてのプラオフォーム(Windows、Linux、およびMac)で使用することができますが、小文字(StyleVisionServer)はWindowsとMacのみで使用することができます。 *LinuxとMac 上ではスラッシュを使用し、Windows 上では、シックスラッシュを使用します。

▼ Windows システム上でのバックスラッシュ、スペース および特別文字

W indows システム (例えば ファイル おけよ フォルダー名、企業名、おけよ 個人、おけよ 製品名内でスペース おけお特別文 字かめ文字列内で発生すると引用符を使用してくたさい 例えば "My File"。しかしなから、二重引用符の後の シクスラシュ (例えば、 "C:\My directory\") は正確に読み取られない場合かおります。これは シクスラシュ文字がエスケープレーケンスの 開始を示すためであり、エスケープレーケンス ¥ は二重引用符文字を示すためです。この文字のレーケンスをエスケープする場合は 以下のようご前に ジクスラシュを次のようご使用します: \\"。要約すると スペースおけま シクスラッシュを含むファイル ひを作成 する場合、以下のような名前を与えます: "C:\My Directory\"。

```
サンプル
```

pdfdata コマドのサンプル:

```
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
```

上記のサンプルはFDF ファイルとXML ファイルをPDF 入力からそれぞれ作成します。

```
オプション
```

outfdf

```
<mark>--outfdf =</mark> FilePath
生成されたFDF ファイルへの ぷ。
```

outxml

<mark>--outxml = FilePath</mark> 生成されたXML ファイルへの ぷ。

--h, --help オプションを使用してコマンドに関する情報を表示します。

(使用できる場合)オプタンは短いおよび長い書式でリストされています。短いおよび長い書式のためコークおくはつのがシュを使用すること ができます。オプタンは植を取る、おくよ、取らない場合かあります。値をとる場合、以下のようしています。--option=value。値は月 用符無して2つの場合で指定することができます。(i)値文字列にスペースが含まれる場合。おくよ、(ii)引用符が必要とするオプションの説 明内で明示的に指示されている場合。オプションのゲール値の値を取り、ブールの値が指定されていない場合、オプノョンのデフォルトの値は TRUE です。--h, --help オプションを使用してコマンドに関する情報を表示します。

4.7 setdeflang

構文と説明

setdeflang コマボ (sdl ど略される)はStyleVision Server.のデフォルトの言語を設定します。英語 (en)、ドイン語 (de)、スペ 个語 (es)、フランス語 (fr)、および日本語 (ja)を使用することができます。コマボ は必須の LanguageCode 引数を取ります。

stylevisionserver setdeflang [options] LanguageCode

- LanguageCode 引数が必須で、StyleVision Server のデフォルの言語を設定します。対応する値は以下の通りです
 :en, de, es, fr, ja.
- --h, --help オプロンを使用してコマドの情報を表示します。

▼ コマボライン上の大文字と小文字の区別とスラッシュ

W indows 上のStyleVisionServer W indows とUnix (Linux, Mac) 上の stylevisionserver

* 小文字 (stylevisionserver) はすべてのプラナフォーム(Windows、Linux、および Mac) で使用することができますが、小文字(styleVisionServer) はWindows とMac のみで使用することができます。 * Linux とMac 上ではスラッシュを使用し、Windows 上では、シックスラッシュを使用します。

例

setdeflang (sdl) コマドのサンプル

stylevisionserver sdl de stylevisionserver setdeflang es

- 最初のコマド StyleVision Server のデオルトの言語を「か語に設定します。
- 2番目のコマドはStyleVision Server のデフォルトの言語をスペイン語に設定します。

オプション

--h, --help オプノコンを使用してコマンドに関する情報を表示します。

4.8 setfopath

構文と説明

setfopath コマド (省略名 sfp) は Style Vision Server パッケージ ご含まれてい る以外の Apache FOP プロセッサーへの ひを 指定します。

stylevisionserver setfopath | sfp [options] Path ?E??

- StyleVision Server 内に含まれている Apache FOP プロセッサーは、デフォルトでFOドキュメトの処理と、PDFの出 カに使用されます。StyleVision Server には提供されるプロセッサー以外の Apache FOP プロセッサーク・スタンスを使用 する場合、使用する FO プロセッサーへの 父を与える Path 引数を持つ setfopath コマイを使用してください。
- 代替 FO プロセッサーがsetfopath コマドを使用して指定されている後は、generate 20 コマドを使用してPDFが 生成される際にこのプロセッサーが使用されます。プロセッサーを変更する場合は、setfopath コマドを再度使用します。 Style Vision Server のFOP プロセッサーごが、替える場合、システム上でFOP フォルダーをロケートし、setfopath の 引数としてののを使用します。
- Windows システム上では、StyleVision Server を使用してインストールされている FOP フォルダーは、 ProgramData\Altova\SharedBetweenVersionsで見つれたいできます。Linux とmacOS システム上では、 StyleVisionServer2021 フォルダーの子孫フォルダー内で見つれたいできます。

FDF ファイルと記入可能な PDF フォームのデザインの詳細に関しては <u>Altova Style Vision (Enterprise Edition)</u> ドキュメトを参照してくたとい。

▼ コマボライン上の大文字と小文字の区別とスラッシュ

W indows 上のStyleVisionServer W indows とUnix (Linux, Mac) 上の stylevisionserver

* 小文字 (stylevisionserver) はすべてのプラナフォーム(Windows、Linux、およびMac) で使用することができますが、小文字(StyleVisionServer) はWindows とMac のみで使用することができます。 * Linux とMac 上ではスラッシュを使用し、Windows 上では デックスラッシュを使用します。

▼ Windows システム上での ックスラッシュ、スペース および特別文字

W indows システム(例えば ファイル おけよ フォルダー名、企業名、おけよ 個人、おけよ 製品名内でスペース おけお特別文 字が約文字列内で発生すると引用符を使用してくたさい 例えば 「My File」。しかしながら、二重引用符の後の、シクスラッシュ (例えば 「C:\My directory」)は正確に読み取られない場合があります。これは シクスラッシュ文字がエスケープシーケンスの 開始を示すけっかであり、エスケーブシーケンス ¥ は二重引用符文字を示すけっかです。この文字のシーケンスをエスケープする場合は 以下のようご前につ、シクスラッシュを次のようご使用します: \\"。要約すると スペースおけお シクスラッシュを含むファイル やを作成 する場合、以下のような名前を与えます: "C:\My Directory\\"。

サンプル

setfopath コマイを実行後、指定されたFO プロセッサーを使用してPDFを生成するためにgenerate 33 コマイドを使用すること ができます:

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

上記のコマドは以下を行います

1. setfopath コマイは c:\FOP\FOP.bat に存在するFO プロセッサーがのPDF 生成コマイド内でPDF を生成する ために使用されるように指定します。 2. generate コマナは、PXF ファイル内に含まれている変換ファイルを使用して、指定された入力 XML から PDF ファイルを 生成します。PDF を生成するために使用された前のコマナド内で FO プロセッサーは指定されています。

オプション

--h, --help オプションを使用してコマンドに関する情報を表示します。

(使用できる場合) オプタンお短いおよび長い書式でしたされています。短いおよび長い書式のためコークおけお つのがシュを使用すること ができます。オプタンは植を取る、おけよ、取られ、場合かあります。値をとる場合、以下のようしなります: --option=value。値は月 用符無して2つの場合で指定することができます:(i)値文字列にてペースが含まれる場合。おけよ、(ii)引用符が必要とするオプタンの説 明内で明示的に指示されている場合。オプタンムゲール値の値を取り、ブールの値が指定されていない場合、オプタンのデフォルトの値は TRUE です。 --h, --help オプタンを使用してコマンドに関する情報を表示します。

4.9 verifylicense (Windows のみ)

構文と説明

verifylicense コマドは、使用中の製品がライセス供与を受けているかチェックします。更に、--license-key オプランにより 特定のライセンスキーが製品に既に割り当て済みかをチェックします。このコマドはWindows 上でのみサポートされています。Linux おけよ Mac システム上ではサポートされていません。

stylevisionserver verifylicense [options]

• Style Vision Server に割り当てられた特定のライセスが、--license-key オプタンの植とてライセスキーを与えるか をチェックします。

