

Altova Authentic 2021 Desktop

ユーザーマニュアル

Altova Authentic 2021 Desktop 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	Authentic Desktop とこのドキュメントについて	10
1.1	Windows ファイルパス.....	11
1.2	このドキュメントについて.....	12
2	インターフェイスと環境	13
2.1	グラフィカルユーザインターフェイス (GUI).....	14
2.1.1	メイン ウィンドウ.....	15
2.1.2	プロジェクト ウィンドウ.....	16
2.1.3	情報ウィンドウ.....	18
2.1.4	入力ヘルパー.....	18
2.1.5	メニューバー、ツールバー、ステータスバー.....	19
2.2	アプリケーション環境.....	20
2.2.1	設定、カスタマイズ.....	20
2.2.2	チュートリアル、プロジェクト、サンプル.....	20
2.2.3	Authentic Desktop 製品機能と Altova 製品.....	21
3	Authentic View チュートリアル	23
3.1	AUTH-VIEW 内で XML ドキュメントを開く.....	25
3.2	Authentic View インターフェイス.....	27
3.3	ノードを操作する.....	30
3.4	Authentic View 内にデータを挿入する.....	33
3.5	属性の値を入力する.....	35
3.6	エンティティを入力する.....	36
3.7	ドキュメントを印刷する.....	37
4	Authentic View インターフェイス	38
4.1	GUI の概要.....	39

4.2	Authentic View ツールバーアイコン.....	41
4.3	Authentic View メインウィンドウ.....	44
4.4	Authentic View エントリヘルパー.....	47
4.5	Authentic View コンテキストメニュー.....	51
5	Authentic View 内で編集する	53
5.1	ファイルを自動的にバックアップする.....	54
5.2	基本的な編集.....	56
5.3	Authentic View 内のテーブル.....	60
5.3.1	SPS テーブル.....	60
5.3.2	CALS/HTML テーブル.....	61
5.3.3	CALS/HTML テーブル編集アイコン.....	65
5.4	DB の編集.....	69
5.4.1	DB テーブル内をナビゲートする.....	69
5.4.2	DB クエリ.....	70
5.4.3	DB テーブルの変更.....	74
5.5	日付と作業する.....	76
5.5.1	日付の選択.....	76
5.5.2	テキストのエントリ.....	77
5.6	エンティティを定義する.....	79
5.7	XML 署名.....	81
5.8	Authentic View 内のイメージ.....	82
5.9	Authentic View 内のキーストロール.....	83
6	Authentic スクリプト	84
7	ブラウザービュー	86
8	Altova グローバルリソース	87
8.1	グローバル リソースの定義.....	88
8.1.1	ファイル.....	89
8.1.2	フォルダー.....	93

8.1.3	データベース.....	95
8.2	グローバルリソースを使用する.....	97
8.2.1	ファイルとフォルダーの割り当て.....	97
8.2.2	構成の切り替え.....	100
9	Source Control	101
9.1	.Setting Up Source Control.....	103
9.2	.Supported Source Control Systems.....	104
9.3	.Local Workspace Folder.....	106
9.4	.Application Project.....	107
9.5	.Add to Source Control.....	109
9.6	.Working with Source Control.....	111
9.6.1	Add to, Remove from Source Control.....	111
9.6.2	Check Out, Check In.....	112
9.6.3	Getting Files as Read-Only.....	114
9.6.4	Copying and Sharing from Source Control.....	116
9.6.5	Changing Source Control.....	119
9.7	.Source Control with Git.....	121
9.7.1	Enabling Git Source Control with GIT SCC Plug-in.....	122
9.7.2	Adding a Project to Git Source Control.....	122
9.7.3	Cloning a Project from Git Source Control.....	123
10	メニューコマンド	126
10.1	.ファイルメニュー.....	127
10.1.1	新規作成.....	127
10.1.2	開く.....	128
10.1.3	再ロード.....	133
10.1.4	エンコード.....	133
10.1.5	閉じる、全て閉じる、非アクティブを全て閉じる.....	134
10.1.6	保存、名前を付けて保存、全て保存.....	134
10.1.7	メールで送信.....	139
10.1.8	印刷.....	140
10.1.9	印刷プレビュー、印刷設定.....	140

10.1.10	最近使用されたファイル、終了.....	141
10.2	編集メニュー.....	142
10.2.1	元に戻す、やり直し.....	142
10.2.2	切り取り、コピー、貼り付け、削除.....	142
10.2.3	全て選択.....	143
10.2.4	検索、次を検索.....	143
10.2.5	置換.....	143
10.3	プロジェクトメニュー.....	145
10.3.1	新規プロジェクト.....	147
10.3.2	プロジェクトを開く.....	147
10.3.3	プロジェクトの再ロード.....	148
10.3.4	プロジェクトを閉じる.....	148
10.3.5	プロジェクトの保存、名前を付けて保存.....	148
10.3.6	ソース管理.....	148
10.3.7	プロジェクトにファイルを追加.....	162
10.3.8	プロジェクトにグローバルリソースを追加.....	162
10.3.9	プロジェクトに URL を追加.....	162
10.3.10	プロジェクトにアクティブなファイルを追加.....	162
10.3.11	プロジェクトにアクティブならびに関係するファイルを追加.....	163
10.3.12	プロジェクトにアクティブならびに関係するフォルダーを追加.....	163
10.3.13	プロジェクトに外部フォルダーを追加.....	163
10.3.14	プロジェクトに外部ウェブフォルダーを追加.....	166
10.3.15	スクリプトの設定.....	171
10.3.16	プロパティ.....	171
10.3.17	最近使用されたプロジェクト.....	174
10.4	XML メニュー.....	175
10.4.1	整形形式のチェック.....	175
10.4.2	XML の検証.....	176
10.4.3	編集中に検証.....	177
10.5	XSL/XQuery メニュー.....	178
10.5.1	XSL 変換.....	178
10.5.2	XSL-FO 変換.....	179
10.5.3	XSL パラメーター / XQuery 実行.....	180
10.6	Authentic メニュー.....	185
10.6.1	新規ドキュメント.....	185

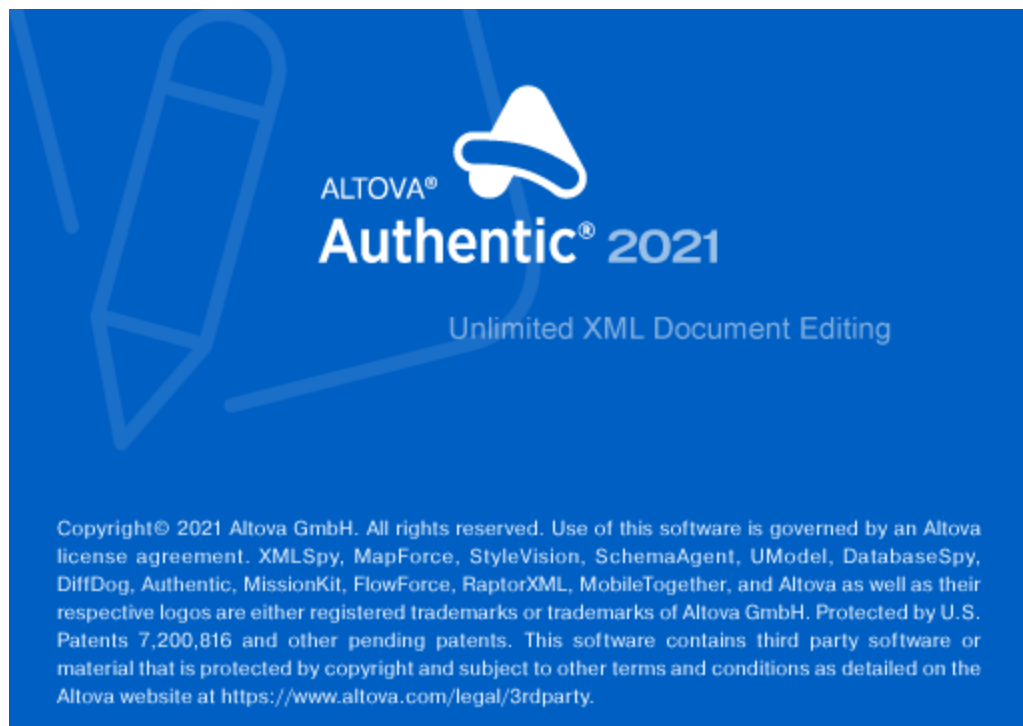
10.6.2	データベースの編集.....	186
10.6.3	StyleVision スタイルシートの編集.....	187
10.6.4	XML データの新たな行を選択し編集.....	187
10.6.5	XML 署名.....	188
10.6.6	XML エンティティの定義.....	190
10.6.7	マークアップの表示.....	191
10.6.8	行の追加/挿入/複製/削除.....	192
10.6.9	上に/下に移動.....	192
10.6.10	HTML、RTF、PDF、Word 2007+ ドキュメントの生成.....	192
10.6.11	信頼された場所.....	193
10.7	表示メニュー.....	194
10.7.1	Authentic View.....	194
10.7.2	ブラウザービュー.....	194
10.8	ブラウザーメニュー.....	195
10.8.1	戻る.....	195
10.8.2	進む.....	195
10.8.3	中止.....	196
10.8.4	最新の状態に更新.....	196
10.8.5	フォント.....	196
10.8.6	別のウィンドウ.....	196
10.9	ツールメニュー.....	197
10.9.1	スペリング.....	197
10.9.2	スペルチェックのオプション.....	200
10.9.3	スクリプトエディター.....	203
10.9.4	マクロ.....	204
10.9.5	グローバルリソース.....	204
10.9.6	アクティブな構成.....	205
10.9.7	カスタマイズ.....	205
10.9.8	ツールバーとウィンドウの復元.....	218
10.9.9	オプション.....	219
10.10	ウィンドウメニュー.....	231
10.10.1	重ねて表示.....	231
10.10.2	上下に並べて表示.....	231
10.10.3	左右に並べて表示.....	231
10.10.4	プロジェクトウィンドウ.....	231

10.10.5	情報ウィンドウ.....	232
10.10.6	入力ヘルパー.....	232
10.10.7	出力ウィンドウ.....	232
10.10.8	プロジェクトおよび入力ヘルパー.....	232
10.10.9	全てオン/オフ.....	232
10.10.10	現在開かれているウィンドウのリスト.....	233
10.11	ヘルプメニュー.....	234
10.11.1	目次、インデックス、検索.....	234
10.11.2	キーボードマップ.....	235
10.11.3	ライセンス登録、注文フォーム、登録、最新情報のチェック.....	235
10.11.4	他のコマンド.....	238
10.12	コマンドライン.....	240
11	Visual Studio 内の Authentic Desktop	241
11.1	Visual Studio のための Authentic Desktop プラグインのインストール.....	242
11.2	スタンドアロンバージョンとの違い.....	243
12	Eclipse 内の XMLSpy	244
12.1	Eclipse のための Authentic Desktop プラグインのインストール.....	245
12.2	Eclipse 内の Authentic Desktop エントリポイント.....	247
13	プログラマーのレファレンス	250
13.1	Scripting Editor.....	252
13.1.1	Creating a Scripting Project.....	253
13.1.2	Built-in Commands.....	265
13.1.3	Enabling Scripts and Macros.....	275
13.2	IDE Plugins.....	277
13.2.1	Registration of IDE Plugins.....	277
13.2.2	ActiveX Controls.....	278
13.2.3	Configuration XML.....	278
13.2.4	ATL Sample Files.....	281
13.2.5	IXMLSpyPlugIn.....	286
13.3	Application API.....	291

13.3.1	Overview.....	292
13.3.2	Interfaces.....	319
13.3.3	Enumerations.....	542
13.4	ActiveX Integration.....	556
13.4.1	必要条件.....	556
13.4.2	ActiveX コントロールをツールボックスに追加する.....	557
13.4.3	アプリケーションレベルの統合.....	559
13.4.4	ドキュメントレベルの統合.....	561
13.4.5	ActiveX 統合のサンプル.....	564
13.4.6	コマンド レファレンス.....	579
13.4.7	オブジェクトレファレンス.....	586
14	付録	608
14.1	技術データ.....	609
14.1.1	OS とメモリ要件.....	609
14.1.2	Altova XML バリデーター.....	609
14.1.3	Altova XSLT と XQuery エンジン.....	609
14.1.4	Unicode のサポート.....	609
14.1.5	インターネットの使用.....	610
14.2	ライセンス情報.....	611
14.2.1	電子的なソフトウェアの配布.....	611
14.2.2	ソフトウェアのアクティベーションとライセンスの計測.....	611
14.2.3	Authentic のための Altova エンドユーザー使用許諾契約書.....	612
	インデックス	613

1 Authentic Desktop とこのドキュメントについて

XMLドキュメントを認証するための画期的かつ視覚的なアプローチである **Altova Authentic 2021 Desktop** を使用するとエンドユーザーをXMLの技術的な側面に煩わすことなく作業することができます。Authentic Desktop はプラットフォーム更新済みのWindows 7 SP1、Windows 8、Windows 10 とプラットフォーム更新済みのWindows Server 2008 R2 SP1 まで以降で動作します。Authentic Desktop Enterprise Edition は64ビット版と32ビット版がご利用になります。



最終更新日: 2021年02月24日

1.1 Windows ファイルパス

Windows 7、Windows 8 および Windows 10 でのファイルパス

このドキュメントのファイルパスはすべてのオペレーティングシステムで同じではありません。以下の通信に注意してください。

- (マイ)ドキュメント フォルダー: デフォルトでは以下の場所に存在します。Example ファイルはこのフォルダのサブフォルダに存在します。

Windows 7/8/10	C:\Users\ <username>\Documents</username>
----------------	---

- アプリケーション フォルダー: アプリケーション フォルダーには Altova アプリケーションが存在します。デフォルトでの アプリケーション フォルダーのパスは以下の通りです。

Windows 7/8/10	C:\Program Files\Altova\
64-bit OS 上での 32 bit バージョン	C:\Program Files (x86)\Altova\

メモ Authentic Desktop はプラットフォーム更新済みの Windows Server 2008 R2 SP1 または以降 上でもサポートされています。

1.2 このドキュメントについて

このユーザーマニュアルには多種の Authentic View 機能についてのチュートリアルと説明が含まれています。また、インターフェイスの様々な機能についての説明が行われます。使用方法のセクションは以下のようになります:

- [イントロ](#) は GUI と Authentic Desktop 環境に説明しています。
- [チュートリアル](#) は Authentic Desktop を使用して開始するための説明をします。
- XML ドキュメントの WYSIWYG ビューである [Authentic View](#) の詳細な説明。Authentic View によりユーザーは XML ドキュメントを単純なテキストドキュメントまたはインタラクティブな書式のように書き込み、編集することができます。XML マークアップはユーザーから隠されており、ユーザーがドキュメントの内容に集中することができます。Authentic View は Authentic Desktop の主要なビューです。
- XML ドキュメントが高速に変換されブラウザビューで表示される [ブラウザビュー](#) の説明。
- リソースを素早く切り替えることのできる Altova の [グローバルリソース](#) 機能の説明。
- [Visual Studio](#) と [Eclipse](#) 内で使用するための Authentic Desktop の説明
- Authentic Desktop 内で使用することのできる全てのウィンドウとメニューコマンドの説明を含む [メニューコマンドレファレンス](#)。

2 インターフェイスと環境

このセクションでは以下の説明を行います:

- [アプリケーション GUI](#)
- [アプリケーション環境](#)

[GUI のセクション](#) では、最初に GUI の概要を説明した後、各 GUI ウィンドウの詳細について解説していきます。ウィンドウならびに GUI のサイズ変更、移動、そして操作方法などについても説明します。

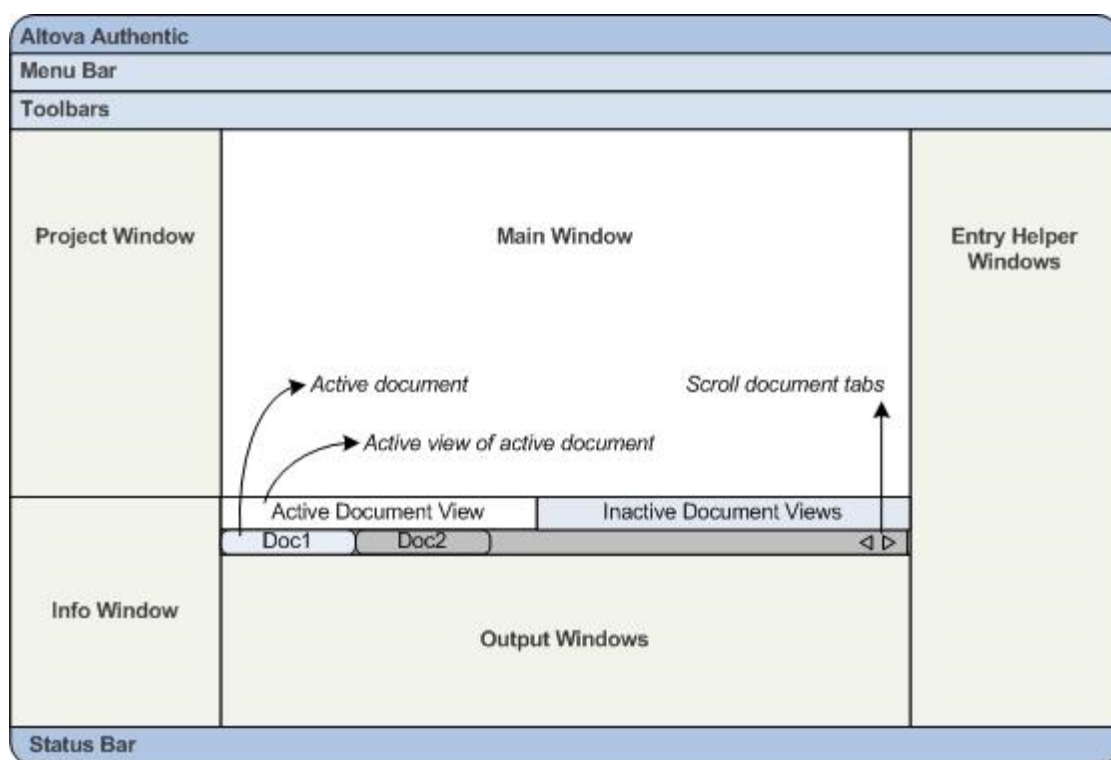
[アプリケーション環境のセクション](#) では、ファイルの管理を行うために必要な、様々な設定の表示ならびに変更方法について説明します。アプリケーションのカスタマイズを行う場所と、その方法についても記述されます。このセクションでは、あなたのコンピュータ内にインストールされている、重要なサンプルファイルやチュートリアルファイルの場所についても言及されています。セクションの最後には [Altova Web サイト](#) へのリンクが掲載されており、アプリケーションが持つ機能の一覧や、様々な形式のユーザーマニュアル、利用可能なサポートオプション、そして Altova が提供するその他の製品に関する情報を参照することができます。

2.1 グラフィカルユーザインターフェース (GUI)

グラフィカルユーザインターフェース(GUI)は、メインウィンドウと、幾つかのサイドバーにより構成されています(以下の図を参照)。デフォルトでは、サイドバーはメインウィンドウの周りに配置されており、以下の様にグループ分けされています:

- プロジェクトウィンドウ
- 情報 ウィンドウ
- 入力ヘルパ: 要素、属性、エンティティなど(現在アクティブなドキュメントの種類により変化)
- 出力ウィンドウ: メッセージ、

このセクションのサブセクションではメインウィンドウならびにサイドバーについて説明します。



GUIにはメニューバーやステータスバー、ツールバーなども含まれており、このセクションにて全て説明されます。

サイドバー表示の切り替え

「ウィンドウ」メニューのコマンドを使用することで、サイドバーグループのウィンドウ(プロジェクトウィンドウ、Info ウィンドウ、入力ヘルパ、出力ウィンドウ)の表示を切り替えることができます。表示されているサイドバー(またはタブによりグループ化されたサイドバー)のタイトルバーを右クリックして、「隠す」コマンドを選択することで、サイドバーを非表示にすることができます。

サイドバーのフロートドッキング

各サイドバーウィンドウは、GUIから自由な状態のフロート、またはGUIに結合したドッキング状態にセットすることができます。フロート状態のウィンドウがドッキング状態になると、最後にドッキングしていた場所ドッキングすることになります。ウィンドウがタブとして別のウィンドウドッキングすることもできます。

以下の方法を使うことで、ウィンドウをフロートまたはドッキング状態にすることができます:

- ウィンドウのタイトルバーを右クリックして、必要なコマンド(フロートまたはドッキング)を選択する。
- ウィンドウのタイトルバーをダブルクリックする。ウィンドウがドッキングしている状態であれば、フロート状態になり、フロート状態であれば、最後にドッキングしていた位置にドッキングします。
- (タイトルバーを使って)ドッキング状態のウィンドウをドラッグすることで、フロート状態に切り替える。フロート状態のウィンドウをドラッグすることで、目的の位置にドッキングすることもできます。2つのセットの矢印が表示されます。外側に表示される4つの矢印により、アプリケーションウィンドウに対して相対的なドッキングを行うことができます。内側に表示される矢印を使うことで、現在カーソルが位置しているウィンドウに対して相対的なドッキングを行うことができます。内側に表示されている矢印の中央でボタンをドロップすると、ドラッグしていたウィンドウが、ドロップされたウィンドウのタブとしてドッキングされます。

タブ化されたウィンドウをフロート状態にするには、そのタブをダブルクリックしてください。タブをドラッグすることで、タブ化されたウィンドウをグループから切り離すことができます。

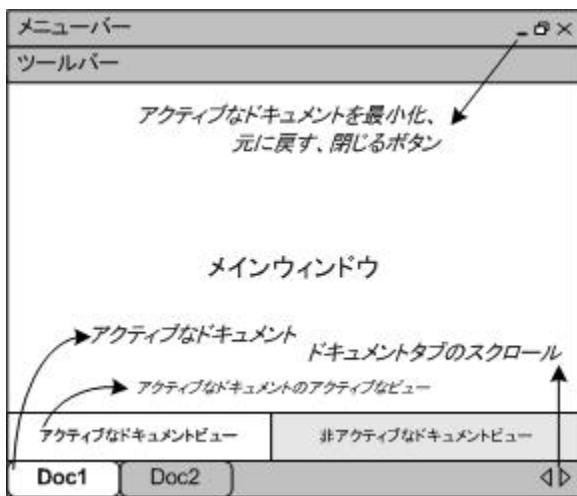
サイドバーを自動的に隠す

自動的に隠す機能により、ドッキング状態のサイドバーがアプリケーションウィンドウの端に寄せられ、サイズを最小限に抑えることができます。この機能により、メインウィンドウやその他のサイドバーを、より大きな空間で使用できるようになります。最小化されたサイドバーにマウスを移動させることで、サイドバーが再度表示されます。

サイドバーを自動的に隠し、その設定を解除するには、サイドバーウィンドウのタイトルバーにある自動的に隠すアイコンをクリックするか、タイトルバーを右クリックして「自動的に隠す」を選択します。

2.1.1 メイン ウィンドウ

(以下の図に表示される)メインウィンドウでドキュメントの編集を行います。



メインウィンドウのファイル

- 開いた編集を行うファイルの数に制限はありません。
- 開かれた各ドキュメントは、各自ウィンドウに表示され、その名前がメインウィンドウの下部にあるタブに表示されます。ドキュメントをアクティブにするには、そのドキュメントのタブをクリックしてください。
- 複数のファイルが開かれている場合、ドキュメントタブバーに収まりきらず、一部のドキュメントタブが表示されないことがあります。目的のドキュメントタブは、(i) ドキュメントタブバーの右にあるスクロールボタンを使うことで、または(ii) [ウィンドウ](#)メニューから目的のドキュメントを選択することでアクティブにすることができます。

- アクティブなドキュメントが最大化されると、「最小化」「元に戻す」「閉じる」ボタンがメニューバーの右側に配置されます。ドキュメントが重ねて表示、または最小化されると、「最大化」「元に戻す」「閉じる」ボタンがドキュメントウィンドウのタイトルバーに配置されます。
- 1つのファイルを最大化すると、開かれている他のファイルも全て最大化されます
- **ウィンドウ** メニューのコマンドを使用することで、開かれているドキュメントを重ねて、または上下(左右)に並べて表示することができます。
- 開かれているファイルは、「Ctrl+Tab」または「Ctrl+F6」により、開かれた順序でアクティブに切り替えることができます。
- ドキュメントタブを右クリックすることで表示されるコンテキストメニューから、「印刷」や「閉じる」といったコマンドを実行することができます。

メインウィンドウ内のビュー

アクティブなドキュメントを表示、編集する方法は可通があります。使用することのできるビューは、(上の図にあるように)ドキュメントタブ上のバーに表示されており、アクティブなビューがハイライトされます。目的のビューボタンをクリック、または **表示** メニューのコマンドを使用することでビューの切り替えを行うことができます。

利用可能なビューは編集またはブラウザビューです:

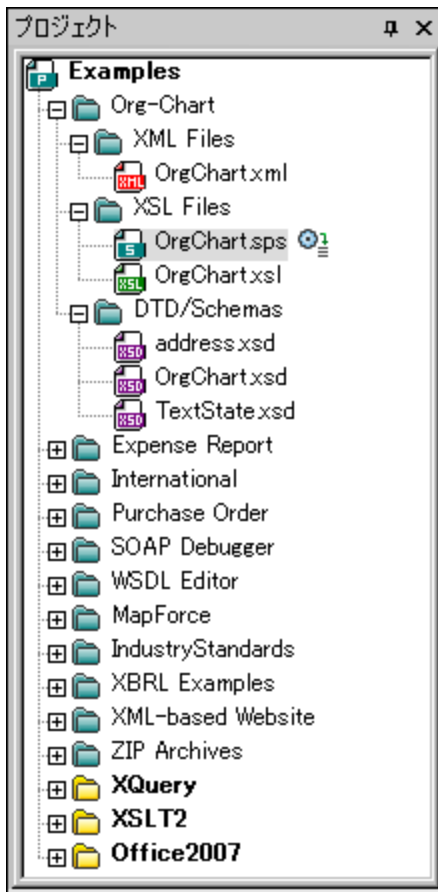
- **Authentic ビュー**: StyleVision Power Stylesheet をベースとしたXMLドキュメントの編集をグラフィカルなインターフェイスで行ないます。
- **ブラウザビュー**: CSS ならびにXSLスタイルシートをサポートする統合されたブラウザビューです。

メモ 各ファイル拡張子のデフォルトビューは、「**ツール|オプション**」ダイアログの、ファイルタブにあるデフォルトビューペインにてカスタマイズすることができます。

2.1.2 プロジェクト ウィンドウ

プロジェクトとは、何らかの形で関係するファイル同士を集めた集合のことです。例えば、以下のスクリーンショットでは、Examples という名前のプロジェクトに、様々なサンプルファイルが、適当なフォルダーに収められており、各フォルダーにはサブフォルダーを作成することもできます。例えば、Examples プロジェクト内のOrgChart サンプルフォルダーにはXML、XSL、そしてスキーマファイルを管理するためのサブフォルダーがあります。

メモ サンプルプロジェクトはデフォルトでプロジェクトウィンドウ内で開かれます。開く必要がある場合、アプリケーションの **ドキュメントフォルダー** 内の Examples フォルダーに移動してファイル Examples.spp をダブルクリックします。



プロジェクトを使用することで、一緒に使用されるファイルをまとめて、ファイルへのアクセスを素早く行うことができます。更に、各フォルダーに対してスキーマやXSLT ファイルを定義することもでき、フォルダーに対するバッチ処理を行うこともできます。

プロジェクトの操作

フォルダーの操作を行うコマンドは、プロジェクトメニュー、またはプロジェクトやフォルダーの(右クリックにより表示される)コンテキストメニューからアクセスすることができます。

- プロジェクトウィンドウで同時に開くことのできるプロジェクトは1つだけです。新規プロジェクトが作成、または既存のプロジェクトが開かれると、現在プロジェクトウィンドウにて開かれているプロジェクトは置き換えられます。
- プロジェクトに変更がなされた場合、そのプロジェクトを保存する必要があります。プロジェクトの保存は「プロジェクト | プロジェクトを保存」メニューコマンドにより行うことができます。保存されていない変更を含むプロジェクトは名前の横にアスタリスクが表示されています(上のスクリーンショットを参照)。
- プロジェクトはフォルダー、ファイル、その他のリソースなどにより構成されるツリー構造により表示されます。リソースを追加する場所や階層に制限はありません。
- プロジェクトフォルダーは、論理的にファイルのグループ化を行う論理フォルダーです。ハードディスク上にある階層構造などに対応する必要はありません。
- フォルダーは、実際のファイルシステムに対して、直接対応するように作成することもできます。このようなフォルダーは外部フォルダーと呼ばれ、プロジェクトウィンドウでは黄色のフォルダーアイコンにより表示されます(通常のプロジェクトフォルダーは緑色のフォルダーアイコンで表示されます)。外部フォルダーは、更新コマンドにより、明示的に同期を行う必要があります。
- フォルダー内に収められるファイルの種類やその組み合わせの制限はありません。また、フォルダーのプロパティダイアログから各フォルダーにファイルの拡張子を定義することで、共通するファイルを1つの場所に収めることができます。ファイルの親フォルダーに追加されると、そのファイル拡張子が定義されたサブフォルダーに、ファイルが自動的に追加されます。

- プロジェクトウィンドウでは、フォルダーを他のフォルダー、または、同じフォルダー内の他のロケーションにドラッグすることができます。ファイルは他のフォルダーにドラッグすることができますが、アルファベット順にファイルが並べられている同じフォルダー内では移動することができません。また、Windows エクスプローラーなどからファイルやフォルダーをプロジェクトウィンドウにドラッグすることで、プロジェクトへの追加を行うこともできます。
- 各フォルダーには、フォルダーのプロパティダイアログにて定義されるプロパティがあります。これらのプロパティには、フォルダーのファイル拡張子、XML ファイルの検証を行うスキーマファイル、XML ファイルの変換を行う XSLT ファイルなどが記述されます。
- フォルダー内にあるファイルのバッチ処理を行うには、フォルダーを右クリックし、コンテキストメニューから選択することのできるコマンドを選択します(例: XML の検証または整形形式のチェック)。

メモ プロジェクトウィンドウの表示はウィンドウメニューから切り替えることができます。

2.1.3 情報ウィンドウ

情報ウィンドウには、現在カーソルが位置している要素や属性に関する情報が表示されます(以下のスクリーンショットを参照)。情報ウィンドウを確認することができます。



情報ウィンドウの表示はウィンドウメニューから切り替えることができます。

- XULE ドキュメントがアクティブなドキュメントの場合、情報ウィンドウの XULE タブが XBRL インスタンスの構造に関する情報のために使用する XBRL タクノミを選択するオプションを提供します。タクノミ情報は XULE エディターの自動補完機能のために使用されます。

2.1.4 入力ヘルパー

入力ヘルパーは、妥当な XML ドキュメントを素早く作成するための、インテリジェントな編集機能です。ドキュメントの編集時、現在カーソルがある位置にて利用可能なオプションが入力ヘルパーに表示されます。入力ヘルパーに表示される情報は、DTD、XML スキーマ、およびまたは StyleVision Power Stylesheet から取得されます。例えば、XML データドキュメントの編集を行っている場合、現在カーソルのある位置で挿入可能な要素、属性、エンティティが、それぞれの入力ヘルパーにて表示されます。

以下の点に留意してください:

- 入力ヘルパーの表示は、メニューオプションの「ウィンドウ | 入力ヘルパー」から切り替えることができます。

2.1.5 メニューバー、ツールバー、ステータスバー

メニューバー

メニューバー ([図を参照](#)) には、様々なアプリケーションメニューが収められています。以下の慣習が適用されます:

- メニュー内のコマンドが、その時のビューや、カーソルの位置に対して適用できない場合、コマンドは無効になります。
- いくつかのメニューコマンドはサブメニューを持っており、更に別のオプションを選択することができます。サブメニューがあるメニューコマンドは、コマンド名の右側に表示される右方向矢印により示されます。
- いくつかのメニューコマンドはサブメニューを持っており、更に別のオプションを選択することができます。サブメニューがあるメニューコマンドは、コマンド名の右側に表示される右方向矢印により示されます。
- メニューコマンドにアクセスするには、メニュー名をクリックした後、コマンドを選択します。メニューアイテムにサブメニューがある場合、そのメニューアイテムにマウスを移動することで、サブメニューが開かれます。その後、目的のサブメニューアイテムをクリックします。
- 適切なキーのコンビネーションにより、キーボードからメニューを開くこともできます。各メニューへのキーコンビネーションは **Alt+KEY** で、**KEY** がメニュー名のキーになります。例えば、「ファイル(F)」メニューへのアクセスを行うには、**Alt+F** を押下します。
- (i) メニューに対するキーの組み合わせ(前の項目を参照)の後に (ii) 特定のコマンドに対するキーの組み合わせを入力することで、(メニュー内に収められているコマンドの)メニューコマンドへアクセスすることができます。例えば、新規ファイルを作成(「ファイル(F) | 新規作成(N)」)するには **Alt+F** を押下した後、**Alt+N** を押下します。
- 特定のメニューコマンドは、ショートカットキーまたは、キーのコンビネーション(**Ctrl+Key**)により直接選択することができます。ショートカットが割り当てられたコマンドは、コマンド名の右側にショートカットキーや、キーのコンビネーションが示されます。例えば、ショートカットキーのコンビネーションである **Ctrl+N** を使用することで、ファイルの新規作成を行うことができ、**F8** によりXMLファイルの検証を行うことができます。カスタマイズダイアログ(「ツール | カスタマイズ」)にあるキーボードタブにて、**独自のショートカットを作成** することもできます。

ツールバー

メニューバー ([図を参照](#)) には、メニューコマンドを選択するためのショートカットとなるアイコンが収められています。マウスポインターをアイコン上に移動させると、コマンドの名前が表示されます。アイコンをクリックすると、コマンドが実行されます。

ツールバーボタンはグループでまとめられています。「[ツール | カスタマイズ | ツールバー](#)」ダイアログでは、表示するツールバーのグループを選択することができます。これらの設定は現在のビューに対して適用されます。他のビューの設定を変更するには、そのビューに切り替えした後、「[ツール | カスタマイズ | ツールバー](#)」にて設定を変更します。GUI では、ツールバーグループのハンドル(またはタイトルバー)を別の位置にドラッグすることで、ツールバーグループの位置を変更することができます。ハンドルをダブルクリックすると、ツールバーがドッキングからフロート状態になり、タイトルバーをダブルクリックすることで、元の場所にドッキングされます。

ステータスバー

ステータスバーはアプリケーションウィンドウの下部に位置 ([図を参照](#)) しており (i) ファイルのロードに関する情報や、(ii) マウスポインターがメニューコマンドや、コマンドショートカットにある時に、それらコマンドに関する情報を表示します。64ビット版のXMLSpyを使用している場合、Authentic Desktop の後に (x64) という表記が加わります。この表記は32ビットバージョンにおいてはなされません。

2.2 アプリケーション環境

このセクションでは、アプリケーションの使用を始めるにあたり重要となる情報を記述します。このセクションを読み進めることで、Authentic Desktop の操作に慣れ親しみ、自信を持ってアプリケーションの使用を始めることができるようになります。設定やカスタマイズに関する重要な情報も含まれており、設定やカスタマイズのオプションにより設定可能な大まかな範囲や、それらの変更方法などについて知ることができます。

このセクションは以下のように構成されます：

- [設定、カスタマイズ](#) 重要な設定やカスタマイズを行う場所と、その変更方法について説明します。
- [チュートリアル、プロジェクト、サンプル](#) アプリケーションパッケージに収められている、様々な非プログラムファイルの場所と、その注釈について記述します。
- [製品機能とドキュメンテーション](#)、[Altova 製品](#)：製品の機能や、その他の形式で記述されたヘルプ、その他の Altova 製品に関する情報が記述された [Altova Web サイト](#)へのリンクが掲載されます。

2.2.1 設定、カスタマイズ

このセクションは、以下のように構成されます：

- [設定](#)
- [カスタマイズ](#)

設定

オプションダイアログのタブには、Authentic Desktop が持つ重要な設定が定義されています。利用可能なオプションがどのようなものか確認してください。

カスタマイズ

ユーザーは GUI の外観など、Authentic Desktop の様々な側面をカスタマイズすることができます。これらのカスタマイズオプションは、カスタマイズダイアログからアクセスすることができます（メニューコマンド「[ツール | カスタマイズ](#)」からアクセスすることができます）。

カスタマイズコマンドに関する詳細は [ユーザーヘルプ](#) のセクションを参照ください。

2.2.2 チュートリアル、プロジェクト、サンプル

Authentic Desktop インストールパッケージには、チュートリアル、プロジェクト、そしてサンプルファイルが収められています。

チュートリアル、プロジェクト、サンプルファイルの場所

Authentic Desktop チュートリアル、プロジェクト、そしてサンプルファイルは以下のフォルダーにインストールされています：

```
C:\Documents and Settings\\My Documents\  
Altova\Authentic2021\AuthenticExamples\
```

The My Documents\Altova\Authentic2021 フォルダは PC 上で登録されたユーザーごとに、ユーザーの<username> フォルダ以下にインストールされます。従って、このインストールシステムでは、各ユーザーが、それぞれの場所に AuthenticExamples フォルダを持つこととなります。

マスター Authentic Desktop フォルダに関する注意

Authentic Desktop がインストールされると、以下の場所に、マスター Altova\Authentic2021 フォルダが次の場所で作成されます。

```
C:\Documents and Settings\All Users\Application Data\
```

このコンピュータ上でユーザーが Authentic Desktop を初めて起動すると、Authentic Desktop は、マスターフォルダのコピーを、ユーザーの<username>\My Documents フォルダに作成します。Authentic Desktop の他のユーザーが次回使用する場合、編集されたファイルがユーザーフォルダにコピーされるため、チュートリアルやサンプルファイルを使用する際には、このマスターフォルダを使用しないようご注意ください。

チュートリアル、プロジェクト、サンプルファイルの場所

チュートリアル、プロジェクト、サンプルファイルは、全て、AuthenticExamples フォルダに収められています。

2.2.3 Authentic Desktop 製品機能と Altova 製品

Altova Web サイト <http://www.altova.com/jp/> には、以下にあるように Authentic Desktop-に関連した様々な情報やリソースが含まれています。

Authentic Desktop 機能の一覧

Altova Web サイトには [Authentic Desktop に含まれている最新機能の一覧](#)などが掲載されています。

Authentic Desktop ヘルプ

このドキュメンテーションは、Altova により作成された Authentic Desktop のヘルプです。ドキュメンテーションは、Authentic Desktop 内に収められており、「ヘルプ」メニューや F1 によりアクセスすることができます。更に、Altova 製品のマニュアルは以下の形式でも入手することができます。

- [オンライン HTML マニュアル](#) Altova Web サイトのサポートページからアクセスすることができます。
- [印刷可能な PDF](#) Altova Web サイトからダウンロードして、印刷することができます。
- [印刷物](#) Altova Web サイト上にあるリンクからお求め頂けます。

サポートオプション

ユーザーマニュアル(本ドキュメンテーション)には記載されていない情報や、Altova 製品に関するお問い合わせは、Altova Web サイトの [サポートセンター](#) をご利用ください。以下のような機能がご利用いただけます。

- [よくある質問と回答ページ](#)へのリンク
- Altova 製品や、XML 全般に関する [ディスカッションフォーラム](#)。
- [オンラインサポートフォーム](#) : サポートパッケージをお持ちである場合、サポートリクエストを送信することができます。サポートリクエストは、サポートチームにより処理されます。

Altova 製品

Altova 製品のラインナップについては、[Altova Web サイト](#)を参照ください。

3 Authentic View チュートリアル

Authentic View 内の視覚的な WYSIWYG インターフェイス内で Microsoft Word などのワードプロセッサアプリケーション内で作業するように XML ドキュメントを編集することができます (下のスクリーンショット)。XML ドキュメントの Authentic View を管理するスタイルシート内で既に定義されているため、ドキュメントの書式設定を心配する必要はありません (StyleVision Power Stylesheet はこのチュートリアル内では SPS と省略されています)。スタイルシートは Altova の StyleVision 製品のスタイルシートデザイナーを使用して作成されます。

Nanonull, Inc.	
Location: <input type="text" value="US"/>	
Street:	119 Oakstreet, Suite 4876
City:	Vereno
State & Zip:	<input type="text" value="DC"/> <input type="text" value="29213"/>
Phone:	+1 (321) 555 5155 0
Fax:	+1 (321) 555 5155 4
E-mail:	office@nanonull.com

Vereno Office Summary: 4 departments, 15 employees.