ライセンス 関する詳細は LicenseServer ドキュメントを確認してけさい (https://www.altova.com/manual/ja/AltovaLicenseServer/)。

▼ コマボライン上の大文字と小文字の区別とスラッシュ

W indows 上のStyleVisionServer W indows とUnix (Linux, Mac) 上の stylevisionserver

```
* 小文字 (stylevisionserver) はすべてのプラナフォーム(Windows、Linux、およびMac) で使用することができますが、小文字(StyleVisionServer) はWindows とMac のみで使用することができます。
* Linux とMac 上ではスラッシュを使用し、Windows 上では、シックスラッシュを使用します。
```

例

verifylicense コマドのサンプル

```
stylevisionserver verifylicenseserver
stylevisionserver verifylicenseserver --license-key=ABCD123-ABCD123-ABCD123-ABCD123-ABCD123-ABCD123-ABCD123-ABCD123-ABCD123
```

- 最初のコマドはStyleVision Server からイセンス供与されているかをチェックします。
- 2番目のコマドはStyleVision Server が--license-key オプタンで指定されているライセスキーと共にライセス供与されているかをチェックします。

オプション

(使用できる場合) オプタンお短いおよび長い書式でリストされています。短いおよび長い書式のオーガコ つまけ おつのがシンコを使用すること ができます。オプタンム植を取る、まけよ、取られ、場合かあります。値をとる場合、以下のよう こなります: --option=value。値は月 用符無して2つの場合で指定することができます:(i)値文字列にてペースが含まれる場合。まけよ、(ii)引用符が必要とするオプタンの説 明内で明示的に指示されている場合。オプションのブール値の値を取り、ブールの値が指定されていない場合、オプションのデフォルトの値は TRUE です。 --h, --help オプションを使用してコマンドに関する情報を表示します。

license-key [l]

--I, --license-key = <u>Value</u> StyleVision Server がのオプションの値と、てライセンスキーと共にライセンス供与されているか確認します。

4.10 version

構文と説明

version コマイはStyleVision Server のデジンを表示します。

stylevisionserver version

```
▼ コマボライン上の大文字と小文字の区別とスラッシュ
```

W indows 上のstyleVisionServer W indows とUnix (Linux, Mac) 上の stylevisionserver

* 小文字 (stylevisionserver) はすべてのプラナフォーム(Windows、Linux、および Mac) で使用することができますが、小文字(StyleVisionServer) はWindows とMac のみで使用することができます。 * Linux とMac 上ではスラッシュを使用し、Windows 上では ジックスラッシュを使用します。

例

version コマドのサンプル

stylevisionserver version

5 StyleVision Server API

Style Vision Server は、.NET、COM、おけよ、Java ベースコードからプログラム的にアクセスすることのできるアプリケーションプログラング インターフェイス (API)を提供します。

このレファレンスセクションは以下のように整理されています

- <u>.NET インターフェイスについて⁵¹</u>
- <u>COM インターフェイスについて⁵²</u>
- <u>Java イノターフェイスこえて⁵³</u>
- コードのサンプル54
- <u>API レファレンス⁶¹</u>

5.1 .NET インターフェイスについて

.NET インターフェイスは、COM インターフェイスの周りのラッドーとして作成されます。これらは Altova にお屠名され、名前空間 Altova.StyleVisionServerを使用するプライマ相互運用機能アセンブルとて提供されます。

手動の登録は必要ないため、インストール中、StyleVision Server は自動的にCOM サーバーオブジェクトとして、自動的に登録されます。アクセスエラー送信されると、コンポーネントサービスを開き、StyleVision Server を含むアプリケーションプールを実行する同じアカウントへのパーミッションを与えます。

.NET プレンエクトで Style Vision Server を使用するコよ Altova.Style Vision Server.dll ファイルコンスを追加します(下の説明を参照してくたさい)。 Altova.Style Vision Server.dll は Style Vision Server インストールフォルダー内の bin フォルダー こかます。この.dll ファイルは、自動的にグロー、シレアセンブノキャッシュ(GAC) にStyle Vision Server インストール 中に追加されます(GAC)は通常以下で見つったことができます C: \WINDOWS\assembly フォルダー).

Style Vision Server がCOM サイトオブンエクトとて登録されると、とAltova.Style Vision Server.dll を.NET イクターフェイスで使用できるようことはます。 Style Vision Server API 機能は、NET プロジェクトので使用できるようことはます。

Visual Studio .NET プロジェクト内のStyleVision Server DLL (コノファレンスを追加する

- 1. Visual Studio 内で開かれている.NET プロジェクトカら、「プロジェクト | レファレンスを追加する」をクトックします。レファレンスの追加ダイアログボックスのポップアップします。
- 2. 参照タブで、<*StyleVisionServer application folder*>/bin フォルダーを参照し、 Altova.StyleVisionServer.dll を選択して、「OK」をクリックします。

(オブジェクトブラウザーを表示するコは、「表示」メニュー内の「オブジェクトブラウザー」をクトックします) Visual Studio オブジェクトブラウザ ーを使用して、Altovaの構造を確認することができます。StyleVisionServer アセンブの構造をビューすることができます。

5.2 COM インターフェイスについて

StyleVision Server は、インストール中にCOM サーバーオブジェクトとて自動的に登録されます。登録が成功したか確認する日まレジストリエディターを開きます(例えば、コマイドラインでregedit.exe を入力するなど)。登録し成功するとレジストリまプラス StyleVision.Server を含んでいます。このプラスは、通常でHKEY_LOCAL_MACHINE\SOFTWARE\Classes 見つたことができます。

COM サード・オブシェクト か登録されると、COM 呼び出しのためのプログランクをサポートするアプリケーションとスクレプト 言語から呼び出 すことができます。 Style Vision Server インストール シケージの場所を変更する場合、Style Vision Server をアンインストール、必要な場所でインストールすることが奨励されています。このように、必要な非登録と登録はインストールの過程で実行されます。

5.3 Java インターフェイスについて

Java コードから Style Vision Server API にアクセスするゴは Java プロジェクトの.classpath ファイルに次のレファレンスを追加します。

StyleVisionServer.jarStyleVision Server と通信するライブラリStyleVisionServer_JavaDoc.zipStyleVision Server API のドキュメナ

両方のファイルは、StyleVision Server インストールbin フォルダーの中で使用することができます。ファイルを元の場所から参照するか、 まさは、プロジェクトのセトアップに合う場合に地の場所にコピーすることができます。

Eclipse 内にライブラリレファレンスを追加する

Eclipse 内では、Java プロジェクトのプロ・ティを編集することによりクラス・マのレファレンスを追加することができます。下のサンプルの処理 は、サンプルEclipse 4.4 に適用することができます。

- 1. Eclipse 内で開かれているファイルのプロジェクト メニューから「プロノティ」をクトックし、 Java ビルドノ やを選択します。
- 2. ライブラタブから外部 JAR を追加する」をクリックして、StyleVision Server インストールフォルダー内の StyleVisionServer.jar ファイルを参照します。
- 3. ビルドパズの下のJAR とクラスフォルダーから、StyleVisionServer.jar レコードを展開し、Javadoc location: (None) レコードをダブルクトックします。
- 4. アーカイブと外部 ファイルオプラン内で Javadoc が選択されていることを確認し、Style Vision Server インストールフォルダ 一内にある Style Vision Server JavaDoc.zip ファイルを参照してくたさい。
- 5. 「OK」をクリックします。StyleVision Server ライブラルとJavadoc アーカイブへのレファレンスは、プロジェクトの .classpath ファイルに追加されます。