The company was established in Vereno in 1995 as a privately held software company. Since 1996, Nanonull has been actively involved in developing nanoelectronic software technologies. It released the first version of its acclaimed *NanoSoft Development Suite* in February 1999. Also in 1999, Nanonull increased its capital base with investment from a consortium of private investment firms. The company has been expanding rapidly ever since.

Authentic View 内で XML ドキュメントを編集するごとのユーザーアクションが含まれます: (i) ドキュメントの構造の編集 (例えば、段落とヘッダーなどのドキュメントの一部を追加、または、削除するなど)、および (ii) データの挿入 (ドキュメントの一部のエントリ)。

このチュートリアルでは次のステップについて説明されています:

- Authentic View 内で XML ドキュメントを開く。Authentic View 編集の主要な必要条件は XML ドキュメントが SPS ファイルに関連付けられていることです。
- Authentic View インターフェイスの主要な編集メカニズムに関する概要を確認してください。
- ノードを挿入または削除することによってドキュメントの構造を編集します。
- XML ドキュメント内にデータ挿入します。
- (i) 属性入力ヘルパーを介して属性の値と (ii) エンティティの値を入力します。
- ドキュメントを印刷します。

このチュートリアルは使用開始の準備をするためであり、簡単にまとめられています。Authentic View インターフェイスセクション内で追加の参照マテリアルと機能の説明を確認することができます。

チュートリアルの必要条件

チュートリアルに必要な全てのファイルは Altova アプリケーションの `AuthenticExamples` フォルダ内にあります。これらのファイルは以下のとおりです:

- `NanonullOrg.xml` (XML ドキュメント)
- `NanonullOrg.sps` (XML ドキュメントがリンクされている StyleVision Power Stylesheet)
- `NanonullOrg.xsd` (XML ドキュメントと StyleVision Power Stylesheet がベースされ、リンクされている XML スキーマ)
- `nanonull.gif` and `Altova_right_300.gif` (チュートリアルで使用されるイメージ)

メモ このチュートリアルのある時点で、(ドキュメントの Authentic View とは異なり) XML ドキュメントの XML テキストを確認するように促されます。(Authentic Desktop と Authentic Browser 同様) Altova 製品エディションにテキストビューが搭載されていない場合、Wordpad または Notepad のようなテキストエディターを使用して XML ドキュメントのテキストを確認してください。

注意点: 問題が発生した際に元のファイルにアクセスできるように、`NanonullOrg.xml` をチュートリアルのためコピーすることが奨励されます。

3.1 AUTH-VIEW 内で XML ドキュメントを開く

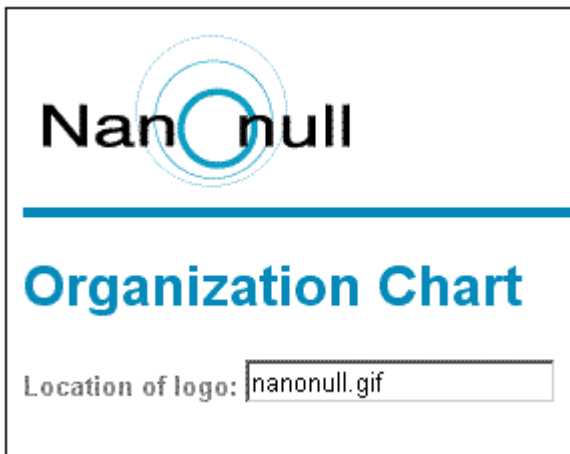
Authentic View 内で、既存のXMLドキュメントを編集、または、新規のXMLドキュメントを作成して編集することができます。(このセクションで説明されているとおり)このチュートリアルでは、Authentic View 内で既存のXMLドキュメントを開き、(次のセクションで)編集の方法を学びます。更に、Authentic View 内での編集のために新規XMLドキュメント作成の方法について説明されています。

既存のXMLドキュメントを開く

NanonullOrg.xml というファイルを開きます。Altova アプリケーションのAuthenticExamples フォルダ内にあり、以下の2つの方法によりNanonullOrg.xml を開くことができます:

- Altova 製品内の「ファイル | 開く」をクリックし、表示されるダイアログ内からNanonullOrg.xml を選択し、「開く」をクリックします。
- ウィンドウExplorer を開き、ファイルを検索し、右クリックして、ファイルを開くためのアプリケーションとして Altova 製品を選択します。

Authentic View 内でファイルNanonullOrg.xml を直接開くことができます(下のスクリーンショット)。



留意点: SPS はAuthentic View 内でXMLドキュメントがどのように表示されるかを表示し管理します。SPS が存在しない場合、ドキュメントのAuthentic View は存在しません。

SPS をベースとした新規のXMLドキュメント

SPS をベースとした新規のXMLドキュメントを作成することができます。これを2つの方法で行うことができます: 「ファイル | 新規作成」メニューコマンドと「Authentic | 新規のドキュメント」メニューコマンドを使用する。SPS が両方の方法で選択されます。

「ファイル | 新規作成」を使用する方法

1. 「ファイル | 新規作成」を選択します。
2. 新規のドキュメントの作成ダイアログ内で、希望するSPS を参照します。

テンプレート XML ファイルがSPS に割り当てられている場合、Authentic View 内で作成されたドキュメントテンプレートの開始データとしてテンプレート XML ファイル内のデータが使用されます。

「Authentic | 新規のドキュメント」を使用する方法

1. 「Authentic | 新規のドキュメント」を選択します。
2. 「新規のドキュメントの作成」ダイアログ内で希望する SPS を参照します。

テンプレート XML ファイルが SPS に割り当てられている場合、テンプレート XML ファイル内のデータは Authentic View. 内で作成される XML ドキュメントテンプレートの開始データとして使用されます。


3.2 Authentic View インターフェイス

Authentic View 編集インターフェイスはドキュメントデータを入力し編集するメインウィンドウと2つの入力ヘルパーにより構成されています。ドキュメントの編集は簡単です。ドキュメントのマークアップを確認する場合、マークアップタグをオン/オフ切り替え、ドキュメントのエディットの入力を開始します。ドキュメントの構造を変更するには、コンテキストメニュー、または要素入力ヘルパーを使用します。

XML ノードタグ(ドキュメントマークアップ)の表示

XMLドキュメントは、基本的には、ノードの階層構造です。例

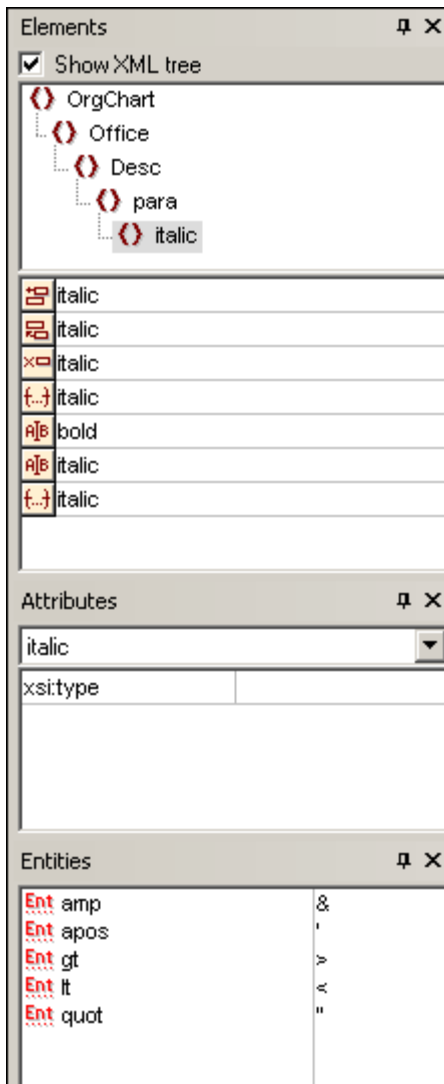
```
<ドキュメントRoot>
  <Person id="ABC001">
    <Name>Alpha Beta</Name>
    <Address>Some Address</Address>
    <Tel>1234567</Tel>
  </Person>
</ドキュメントRoot >
```

デフォルトでは、ノードタグ Authentic View 内では表示されません。メニューアイテム「Authentic | 大きなマークアップを表示する」(または  ツールバーアイコン)を選択してノードタグをオン/オフ切り替えることができます。大きなマークアップタグはそれぞれのノードの名前が含まれています。更に(タグ内にノード名の無い)小さなマークアップと複合型マークアップを選択することができます(スタイルシートデザイナーにより大きい、小さい、または、マークアップタグの無い複合型マークアップが定義されます。ドキュメントのためのデフォルトの複合型マークアップはマークアップが存在しません)。

Altova 製品 またはテキストエディターのテキストビュー内でXMLドキュメントのテキストを確認することができます。

入力ヘルパー

インターフェイス内には、デフォルトでアプリケーションウィンドウの右端にある3つの入力ヘルパーが存在します(下のスクリーンショット)。これらは、要素、属性、および、エンティティ入力ヘルパーです。



要素入力ヘルパー: 要素入力ヘルパーは、カーソルの現在の場所、または、メインウィンドウ内の選択された場所に挿入、または、から削除することができる要素を表示します。入力ヘルパーは状況依存であることにご注意してください。コンテキストはカーソルの位置、または、選択範囲により異なります。入力ヘルパーのエンティティは以下の方法で変更することができます。要素入力ヘルパーのXML ツリー内で他のノードが選択された場合、そのノードに関連する要素が入力ヘルパー内で表示されます。入力ヘルパーの上のXML ツリーの表示チェックボックスをチェックして、XML ツリーを表示するために要素入力ヘルパーを展開することができます（上のスクリーンショットを参照してください）。XML ツリーは、トップレベル要素ノードからメインウィンドウ内で選択されたノードまでのノードの階層構造を表示します。

属性入力ヘルパー: メインウィンドウ内で選択されている要素の属性、および、属性の値を表示します。属性の値は、属性入力ヘルパー内に入力し編集することができます。属性入力ヘルパーのコンボボックス内の選択のためにトップレベルの要素選択された要素までの要素ノードを使用することができます。

コンボボックスのドロップダウンリストから要素を選択すると、要素の属性が入力ヘルパー内に表示されます。入力ヘルパー内で要素の属性を編集することができます。

エンティティの入力ヘルパー: エンティティの入力ヘルパーは状況依存ではなく、ドキュメントのために選択された全てのエンティティを表示します。エンティティをダブルクリックすると、カーソルの位置にエンティティを入力することができます。ドキュメントへのエンティティの追加方法はセクション [Authentic View インターフェイス](#) で説明されています。

コンテキストメニュー

Authentic View ドキュメント内を右クリックすると、その場所（ノード）に関連したコンテキストメニューがポップアップされます。コンテキストメニューは以下を行うことのできるコマンドを提供します:

- 選択されたノードの前、または、あとノードを挿入します。サブメニューはそれぞれの挿入場所で使用を許可されたノードのリストを表示します。
- (スキーマで許可されている場合) 選択されたノード、または、削除することのできる祖先要素を削除します。(スキーマに従い) 削除される可能性のあるノードはサブメニューにリストされています。
- エンティティとCDATA セクションを挿入します。ドキュメントのために宣言されているエンティティはサブメニュー内にリストされています。CDATA セクションはテキスト内でのみ挿入することができます。
- ドキュメントのコンテンツの切り取り、コピー、(XML、または、テキストとしての貼り付けを含む) 貼り付け、および削除。

メモ インターフェイスに関する詳細は、次を参照してください: [Authentic View インターフェイス](#)

3.3 ノードを操作する

Authentic View XMLドキュメント内で発生するノードには2つの主要な種類があります: 要素ノードと属性ノード。これらのノードは[オンラインで切り替えることのできるタグ](#)によりマークアップされます。(マークアップされていない)テキストノードと周りのテキストから区別するためにマークアップされているCDATAセグメントノードなどドキュメント内に他のノードも存在します。

このセクションで説明されるノードの操作は要素ノードと属性ノードのみを参照します。このセクション内で説明されるオペレーションを試す場合、[大きなマークアップがオンラインで切り替えられていることが勧められます](#)。




メモ 同じレベル、または、**ハイレベルの要素のみが**選択された要素の前、または、後に挿入されることにご注意してください。同じレベルの要素は兄弟です。段落要素の兄弟は、他の段落要素ですが、リスト、テーブル、イメージであることができます。兄弟は要素の前、または、後に発生することができます。ハイレベルの要素は祖先要素と祖先の兄弟です。段落要素に関しては、祖先要素はセグメント、チャプター、アーティクルなどである場合があります。有効なXMLファイル内の段落には既に祖先が存在します。結果として、Authentic View内に高いレベルの要素を追加すると、関連する祖先の兄弟として新規の要素が作成されます。例えば、セグメント要素が段落の後に挿入されると、現在の段落要素を含むセグメントの兄弟として作成されます。

ノードの操作の実行

ノードの操作は、コンテキストメニュー内でコマンドを選択、または、[要素入力ヘルパー](#)内でノード操作エントリをクリックすることで行うことができます。一部の場合、要素、または、属性はドキュメントのAuthentic View内の[ノードリンクの追加](#)をクリックして追加することができます。段落またはリストアイテムとして定義されている要素の特別な場合は、このような要素内を作成する際、[Enterキー](#)を押すと、この種類の新規の兄弟要素が作成されます。このセクションは[要素の適用](#)、[ノードの削除](#)、および[要素をクリアする](#)メカニズムを使用して、ノードを作成、または、削除する方法について説明されています。

要素の挿入

要素は次の場所に挿入することができます:

- 要素ノード内のカーソルの場所に挿入することができます。コンテキストメニューの「挿入」コマンドのサブメニュー内にこの場所に挿入することのできる要素がリストされます。要素入力ヘルパー内では、その場所に挿入することのできる要素は  アイコンにより表示されます。NanonullOrg.xmlドキュメント内では、para要素内にカーソルを置き、コンテキストメニューと要素入力ヘルパーを使用して bold と italic 要素を作成します。
- スキーマにより許可されている場合、選択された要素、または、その祖先の前、または、後に挿入することができます。必要とされる要素をロールアウトされるサブメニューから選択します。要素入力ヘルパー内では、選択された要素の前、または、後に挿入することのできる要素は  および  アイコンにより表示されます。要素入力ヘルパー内では、選択された要素の前後のみに要素を挿入することができます。祖先要素の前後に要素を挿入することはできません。para要素と社員をリストするテーブル内にカーソルを置いてこのコマンドを試してください。

ノードリンクの追加

要素、または、属性がドキュメントデザイン内に含まれ、XMLドキュメント内で存在しない場合、ノードが指定されているドキュメント内の場所でノードリンクの追加が表示されます。このリンクを確認するために、テキスト、ロゴの場所を持つライン内で、CompanyLogo要素内で @href ノードを選択し、「削除」キーを押して削除します。編集されたCompanyLogo要素内で add @href リンクが表示されます(下のスクリーンショット)。リンクをクリックすると、@href ノードがXMLドキュメントに追加されます。@href ノードがこのように表示されるようにデザインを指定しているため、@href タグ内のテキストボックスは、はこのような表示されます。@href ノード(または、コンテンツ)の値を入力する必要があるため、テキスト nanonull.gif を入力します。



要素のコンテンツモデルが曖昧な場合、例えば、子要素のシーケンスを指定する場合、順序が決まりはありませんので `add...` が表示されます。ノード名が指定されていないことにご注意ください。リンクをクリックすると、有効に挿入することのできる要素のリストが表示されます。

メモ ノードリンクの追加がキムメントテンプレート内に直接表示されます。コンテキストメニュー、または、要素入力ヘルパーに対応するエントリはありません。

Enter キーを使用して新規要素を作成する

要素がスタイルシート デザイナーにより段落、または、リストアイテムとして書式設定されている場合、このようなノード内で「Enter」キーを押すと、点在のノードの後、このような種類の新規のノードが挿入されます。NanonullOrg.xml ドキュメント内でこのメカニズムを試すことができます。(終了タグの前の) `para` ノードの終わりに移動し、「Enter」を押すことで実行することができます。

要素の適用

(テキストと子要素の両方を含む) 複合型 コンテンツの要素内では、テキストコンテンツの一部を選択し、許可されている子要素を適用することができます。選択されたテキストは適用された要素のコンテンツになります。要素を適用するには、コンテキストメニュー内で、「適用」を選択し、適用することのできる要素から選択します(選択されたテキストに要素が適用されない場合、「適用」コマンドはコンテキストメニュー内に表示されません)。要素入力ヘルパー内では、選択のために適用することのできる要素は、**RB** アイコンで表示されます。

NanonullOrg.xml ドキュメント内で、複合型 コンテンツ `para` 要素内のテキストを選択し、**bold** と **italic** 要素の適用を試してください。

スタイルシート デザイナーが要素を適用するツールバーアイコンを作成する場合があります。NanonullOrg.xml ドキュメント内では、**bold** と **italic** 要素をアプリケーションの Authentic ツールバー内の太字と斜体アイコンをクリックすることで適用できます。

要素の削除

ドキュメントを無効にしない場合、ノードを削除することができます。ノードの削除はその全てのコンテンツを削除します。コンテキストメニュー内の「削除」コマンドを使用してノードを削除することができます。「削除」コマンドがハイライトされると、選択されたノードからドキュメントのトップレベルのノードまで削除する全てのノードを含むサブメニューが表示されます。削除するためノードを選択するには、カーソルをノード内に置く、または、ノード(または、ノードの一部)をハイライトします。要素入力ヘルパー内では、削除することのできるノードは、**✕** アイコンで表されます。削除可能なノードは選択後に「削除」キーを押すことにより削除することができます。NanonullOrg.xml ドキュメント内で、上記のメカニズムを使用していくつかのノードを削除してください。**Ctrl+Z** を使用して変更を元に戻すことができます。

要素の作成

複合型 コンテンツ(テキストと要素の子)を持つ要素の子である要素ノードをクリアすることができます。ノードが選択されると、または、入力ポイントとしてノードにカーソルがノード内に置かれると、要素の全体をクリアすることができます。要素内のテキストフラグメントはテキストフラグメントをハイライトすることにより要素マークアップをクリアすることができます。選択を決定し、コンテキストメニュー内の「クリア」を選択して、要素をクリアします。要素入力ヘルパー内では、特定の選択のためにクリアする要素は、**f.t** アイコン(挿入ポイント 選択)と **f.t** アイコン(範

囲の選択)により示されます。NanonullOrg.xml ドキュメント内で、(複合型 コンテンツを持つ) para の bold と italic 子要素を使用して、クアメカニズムを試してください。

テーブルとテーブルの構造

Authentic View テーブルには2つの型が存在します:

- *SPS* テーブル(静的と動的)。スタイルシート デザイナーによりSPS テーブルの大きな構造は決定されます。この広義な構造では、許可されている唯一の構造的な変更はコンテンツ優先のみです。例えば、新規の行を動的な SPS テーブルに追加することができます。
- *XML* テーブルは特定のノードのコンテンツを表示します。(個人の特定の情報などを含んだ) テーブルなど。スタイルシート デザイナーがXML テーブルとしてこのノードの作成を有効化している場合、テーブルの構造を決定し、コンテンツを編集することができます。XML テーブルは [Authentic View 内のテーブル](#) セクションで詳細に説明されています。

3.4 Authentic View 内にデータを挿入する

Authentic View メインウィンドウ内のXMLドキュメントに直接データが入力されます。追加して、属性のためにデータ(属性の値を属性入力ヘルパーに挿入)することができます。データは(i) 直接テキストとして入力されます。または(ii) 定義済みのテキストエントリにマップされるデータ入力デバイス内のオプションを選択することにより入力されます。