5.4 コードのサンプル

このセクションのサンプルは以下のプログラム言語のために説明されています

- C++ 54
- C#⁵⁵
- VBScript⁵⁷
- Visual Basic⁵⁸
- Java ⁵⁹

5.4.1 C++

下のサンプルは PXF ファイルと入力 XML ファイルを使用して、出力 RTF ファイルを生成するためにC++ コードを使用する方法を表示 しています。StyleVision Server がインストール、および、ライセンスされていること、おこ、COM サードーオブジェクトとして使用できること を確認してくたさい、COM サードーオブジェクトとしての登録は、StyleVision Server のインすオール中に行われます。登録の成功を確 認するために<u>COM インターフェイスについてを参照してくたさい、⁶²⁰。</u>

```
// StyleVisionServerAPI Sample.cpp : Defines the entry point for the console application.
11
#include <iostream>
#include "atlbase.h"
// The following import statements require the corresponding C++ tool-chain to be
selected in the project configuration file.
#ifndef WIN64
// 32-bit StyleVisionServer
#import "progid:StyleVision.Server"
#else
// 64-bit StyleVisionServer
#import "progid:StyleVision x64.Server"
#endif
int tmain(int argc, TCHAR* argv[])
ł
      CoInitialize( NULL );
      try
      {
             //Create a StyleVision Server object
             StyleVisionServerLib::IServerPtr pSVS;
             CoCreateInstance( uuidof( StyleVisionServerLib::Server ), NULL,
CLSCTX ALL,
            uuidof( StyleVisionServerLib::IServer ), reinterpret cast< void**</pre>
>( &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
             11
                   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
             11
             // pSVS->InputXML = "ExpReport.pxf|zip\\ExpReport.xml";
             11
                 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"))
              ł
                    std::cout << pSVS->LastExecutionMessage << std::endl;</pre>
                    std::cout << "Success - finished execution" << std::endl;</pre>
              }
             else
                    std::cout << pSVS->LastExecutionMessage << std::endl;</pre>
      }
      catch (_com_error& err )
       Ł
             BSTR bstrMessage;
             (err).ErrorInfo()->GetDescription( &bstrMessage );
             std::cout << "Exception occurred: " <<</pre>
com util::ConvertBSTRToString( bstrMessage ) << std::endl;</pre>
      }
      CoUninitialize();
      return 0;
}
```

5.4.2 C#

下のサンプルは PXF ファイルと入力 XML ファイルを使用して、出力 RTF ファイルを生成するためにC#コードを使用する方法を表示しています。StyleVision Server がインストール および、ライセンスされていること、また、 COM サードーオブジェイト としての登録しよ StyleVision Server のインすオール中に行われます。登録の成功を確認 確認してくたさい。COM サードーオブジェイトとしての登録しよ StyleVision Server のインすオール中に行われます。登録の成功を確認 するために COM インターフェイスについてを参照してくたさい、⁶²。

```
namespace StyleVisionServerAPI sample
ſ
    class Program
    ł
        static void Main(string[] args)
        ł
            try
            ł
               //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
               11
externally (not inside PXF)
               // objSVS.InputXML = "ExpReport.xml";
                     Or it can contain the path INSIDE the PXF
               11
               // objSVS.InputXML = "ExpReport.pxf|zip\\ExpReport.xml";
               11
                     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)
            ł
                // some general error like an invalid license happened
                System.Console.WriteLine("Internal Error - " + ex.Message);
            }
        }
   }
}
```

5.4.3 VBScript

下のサンプルは PXF ファイルと入力 XML ファイルを使用して、出力 RTF ファイルを生成するために VB Script コードを使用する方法 を表示しています。 Style Vision Server がインストール および、ライセンスされていること、おこ、 COM サード・オブジェクト として使用で きることを確認してくたさい。 COM サード・オブジェクトとしての登録は、 Style Vision Server のインすオール中に行われます。登録の成 功を確認するために <u>COM インターフェイスについてを参照してくたさい</u>⁶²⁰。

```
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
                          ' ignore any COM errors avoiding uncontrolled script
On Error Resume Next
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
                           ' re-enble default error promotion
On Error Goto 0
'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-enble default error promotion
```

5.4.4 Visual Basic

下のサンプルは PXF ファイルと入力 XML ファイルを使用して、出力 RTF ファイルを生成するかとVisual Basic コードを使用する方 法を表示しています。Style Vision Server がんノストール および、ライセノされていること、おこ、COM サード・オブジェケトとして使用 できることを確認してくたさい。COM サード・オブジェケトとしての登録は Style Vision Server のインすオール中に行われます。登録の成 功を確認するかり、COM インターフェイスについてを参照してくたさい、⁶²²。

```
Option Explicit On
```

Module Program

Sub Main()

```
Try
```

'Create a StyleVision Server object

```
Dim objSVS As Altova.StyleVisionServer.Server = New
Altova.StyleVisionServer.Server
            'Set a working directory - used for output and for intermediate files
            'objSVS.WorkingDirectory = "C:\Program Files (x86)
\Altova\MapForceServer2020\etc\Examples"
            objSVS.WorkingDirectory = "..\..\.."
            'Default path to the StyleVision Server executable is the installation path
(same dir with the StyleVisionServer.dll)
            'In case you moved the binaries on the disk, you need to explicitly set the
path to the .exe file
            'objSVS.ServerPath = "C:\Program Files (x86)
\Altova\StyleVisionServer2020\bin\StyleVisionServer.exe"
            'objSVS.ServerPath = "C:\Program
Files\Altova\StyleVisionServer2020\bin\StyleVisionServer.exe"
            'Prepare the name of the working XML
                 This can be an absolute/relative path if the file is stored externally
(not inside PXF)
            ' objSVS.InputXML = "ExpReport.xml"
                Or it can contain the path INSIDE the PXF
            objSVS.InputXML = "ExpReport.pxf|zip\ExpReport.xml"
                Easiest way is to refer to the file as being embedded in the
transformation file
            ' objSVS.InputXML = "altova://packagedfile/ExpReport.xml"
            'Add output paths (absolute or relative to WorkingDirectory) for all formats
that should be generated
            objSVS.OutputRTF = "C:\tmp\ExpReport.rtf"
            objSVS.OutputPDF = "C:\tmp\ExpReport.pdf"
            objSVS.OutputHTML = "C:\tmp\ExpReport.html"
            'Prepare the parameters, if your design uses parameters
            'objSVS.AddParameter( "testparam1", "value 1" )
            ' Run the transformation; the output will be stored at C:\temp
            If (objSVS.Generate("ExpReport.pxf")) Then
                System.Console.WriteLine(objSVS.LastExecutionMessage)
                System.Console.WriteLine("Success - finished execution")
            Else
                System.Console.WriteLine(objSVS.LastExecutionMessage)
           End If
        Catch ex As Exception
            System.Console.WriteLine("Internal Error - " & ex.Message())
        End Try
   End Sub
```

```
End Module
```

5.4.5 Java

下のサンプルは、PXF ファイルと入力 XML ファイルを使用して、出力 RTF ファイルを生成するように Java コードを使用する方法を表示 しています。Style Vision Server がインストール、および、ライセンスされていること、およ、サーバーオブジェイトとして使用できることを確認し てくけさい、サーバーオブジェイトとしての登録は、Style Vision Server 登録の成功を確認するようはこ、次を参照してくけさい、Java イン ターフェイスについて

```
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();
       }
   }
}
```

API レファレンス 5.5

このセグションは Style Vision Server API のためのユーザーのレファレンスです。

- COM と.NET⁶¹ Java⁶⁸ •
- •

COM と .NET 5.5.1

Style Vision Server API は新しい Style Vision Server オジェクト インスタンスを作成し、Style Vision Server へのアクセスを提 供する IServer インターフェイスを公開します。

IServer インターフェイス には 以下のメノボとプロ 守か存在します。

メンド

AddParameter

PXF ファイル内で定義されたパラメーターに値を割り当てます。

■ 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

パラメーターのノストを作成します。

🖃 C#

void ClearParameterList()

□ C++

HRESULT ClearParameterList();

🖃 VB

Sub ClearParameterList()

Generate

TransfPath 内で指定されているPXF ファイルを使用していつおけば複数の出力ファイル(HTML、PDF、RTF、および/おけよ DOCX)を生成します。PXF ファイル内に含まれているXSLTドキュメトを使用して、PXF ファイル内の作業 XML ファイル入力 XML ファイルを変換します。成功の場合は、TRUE を、それ以外は、FALSE を返します。

■ 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

プロッティ

APIMajorVersion

Style Vision Server API のメジャーなバージョンを取得します。API が他のサーバーは接続されている場合、製品のバージョンとは異なる場合がおけます。

■ C#

int APIMajorVersion { get; }

🖃 C++

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

🖃 VB

ReadOnly Property APIMajorVersion As Integer

APIMinorVersion

Style Vision Server API のマイナーなバージョンを取得します。API が他のサーバーは接続されている場合、製品のバージョンとは異なる場合があります。