テキストコンテンツの追加

Authentic View のメインウィンドウ内にテキストとして直接入力することができます。コンテンツを挿入するために、カーソルをテキストを入力する箇所にポイントし、入力を開始します。クリップボードからテキストをコピーして、ドキュメントに貼り付けることもできます。コンテンツは「Caps」と「削除」キーなどの通常の編集機能を使用して編集することができます。例えば、編集するテキストをハイライトして、「Caps」キーを押したまま変換のテキストを入力することができます。

例えば、企業の名前を変更する場合は、Office のName フィールド内でカーソルをNanonull の後にポイントし、Nanonull, Inc. からNanonull USA, Inc. に名前を変更するためにUSAを入力します。



テキストを編集することができる場合、カーソルをポイントしてハイライトします。それ以外の場合、ハイライトすることはできません。アドレスブロック内の「Street」、「City」、または「State/Zip」などのフィールドの値ではなく、フィールド名を変更しようと試みてください。StyleVision Power Stylesheet から生成されたテキストはXMLコンテンツではないため、このテキスト内にはカーソルをポイントすることはできません。

特別文字とエンティティの挿入

データの挿入する場合、次のコンテンツの型は特定の方法で扱われます:

- 特殊文字は、XML マークアップ(アンパサンド、アポストロフ、より大記号(>)、より小記号(<)、および引用符)のために使用されます。これらの文字は内蔵のエンティティとして使用することができ、エンティティの入力ヘルパー内の対応するエンティティをダブルクリックすることによりドキュメント内に入力することができます。(例えば、プログラムコードリスティング内ではCDATA セクション内に挿入することができます。CDATA セクションを入力する場合は、CDATA セクションを入力する場所を右クリックして、テキストメニューから「CDATA セクションの挿入」を選択します。XML プロセッサはCDATA セクション内の全てのマークアップ文字を無視します。CDATA セクション内で特殊文字を使用する場合、エンティティの参照ではなくその文字を入力します。
- キーボードを使用して入力できない特殊文字はシステムの文字マップからコピーしてドキュメント内の必要とする場所に貼り付けます。
- 頻繁に使用される文字列は、エンティティの入力ヘルパー内に表示されるエンティティとして定義することができます。エンティティは必要な場所にカーソルをポイントして、入力ヘルパーとして挿入することができます。テキスト文字列の値は一つの場所に保管されるため、これは管理のために役に立ちます。値を変更する場合、エンティティの定義を変更するだけで行うことができます。

メモ Authentic View 内でマークアップが非表示の場合、空の要素を簡単に見逃す可能性があります。空の要素を見逃すことを回避するため、大きなまたは小さなマークアップをオンスクリーンに替える。

上記のテキストコンテンツの各種類の使用法を試してください。

データ入力デバイスを使用してコンテンツを追加する方法

上記で学んだコンテンツの編集内で、テキスト内コンテンツとして直接入力することにより追加することができます。要素のコンテンツ(または、属性の値)を以下の方法以外でAuthentic View 内に入力することはできません。データ入力デバイスを使用して入力する

Authentic View 内のデータ入力デバイスのリストと各デバイスのためのXML ファイルごとのようにデータが入力されるかの説明と共に下に示されています。

データ入力デバイス	XML ファイル内のデータ
入力フィールド (テキスト ボックス)	ユーザーにより入力されたテキスト
複数の入力フィールド	ユーザーにより入力されたテキスト
ドロップボックス	値にマップされているユーザーの選択
チェックボックス	値にマップされているユーザーの選択
ラジオボタン	値にマップされているユーザーの選択
ボタン	値にマップされているユーザーの選択

アドレスフィールドを含む静的な テーブル内で、2つのデータ入力デバイスが扱われます (下に表示される): zip フィールドの入力フィールドと州のフィールドのドロップボックス。テキストフィールド内に入力する値は、対応する要素のXML コンテンツとして直接入力することができます。他のデータ入力デバイスに関しては、選択された内容は値にマップされます。

Street: 119 Oakstreet, Suite 4876
 City: Vereno
 State & Zip: DC 29213
 Vereno Office Summary artments, 15 employees.

上に示される Authentic View のための対応するXML テキストは以下のとおりです:

```
<Address>
  <ipo:street>119 Oakstreet, Suite 4876</ipo:street>
  <ipo:city>Vereno</ipo:city>
  <ipo:state>DC</ipo:state>
  <ipo:zip>29213</ipo:zip>
</Address>
```

ドロップボックスの選択 DC は DC の値にマップされていることに注意してください。zip フィールドの値は ipozip 要素のコンテンツとして直接入力されます。

3.5 属性の値を入力する

属性は要素のプロパティで、要素は無制限の数の属性を持つことができます。属性には値があります。XML データを属性の値として入力する場合があります。Authentic View 内では、属性の値を2つの方法で入力することができます:

- 属性が値を受け入れるように作成されている場合、メインウィンドウ内のエディットボックスとして入力する
- 属性入力ヘルパー内で入力する

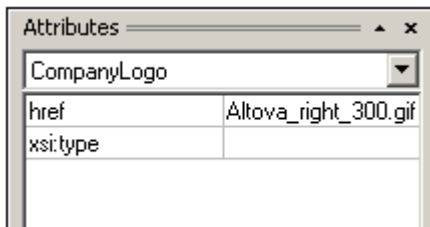
メインウィンドウ内の属性の値

属性の値を通常のテキスト、または、入力フィールド内にテキストとして、または、XML 値にマップされるユーザー選択として入力することができます。これは要素のエディットボックスが入力される方法と同様に入力されます: 次を参照してください: [Authentic View 内のデータの挿入](#)。このような場合、要素のエディットボックスと属性の値がStyleVision Power Stylesheet により区別され、データが適切に処理されます。

属性入力ヘルパー内の属性の値

属性の値を入力、または、変更する場合、属性入力ヘルパー内で行うこともできます。最初に、属性ノードを Authentic View 内で選択すると、属性の値が属性ヘルパー内に入力、または、属性ヘルパー内で編集されます。NanonullOrg.xml ドキュメント内では、の場所は、CompanyLogo 要素の href 属性の値として保管されます。使用されるを変更するには、以下を行います:

1. CompanyLogo タグをクリックして、CompanyLogo 要素を選択します。CompanyLogo 要素の属性は、属性入力ヘルパー内に表示されます。
2. 属性入力ヘルパー内で、href 属性をnanonull.gif からAltova_right_300.gifに変更します(AuthenticExamples フォルダー内のイメージ)。

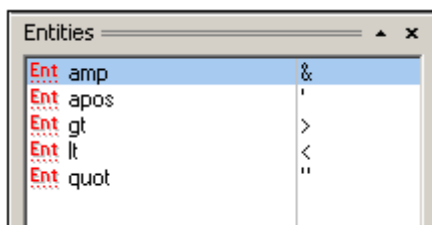


Nanonull がAltova により置き換えられます。

メモ 属性入力ヘルパー内にエンティティを入力することはできません。

3.6 エンティティを入力する

Authentic View 内のエンティティは必ずしもそうではありませんが通常、単一のテキスト文字列、およびXMLドキュメントのフラグメントのようなXMLデータです。エンティティはイメージファイルのようなバイナリファイルであることができます。すべてのエンティティは特別のドキュメントのために使用することができ、エンティティの入力ヘルパー内で表示されます(下のスクリーンショット)。エンティティを入力するときはドキュメント内の入力する箇所カーソルをポイントし、エンティティ内のエンティティをダブルクリックします。属性入力ヘルパー内エンティティを入力することはできないことに注意してください。



(アポストロフ、より大記号(>)、より小記号(<) シンボル、および二重引用符と同様)アンパサンド文字(&)はXML内で特別な意味を有します。XML特有の文字と区別するために、これらの文字を挿入するために、エンティティは使用されます。これらの文字はAuthentic View内のエンティティとして使用することができます。

NanonullOrg.xml 内で、Joe Martin (Marketing) のタイトルを Marketing Manager Europe & Asia に変更してください。以下を行ってください!

1. アンパサンドが入力される箇所にカーソルをポイントしてください。
2. 「amp」としてリストされているエンティティをダブルクリックします。アンパサンドが入力されます(下のスクリーンショット)。

First	Last	Title
Joe	Martin	Marketing Manager Europe &

メモ エンティティの入力ヘルパーは状況に依存しません。全ての使用することのできるエンティティは、カーソルの位置に関係なく表示されます。これは、ドキュメント内の全ての場所からエンティティを入力できることを意味しません。わからない場合は、エンティティを入力した後、ドキュメントを検証してください!「XML | 検証」(F8)。

自身のエンティティの定義

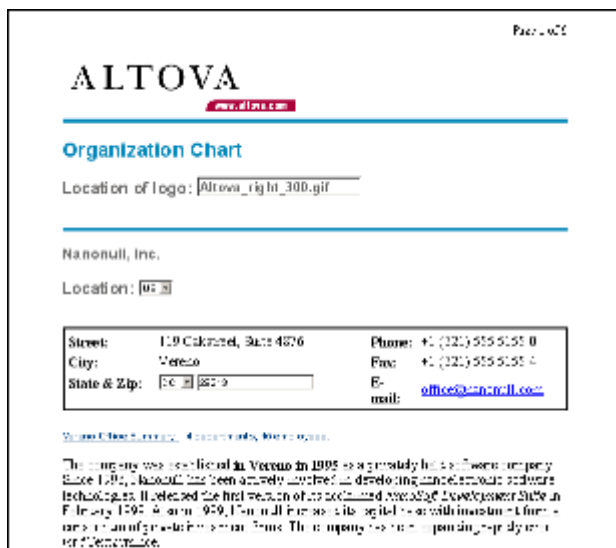
ドキュメントエディターとして、ドキュメントエンティティを定義することができます。この点についてセクション [Authentic View 内でエンティティを定義する](#)内で詳しく説明されています。

3.7 ドキュメントを印刷する

XMLドキュメントのAuthentic View からの印刷は、Authentic View 内の書式設定を保持します。

NanonullOrg.xml を印刷するに以下を行います:

1. おざりしていない場合は、マークアップモードを非表示にするに切り替えます。マークアップを印刷しない場合は、これを行う必要がありません。
2. 「ファイル | 印刷をプレビュー」を選択して、全てのページをプレビューします。50% に縮小された印刷のプレビューページが下に表示されています。ページの書式設定はAuthentic View と同じであることを注意してください。



3. ファイルを印刷するには、「ファイル | 印刷」をクリックします。

マークアップを表示するドキュメントのバージョンを印刷することもできます。これを行うには、Authentic View を小さなマークアップモードの表示、または、大きなマークアップモードの表示に切り替えて印刷します。

4 Authentic View インターフェイス

Authentic View はアクティブなドキュメントのAuthentic タブをクリックすると有効化されます。SPS がXMLドキュメントに割り当てられていない場合、割り当てるようプロンプトされます。

このセクションでは以下について説明されています:

- インターフェイスの概要
- Authentic View に特有のツールバーの説明
- メインのAuthentic View ウィンドウで使用することのできるモードの表示に関する説明
- 入力ヘルパーの使用方法的説明
- コンテキストメニューの説明は、XMLドキュメント内のAuthentic View で見つけることができます。

Authentic View 情報の追加ソースは以下のとおりです:

- Authentic View チュートリアルはAuthentic View インターフェイスの使用法を表示します。このチュートリアルはAltova XMLSpy とAltova Authentic Desktop 製品のドキュメント (チュートリアルセクションを参照してください)、および[オンライン](#)で見つけることができます。
- Authentic View メニューコマンドの詳細に関しては、次を参照してください: 製品ドキュメントのユーザーファレンスセクション。

Altova Web サイト: [XML コンテンツの編集, XML の作成](#)

4.1 GUI の概要

Authentic View は、ウィンドウの上部のメニューバーとツールバー、および、残りのインターフェイスを占める下記の3つの部分により構成されています: プロジェクトウィンドウ、メインウィンドウ、と入力ヘルパーウィンドウ。これは下記に示されるとおりです。



メニューバー

メニューバー内で使用することのできるメニューは、製品ドキュメントのユーザーファレンス セクション内で詳しく説明されています。

ツールバー

ツールバー内に表示されるシンボルとアイコンは、[Authentic View ツールバーアイコン](#)のセクションで説明されています。

プロジェクトウィンドウ

XML、XSL、XMLスキーマ、とエンティティファイルプロジェクト内でグループ化することができます。プロジェクトファイルのリストを作成し変更するには「プロジェクト」メニュー内のコマンドを使用します(この点に関しては、製品ドキュメントのユーザーファレンス セクションで詳しく説明されています)。プロジェクトファイルのリストはプロジェクトウィンドウ内に表示されています。プロジェクトウィンドウ内のファイルは、ダブルクリックすることによりアクセスすることができます。

情報ウィンドウ

このウィンドウは現在 Authentic View 内で選択されているノードに関する情報を提供します。

メインウィンドウ

XMLドキュメントが表示され編集されるウィンドウです。セクション [Authentic View メインウィンドウ](#)内で詳しく説明されています。

入力ヘルパー

次の3つの入力ヘルパーウィンドウがエリア内に存在します: 要素、属性、および、エンティティ。ウィンドウ(要素と属性 入力ヘルパー) 内に表示される内容は状況に依存します。すなわち、ドキュメントカーソルが存在する位置により異なります。要素またはエンティティをドキュメントに入力ヘルパーをダブルクリックすることにより挿入することができます。属性入力ヘルパー内のその属性の値フィールドに属性の値が入力されます。詳細に関しては、次を参照してください! [Authentic View 入力ヘルパー](#)

ステータスバー

ステータスバーは現在選択されているノードへのXPath を表示します。

コンテキストメニュー

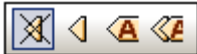
メインウィンドウ内で右クリックすると表示されるメニューです。使用することのできるコマンドは、状況に依存した編集コマンドです。すなわち、選択されたノードに関連する構造とコンテンツを操作することができます。このような操作は、ノードの挿入、追加、または、削除、エンティティの追加、またはコンテンツの切り取りと貼り付けを含みます。





4.2 Authentic View ツールバーアイコン

Authentic View ツールバー内のアイコンは、コマンドのショートカットです。アイコンの一部は、Windows アプリケーション、または Altova 製品で既に慣れ親しんだアイコンであり、その他は全く新しいアイコンです。このセクションでは、Authentic View の固有のアイコンについて説明されています。下の説明では、アイコンはグループ化されています。

XML マークアップの表示/非表示

Authentic View 内では、XML 要素、または 属性 の全部 (大きなマークアップ) または一部 (小さなマークアップ) を名前と共に、または、名前をなしで表示することができます。4つのマークアップアイコンがツールバー内に表示され、Authentic メニュー内で対応するコマンドを使用できるようになります。



	マークアップを非表示にします。すべてのXML タグは既に折りたたまれたものを除いて非表示になります。(展開する通常の方法である)「マークアップモードを非表示にする」内の折りたたまれたタグをクリックすると、ノードのエントレンが表示されタグが非表示になります。
	小さなマークアップを表示します。XML 要素/属性 タグは名前と共に表示されます。
	大きなマークアップを表示します。XML 要素/属性 タグは名前無しで表示されます。
	複合型マークアップを表示します。StyleVision Power Stylesheet 内では、各 XML 要素、または 属性 を表示するために大きなマークアップ、または、小さなマークアップ指定することができます。また、すべてを非表示にすることも出来ます。要素の一部がマークアップと共に表示、または、マークアップを隠して表示するように指定されているため、複合型マークアップモードと呼ばれます。複合型マークアップモード内では、Authentic View ユーザーはカスタム化されたマークアップを確認することができます。しかしながら、このカスタム化は StyleVision Power Stylesheet をデザインした個人により作成され、Authentic View ユーザーにより定義することはできません。





動的なテーブル構造の編集



動的な SPS テーブル内の行はデータ構造の繰り返しです。各行は単一の要素の発生を表示しています。各行は、結果として、次と同じ XML サブ構造を持ちます。

動的な テーブルの編集コマンドは、動的な SPS テーブルの行を操作します。要素の発生の数と順序を変更することができます。しかしながら、個別の要素の発生のサブ構造の変更を意味するため、動的な SPS テーブルの列が編集されます。

動的な テーブル編集コマンドのためのアイコンはツールバー内に表示され、Authentic メニュー内でも使用できます。



	テーブル1行を追加する
	テーブル1行を挿入する
	現在のテーブル行を複製する(すなわち、セルエントレンが複製されます)
	現在の行を上1つ移動する

	現在の行を下に1つ移動する
	現在の行を削除する

メモ これらのコマンドは、動的な SPS テーブルのみに適用されます。静的な SPS テーブル内で使用することはできません。Authentic View 内で使用されるテーブルの異なる種類は、このドキュメントの [Authentic View 内のテーブルの使用](#) セクション内で説明されています。

XML テーブルの作成と編集

データをテーブルとして表示するために、自身のテーブルを挿入することができます。このようなテーブルは、XML テーブルとして挿入されます。XML テーブルと書式 テーブルの構造を変更することができます。XML テーブルの作成と編集のためのアイコンは、下に表示されるようにツールバー内で使用することができます。セクション [XML テーブル編集アイコン](#) で詳細が説明されています。



アイコンに対応するコマンドは、メニューアイテムとしては使用することはできません。XML テーブルを使用するには、この機能は有効化され、StyleVision Power Stylesheet 内で適切に構成されている必要があります。

Authentic View 内で使用されているテーブルの型と XML テーブルの作成方法、及び、編集方法は [Authentic View 内のテーブルの使用](#) 内で説明されています。


テキストのフォーマット アイコン

Authentic View 内のテキストは、XML 要素を適用して、または、必要とされる書式設定を持つ属性を適用することにより書式設定することができます。このような書式設定が定義されると、StyleVision Power Stylesheet のデザイナーは、書式設定を適用するために Authentic View ツールバー内でアイコンを提供します。テキストのフォーマット アイコンを使用してテキストのフォーマットを適用するには、書式設定するテキストをハイライトして、適切なアイコンをクリックします。

DB 行のナビゲーションアイコン



矢印アイコンは(左側から右側) 最初のレコードに移動する、前のレコードに移動する、レコード # ダイアログに移動する、次のレコードに移動する、最後のレコードに移動する、です。

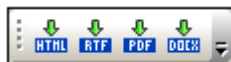
	このアイコンはエディタに入力することのできる「データベースレコードの変更」ダイアログを開きます。Authentic View はエディタレコードを表示します。
---	---

XML データベースの編集

「編集のための XML データを使用して新規の行を選択する」コマンドにより、IBM DB2 などの XML DB 内の関連するテーブルから新規の行を選択することができます。この行は、Authentic View 内で表示され、編集することができ、DB に戻し保存することができます。

ポータブルな XML 書式 (PXF) ツールバーボタン

XMLSpy と Authentic Desktop の Authentic View で次の PXF ツールバーボタンを使用することができます:



個別のボタンをクリックすると、HTML、RTF、PDF、およびまたはDocX 出力が生成されます。

これらのボタンはAuthentic View 内でPXF ファイルが開かれるとこれらのボタンを使用することができます。特定の出力書式のためにPXF ファイルがXSLT スタイルシートを含むように構成されている場合、個別のボタンは有効化されます。例えば、PXF ファイルがXSLT スタイルシートをHTML とRTF のために含むように構成されている場合、HTML とRTF 出力のためのツールバーボタンのみが有効化され、PDF とDocX (Word 2007+) 出力のためのボタンは無効化されます。

4.3 Authentic View メインウィンドウ

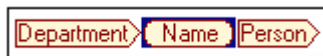
Authentic View には4つの表示モードが存在します: 大きなマークアップ、小さなマークアップ、複合型マークアップ、すべてのマークアップを隠す。マークアップ情報の多種のレベルをもつドキュメントをこれらのモードで確認することができます。モード間を切り替えるには **Authentic** メニュー内のコマンド、または ツールバー内のアイコンを使用します(前のセクションを参照してください: [Authentic View ツールバーアイコン](#))。

大きなマークアップ

タグ内に要素/属性名を持つ要素と属性の開始と終了タグを表示します:



上の図の中では、要素 Name が展開されており、すなわち、開始と終了タグ、および要素のコンテンツが表示されています。要素/属性は、開始、または、終了タグをダブルクリックすることにより、折りたたむことができます。短くされた要素/属性を展開するには、短くされたタグをダブルクリックします。



大きなマークアップ内では、属性は属性の開始と終了タグ内で等号シンボルとして認識されます:



小さなマークアップ

名前の無い要素/属性の開始と終了タグを表示します :

Nanonull, Inc.	
Location: US	
Street: 119 Oakstreet, Suite 4876 City: Vereno State & Zip: DC 29213	Phone: +1 (321) 555 5155 Fax: +1 (321) 555 5155 E-mail: office@nanonull.com
Vereno Office Summary: 4 departments, 15 employees.	
<p>The company was established in Vereno in 1995 as a privately held software company. Since 1996, Nanonull has been actively involved in developing nanoelectronic software technologies. It released the first version of its acclaimed NanoSoft Development Suite in February 1999. Also in 1999, Nanonull increased its capital base with investment from a consortium of private investment firms. The company has been expanding rapidly ever since.</p>	

開始タグ内にボソボソが存在し、終了タグは空であることにご注意してください。また、要素タグには角かっこが使用されており、属性タグには等号サインがシンボルとして使用されていることにご注意してください（下のスクリーンショットを参照してください）。

2006-04-01; Boston, USA

要素/属性を展開、または、折りたたみには適切なタグをダブルクリックしてください。下のサンプルは（青色でハイライトされている折りたたまれた要素を表示しています。折りたたまれた要素と展開された要素の開始タグの形にご注意してください）。

Office Summary: 4 departments, 15 employees.

複合型マークアップ

複合型マークアップは、カスタム化されたマークアップのレベルを表示します。StyleVision Power Stylesheet をデザインした個人は、ドキュメント内の個々の要素/属性のために大きなマークアップ、小さなマークアップ、または、マークアップ無しを決定することができます。Authentic View ユーザーは、内でカスタム化されたマークアップを複合型マークアップ表示モードで確認することができます。

すべてのマークアップを非表示にする

すべてのXML マークアップは格差されています。Authentic View 内で確認することのできる書式設定は、印刷されたドキュメントの書式であるため、この表示モードはドキュメントのWYSIWYG 表示です。

コンテンツの表示

Authentic View 内では、コンテンツおつの方法で表示されています。

- テキスト形式。テキストを入力し、このテキストが要素のコンテンツ、または、属性の値になります。



- データ入力デバイス。表示は、入力フィールド（テキストボックス）、複数の入力フィールド、コンボボックス、チェックボックス、または、ラジオボタンのいずれかを含んでいます。入力フィールドと複数の入力フィールドの場合、フィールドに入力するテキストが要素の XML コンテンツ、または、属性の値になります。

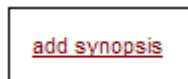


データ入力デバイスの場合、選択は StyleVision Power Stylesheet 内で指定される対応する XML 値を生成します。

このため、コンボボックス内では（コンボボックスのドロップダウンリスト内で使用することができる）「許可された選択」選択が「1」の XML 値、または、「許可された選択」、または、それをマップすることができます。「許可されていない選択」が「0」、「許可されていない選択」、または、その他にマップすることができます。

任意のノード

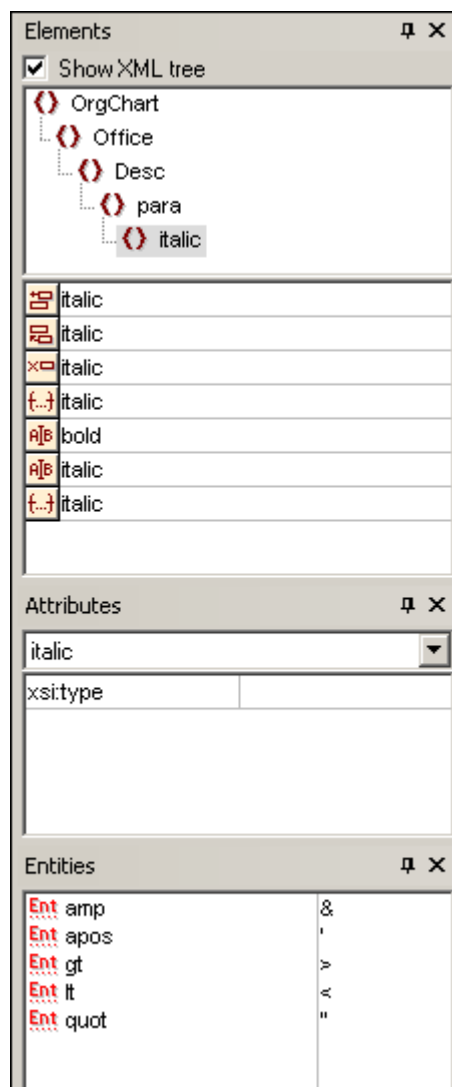
（参照されたスキーマに従い）要素、または、属性が任意の場合、型 `add [element/attribute]` のプロンプトが表示されます。



プロンプトをクリックすると、要素が追加され、カーソルがデータの输入のために表示されます。複数の任意のノードが存在する場合、プロンプト `add...` が表示されます。プロンプトをクリックすると、任意のノードのメニューが表示されます。

4.4 Authentic View エントリヘルパー

Authentic View 内におつの入力ヘルパーが存在します: 要素、属性、および、エンティティ。Authentic View インターフェイスの右下のウィンドウとしてこれらは表示されています(下のスクリーンショットを参照してください)。



要素と属性 入力ヘルパーは状況に依存しています。すなわち、入力ヘルパー内に表示される内容は、ドキュメント内のカーソルの位置により異なります。エンティティの入力ヘルパー内に表示されているエンティティは、状況に依存しません。ドキュメントのために許可されている全てのエンティティはカーソルの場所に関わらず表示されます。

それぞれの入力ヘルパーの説明に関しては下で説明されています。

要素入力ヘルパー

要素入力ヘルパーは2つの部分から構成されています:

- 上の部分には「XML ツリーを表示」チェックボックスを使用してオンとオフを切り替えることのできるXML ツリーが含まれています。XML ツリーは、現在の要素のドキュメントのルート要素までの祖先を表示しています。XML ツリー内の要素をクリックする

と、その要素に対応する要素（このリストの次のアイテム内で説明されているとおり）は要素入力ヘルパーの下の部分に表示されま

- 挿入することのできるノードのリストを含む下の部分は、Authentic View 内の選択された要素、または、テキストの範囲から挿入、削除またはクリアすることができます。入力ヘルパー内の要素名の左側のアイコンにより入力ヘルパー内にリストされる要素が表示されます。要素入力ヘルパー内に発生するアイコンは意味を説明する詳細と共に下にリストされます。

入力ヘルパーからノードを使用するには、アイコンをクリックします。



要素の後に挿入

入力ヘルパー内の要素は、選択された要素の後に挿入されます。正確な階層レベルに追加されている場合、例えば、//sect1/para 要素内にカーソルが存在する場合、sect1 要素を追加すると、新規のsect1 要素が//sect1/para の兄弟の後ではなく、para 要素の親であるsect1 要素の兄弟の後に追加されます。



要素の前に挿入

入力ヘルパー内の要素は選択された要素の前に挿入されます。要素の後に挿入 コマンドと同様、要素は正確な階層レベルとして挿入されます。



要素の削除

要素とコンテンツを削除します。



要素の挿入

入力ヘルパーからの要素を要素内に挿入することもできます。要素内にカーソルが置かれると、その要素の許可されている子要素を挿入することができます。許可されている子要素は要素のみのコンテンツモデルの一部コンテンツモデル、および、複合型 コンテンツモデル(テキストと子要素)とすることができます。

テキストの範囲が選択されていると、許可されている子要素がテキスト内で入力ポイントとしてカーソルがポイントされている場所に挿入されます。

- テキストの範囲が選択され、要素が挿入されると、テキストの範囲が挿入された要素のコンテンツになります。
- 要素が入力ポイントに挿入されると、要素はそのポイントに挿入されます。

要素の挿入後、これらのインラインの要素のオグの表示される要素入力ヘルパー内の要素アイコンをクリックすることによりクリアすることができます。テキストの範囲、または、入力ポイントとしてテキスト内にカーソルをポイントするかにより表示されるアイコンが決定されます（下を参照してください）。



要素の適用

（大きなマークアップの表示 ビュー内の開始、および終了タグをクリックすることにより）ドキュメント内の要素を選択する場合、および、要素が他の要素により置き換えられる場合（例えば、para などの複合型 コンテンツ 要素内では、italic 要素 がbold 要素と置き換えられることができます）、このアイコンは入力ヘルパー内の要素が選択された（オリジナルの）要素に適用することができることを示します。要素の適用 コマンドを複合型 コンテンツの要素内のテキストの範囲に適用することもできます。テキストの範囲は適用された要素のコンテンツとして作成されます。

- 適用された要素が元の要素の子と同じ名前を持つ子要素を持つ場合、また、元の要素内でこの子要素のインスタンスが存在する場合、オリジナルの子要素が新規の要素のコンテンツ内で保持されます。
- 適用された要素が元の要素のインスタント化された子と同じ名前を持たない子要素を持つ場合、元の要素のインスタント化された子が新規の要素が持つ子要素、または、要素の兄弟として追加されます。
- 適用された要素が元の要素のコンテンツモデル内に等価が存在しない子要素を持つ場合、この子要素は直接作成されず、Authentic View が挿入するオプションを提供します。

要素ではなくテキストの範囲が選択されている場合、要素の選択への適用は、コンテンツとして選択されたテキストの範囲と共にその場所で適用された要素を作成します。カーソルが入力ポイントの場合要素の適用は許可されていません。



要素をクリアする

このアイコンは、複合型コンテンツの要素内のテキストが選択されると表示されます。アイコンをクリックすると選択されたテキストの範囲の周りに要素をクリアします。



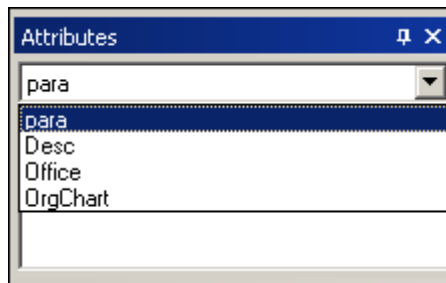
要素をクリアする (挿入ポイントが選択されている場合)

複合型の子コンテンツ要素である要素内にカーソルが置かれるとこのアイコンが表示されます。アイコンをクリックするとインラインの要素がクリアされます。

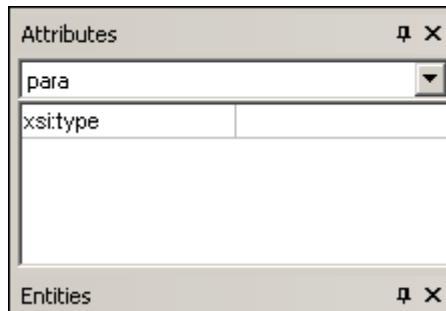
属性入力ヘルパー

属性入力ヘルパーはドロップダウンボックスと属性のリストから構成されています。選択した要素がドロップダウン内に表示されます (開始、または、終了タグをクリック、または、要素コンテンツ内にカーソルをポイントして選択します)。

下に示されている属性入力ヘルパーはコンボボックス内の para 要素が表示されています。コンボボックス内の矢印をクリックすると、この場合 OrgChart である para 要素の祖先ドキュメントのルート要素までの祖先を表示するドロップダウンリストが表示されます。



コンボボックスの下に、要素のための有効化属性のリストが表示されます。この場合、para のためのリストが表示されます。要素で属性が必須の場合、太字で表示されます。(下のサンプルでは、内蔵の属性 xsi:type 以外の必須の属性は存在しません)。



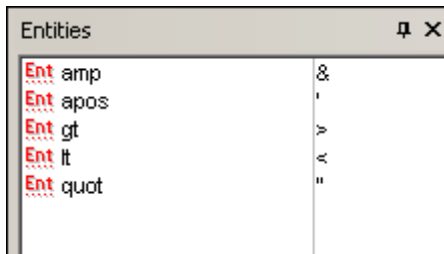
属性のための値を入力するには、属性の値フィールドをクリックして、値を入力します。これにより属性とその値がXMLドキュメント内に作成されます。

以下に注意してください!

- nillable 要素が選択された場合属性入力ヘルパー内に表示される xsi:nil 属性の場合、xsi:nil 属性の値は属性の値のためのドロップダウンリストから許可されている値 (true、または、false) の一つを選択して入力することができます。
- xsi:type 属性を属性の値フィールド内をクリックして変更することができ、ドロップダウンリストからリストされた値を選択することができます。リストされた値は、Authentic View ドキュメントがベースとされる XML スキーマ内で定義された使用することのできる抽象型です。

エンティティの入力ヘルパー

エンティティの入力ヘルパーによりドキュメントにエンティティを挿入することができます。エンティティを使用してドキュメントに企業名など頻繁に発生する特殊文字、または、テキスト フラグメントを挿入することができます。エンティティを挿入するときは、エンティティを挿入するテキストにカーソルを置き、エンティティの入力ヘルパー内のエンティティをダブルクリックします。



メモ 内部のエンティティは DTD 内で定義された値を持ちます。外部エンティティは他の XML ファイルなどの外部ソース内に含まれる値を持ちます。内部と外部エンティティはエンティティの入力ヘルパー内にリストされています。エンティティを挿入すると、値ではなく、内部の、または、外部のエンティティが XML テキストに挿入されます。エンティティが内部のエンティティの場合、Authentic View エンティティの値を表示します。エンティティが外部エンティティの場合、Authentic View は値ではなくエンティティを表示します。これは、Authentic View 内で表示される外部エンティティである XML ファイルを意味します。コンテンツは Authentic View 表示内のエンティティを置き換えません。

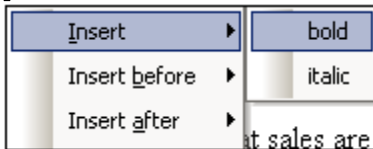
Authentic View 内で自身のエンティティを定義することができ、これらは入力ヘルパー内で表示されます: Authentic View セクション内の編集の[エンティティの定義](#)を参照してください

4.5 Authentic View コンテキストメニュー

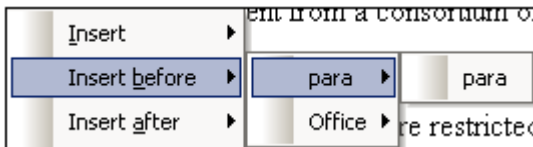
選択されたドキュメントのコンテンツ、選択に関連したコマンドを使用して表示されるコンテキストメニュー、またはカーソルの場所を右クリックします。

要素の挿入

下の図は現在のカーソルの位置で挿入することのできる全ての要素リストであるサブメニューの挿入を表示しています。サブメニューの前に挿入は現在の要素の前に挿入することのできる全ての要素をリストします。サブメニューの後に挿入 サブメニューは現在の要素の後に挿入することのできる全ての要素をリストします。下の図では現在の要素は para 要素です。bold と italic 要素は、現在の para 要素内に挿入することができます。



下で確認することができるように、para と Office 要素は現在の para 要素の前に挿入することができます。



ノードの挿入、置換え、(適用)、およびマークアップの削除 (クリア) コマンドはコンテキストメニュー内、および [Authentic View 入力レベル 2](#) 内で使用することができますそれぞれのセクションで説明されています。

エンティティの挿入

「エンティティの挿入」コマンドにカーソルをポイントすると、宣言されている全てのエンティティのリストを含むサブメニューが表示されます。エンティティをクリックすると、選択された場所エンティティが挿入されます。次を参照してください! ドキュメントのためにエンティティを定義する方法に関しては [エンティティの定義](#) を参照してください。

CDATA セグメントの挿入

このコマンドは、カーソルがテキスト内にポイントされると有効化されます。クリックするとカーソルの位置で CDATA セグメントが挿入されます。CDATA セグメントは開始と終了タグにより区切られています。大きな、小さなマークアップをオンにしてこれらのタグを確認することができます。CDATA セグメント内では、XML マークアップと解析は無視されます。XML マークアップ文字 (アンパサンド、アポストロフィ、より大記号 (>)、より小記号 (<)、および引用符) はマークアップとして扱われず、リテラルとして扱われます。ですから、CDATA セグメントは、XML マークアップ文字を持つプログラムコードリスティングなどのテキストのために役立ちます。

ノードの削除

「削除」コマンドにマウスのカーソルをポイントすると、選択されたノードにより構成されたメニューリストとドキュメントを無効化しない) 削除可能な祖先すべてが表示されます。削除する要素をクリックします。これは要素、または、削除可能な祖先を簡単に削除する方法です。祖先要素をクリックすると、選択されて要素を含む全ての子孫が削除されます。

クリア

「クリア」コマンドは要素マークアップを選択の周りからクリアします。ノード全体が選択されると、要素 マークアップはノード全体のためにクリアされます。テキストセグメントが選択されると、要素 マークアップがそのテキストセグメントのみのためにクリアされます。

適用

「適用」コマンドはメインウィンドウ内の選択の選択された要素に適用されます。詳細に関しては、次を参照してください: [Authentic View 入力ヘルプ](#)。

コピー、切り取り、貼り付け

これは標準ウィンドウコマンドですが「貼り付け」コマンドはコピーされたテキストを SPS 全体のために指定されたスタイルシートデザイナーにより XML、または、テキストとして貼り付けます。「XML としてコピー」と「テキストとしてコピー」コマンドの詳細に関しては、次を参照してください: 下記の「貼り付け」コマンドの詳細。

貼り付け

「貼り付け」コマンドは Authentic View XML フラグメントにクリップボードにコピーされた内容を XML、または、テキストとして張り付けるオプションを提供します。コピーされたフラグメントが XML として貼り付けられると、XML マークアップと共に貼り付けられます。テキストとして貼り付けられると、(XML マークアップではなく)コピーされたフラグメントのテキストコンテンツのみが貼り付けられます。次のシチュエーションが可能です:

- **マークアップタグと共にノード全体**が Authentic View 内でイライトされ、クリップボードにコピーされます。(i) ノードはこのノードが有効な場所で XML として解析されます。無効な場所では解析されません(ii) ノードがテキストとして解析されると (マークアップではなく)ノードの **テキストコンテンツのみ**が貼り付けられます。テキストを貼り付けることのできる XML ドキュメント内にテキストコンテンツを貼り付けることができます。
- **テキストフラグメント**が Authentic View 内でイライトされ、クリップボードにコピーされます。(i) フラグメントが貼り付けられる場所で XML ノードが有効な場合のみ、このフラグメントが XML として貼り付けられる場合、テキストの XML マークアップタグは、テキストフラグメントと共に明示的にコピーされず、テキストと共に貼り付けられます。(ii) フラグメントがテキストとして貼り付けられる場合、テキストが貼り付けられる XML ドキュメント内の場所に貼り付けられます。

メモ テキストは、テキストが許可されているノードにコピーされます。コピーされたテキストがドキュメントを無効化しないようご注意ください。コピーされたテキストは以下のようになります:

- (i) 新規の場所で構文的に有効な場合 (例えば、数字のノード内の数字以外の文字が無効になります)。
- (ii) 以外はノードを無効化する場合 (例えば、3つの桁のみ受け入れるノード内の4つの桁はノードを無効化します)。

メモ 解析されたテキストがドキュメントを無効化する場合、テキストが赤い色で表示されます。

削除

「削除」コマンドは選択されたノードとそのコンテンツを削除します。ノード内にカーソルをポイント、または、ノードの開始、または、終了タグをクリックすることによりノードは選択されているとして考えられます。

5 Authentic View 内で編集する

このセクションは Authentic View の重要な機能について詳しく説明しています。機能は頻繁に使用されるため、または、メカニズム、または、概念について説明が必要なため、これらの機能はこのセクションに含まれています。

セクションでは以下について説明されています:

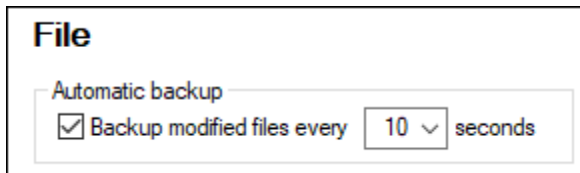
- Authentic View 内で使用されているテーブルは3つの種類が存在します。セクション [Authentic View 内のテーブルを使用する](#) はテーブルの3つの種類 (静的な SPS、動的な SPS、および XML) がいつどのように使用されるかについて説明しています。最初に広義および概念についての説明がされ、使用方法の詳細について説明されます。
- 日付の選択は、日付をクリックすると、正確な XML 書式で日付を入力する視覚的なカレンダーです。次を参照してください! [日付の選択](#)。
- エンティティは、特殊文字、または、テキスト文字列の短縮形です。これらの特殊文字、または、テキスト文字列を挿入するために、対応するエンティティを挿入することにより、自身のエンティティを定義することができます。詳細に関しては、次を参照してください! [エンティティの定義](#)
- Altova 製品の Enterprise と Professional エディションでは、Authentic View ユーザーは、[デジタルXML 署名](#)を使用して XML ドキュメントを署名し、署名を検証することができます。
- Authentic View で表示することのできる [イメージ](#)。

Altova Web サイト: [XML コンテンツの編集、XML の作成](#)

5.1 ファイルを自動的にバックアップする

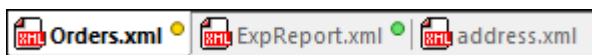
Authentic Desktop 内で変更されたファイルは定期的な頻度で自動的にバックアップされます。[\(「ツール」オプション「ファイル」\)](#) オプションダイアログの「ファイルタブ」で以下のスクリーンショットで示されている通り以下を行うことができます。

- 自動バックアップを自動的にオン/オフに切り替える
- バックアップの頻度 (5 秒から 300 秒まで) を指定します。



インジケータ

メインウィンドウの下のファイルタブにはファイル名の右側に表示されるシンボルが含まれています。これらのシンボルはファイルの保存済み/未保存状態およびバックアップ状態を表示しています (下のスクリーンショットを参照)。



保存済み / 未保存

色付けされたサークルのシンボルはファイルの変更済みの場合表示されます。このようなシンボルが存在しない場合、ファイルは開かれた時点または最後の保存時から変更されていません。上のスクリーンショットでは、例えば address.xml を参照してください。

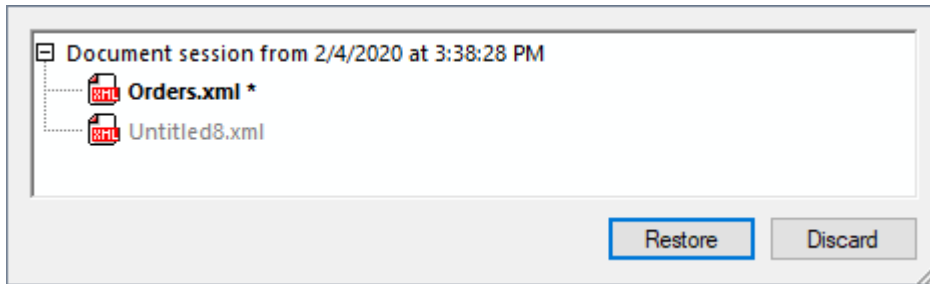
バックアップの状態

サークルのシンボルはファイルのバックアップの状態を示しています。

- **黄色:** ファイルの変更済みで、最後の変更時からバックアップ (または未保存) の場合。
- **緑:** ファイルのバックアップ済みで、最後のバックアップ時から変更されていません。しかしながら、ファイルが保存されていない場合。(保存されている場合、赤いサークルのシンボルは表示されません。)
- **赤:** バックアップはこのファイルのためにサポートされていません。またはバックアップに失敗した場合。
- **グレー:** 自動的なバックアップ機能が無効化されている場合 (via the [オプションダイアログ](#)の使用。上記を参照してください)。このシンボルの存在はしかしながらファイルが最後の変更時から保存されていないことを意味します。(保存されている場合、赤いサークルのシンボルは表示されません。)

バックアップの復元

Authentic Desktop が予期せず終了した場合、アプリケーションが次回起動されるとアプリケーションの終了時に開かれていたすべてのドキュメントのリストを含む「ドキュメントの復元」ダイアログが表示されます (下のスクリーンショット)。リストを確認するために各ファイルにマウスをポイントすることができます。保存されていない一時ファイルの場合、ファイルリストはそのファイルのために開かれている名前を付けて保存ダイアログが現在のデフォルトのリストになります。



リスト内の各ファイルのためのフォントスタイルとアスタリスクの存在/不在により以下の情報を提供されます:

- 太字とアスタリスクはファイルに保存されていない変更が含まれていることを示しています。このようなファイルは最後にバックアップされた状態で復元されます。
- 標準のスタイルはファイルが保存済みで未保存のファイルが存在しないことを示しています。このようなファイルは保存された状態で復元されます。
- 灰色表示されているスタイルはファイルが保存されておらず、またバックアップされていないことを表示しています (例えば、編集されていない新規のファイルなどが挙げられます)。このようなファイルは復元されません。

以下の一つを行うことができます:

- 「復元」をクリックして GUI 内のファイル最後のバックアップの状態に復元します。
- 「破棄」をクリックしてリストされているファイルを開くことなく、使用可能なバックアップを破棄します。

5.2 基本的な編集

Authentic View 内で編集を行う場合、XMLドキュメントが編集されます。Authentic View はドキュメントの構造的な XML マークアップを非表示にすることができ、ドキュメントのコンテンツのみを表示することができます（下最初のスクリーンショット）。XML の技術的な側面に触れることなく、ドキュメントをノーマルなテキストドキュメントと同様に編集することができます。希望する場合、編集中にマークアップをオンスイッチ替えることもできます（下の 2 番目のスクリーンショット）。

Vereno Office Summary: 4 departments, 16 employees.

The company was established **in Vereno in 1995** as a privately held software company. Since 1996, Nanonull has been actively involved in developing nanoelectronic software technologies. It released the first version of its acclaimed *NanoSoft Development Suite* in February 1999. Also in 1999, Nanonull increased its capital base with investment from a consortium of private investment firms. The company has been expanding rapidly ever since.

XML マークアップの存在しない編集することのできる Authentic View ドキュメント

Address ipo:city **Vereno** ipo:city Address **Office Summary:**
4 departments, 16 employees. Desc para

The company was established **in Vereno in 1995** as a privately held software company. Since 1996, Nanonull has been actively involved in developing nanoelectronic software technologies. It released the first version of its acclaimed *NanoSoft Development Suite* in February 1999. Also in 1999, Nanonull increased its capital base with investment from a consortium of private investment firms. The company has been expanding rapidly ever since.

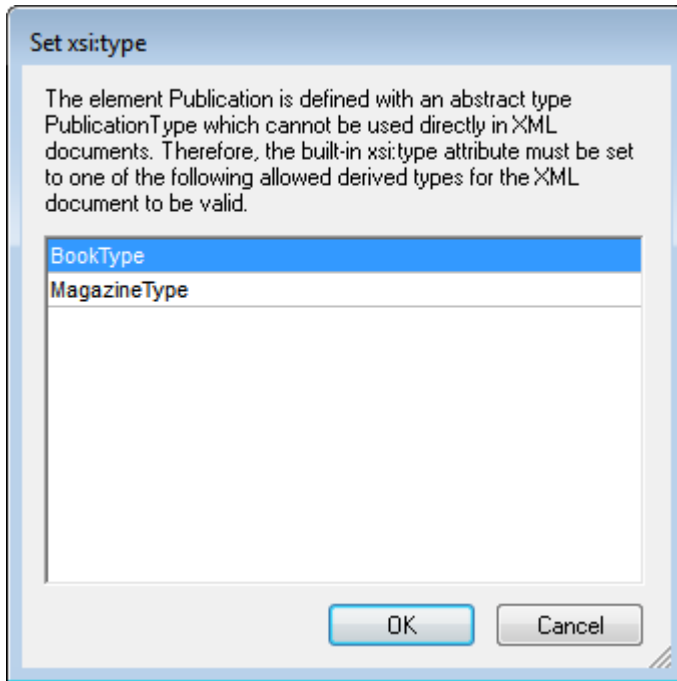
para para

XML マークアップタグが存在する編集することのできる Authentic View ドキュメント

ノードの挿入

新規のノードを Authentic XML ドキュメントに追加する必要がある場合があります。例えば、新規の `Person` 要素はドキュメントの `PersonList` の型に追加される必要があります。このような場合、XML スキーマは新規の要素の追加を許可します。新規のノードを追加する箇所の後に Authentic View ドキュメント内でノードをクリックするだけです。表示されるコンテキストメニュー内で、必要に応じて「前に挿入」、または「後に挿入」を選択します。ドキュメントに挿入することのできるノードはサブメニュー内に表示されます。必要とされるノードをクリックして挿入します。全ての必須の子孫ノードも挿入されます。子孫ノードが任意の場合、クリックすることのできるリンク `Add NodeName` が表示され、任意のノードを追加することができます。

追加されるノードが抽象型の要素の場合、(下のスクリーンショットで表示されるような) XML スキーマ内で使用することのできる生成された型のリストを含むダイアログが表示されます。



Publication 要素が追加されると、上のスクリーンショットが表示されます。Publication 要素は抽象的な複合型である型 PublicationType です。2つの複合型 BookType と MagazineType は抽象型 PublicationType から派生します。結果として、Publication 要素が XML ドキュメントに追加されると、Publication の抽象型から派生した2つの具体型の1つが指定される必要があります。新規の Publication 要素は xsi:type 属性に追加されます:

```
<Publication xsi:type="BookType"> ... </Publication>
<Publication xsi:type="MagazineType"> ... </Publication>
...
<Publication xsi:type="MagazineType"> ... </Publication>
```

使用することのできる生成された型の1つを選択し、「OK」をクリックすると以下を行うことができます:

- 選択された生成された型を要素の xsi:type 属性の値として設定します。
- 選択された生成された型のコンテンツモデル内で定義された子孫ノードと共に要素を挿入します。

属性入力ヘルパー内の要素の xsi:type 属性の値を変更することにより、選択された生成された型は後で変更することができます。要素の型がこのように変更されると、前の型のコンテンツモデルのノードは、削除され新しいコンテンツモデルのノードが挿入されます。

テキストの編集

Authentic View ドキュメントは、基本的にテキストとイメージにより構成されています。ドキュメント内のテキストを編集するには、テキストを挿入する箇所にカーソルを置き、編集します。「削除」キーなどのキーストロークとドラッグアンドドロップ機能を使用して、テキストのコピー、移動、削除を行うことができます。「Enter」キーは唯一の例外です。Authentic View ドキュメントは書式設定済みのため、追加のラインスペースをアイテムの間に追加することができません。Authentic View 内の「Enter」キーは、このため、現在編集されている要素の他のインスタンスに追加するのではなく、この目的のみで使用することができます。

XML またはテキストとしてコピーする

XML、または、テキストとしてコピーし貼り付けることができます。

- テキストがXMLとして解析されると、XML マークアップは、ノードのテキスト コンテンツと共に解析されます。XML マークアップはノードのコンテンツの一部がコピーされた場合でも、貼り付けられます。マークアップを貼り付けるには、貼り付ける場所ですクリーンに従い、貼り付けが許可されている必要があります。
- テキストがテキストとして解析されると、XML マークアップ破壊されません。

XML、またはテキストとして貼り付けるには、テキスト (Ctrl+C) をコピーし、テキストを貼り付ける箇所を右クリックし、コンテキストメニュー コマンド「貼り付け | XML」、または「貼り付け | テキスト」を選択します。ショートカット「Ctrl+V」が使用される場合、テキストはデフォルトのSPSのデフォルトの貼り付けモードで貼り付けられます。SPSのデザイナーにより貼り付けモードは指定することができます。詳細に関しては、次を参照してください: [セグメン コンテキスト メニュー](#)。

更に、ハイライトされたテキストを貼り付ける箇所をドラッグします。テキストがドロップされるとポップアップが表示され、テキストをテキスト または XML として貼り付けるかを問われます。希望するオプションを選択します。

テキストのフォーマット

XMLドキュメントのシステムの基本的な概念は、プレゼンテーションとコンテンツは区別されるということです。XMLドキュメントにはコンテンツが含まれますが、スタイルシートにはプレゼンテーション (書式設定) が含まれています。Authentic View 内では XMLドキュメントはスタイルシートにより表示されています。Authentic View 内で表示されるすべての書式設定はスタイルシートにより作成されていることを意味します。太字のテキストが表示されると、スタイルシートにより太字の書式設定が提供されます。リスト書式、または、テーブル書式を表示するリスト、または、テーブルはスタイルシートにより提供されています。Authentic View 内で編集するXMLドキュメントにはコンテンツのみが含まれており、書式設定は含まれていません。スタイルシートに書式設定が含まれています。Authentic View ユーザーは編集するテキストの書式設定をする必要が無く、または、できないことを意味します。コンテンツに自動的に適用される書式設定は、データのセマンティクスおよびまたは構造的な値にリンクされています。例えば、(セマンティックな単位として考えることのできる) 電子メールアドレスは電子メールであるため、特定の手法で自動的に書式設定されます。同様に、ヘッラインドキュメント (構造的な及び構文的なユニットを含む) 内の特定の場所で発生する必要があり、スタイルシートデザイナーがそのヘッラインのため指定する書式に自動的に書式設定されます。電子メールアドレス、または、ヘッラインの書式設定を変更することができます。電子メールアドレス、または、ヘッラインのコンテンツを編集することができます。

一部の場合、コンテンツは特別に表示される必要があります。例えば、テキスト文字列を太字で表示するなど。すべてのこのような場合、プレゼンテーションはドキュメントの構成要素に結び付けられている必要があります。例えば、表示するテキスト文字列を太字で表示するなど、太字で表示することのできるスタイルシートデザイナーによるマークアップにより周りのコンテンツから構造的に区別される必要があります。Authentic View ユーザーとして、このようなテキスト文字列を使用する場合テキスト文字列を適切な要素マークアップで囲む必要があります。これを行うための情報に関しては、次を参照してください: [ドキュメントの要素入力ヘルパー](#) セグメント内の要素の挿入 コマンド

Authentic View 内で RichEdit を使用する

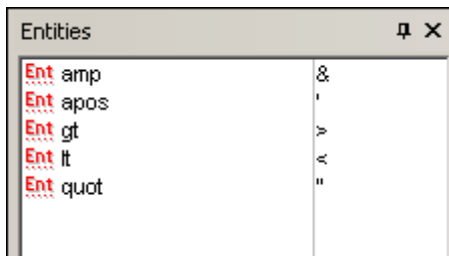
Authentic View 内で、RichEdit ツールバーとして作成された要素の中にカーソルが置かれている場合、ボタンとコントロールは有効化されます (下のスクリーンショット)。それ以外の場合、グレーアウトされます。



RichEdit ツールバーのボタンとコントロールを使用して適用するスタイルを指定し、スタイルするテキストを選択します。RichEdit により Authentic View ユーザーはフォント、フォントの大きさ、フォントのスタイル、フォントの飾り、色、テキストの背景色と配置を指定することができます。スタイルされたテキストはスタイル要素のタグで囲まれています。

エンティティの挿入

XMLドキュメント内では、文字の一部はマークアップのため保管されており、ノーマルなテキストでは使用することができません。これらの文字はアンパサンド (&)、アポストロフィ (')、より小記号 (<)、より大記号 (>)、引用符 (") 文字です。これらの文字をデータ内で使用する場合は、[エンティティの入力ヘルパー](#) を使用して、エンティティリファレンスとして挿入します (下のスクリーンショット)。



XML はカスタムのエンティティを作成する機会を与えます。

これらは以下であることができます: (i) キーボードで使用することのできない特殊文字。(ii) ドキュメントのコンテンツ内で再利用するテキスト文字列。(iii) XML データフラグメント、または (iv) イメージなどの他のリソース。Authentic View アプリケーション内で [エンティティを自身で定義する](#) ことができます。定義後、これらのエンティティは [エンティティの入力ヘルパー](#) 内に表示され、ドキュメントに挿入されます。

CDATA セクションの挿入

CDATA セクションはXML データとしてXML パーサーが処理しないXMLドキュメント内のテキストのセクションです。エンティティ参照による特殊文字の置換えを希望しない場合テキストの大きなセクションをエスケープするために使用することができます。例えば、マークアップタグを使用して再生成するプログラムコード、またはXML フラグメントです。要素のコンテンツ内でCDATA セクションは発生することができます。開始と終了で<![CDATA[and]]> を使用してそれぞれ区切ることができます。この結果、テキスト文字列]]> はセクションの終わりを完了前に示すためCDATA セクションは発生しません。この場合、より大きい文字はエンティティ参照(>) によりエスケープされるべきです。要素内にCDATA セクションを入力するときは、希望する場所にカーソルを置き、右クリックし、コンテキストメニューからCDATA セクションの挿入を選択します。CDATA セクションタグを Authentic View 内で確認するには、[マークアップの表示オンに切り替えます](#)。更に、CDATA セクション内に含まれるテキストをハイライトし、CDATA セクションの挿入 コマンドを選択します。

メモ CDATA セクションを入力フィールド(すなわち、テキストボックスと複数のテキストボックス)に挿入することはできません。CDATA セクションは、Authentic View にテキストコンテンツコンポーネントとして表示される要素内でのみ入力することができます。

リンクの編集

ハイパーリンクは2つの部分から構成されています: リンクテキストとリンクのターゲット。リンクテキストをテキスト内をクリックして編集することにより編集することができます。リンクのターゲットを編集することはできません。(静的なターゲットアドレス、またはターゲットアドレスをXMLドキュメント内に含まれるデータを生成することにより)スタイルシートのデザイナーによりリンクのターゲットが設定されます。) Authentic View からは、「Ctrl」を押してをクリックしてリンクのターゲットに移動することができます。(留意点: リンクをクリックするだけで、リンクテキストを編集するためのセットアップを行うことができます)。

5.3 Authentic View 内のテーブル

3つのテーブルの種類は、2つのカテゴリに分けることができます: SPS テーブル(静的と動的) と CALS/HTML テーブル。

SPS テーブルには2つの種別があります: 静的と動的。XMLドキュメントがリンクされている StyleVision Power Stylesheet のデザイナーにより SPS テーブルはデザインされます。XMLドキュメント内に SPS テーブルを挿入することまではできず、SPS テーブルフィールドにデータを挿入し、動的な SPS テーブルは行を追加、または削除することができます。[SPS テーブル](#) のセクションでは、これらのテーブルについて説明されています。

CALS/HTML テーブルは Authentic View のユーザーにより挿入されます。目的は、ドキュメント階層構造内で希望する場所にテーブルを挿入することです。[CALS/HTML テーブル](#) と [CALS/HTML テーブル編集アイコン](#) の編集機能に関しては、下で説明されています。

5.3.1 SPS テーブル

Authentic View 内で2つの種類の SPS テーブルが使用されます: 静的なテーブルと動的なテーブル。

静的なテーブル

静的なテーブルは構造内およびセルのエントリ型内で固定されています。Authentic View のユーザーは、テーブルセルにデータを挿入することができますが、これらのテーブルの構造を変更(すなわち、行や列の追加)すること、セルのエントリ型の変更することまではできません。テキスト内にタイプしてデータを入力、または、チェックボックス、または、ラジオボタンのフォーム内のオプションから選択して、または、コンボボックスのリストから選択してデータを入力することができます。データを入力すると編集することができます。

Nanonull, Inc.	
Street:	119 Oakstreet, Suite 4876
City:	Vereno
State & Zip:	DC 29213
Phone:	+1 (321) 555 5155
Fax:	+1 (321) 555 5155 - 9
E-mail:	office@nanonull.com

メモ 動的なテーブルを編集するためのアイコン、または、コマンドは、静的なテーブルを使用する際に使用することはありません。

動的なテーブル

動的なテーブルには繰り返しデータ構造を表す行が存在します。すなわち、(静的なテーブルの場合は異なりますが)各行は同一のデータ構造を持ちます。結果として、行操作を行うことができます: 行の追加、行の挿入、行を上へ移動する、行を下へ移動する、行の削除。これらのコマンドは **Authentic** メニューとツールバー内のアイコンとして使用することができます(下に表示される)。



これらのコマンドを使用するには、適切な行内にカーソルをポイントし、必要とするコマンドを選択します。

Administration								
First	Last	Title	Ext	EMail	Shares	Leave		
						Total	Used	Left
Vernon	Callaby	Office Manager	581	v.callaby@nanonull.com	1500	25	4	21
Frank	Further	Accounts Receivable	471	f.further@nanonull.com	0	22	2	20
Loby	Matise	Accounting Manager	963	l.matise@nanonull.com	add Shares	25	7	18
Employees: 3 (20% of Office, 9% of Company)					Shares: 1500 (13% of Office, 6% of Company)			
Non-Shareholders: Frank Further, Loby Matise.								