■ C#

int APIMinorVersion { get; }

■ C++

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

🖃 VB

ReadOnly Property APIMinorVersion As Integer

APIServicePackVersion

StyleVision Server API のサービスパックバージョンを取得します。API が他のサーバーは接続されている場合、製品のバージョン とは異なる場合がおります。

□ C#

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

□ C++

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

🖃 VB

ReadOnly Property APIServicePackVersion As Integer

InputXML

(PXF ファイル内の作業 XML ファイル)処理する XML ファイルの ひを名前を設定します。

```
⊟ C#
```

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

```
■ C++
```

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

🖃 VB

```
Property InputXML As String
```

Is64Bit

StyleVision Server エンジンが64ビナ実行可能ファイルの場合、TRUE を返します。

■ C#

bool Is64Bit { get; }

□ C++

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

🖃 VB

ReadOnly Property Is64Bit As Boolean

LastExecutionMessage

最後のGenerate コマドの取得中に受信したメッセージを取得します。

■ C#

string LastExecutionMessage { get; }

□ C++

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

🖃 VB

ReadOnly Property LastExecutionMessage As String

MajorVersion

Style Vision Server のバジャーなバージョンを取得します。

```
■ C#
```

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

□ C++

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

🖃 VB

ReadOnly Property MajorVersion As Integer

MinorVersion

StyleVision Server のマイナーない、ージョンを取得します。

■ C#

```
int MinorVersion { get; }
```

■ C++

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

🖃 VB

ReadOnly Property MinorVersion As Integer

OutputDOCX

出力 DOCX ファイルの やと名前を設定します。

```
■ C#
string OutputDOCX { set; }
```

■ C++

HRESULT OutputDOCX([in] BSTR bstrPath);

🖃 VB

Property OutputDOCX As String

OutputFO

出力 FO ファイルの やと名前を設定します。

■ C#

string OutputFO { set; }

■ C++

HRESULT OutputFO([in] BSTR bstrPath);

🖃 VB

Property OutputFO As String

▼ OutputHTML

出力 HTML ファイルの やと名前を設定します。

```
🖃 C#
```

string OutputHTML { set; }

■ C++

HRESULT OutputHTML([in] BSTR bstrPath);

🖃 VB

Property OutputHTML As String

✓ OutputPDF

出力 PDF ファイルの マと名前を設定します。

■ C#

string OutputPDF { set; }

■ C++

HRESULT OutputPDF([in] BSTR bstrPath);

🖃 VB

Property OutputPDF As String

OutputRTF

出力 RTF ファイルの マと名前を設定します。

■ C#

string OutputRTF { set; }

□ C++

HRESULT OutputRTF([in] BSTR bstrPath);

🖃 VB

Property OutputRTF As String

ProductName

製品の名前を取得します: "StyleVision Server"

■ C#

string ProductName { get; }

■ C++

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

🖃 VB

ReadOnly Property ProductName As String

ProductNameAndVersion

バージョン番号を含む製品の完全な名前を取得します: "StyleVision Server 2014r2 sp1 (x64)"。

```
Government CH
String ProductNameAndVersion { get; }
C++
```

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

🖃 VB

ReadOnly Property ProductNameAndVersion As String

ServerPath

StyleVision Server 実行可能ファイルへの ひを取得、おけよ 設定します。

■ C#

string ServerPath { set; get; }

□ C++

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

🖃 VB

Property ServerPath As String

ServicePackVersion

StyleVision Server のサービスパックバージョンを取得します(例: 1 Altova StyleVision Server 2014 r2 sp1 (x64)のため)

■ C#

int ServicePackVersion { get; }

□ C++

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

🖃 VB

ReadOnly Property ServicePackVersion As Integer

WhereClause

処理するDB-XML スキーマノースの行を決定する SQL WHERE 句を設定します。

■ C#

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

■ C++

HRESULT WhereClause([in] BSTR bstrPath);

🖃 VB

Property WhereClause As String

WorkingDirectory

ジョブの実行のための現在のディレクトリを取得、おけよ、設定します。相対パスは作業ディレクトリーマオして評価されます。

□ C#

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

□ C++

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

🖃 VB

```
Property WorkingDirectory As String
```

5.5.2 Java

パッケージ com.altova.stylevisionserver は 以下のクラスから構成されて ます:

- public class StyleVisionServer (説明は以下を参照してください)
- public class **StyleVisionServerException** extends **Exception**

StyleVisionServer クラス

StyleVisionServer クラスは新しいStyleVision Server オブジェクトインスタンスを作成し、StyleVision Server へのアクセスを提供します。StyleVisionServer インターフェイスのメンドは下で説明されています。

StyleVisionServer クラスの メンド

StyleVisionServer クラスのメノボはアルファベナ順にしていたれて、ます。

addParameter

```
public void addParameter(String name, String value)
新規・デメーターの名前と値を追加します。各・デメーターとその値は、メノ・ドへの個別の呼び出し内で指定されています。・デメータ
ーはXSLTドキュメト内で宣言される必要があります。
<u>パテメーター</u>
name: パデメーターの名前を文字列としてホールドします。
value: パデメーターの値を文字列としてホールドします。
```

clearParameterList

```
public void clearParameterList()
パラメーターのJストを作成します。
```

generate

```
public boolean generate(String transfPath)
transfPath 内で指定されているPXF ファイルを処理します。StyleVisionServerException をスローします。
<u>パラメーター</u>
transfPath: PXF ファイルを与える絶対 URL。
<u>戻い値</u>
```

true() 実行が成功した場合 false() 実行が失敗した場合 エラーの場合、getLastExecutionMessage()を使用してたさい

getAPIMajorVersion

public int getAPIMajorVersion() StyleVision Server API のメジャーなバージョンを取得します。API が他のサーバーは接続されている場合、製品のバージョンと は異なる場合があります。

getAPIMinorVersion

public int getAPIMinorVersion() StyleVision Server API のマイナーなバージョンを取得します。API が他のサーバーは接続されている場合、製品のバージョンとは 異なる場合がおります。

getAPIServicePackVersion

public int getAPIServicePackVersion() StyleVision Server API のサービス シクバージョンを取得します。 API が他のサーバーはま続されている場合、製品のバージョン とは異なる場合がおります。

getLastExecutionMessage

public String getLastExecutionMessage() コマナの最後の生成中に受け取オシッセージを取得します。

getMajorVersion

public int getMajorVersion() アプリケーションのメジャーない、ージョンを取得します。

getMinorVersion

public int getMinorVersion() アプリケーションのマイナーなバージョンを取得します。

getProductName

public String getProductName() 製品の名前を取得します。

getProductNameAndVersion

public String getProductNameAndVersion() 製品の完全な名前とデジン番号を取得します。

getServicePackVersion

public int getServicePackVersion() StyleVision Server のサービス ツカの デンタンを取得します。

▼ is64bit

public boolean is64bit() 実行可能ファイルが64ビットであるかチェックします。 戻り値: StyleVision Server (x64)のためこ true () を返します。それ以外の場合は false () を返します。

setInputXML

public void setInputXML (String path) 処理する XML ファイルを設定します。これは、PXFファイルで指定されている作業 XML ファイルの 次である必要が決ます。 <u>パラメーター</u> path: PXF ファイル内の作業 XML ファイルの 次をホールドしています。

setOutputDOCX

public void setOutputDOCX (String path) 生成する DOCX ファイルの や名前を設定します。 <u>パラメーター:</u> path: 生成する DOCX ファイルの やと名前を設定します。

setOutputFO

public void setOutputFO(String path) 生成する FO ファイルの なと名前を設定します。 ハラメーター: path: 生成する FO ファイルの なと名前を設定します。

setOutputHTML

public void **setOutputHTML** (String path) 生成するHTML ファイルの、なと名前を設定します。 <u>パラメーター:</u> path: 生成するHTML ファイルの、なと名前を設定します。

setOutputPDF

public void **setOutputPDF**(String path) 生成する PDF ファイルの なと名前を設定します。 <u>パラメーター</u> path: 生成する PDF ファイルの なと名前を設定します。

setOutputRTF

public void setOutputRTF(String path) 生成するRTF ファイルの 次と名前を設定します。 <u>パラメーター</u> path: 生成するRTF ファイルの 次と名前を設定します。

setServerPath

```
public void setServerPath(String serverFile)
StyleVisionServer 実行可能ファイルへの 次を設定します。
<u>パラメーター:</u>
serverFile: StyleVisionServer 実行可能ファイルへの 次。
```

setWhereClause

```
public void setWhereClause(String whereClause)
処理するDB-XML スキーマノースの行を決定する SQL W HERE 句を設定します。
<u>パラメーター</u>
```

whereClause: 処理するDB-XML スキーマノースの行を決定する SQL W HERE 句。

setWorkingDirectory

public void setWorkingDirectory(String workingDirectory) デフォルのディレクトリを設定します。相対的ないなしてのディレクトリーマ対して相対的に解決されます。 <u>パラメーター:</u> workingDirectory: デフォルトの(作業)ディレカリの &。

6 XBRL タクソノミ マネジャー

XBRL タクノノミマネジャーはStyleVision Server を含むAltova XBRL-か有効化されているアプリケーション全体で使用するために XBRL タクノノミをインストールし管理する中央化した方法を提供するソールです。Windows ではXBRL タクノノミマネジャーには視覚 的なユーザーインターフェイスかあり、コマンドラインでも使用可能です。Linux とMac* 上ではコマンドラインでのみソールを使用することができます。

* Linux とmacOS オペレーティングシステムをこれらのオペレーティングシステム上で XBRL タクノノミ マネジャーを MapForce Server、Style Vision Server、おけは Raptor XML+ XBRL Server などの Altova クロスプラオ フォームサーバーと共に実行し ている場合に適用可能です。

XBRL タケノノミマネジャーは以下の機能を搭載しています:

- 使用中のエピューターにインストールされている XBRL タケノノミをビューして、新規の、ージョンをダウレロード可能か確認してください。
- Altova 製品のリレースサイクルは関わらず XBRL タケノノミの新しし V デンシンをダウンロードします。全てのタケノノミは XBRL タケ ソノミマネジャーによりアクセス可能なオンラインベースのV シケージストレージ上で Altova により管理されており、使用可能になる とすくにダウンロードすることが可能です。
- タケノノミの複数の、ージョン、おと、必要な場合全ての、ージジョン)をインストールまとはアンインストールする
- 単一のXBRL タケノノミは[パッケージ]を表しており、他のタケノノミトロな存関係が存在する場合があります。特定のタケノノミを インストールませまアンインストールすることを選択すると、依存関係を持つタケノノミが検知され、自動的にインストールませまアンイ ンストールされます。グラフィカルなユーザーインターフェイス(ませま適用できる場合コマンドラインは)依存が追加ませまず。 どを通知します。
- XBRL タケノノミマネジャーにより管理される XBRL タケノノミはインスタンス、おけはスキーマドキュメトインの URI レファレンスが インターネットからファイルを抽出することは異なり、ローカルファイルから解決されることを有効化する XML カタログ メカニズムから 利益を得ることができます。通常パンォーマンス上の理由からノモート URI からのスキーマのレゾルーションが実質的ではない場合、 大きな XBRL タケノノミの場合これはとても重要です。

XBRL タウノノミマネジャーはXBRL タウノノミの管理者が<u>Altova XBRL-有効化されたアプリケーション</u>²²の一つを使用していることを 必要とます。欧州銀行監督局レポーティングスレームワークタケノノミ、US-GAAP 金融レポーティングタウノノミ、他の多種の国別、おこ は、ドメイン固有のXBRL タウノノミが含まれます。完全なリストを確認するコはXBRL タウノノミマネジャーを実行おけましょしま コマンドラインで実行します。を参照してくたさい。

カスタム XBRL タクノノミ

XBRL タクノノミマネジャーに含まれるカスタム XBRL タクノノミと作業する場合、これらを以下のように有効化します:

- Altova XBRL-有効化済みのデスケトップアプリケーションカらツール オプション メニューコマンドを実行し、XBRL | タケノノミノッシケージ 設定ページご移動します。設定ページからカスタム XBRL タケノノミの.zip パッケージを参照してください
- サポトするコマボ おうま API メノバを実行する場合 Altova XBRL-有効化済みサーバーアプリケーションカらー taxonomy-package おうまー-taxonomy-package-config-file オブションを提供してください。例えば これらのオ プランは valxbrl おうま valxbrltaxonomy などの Raptor XML+XBRL Server 内の XBRL 検証コマボ おうま MapForce Server 内の run コマド(おうよ関連する API メノバ)によりサポトされています。

Altova XBRL-有効化済みアプリケーション

以下のAltova アプリケーションはXBRL-有効化済みで、XBRL タクノノミマネジャーによ、提供される機能によりメリットを受けています:

- Excel のためのAltova XBRL アドイン(EBA, Solvency II)
- MapForce Enterprise Edition
- MapForce Server
- MapForce Server Advanced Edition
- RaptorXML+XBRL Server
- StyleVision Server
- StyleVision Enterprise Edition
- XMLSpy Enterprise Edition

XBRL タケノノミマネジャーを使用した XBRL タケノノミ内での変更は同じコンピューターにインストールされている場合、上にリストされたすべてのアプリケーションで反映されます。

インストール

Altova XBRL-有効化済みアプリケーション、おける Altova Mission Kit Enterprise Edition をインストールすると XBRL タウノノ ミマネジャー は自動的にインストールされます。同様に、Altova XBRL-有効化済みアプリケーション、おける Altova Mission Kit Enterprise Edition をエピューターからアンインストールすると自動的にアンインストールされます。

しくみ

Altova は製品内で使用されている全てのXBRL タケノノミを保管するオンラインタケノノミストレージを管理しています。このタケノノミストレージは定期的に更新されます。例えば、発行機関により関連するタケノノミの新規の、レジョンがリレースされた後などに更新されます。

XBRL タケノノミマネジャーをグラフィカルなユーザーインターフェイスで実行すると、タケノノミをインストール、更新ませるエンインストールすることができるダイアログボックス内に使用可能な最新のタケノノミの情報が表示されます。コマンドラインインターフェイスで同じアクションを実行することができます。

🌆 XBRL タクソノミ パッケージ	×
ACPR COREP - Common Reporting with the modules LCR_CON, LCR_DA, NSFR_	CON an
ACPR LCB-FT - Combating Money Laundering and the Financing of Terrorism	
BBK - EBA-ITS-Reporting extended by the German base taxonomy	
🗄 🔲 BDP - Financial and Common Reporting framework	
BOE BANKING - Regulatory reporting for the banking sector	
BOE INSURANCE - Regulatory reporting for the insurance sector	
BOE STATISTICS - Statistical reporting obligations	
🛨 🗌 CBI - Solvency II National Specific Templates	
🗄 🔳 EBA - Reporting framework provided by the European Banking Authority	
EIOPA PENSIONFUNDS - Supervisory reporting for Pension Funds	
EIOPA SOLVENCY-II - Supervisory Solvency II reporting	
🕀 🔲 ESEF - European Single Electronic Format taxonomy	
🖽 🗹 Eurofiling - European XBRL taxonomy architecture	
FTK - Financial Assessment Framework for DNB FTK reports	-
4	•
タクソノミ eurofiling-1.0.0 は既にインスト ールされています。 eiopa-solvencyII-2.5.0 をインスト ールする。	^
フオルダー C:¥ProgramData¥Altova¥pkgs¥xbrl¥eiopa-solvencyII¥の作成 Altova タクソリミフトアからダウンロードする:eiopa-solvencyII+2.5_rules_0.txt	
Altova タクソノミストアからダウンロードする: EIOPA_SolvencyII_XBRL_Taxonomy_2.5.0_0.zip。	
UK	
	\sim
<	>
パッチの選択(P) 全ての解除(D) 選択のリセット(R) 適用(A)	閉じる(C)

Altova W eb サイト (<u>https://www.altova.com/download</u>)からダウレロードされたrunning .altova_taxonomies ファイルを実行し てタウノノミをインストールすることができます。更に XBRL タクノノミをロード するアウションを取る場合 Style Vision Server は不足する XBRL タクノノミをインストールするようこプロンプトします。

タケノノミのインストール方法に関わらず、インストール済みのタケノノミニ関する情報はエンピューター内で中央化された場所でトラックされてい

ます。これはローカルキャッシュディレクトリとして知られています。ローカルキャッシュディレクトリコはオンデマンドでインストールされる実際のタクノ ノミファイルを除き)Altova パッケージン関する情報が含まれています。ローカルキャッシュディレクトリは以下の。父で見つけることができます:

Linux	/var/opt/Altova/pkgs	
macOS	/var/Altova/pkgs	
Windows	C:\ProgramData\Altova\pkgs	

ローカルコンユーターにオンラインストレージの最新の状態を作成するながコローカルキャッシュディレクトリは時々自動的に更新されます。更に具体的には、キャッシュ」は以下のように更新されます

- XBRL タクノノミマネジャーを実行 する場合。
- 同じカレンダーの日付で Style Vision Server を実行する場合。
- StyleVision Server か既に実行中の場合、キャッシュディレクトリは24 時間ごとこ更新されます。
- コマボラインインターフェイスから更新を実行しローカルキャッシュをオンラインストレージからオンデマボで更新することができます。

タケノノミをインストールはたまアンインストールすると、ローカルキャッシュディレクトリは使用可能な情報、およびインストールされているタケノノミの情報、およびタケノノミファイル自身と共に自動的に更新されます。

ローカルキャシュディレクトリはインストールおけばアンインストールはるタクノノミをベースに自動的に管理されます。手動で変更おけば削除されてはないません。XBRL タクノノミマネジャーを元の状態にしたよする必要がある場合、コマイドラインインターフェイスのreset コマイドを実行してからinitialize コマイドを実行します。おけまreset コマイドを - i オプノョンと共に実行してくたさい。

6.1 XBRL タクソノミ マネジャー の実行

XBRL タクノノミマネジャーを下にリストされるアプローチで適用可能な場合実行することができます。

グラフィカルなユーザーインターフェイス(Windows のみ)

StyleVision Server がW indows 上で作動する場合 XBRL タケノノミ マネジャーを以下のように作動することができます:

- Altova Web サイトからダウロードされた.altova_taxonomies 拡張子を持つファイルを(ダブルクトックして) 実行します。
- Windows エトロール やりから Altova タケノノミマネジャー エトリを右クリックして、変更 おっぱ アンインストールをコンテキストメニューから選択します。

更に、StyleVision Server のインストール後チェックボックス Altova タクノノミマネージャーの呼び出し がインストールウィザードの最後のページで使用可能にないます。

コマンドライン インターフェイス(Windows)

XBRL タケノノミ マネジャーをコマンドインターフェイスから実行する方法

- 1. コマドプロンプを開きデルクトレを C:\ProgramData\Altova\SharedBetweenVersions に変更します。
- 2. コマボラインでへいた表示するコよ 以下を実行します:

TaxonomyManager.exe --help

コマンドライン インターフェイス (Linux, macOS)

XBRL タケノノミマネジャーをコマンドインターフェイスから実行する方法

- 1. ターミナルウインドウを開き % INSTALLDIR% がプログラムのインストールディレクトリである % INSTALLDIR% /bin にディレクトリを変更します。を参照します。
- 2. コマボラインでへいた表示するコよ 以下を実行します:

sudo ./taxonomymanager --help

6.2 タクソノミのインストール

タケノノミを下にリストされるアプローチで適用可能な場合インストールすることができます。

以下の点に注意してくたさい

- XBRL タケノノミマネジャーからタケノノミをインストールはオゴエアンインストールすると同じエンピューター上のすべてのユーザーアカナントに影響を及ぼします。
- XBRL タケノノミマネジャーからタケノノミをインストールまけはアンインストールすると同じエンピューター上にインストールされている Altova XBRL-有効化済みのすべてのアプリケーションイ影響を与えます。
- 現在のタケノノミトは他のタケノノミトに依存関係があり、依存するタケノノミもインストールされます(おける適用可能な場合アンインストールされます)。

グラフィカルなユーザーインターフェイス(Windows のみ)

タケノノミをインストールする方法

- 1. 実行する。
- 2. インストールするタケノノミオビオタケノノミバージョンの横のチェックボックスを選択し「適用」をクリックします。

ませは.altova_taxonomies 拡張子を持つファイルを Altova Web (<u>https://www.altova.com/taxonomy-manager</u>)サイトから ダウンロードし、.altova_taxonomies ファイルを実行するためごダブルクトックします。.altova_taxonomies ファイルを実行する際に XBRL タクノノミ マネジャー か開かれます。

コマボライン インターフェイス(Windows)

タケノノミをインストールするコよ、以下を実行します。

TaxonomyManager.exe install FILTER...

FILTER 加以下を意味する箇所:

- 1. フォーマホ <name>-<version> 内のタクノノミ識別子の別 eba-2.10、us-gaap-2020.0。使用可能なすべてのタクノ ノミ識別子をビューするゴはlist コマイを実行します。
- 2. Altova Web サイトからダウレードされたaltova_taxonomies ファイル。

コマンドライン インターフェイス (Linux, macOS)

タケノノミをインストールするコよ、以下を実行します。

sudo ./taxonomymanager install FILTER...

FILTER 加以下を意味する箇所

- 1. フォーマホ <name>-<version> 内のタケルミ識別子の別 eba-2.10、us-gaap-2020.0。使用可能なすべてのタケル ノミ識別子をビューするコはlist コマイを実行します。
- 2. Altova Web サイトからダウノロードされたaltova_taxonomies ファイル。

オンデマンドでタクノノミをインストールする

StyleVision Server か特定のXBRL タクノノミを検知すると使用中のコンピューターにインストールされる必要かあります。オンデマンドでタ

クノノミをインストールするようこプロレプトされる場合があります。例えば、XBRL タクノノミのロードが必要なコマンドを実行し、XBRL タクノノノミが現在インストールされていない場合でも、コマンドはキャンセルされ、いかに類似したエラーメッセージが表示されます。

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

不足するタクノノミをインストールコはinstall コマンドをエラーメッセージには提案された通り実行します。list コマンドを実行して以前 にインストールされたすべてのタクノノミを常に確認することができます。

パッチの適用

不定期にXBRL タクノノミは発行元から、シチを受け取ります。XBRL タクノノミマネジャーか、シチが使用可能であることを検知すると、以下か表示されます。

- Windows の規覚的なユーザーインターフェイスを介して XBRL タクノノミマネジャーを使用すると、対応する XBRL タクノノミは アイコンと共に表示されます。
- コマドラインおけるLinux/macOS システムを使用すると、list -u コマドと共に実行可能ファイルを実行する際にしたされている。 かチを持つXBRL タクノノミがしてたわます。

W indows 上でパッチを適用する方法:

1. [パッチの選択] ボタンをクトックします。 🛃 から 😫 への変更を修飾する各 XBRL タクノノミのアイエンとダイアログボックス は適用される やチニス て通知します。例:

🔯 XBRL Taxonomy Manager	×
🖽 🗌 ACPR COREP - Common Reporting with the modules LCR CON, LCR DA, NSFR CON and ALM	
ACPR LCB-FT - Combating Money Laundering and the Financing of Terrorism	
BBK - EBA-ITS-Reporting extended by the German base taxonomy	
BDP - Financial and Common Reporting framework	
BOE BANKING - Regulatory reporting for the banking sector	
BOE INSURANCE - Regulatory reporting for the insurance sector	
BOE STATISTICS - Statistical reporting obligations	
CBI - Solvency II National Specific Templates	
EBA - Reporting framework provided by the European Banking Authority For EBA relevant data transmission or collecting data for credit institutions and investment firr	
The st 2 10 Phase 2	
	-
The following packages will be patched: EBA 2. 10 Phase 2 - Reporting framework provided by the European Banking Authority EIOPA SOLVENCY-II 2.5 - Supervisory Solvency II reporting WIP 2016 - US Work In Process taxonomy	
<	
Patch Selection Deselect All Reset Selection Apply Cancel	

メモ パッチの選択ボタイは現在インストールされているXBRL タケノノミのためコッシチが使用な箇所でのみ有効化されています。

2. [適用]をクルクします。

コマンドラインインターフェイスでパッチを適用する方法:

- 1. list -u コマンドを実行します。パッチ更新が使用可能なタクノノミをリストします。
- 2. upgrade コマイを実行して ジチを インストールします。

更に詳しい情報に関してはコマイドラインインターフェイスを参照してくたさい。

6.3 インストール済みのタクソノミをビューする

現在インストールされているタケノノミを下にリストされるアプローチで適用可能な場合ビューすることができます。

グラフィカルなユーザーインターフェイス(Windows のみ)