テーブル内のセルを移動するには、「上」、「下」、「右」、「左」矢印キーを使用します。前方の次のセルに移動する場合は、「タブ」キーを使用します。最後の行の最後のセル内の「タブ」キーを押すと、新規の行が作成されます。

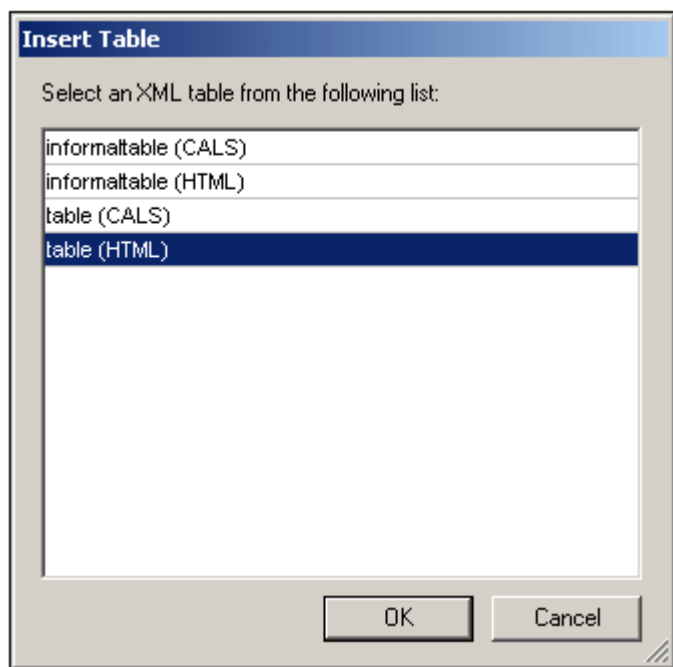
5.3.2 CALS/HTML テーブル

一部のXML データ構造がテーブル書式を表示するように指定されているため、Authentic View のユーザーはCALS/HTML テーブルを挿入することができます。CALS/HTML テーブルと作業するには3つのステップが含まれています: テーブルの挿入。書式設定。データの挿入。CALS/HTML テーブルと作業するためのコマンドはツールバー内のアイコンとして使用することができます(次を参照してください) [CALS/HTML テーブル編集アイコン](#)。

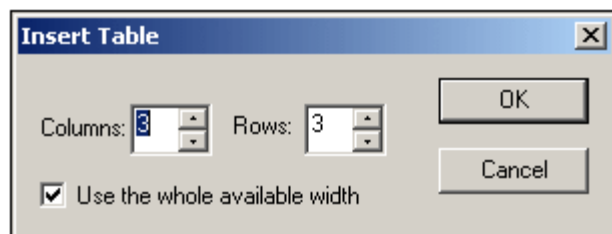
テーブルの挿入

CALS/HTML テーブルを挿入するには、以下を行います:

1. テーブルを挿入する箇所にカーソルをポイントし、 アイコンをクリックします。(スキーマによりテーブルを挿入する箇所は決定されることにご注意ください)。テーブルの挿入 ダイアログが表示されます(下のスクリーンショット)。このダイアログはテーブル構造が定義されているすべてのXML 要素 データ構造をリストします。例えば、下のスクリーンショットではinformal テーブル要素とテーブル要素はCALS テーブル、および、HTML テーブルと定義されています。



2. 挿入する要素とテーブルモデルを含むエントリを選択し、「OK」をクリックします。
3. 次のダイアログ内では(下のスクリーンショット)列と行の数量を選択し、テーブルに追加されるヘッダーおよびフッターを選択し、使用することのできる幅全てをテーブルに使用するかを指定します。完了すると、「OK」をクリックします。



上に示されるダイアログボックス内の仕様のために、以下のテーブルが作成されました。

テーブルメニューコマンドを使用して、列を追加、または削除、列を作成、列をジョインし分割することができます。開始するには、広義の構造を作成する必要があります。

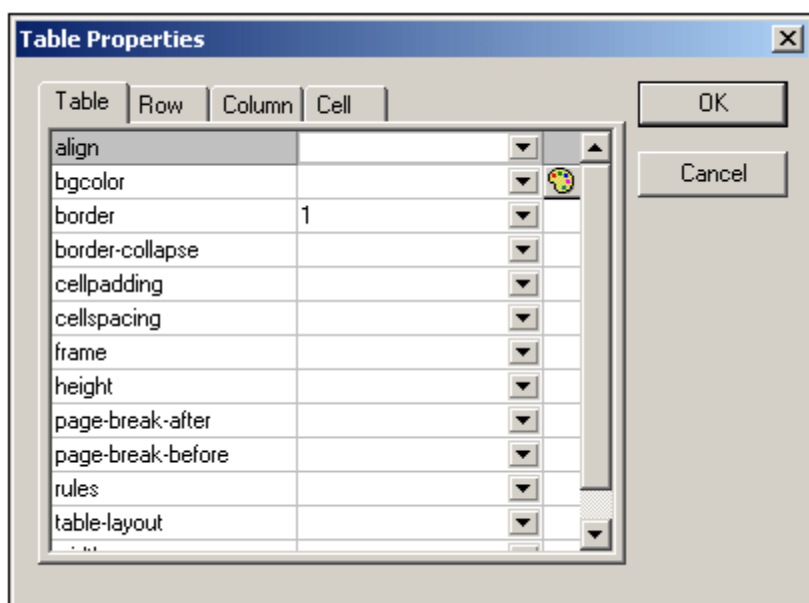
書式設定 テーブルとデータの挿入

テーブル書式設定はドキュメントデザインに既に割り当てられていますが、特定の環境ではテーブル書式設定を変更することができます。これらの環境は、以下のとおりです:

- 多種のテーブル構造要素に対応する要素は属性（基となるXMLスキーマ内の）として定義されている適切なCALSS、またはHTMLテーブルプロパティを持つ必要があります。定義されている属性のみが書式設定のために使用できます。デザイン内で、値がこれらの属性のために設定されている場合、Authentic View内でこれらの値をオーバーライドすることができます。
- デザイン内では、CSSスタイルを含むスタイル属性を設定してはなりません。CSSスタイルを持つスタイル属性が要素のために指定されている場合、スタイル属性その要素の他の書式設定の属性設定に対して優先順位が与えられます。この結果、Authentic View内で指定される書式設定は上書きされます。

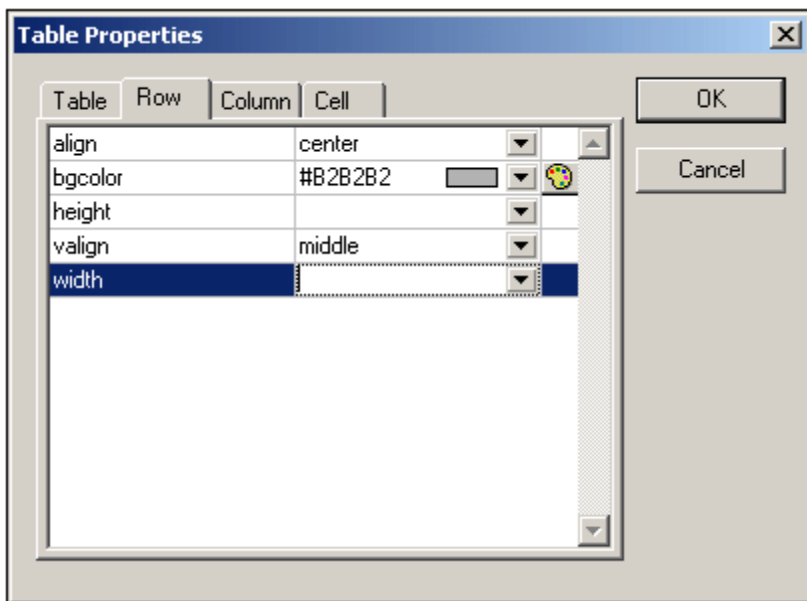
テーブル、行、列、またはセルの書式設定を行うには

1. テーブル  (テーブルプロパティ) アイコンをクリックを置きクリックします。テーブル、または行、列、またはセルのための書式設定を行うテーブルプロパティダイアログが開かれます (スクリーンショットを参照してください)。



2. セルスペースとセルインディングプロパティを「0」に設定します。テーブルは以下のようになります:


3. 最初の行にカーソルをポイントし、 (テーブルプロパティ) アイコンをクリックします。「行」タブをクリックします。




最初の行がヘッダ行になるため、この行を他の行と区別するために背景色を設定します。上の図でプロパティが設定されていることにご注意してください。列ヘッダテキストを挿入します。テーブルは以下のようになります。

Name	Telephone	Email

配置は指定されているとおり中央に揃えられています。

4. 「Telephone」列をサブ列「Office」と「Home」に分割する場合、「Telephone」列の幅を水平2つの列に分割する必要があります。最初にかしなから、サブヘッダ行を作成するためにヘッダセルの垂直に分割します。「Telephone」セル内にカーソルを置き、（垂直に分割）アイコンをクリックします。テーブルは以下のようになります。

Name	Telephone		Email

5. 「Telephone」を含むセルの下にセルカーソルを置き、（水平に分割）アイコンをクリックします。列ヘッダ「Office」と「Home」を入力します。テーブルは以下のようになります。

Name	Telephone		Email
	Office	Home	

「Telephone」列内の各セルの幅を水平に分割します。

テーブル編集アイコンを使用して、列と行を追加、削除、垂直に整列することもできます。CAL S/HTML テーブル編集アイコンは [CAL S/HTML テーブル編集アイコン](#) 内で説明されています。

テーブル内のセルを移動する

CAL S/HTML テーブル内でセル間を移動するには、「上」、「下」、「右」、「左」矢印キーを使用します。

セルにデータを挿入する

セルにデータを挿入するには、セル内にカーソルを置き、データを入力します。

テキストの書式設定

他のテキストXMLドキュメント内同様、CAL S/HTML テーブル内のテキストは、XML 要素、または、属性を使用して書式設定される必要があります。要素を追加するには、テキストをイライトし、要素入力レベル内の必要とされる要素をダブルクリックします。属性の値を指定するには、テキストフラグメント内にカーソルを置き、属性入力レベル内に必要とされる属性の値を挿入します。書式設定ヘッダーテキストを太字に設定すると、テーブルは以下のようになります。

Name	Telephone		Email
	Office	Home	

上のテキストはテキストのイライトと太字がフォントの書式として指定されているグローバルテンプレートである要素 `strong` のダブルクリックにより書式設定されています。

メモ Authentic View 内で表示されるテキストの書式に関しては、必要とされるテキストのフォーマットを持つグローバルテンプレートが要素のためにStyleVision 内で作成される必要があります。

5.3.3 CAL S/HTML テーブル編集アイコン

CAL S/HTML テーブルを編集するために必要なコマンドは、ツールバー内のアイコンとして使用することができ、下にリストされています。これらのアイコンのために対応するメニューコマンドは存在しません。

CAL S/HTML テーブルをいづつどのように使用するかについての詳細は [CAL S/HTML テーブル](#)。

テーブルの挿入



「テーブルの挿入」コマンドは現在のカーソルの場所にCAL S/HTML テーブルを挿入します。

テーブルの削除



「テーブルの削除」コマンドは現在アクティブなテーブルを削除します。

行の追加



「行の追加」コマンドは現在アクティブなテーブルの最後に行を追加します。

列の追加



「列の追加」コマンドは現在アクティブなテーブルの最後に列を追加します。

行の挿入



「行の挿入」コマンドは現在アクティブなテーブル内の現在のカーソルの場所の上に行を追加します。

列の追加



「列の追加」コマンドは現在アクティブなテーブル内の現在のカーソルの場所の左に列を挿入します。

セルを左側にジョインする



「セルを左側にジョインする」コマンドは、現在のセル(現在のカーソルの場所)を左側のセルとジョインします。両方のセルのタグは新規のセル内に留まり、列ヘッダーは変更されず、結合されません。

セルを右側にジョインする



「セルを右側にジョインする」コマンドは、現在のセル(現在のカーソルの場所)を右側のセルとジョインします。両方のセルのエントリは新規のセルに連結されます。

下のセルとジョインする



「下のセルとジョインする」コマンドは、現在のセル(現在のカーソルの場所)を下のセルとジョインします。両方のセルのエントリは新規のセルに連結されます。

上のセルとジョインする



「上のセルとジョインする」コマンドは、現在のセル(現在のカーソルの場所)を上側のセルとジョインします。両方のセルのエントリは新規のセルに連結されます。

セルを左右に分割する



「セルを左右に分割する」コマンドは、現在アクティブなセルの右に新規のセルを作成します。両方のセルのサイズは元のセルと同じサイズです。

セルを上下に分割する



「セルを上下に分割する」コマンドは、新規のセルを現在アクティブなセルの下に作成します。

上揃え



このコマンドはセルコンテンツをセルの上揃えに揃えます。

上下中央揃え



このコマンドは、セルコンテンツを上下中央に揃えます。

下揃え

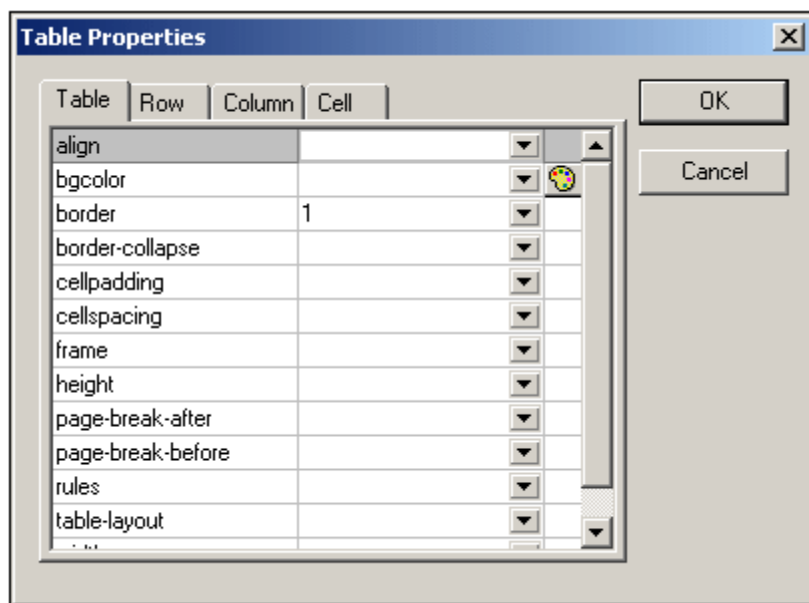


このコマンドはセルコンテンツをセルの下揃えに揃えます。

テーブルプロパティ



「テーブルプロパティ」コマンドは、テーブルプロパティダイアログボックスを開きます。このアイコンはHTML テーブルのためこのみ有効化することができますが、CALS テーブルのためクリックすることはできません。



5.4 DB の編集

Authentic View 内で、データベース(DB) テーブルを編集し、データを DB に戻すことができます。このセクションには、DB テーブルを編集する際に使用することのできるインターフェイスの説明が含まれています。次の一般的な点について注意してください！

- デザインをコンパクトにまとめるために、Authentic View 内で表示される DB テーブル内のレコードの数量は StyleVision Power Stylesheet のデザイナーにより意図的に制限されている場合があります。このような場合、Authentic View にロードされるレコードの数量を制限することができます。DB テーブル行 ナビゲーションアイコンを使用して DB テーブル内に他のレコードをロードすることができます (次を参照: [DB テーブルのナビゲート](#))。
- 特定のレコードを表示するために [DB をクエリ](#) することができます。
- DB レコードを追加、変更、削除し、DB にデータを戻すことができます。次を参照してください! [DB テーブルの変更](#)。

Authentic View 内で DB ベースの StyleVision Power Stylesheet を開くには、「Authentic | データベースデータの編集」をクリックして、必要とされる StyleVision Power Stylesheet を参照します。

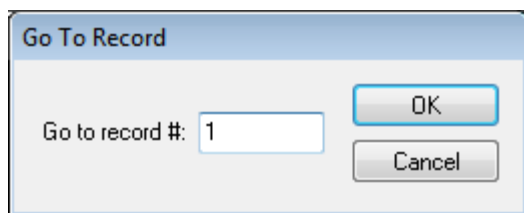
メモ Authentic View 内では SQLite データベースからのデータは編集することができません。Authentic View から SQLite データを保存しようと試みると、メッセージボックスにより既知の制限が通知されます。

5.4.1 DB テーブル内をナビゲートする

DB テーブル行をナビゲートするコマンドは、Authentic View ドキュメント内でボタンとして使用することができます。通常、4-5つのボタンを持つナビゲーションパネルが各 DB テーブルに含まれます。



矢印アイコンは、左側から右側、内の最初のレコードに移動する、前のレコードに移動する、レコードダイアログに移動する、次のレコードに移動する、最後のレコードに移動する、などがあります (スクリーンショットを参照してください)。



DB テーブルをナビゲートするには、必要なボタンを押してください。

XML データベース


IBM DB2 などの XML DB の場合、セル(または行)には、単一の XML ドキュメントが含まれており、このため、単一の行が Authentic View にロードされます。他の行内の XML ドキュメントをロードする場合は、[「Authentic | 編集のために XML データを持つ新規の行を選択する」](#) メニューコマンドを使用します。

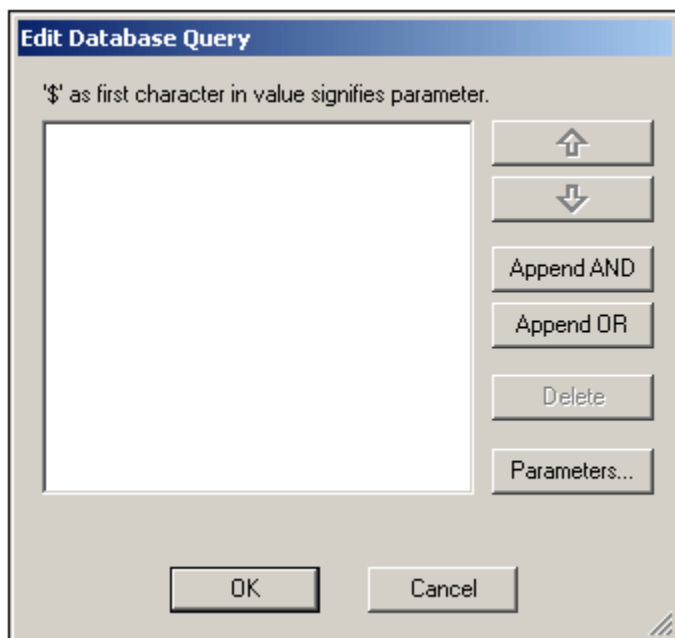
5.4.2 DB クエリ

DB クエリにより Authentic View 内に表示されるテーブルのレコードをクエリすることができます。クエリは個別のテーブルのために作成され、1つのテーブルにつき1つのクエリを作成することができます。クエリの送信時に Authentic View ドキュメント内の変更が保存されない場合、ドキュメント内の全ての変更を保存、または全ての変更を破棄するか問われます。他のテーブル内の変更も保存/破棄されます。クエリの送信後、テーブルはクエリの条件を使用して再ロードされます。