視覚的なユーザーインターフェイスからインストール済みのすべてのタクノノミをビューするコはXBRL タクノノミマネジャーを実行します。タクノ ノミ(まけよ、タクノノミのバージョン)の横の選択済みのチェックボックスはそのタクノノミがインストール済みであることを示しています。

コマンドライン インターフェイス(Windows)

コマボラインインターフェイスからすべての使用可能なタクノノミをビューするコよ、以下を実行します

TaxonomyManager.exe list

インストール済みのタケノノミのみをビューするコよ 以下を実行します:

TaxonomyManager.exe list -i

新規のデジョンが使用可能なタクノノミのみをビューするコよ、以下を実行します

TaxonomyManager.exe list -u

コマンドライン インターフェイス (Linux, macOS)

使用可能な全てのタクノノミをビューするコよ、以下を実行します

sudo ./taxonomymanager list

インストール済みのタケノノミのみをビューするコよ 以下を実行します

sudo ./taxonomymanager list -i

新規の、シジョンが使用可能なタケノノミのみをビューするコよ、以下を実行します。

sudo ./taxonomymanager list -u

6.4 タクソノミのアンインストール

タケノノミを下にリストされるアプローチで適用可能な場合アンインストールすることができます。

グラフィカルなユーザーインターフェイス(Windows のみ)

タケノノミをアンインストールする方法

- 1. 実行する。
- 2. アンインストールするタケノノミオゴコタケノノミバージョンの横のチェックボックスのチェックを解除します。選択されたタケノノミが他のタク ソノミパッケージコな存する場合、依存関係も削除されることを通知するダイアログボックスの開かれます。例:

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

3. [適用]をクリックしてタクノノミとその旅存関係を削除します。

コマンドライン インターフェイス(Windows) タケノノミをアンインストーリオるコよ 以下を実行します:

ダノノノミをアンインストーフレタロコよ以下を美行します:

TaxonomyManager.exe uninstall FILTER...

FILTER 加以下を意味する箇所

- 1. フォーマホ <name>-<version> 内のタケノノミ識別子の別: eba-2.10、us-gaap-2020.0。使用可能なすべてのタケノ ノミ識別子をビューするコはlist コマナを実行します。
- 2. Altova Web サイトからダウノロードされたaltova_taxonomies ファイル。

コマンドライン インターフェイス (Linux, macOS)

パッケージをアンインストールするコよ以下を実行します。

sudo ./taxonomymanager uninstall FILTER...

FILTER が以下を意味する箇所:

- 1. フォーマホ <name>-<version> 内のタケノノミ識別子の別: eba-2.10、us-gaap-2020.0。使用可能なすべてのタケノ ノミ識別子をビューするコはlist コマンドを実行します。
- 2. Altova Web サイトカらダウレードされたaltova_taxonomies ファイル。

6.5 コマンドライン インターフェイス

XBRL タクノノミマネジャーをコマ、ドラインで呼び出す」コは場合実行可能ファイルの ひを知っている必要がみます。デフォルトで XBRL タクノノミマネジャー実行可能ファイルを以下の ひで見つけることができます。

Linux*	/opt/Altova/StyleVisionServer2021/bin/taxonomymanager	
macOS*	/usr/local/Altova/StyleVisionServer2021/bin/taxonomymanager	
Windows	C:\ProgramData\Altova\SharedBetweenVersions\TaxonomyManager.exe	

* Linux とmacOS パンをこれらのオペレーティングシステム上で XBRL タクノノミ マネジャーを MapForce Server、 Style Vision Server、おけよ Raptor XML+ XBRL Server などの Altova クロスプラナ フォームサーバーと共に実行している場合に適用可能です。

規則としては、このドキュメンテーションは特定のコマンドを説明する際に実行可能ファイルのフルルマを省略し、実行可能ファイルのなめを使用する代わりに、exec>を使用します。例

<exec> help

<exec> は実行可能ファイルの やませる前です。

XBRL タケノノミ マネジャーをインストールディレクトリから呼び出す方法

実行可能ファイルをフルプを入力することなく呼び出すコよ現在のディレクトリを以下のコン変更します。

Linux	cd /opt/Altova/StyleVisionServer2021/bin	
macOS	cd /usr/local/Altova/StyleVisionServer2021/bin	
Windows	cd "C:\ProgramData\Altova\SharedBetweenVersions"	

相対的ないなを使用してコマンドを実行し実行可能ファイルを呼び出すことができます。

Linux	sudo ./taxonomymanager help	
macOS	sudo ./taxonomymanager help	
Windows	TaxonomyManager.exe help	

メモ Linux とmacOS システム上では、プレフィックス./ は実行可能ファイルが現在のディレクトリ内にあることを示しています。sudo はコマンドがレート特権と共に実行されなければならないことを示しています。

XBRL タケノノミ マネジャーをディレクトリから呼び出す方法

実行可能ファイルをディレクトリめら呼び出すココ絶対ルズを使用して参照してくたさい。おけよ、プログラムを実行可能ファイル名を入力する だけで呼び出す場合、使用中のオペレーティングシステムのPATH環境変数を編集することにより、実行可能ファイルのディレクトリーのフル パマか含まれるようしています。PATH環境変数を変更する方法に関しては、使用中のオペレーティングシステムのドキュメンテーションを参照

してくたい。

灹

- PATH 環境変数の変更後、ターミナルウイドウを閉じ、変更の効果が反映されるように新規のウイドウを開く必要かある場合 があります。
- Linux とmacOS 上では sudo の使用はユーザーのPATH を考慮していません。

コマボライン構文

コマボラへを使用するための一般的な構文は以下の通りです

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

上のノスト内で垂直バー」は相互に排他的なアイテムを分割します。角かっては「」は任意のアイテムを示しています。基本的には、--h, --help、おけよ--versionオプタンおけよコマバが後に続く実行可能ファイルりを入力することができます。各コマバはオプタンと 引数を持つ場合があります。コマバのノストは以下のセグタンで説明されています。

6.5.1 help

このコマンドはXBRL タケノノミマネジャー実行可能ファイルは関連するコマンドに関するコンテキストのヘルレを提供します。

構文

<exec> help [command]

[command] が有効なコマド名を指定するオプションの引数である個所。

リマーク

-h おけよ--help が後に続くコマイを入力してへいプを呼び出すことができます。例:

<exec> list -h

実行可能ファイルの後に-h おけま--help を入力して一般的なヘルプを呼び出すことができます。例:

<exec> --help

サンプル

次のコマドはlist コマドに関するへいを表示します:

Linux, macOS	./taxonomymanager help list
Windows	TaxonomyManager.exe help list

6.5.2 info

このコマトーは引数として与えられたタケノノミのそれそれの詳細を表示します。詳細にはタイトル、バージョン、詳細、後悔元、依存レファレンスが含まれています。

構文

<exec> info [options] FILTER...

複数のタクノノミを指定するコはFILTER を必要な回数繰り返します。

引数

FILTER	名前内にこの文字列を含むタケノノミをリストします(例えば、eba ませまus-gaap-2020.0)。この引数を複数回指定することができます。
	識別子のフルリストをビューするコはlist コマイを実行します。

オプション

help,h	コマボラインでこのコマンドに関するへいとを表示します。
--------	-----------------------------

例

以下のコマイはeba-2.10とus-gaap-2020.0 タケノノミに関する詳細を表示します:

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

6.5.3 initialize

このコマイドはXBRL タケノノミマネジャー環境を初期化します。タケノノミニ関するすべての情報が保管されている個所でキャッシュデルクトリを作成します。XBRL タケノノミマネジャーを含む Altova アプリケーションをインストールすると初回に初期化が実行されるため、通常このコマイドを実行する必要はありません。reset コマイドを実行後に通常このコマイドを実行する必要があります。

構文

<exec> initialize [options]

このコマンドのコイリアスはinit です。

オプション

--help, --h コマボラインでこのコマボに関するへいた表示します。

silent,s	エラーメッセージのみを表示します。デフォハトの値はfalseです。
verbose,v	実行中の追加情報を表示します。デフォルトの値はfalse です。

6.5.4 install

このコマドはつませる複数のタケノノミをインストールします。以下の点に注意してくたさい

- XBRL タケノノミマネジャーからタケノノミをインストールはオゴムアンインストールすると同じエンピューター上のすべてのユーザーアカナントに影響を及ぼします。
- XBRL タケノノミマネジャーからタケノノミをインストールまたはアンインストールすると同じエンピューター上にインストールされている Altova XBRL-有効化済みのすべてのアプリケーションイ影響を与えます。
- 現在のタケノノミレオルのタケノノミトに依存関係があり、依存するタケノノミもインストールされます(おける適用可能な場合アンインストールされます)。

構文

<exec> install [options] FILTER...

複数のタクノノミを指定するコはFILTER を必要な回数繰り返します。

引数

FILTER	FILTER が以下を意味する箇所
	 フォーマト <name>-<version> 内のタクノノミ識別子の例: eba-2.10、us- gaap-2020.0。使用可能なすべてのタクノノミ識別子をビューするゴはlist コマ・ド を実行します。</version></name> Altova W eb サイトからダウノロードされたaltova_taxonomies ファイル。
	ー意の場合省略された識別子を使用することができます、例えば、eba。省略された識別子を使用する場合、そのタケノノミの使用可能な最新バージョンがインストールされます。

オプション

help,h	コマボラインでこのコマボに関するへいた表示します。
silent,s	エラーメッセージのみを表示します。デフォルトの値はfalseです。
verbose,v	実行中の追加情報を表示します。デフォノトの値はfalse です。

サンプル

以下のコマイは最新のeba (欧州銀行監督局) とus-gaap (米国会計基準) タケノノミをインストールします。

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

6.5.5 list

このコマボを使用して以下の方法のいつこと」ロマボラインでタクノノミをリストします。

- 使用可能なタクノノミを全てリストします
- 特定のタケノノミをリストします
- インストール済みのタケノノミのみをリストします
- アップグレードが必要なタクノノミのみをリストします

構文

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

このコマドは1sを使用して省略することができます。

引数

1		
	FILTER	名前内にこの文字列を含むタケノノミのみをリストします。この引数を複数回指定することができま
		9.0

オプション

help,h	コマボラインでこのコマボに関するへいた表示します。
installed,i	インストール済みのタケノノミのみをリストしますデフォルトの値はfalse です。
upgradeable,u	パッチ更新が使用可能なタクノノミのみをリストします。デフォルトの値はfalse です。

例

使用可能な全てのタケノノミをリストするコよ 以下を実行します:

Linux, macOS	./taxonomymanager list
Windows	TaxonomyManager.exe list

インストール済みのタケノノミのみをリストするコよ 以下を実行します:

Linux, macOS	./taxonomymanager list -i
Windows	TaxonomyManager.exe list -i

名前内に "eba" おけは "us-gaap" を含むすべてのタケノノミをリスト するけは 以下を実行します:

ux, macOS	./taxonomymanager list eba us-gaap	
-----------	------------------------------------	--

Windows

TaxonomyManager.exe list eba us-gaap

6.5.6 reset

このコマドはすべてのインストールされたタケノノミとキャッシュディレクトリを削除します。

警告: このコマンドはすべてのインストールされたタケノノミとその情報を削除します。

このコマドの実行後、キャシュデルケーを再作成するためにinitializeコマドを実行してくたさい。おはresetコマドを-iオプランと共に実行してくたさい。

reset -i は製品オバナルのインストールを復元するため、リセオ後に更にupdate コマイを実行することが奨励されます。おけよ -i と-u オプロンを使用してコマイをルナします。

構文

<exec> reset [options]

オプション

help,h	コマボラインでこのコマボに関するヘルプを表示します。
init,i	XBRL タクノノミマネジャー環境をルセナ後初期化してくたさい。有効な値はtrue おけはfalse です。デフォルトの値はfalse です。
silent,s	エラーメッセージのみを表示します。デフォルトの値はfalseです。
update,u	XBRL タケノノミマネジャー環境をルセナ後に初期化し更新してください。有効な値はtrue また はfalse です。デフォルトの値はfalse です。
verbose,v	実行中の追加情報を表示します。デフォハトの値はfalse です。

例

XBRL タクノノミマネジャーをやけするコよ以下を実行します:

Linux, macOS	./taxonomymanager reset
Windows	TaxonomyManager.exe reset

6.5.7 setdeflang

このコマドはXBRL タケノノミマネジャーの言語を設定します。

構文

<exec> setdeflang language

language か言語コードを提供する必須の引数の箇所。このコマトのエイリアスはsdl です。

引数

language	設定する言語有効な言語は英語(en)、プランス語、(fr)、ドイン語(de)、日本語(ja)、およ びズペイン語(es)です。
----------	--

オプション

help,h	コマボラインでこのコマンドに関するヘルレを表示します。
--------	-----------------------------

サンプル

言語をスペイン語に設定するコよ 以下を実行します:

Linux, macOS	./taxonomymanager setdeflang es
Windows	TaxonomyManager.exe setdeflang es

6.5.8 uninstall

このコマンドはつまけは複数のタケノノミをアンインストールます。デフォルトでは現在のタケノノミにと参照されるタケノノミもアンインストールされます。現在のタケノノミゼノをアンインストールし、参照されたタケノノミを保つコンオプション -- k を設定します。

構文

<exec> uninstall FILTER...

複数のタクノノミを指定するコは FILTER を必要な回数繰り返します。

引数

FILTER	FILTER か以下を意味する箇所:
	1. フォーマト <name>-<version> 内のタクノノミ識別子の例: eba-2.10、us- gaap-2020.0。使用可能なすべてのタクノノミ識別子をビューするコはlist コマナ を実行します。</version></name>

2. Altova Web サイトからダウレードされたaltova_taxonomies ファイル

オプション

help,h	コマボラインでこのコマボに関するへいを表示します。
keep-references, k	このオプションか設定されると、参照されたタケノノミはアンインストールされません。デフォルトの値は false です。
silent,s	エラーメッセージのみを表示します。デフォルトの値はfalseです。
verbose,v	実行中の追加情報を表示します。デフォルトの値はfalse です。

サンプル

以下のコマイはeba-2.10とus-gaap-2020.0 タケノノミとその 旅存関係をアンインストール ます:

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

6.5.9 update

このコマボクエルオンラインストレージから使用可能なタクノノミのノストをクエル、ローカルのキャッシュディレクトリを更新します。この精報の 更新は明示的に行われ、reset とinitializeを実行した場合を除いてこのコマボを実行する必要はありません。

構文

<exec> update [options]

オプション

help,h	コマボラインでこのコマボに関するへいた表示します。
silent,s	エラーメッセージのみを表示します。デフォルトの値はfalseです。
verbose,v	実行中の追加情報を表示します。デフォルトの値はfalse です。

サンプル

以下のコマンドはタクノノミの情報を明示的に更新します。

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

6.5.10 upgrade

このコマンドは全ての対象のタケノノミを使用可能な最新 patch バージョンにアップグレードします。すなオタ、特定のリースの やチレベル でのみ更新を行います。このコマンドを実行することは使用可能なタケノノミを更新する場合のみ意味かあります。list -u コマンドを実行 して更新可能なタケノノミを識別することができます。

構文

<exec> upgrade [options]

オプション

help,h	コマボラインでこのコマボに関するヘルプを表示します。
silent,s	エラーメッセージのみを表示します。デフォハトの値はfalseです。
verbose,v	実行中の追加情報を表示します。デフォルトの値はfalse です。

インデックス

F

FlowForce Server, & StyleVision Server, 9

L

Linux, でのライセンス, 22 へのインストール, 19 Linux でのライセンス, 22 Linux へのインストール, 19

Μ

macOS, でのライセンス, 27 へのインストール, 25 macOS でのライセンス, 27 macOS へのインストール, 25

S

StyleVision Server, FlowForce ワークフロー内, 9 機能, 8

W

Windows, でのライセンス, 16 へのインストール, 13 Windows でのライセンス, 16 Windows へのインストール, 13 Х

XBRL タクソノミ マネジャー, 72

Ζ

コア, 12 セットアップ, Linux での, 19 Mac OS X での, 25 プロセッサーコア, 12 ライセンスとプロセッサーコア, 12 設定, Windows 上, 13 単ースレッド実行, 12