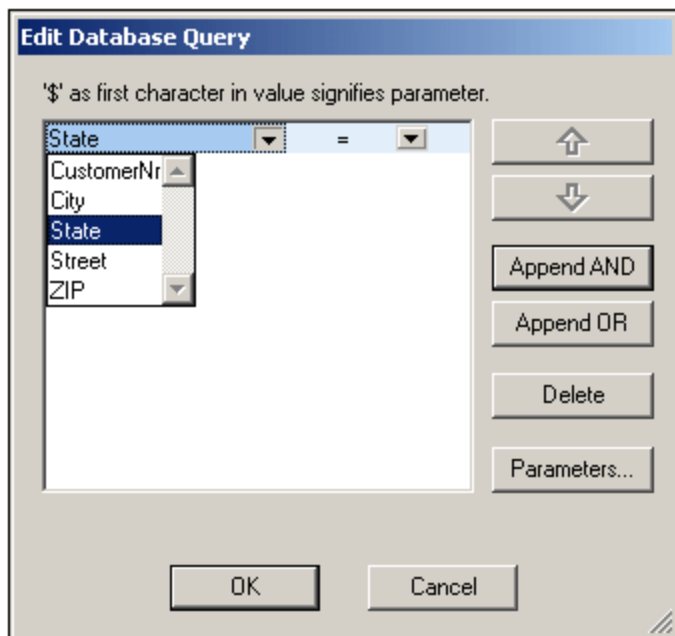
メモ 開かれているテーブルの数量が多すぎることを通知するメッセージが表示されると、開かれているテーブルの数量をクエリを使用してテーブル一部をフィルターし削減することができます。

クエリを作成して送信する方法

1. データベースクエリの変更 ダイアログを開くために必要とされるテーブルのためのクエリボタン  をクリックします (スクリーンショットを参照してください)。各 DB テーブルの上、または下このボタンが表示されます。クエリボタンがテーブルに存在しない場合、StyleVision Power Stylesheet のデザイナーはそのテーブルに対して DB クエリ機能を有効化していません。



2. 「AND を追加」、または「OR を追加」ボタンをクリックします。これにより (下に表示されるように) クエリのための空の条件が追加されます。



3. 条件のため式を入力します。式は以下により構成されています: (i) (関連付けられたコンボボックスで使用するこことできる) フィールド名、(ii) (関連付けられたコンボボックスで使用するこことできます) 演算子、(iii) (直接入力することこできる) 値。式の構築方法に關しては、[必要条件内の式](#) セクションを参照してください。
4. 他の必要条件を追加する場合、2つの条件をJOINするため使用するロジカルな演算子 (AND、または OR) に従い、「AND を追加」、または「OR を追加」ボタンをクリックします。新規の必要条件を追加します。ロジカルな演算子の詳細に關しては、[次を参照してください](#): セクション [DB クエリ内の必要条件を再度並べ替える](#)。

条件内の式

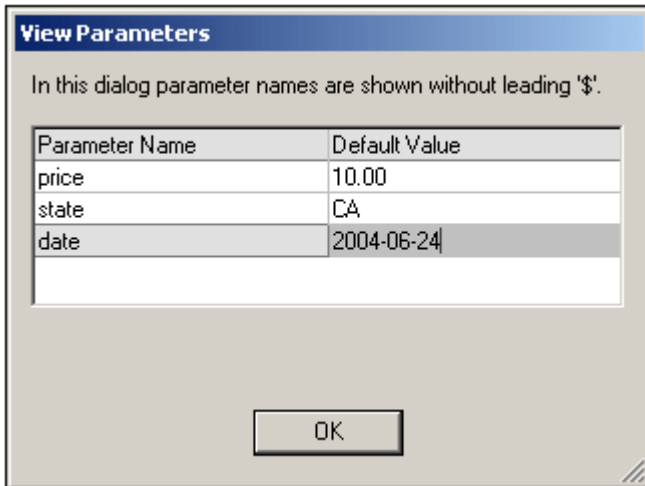
DB クエリの必要条件内の式はフィールド名、演算子、および値により構成されています。使用することこできるフィールド名は、選択されたトップレベルのデータテーブルの子要素です。これらのフィールドの名前は、コンボボックス内リストされます (上のスクリーンショットを参照してください)。「演算子」は下リストされています:

=	等しい記号
<>	等しくない記号
<	より小記号
<=	より小記号、または 等しい記号
>	より大記号
>=	より大記号、または 等しい記号
LIKE	発音が似ている
NOT LIKE	発音が似ていない
IS NULL	空である
NOT NULL	空ではない

IS NULL、または NOT NULL が選択される場合、値フィールドは無効化されます。値は引用符 (または、他の区切り文字) 無しに入力される必要があります。値は対応するDB フィールドと同じ書式設定を持つ必要があります。それ以外の場合、式は FALSE を評価します。例えば MS Access DB 内の date データ型のフィールドのための必要条件が式 StartDate=25/05/2004 を持つ場合、式は FALSE を結果として評価します。これは、MS Access DB 内の date データ型は書式 YYYY-MM-DD の書式を持つからです。

DB クエリを持つパラメータの使用

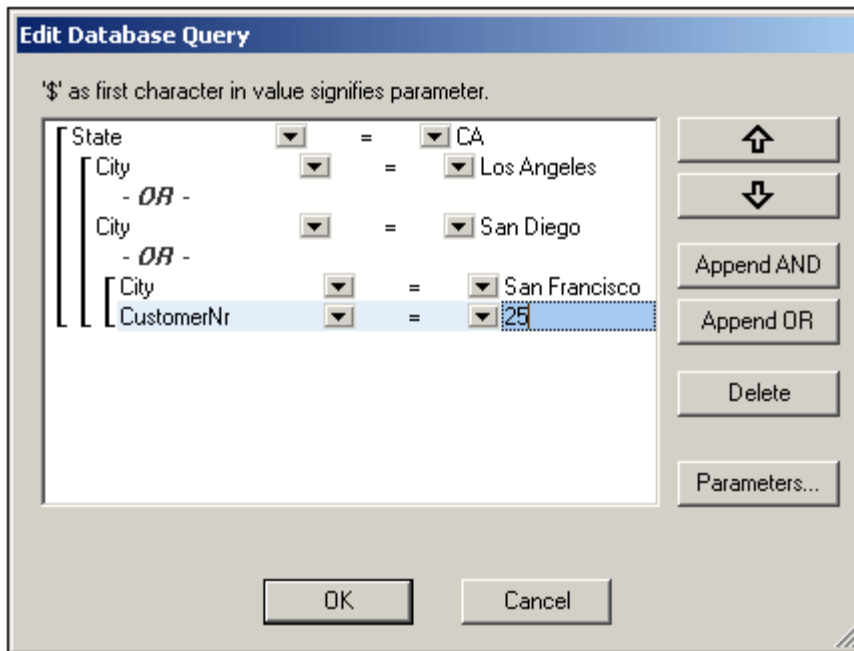
クエリを作成する際、式の値としてパラメータの名前を入力することができます。パラメータはクエリ内のリテラルの値の代わりに使用することができる変数です。式内に入力すると、式内で値が使用されます。使用することができるパラメータはSPS 内のSPS デザイナーにより定義されており、パラメータのビューダイアログ内で確認することができます（下のスクリーンショットを参照してください）。（出力ドキュメントがコマンドラインでコンパイルされる場合）コマンドラインからパラメータの値をバシオーバーライドすることができるSPS 内のデフォルトの値にパラメータが割り当てられます。SPS のために定義されたパラメータをビューするには、データベースクエリの変更 ダイアログ内のパラメータボタンをクリックします。これにより、パラメータのビューダイアログが開かれます（スクリーンショットを参照してください）。



SPS とパラメータ内でスタイルシートのために定義されているすべてのパラメータを含むパラメータのビューダイアログはスタイルシートのデザイン内で編集される必要があります。

DB クエリ内の必要条件を再度並べ替える

DB クエリのロジカルな構造と2つの必要条件、または、必要条件のセット間のルーションシップが視覚的に示されます。ロジカルな構造の各レベルは角かっこで示されています。2つの隣接する必要条件、または、必要条件のセットはAND 演算子を示し、2つの必要条件がOR で区切られている場合、演算子が示されます。必要条件はDB クエリのロジカルな構造のグラフィックな概要を与えることを目的としています。



上のスクリーンショットで表示されているDB クエリはテキストでは以下の用に表されています:

```
State=CA AND (City=Los Angeles OR City=San Diego OR (City=San Francisco AND CustomerNr=25))
```

必要条件を移動して、または、必要条件のセットをDB クエリ内の他の条件に相対して上下に移動して、DB クエリの順番を変更することができます。必要条件、または、必要条件のセットを移動するには、以下を行います:

1. クリックして必要条件を選択し、そのレベルを示すカッコをクリックして、レベル全体を選択します。
2. ダイアログ内の「上へ」、または「下へ」矢印ボタンをクリックします。


次の点について注意してください:

- 進行方向内の隣接する必要条件が同じレベルの場合、2つの条件は場所を交換することができます。
- 必要条件のセット(すなわち、カッコ内の必要条件)は同じレベル内で位置を変更することができますが、レベルは変更しません。
- 同じレベル内で個別の条件は位置を変更します。隣接する必要条件が更に外向き/内向き(すなわち、同じレベルではない)の場合、選択された必要条件は外向き/内向きに1度1つのレベル移動します。

DB クエリ内の条件を削除するには、条件を選択して「削除」をクリックします。

DB クエリの変更の方法



DB クエリの変更方法

1. クエリボタン  をクリックします。データベースクエリの変更 ダイアログボックスが開かれます。式の条件内の式を編集し、新規の条件を追加し、条件を並べ替え、DB クエリ内の条件を削除することができます。
2. 「OK」をクリックします。DB からのデータは、DB クエリへの変更が反映されると Authentic View に自動的に再ロードされます。

5.4.3 DB テーブルの変更

レコードを追加する方法

DB テーブルにレコードを追加する方法

1. (行を追加するため)DB テーブル行  アイコン内にカーソルを置き、または(行を入力するため)  アイコンをクリックします。これにより一時的な XML ファイル内に新規のレコードが作成されます。
2. 「ファイル|保存する」コマンドをクリックして、DB に新規のレコードを追加します。Authentic View 内では、新規のレコード内の行がDB テーブルに追加されます。このレコードのために AltovaRowStatus が A (追加済み) に設定されます。

新規のレコードのためのデータを入力すると、太字と下線と共に入力されます。これにより、既存のレコードとレコードを区別することができます。既存のレコードがテキストのフォーマットプロパティを使用して書式設定されていない場合、赤字で表示されているデータ型エラーがフラグされます。

「ファイル|保存する」をクリックして新規のレコードはDB に追加されます。新規のレコードがDB 保存されると、— により示された AltovaRowStatus フィールドが初期化され、Authentic View 内で通常のレコードとしてレコードが表示されます。

レコードを変更する方法

レコードを変更するには、必要に応じてDB テーブル内の必要とされる箇所カーソルを置き、レコードを編集します。表示されるレコードの数量は制限され、必要とされるレコードをナビゲートする必要がある可能性があります(次を参照してください! [DB テーブルのナビゲート](#))。

レコードを変更すると、レコードの全てのフィールド内のエンティリ下線が引かれ、このレコードのすべてのプライマリインスタンスの AltovaRowStatus は U (更新済み) に設定されます。AltovaRowStatus を持つこのレコードのすべてのセカンダリインスタンスは u (小文字) に設定されます。レコードのプライマリとセカンダリインスタンスはDB の構造と生成されたXML スキーマに従って定義されます。例えば、Address テーブルが Customer テーブルに含まれる場合、Address テーブルはインスタンス化の2つの型内のドキュメントのデザイン内で発生することができます。これは Address テーブル自身が Customer テーブルのインスタンス化内に存在するためです。これらの2つの型の片方が変更されると最初に変更される型となります。他の種類としては、1つ以上の他の型とセカンダリ型が存在する場合があります。データ型エラーは赤い色で表示されフラグされます。


「ファイル|保存する」をクリックして変更はDB に保存されます。変更されたレコードがDB に保存されると、— により示された AltovaRowStatus フィールドが初期化され、Authentic View 内で通常のレコードとしてレコードが表示されます。

以下の点に注意してください:

- Authentic View 内のレコードの単一のフィールドが変更され、データがDB に保存されると、レコード全体が更新されます。
- 日付値 0001-01-01 が一部のDB ために NULL 値として定義されます。エラーメッセージが発生する場合があります。

レコードを削除する方法

レコードの削除の方法

1. 削除するレコード内にカーソルを置いて、 アイコンをクリックします。削除されるレコードは取り消し線と共にマークされます。AltovaRowStatus は以下のとおりです: レコードのプライマリインスタンスは D に設定され、セカンダリインスタンスは d に設定され、間接的に削除されるレコードは X に設定されます。間接的に削除されるレコードは個別のテーブル内に保管され削除されたレコード内のフィールドです。例えば Address テーブルが Customer テーブルに含まれる場合があります。Customer レコードが削除される場合、対応する Address レコードが直接削除されます。Customer テーブル内の Address レコードが削除された場合、Customer テーブル名の Address レコードが主に削除されますが、インスタンス化されている場合、独立した Address テーブル内の同じレコードが順番に削除されます。
2. 「ファイル|保存する」をクリックして変更をDB に保存します。

メモ DB にデータを保存することは、元に戻すコマンドをヒットします。このため、保存前に行われたアクションを元に戻すことはできません。

5.5 日付と作業する

Authentic View 内で日付を編集するには2つの方法があります:

- 日付の選択を使用して日付が入力、または、変更されます。
- [値を入力して](#)日付が入力、または、変更されます。

Authentic View ユーザーが使用するメソッドはSPS 内で定義されています。両方のメソッドはこのセクションの2つのサブセクションで説明されています。

日付の書式に関するメモ

XMLドキュメント内では、日付は複数の日付データ型の1つに保管することができます。これらのデータ型のそれぞれは、XMLドキュメントが有効であるために日付は特定の文法の書式内で保管される必要があります。例えば、`xs:date` データ型は `YYYY-MM-DD` の書式を必要とします。`xs:date` オブジェクト内の日付がこの書式以外の書式で入力されると、XMLドキュメントは無効になります。

正確な書式で日付が入力されるように、SPS デザイナーは視覚的な日付の選択をデザイン内に含むことができます。これにより、日付の選択で選択された日付が、正確な文法の書式であることを確認できます。日付の選択が存在しない場合、Authentic View は正確な文法の書式で日付の挿入を行います。XMLドキュメントの検証により必要とされる文法の書式に関するヒントを得ることができます。

5.5.1 日付の選択

XMLドキュメントに標準の書式で日付を挿入するために視覚的なカレンダーである日付の選択が使用されます。ドキュメント内のデータの処理のために標準の書式は重要です。日付の選択 アイコンが変更された日付フィールドの近くに表示されます(スクリーンショットを参照してください)。



日付の選択を表示するには、日付の選択アイコンをクリックしてください(スクリーンショットを参照してください)。

日付を選択するときは、希望する日付、付き、および、年度を選択します。XMLドキュメントに日付が挿入され、表示される日付は必要に応じて変更されます。必要な場合は、タイムゾーンも入力してください。

5.5.2 テキストのエントリ

日付の選択が存在しない日付フィールドの場合、新規の値を直接入力することにより日付を編集することができます（スクリーンショットを参照してください）。

エラー

以下の種類のエラーがフラグされます:

- 日付を編集し、日付が有効な日付の範囲外になった場合、日付は赤い色で表示されエラーを通知します。マウスのカーソルを無効な日付ポイントすると、エラーメッセージが表示されます（スクリーンショットを参照してください）。

- 日付の書式を変更しようとする時、日付は赤い色で表示されエラーを通知します（スクリーンショットを参照してください。ハイフンの代わりにスラッシュが使用されています）。

5.6 エンティティを定義する

エンティティに関して

ドキュメントが DTD、または XML スキーマをベースとしているかに関わらず、Authentic View 内で使用するためエンティティを定義することができます。定義後、これらのエンティティはエンティティの入力ヘルパー内に表示され、コンテキストメニューの「エンティティの挿入」サブメニュー内に表示されます。エンティティの入力ヘルパー内のエンティティをダブルクリックすると、カーソル挿入ポイントにエンティティが挿入されます。

テキスト文字列、XML フラグメント、または他の外部リソースをドキュメント内の複数の場所で使用する場合、エンティティが役立ちます。必要とされるデータのための短い名前であるエンティティを定義するには、エンティティの定義ダイアログを使用します。エンティティの定義後、ドキュメント内の複数の場所で使用することができます。これにより時間を節約し、効率よくメンテナンスすることができます。

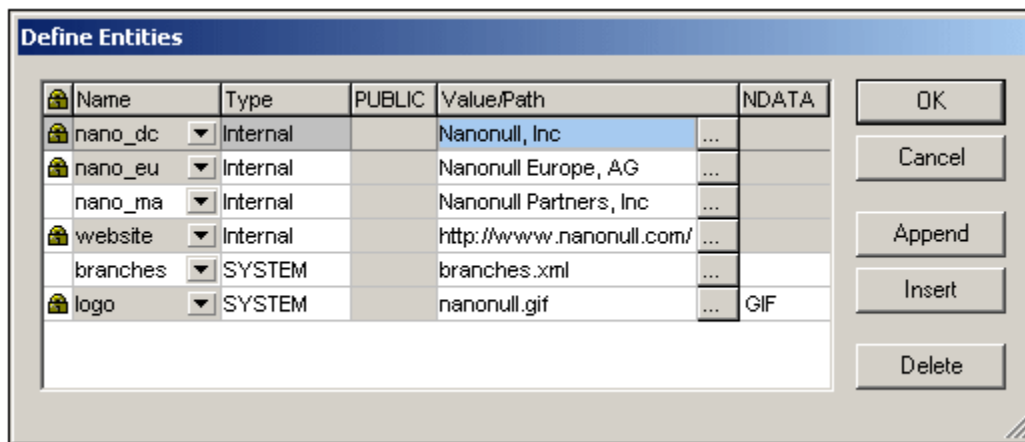
エンティティの型

ドキュメント内で使用することのできるエンティティは 2 つの広い型が存在します: XML データ (テキスト文字列、または XML ドキュメントのフラグメント) である「解析済みのエンティティ」、または非-XML データ (グラフィック、サウンド、またはマルチメディアオブジェクトなどのバイナリファイル) である「解析されていないエンティティ」。各エンティティには名前と値が存在します。解析されたエンティティの場合、エンティティは XML データのためのプレースホルダです。エンティティの値は XML データ自身、または、データを含む .xml ファイルをポイントする URI です。解析されていないエンティティの場合、エンティティの値は非-XML データ ファイルをポイントする URI です。

エンティティの定義

エンティティを定義する方法

1. 「Authentic | XML エンティティの定義...」をクリックすると、エンティティの定義ダイアログが開かれます (下のスクリーンショット)。



2. 名前フィールドにエンティティ名を入力します。これは、エンティティの入力ヘルパーに表示される名前です。
3. エンティティの型を型フィールドのドロップダウンリストから入力します。次の型を使用することができます: テキストが使用する **Internal** エンティティが XML ドキュメント自身内で保管されています。 **PUBLIC** または **SYSTEM** の選択はリソースが XML ファイルの外部にあることを指定し、公開識別子またはシステム識別子を使用してロケートされます。システム識別子はリソースの場所を与える URI です。公開識別子はプロセッサがリソースを識別するためのロケーションに影響されない識別子です。公開とシステム識別子の両方を指定すると、公開識別子はシステム識別子、システム識別子を使用して解決します。
4. **PUBLIC** が Type として選択されている場合、**PUBLIC** フィールド内にリソースの公開識別子を入力します。 **Internal**、または、型として **SYSTEM** を選択すると、**PUBLIC** フィールドは無効化されます。
5. Value/Path フィールド内に、次を入力することができます:

- エンティティの型がInternal の場合、エンティティの値として使用するテキスト文字列を入力します。エントリ引用符を入力しないでください。入力する引用符はテキスト文字列の一部として扱われます。
 - SYSTEM がエンティティの型である場合、参照ボタンの使用してリソースのURI、または、ローカルネットワーク上のリソースを入力します。解析されたデータがリソースに含まれている場合、XML ファイルである必要があります（すなわち、ファイルは.xml 拡張子を持つ必要があります）。更に、リソースはGIF ファイルなどのバイナリファイルであることができます。
 - エンティティの型がPUBLIC の場合、このフィールドにシステム識別子を追加で入力する必要があります。
6. NDATA エントリはプロセッサがこのエンティティを解析せず適切なプロセッサに送信されるように命令します。NDATA フィールドはエンティティが解析されていないエンティティを示す値を含む必要があります。

ダイアログの機能

エンティティの定義 ダイアログ内で以下の機能を使用することができます:

- エンティティの追加
- エンティティの挿入
- エンティティの削除
- 列ヘッダーをクリックしてエンティティをアルファベット順の値に並べ替えます。1度クリックすると昇順に並べ替え、2度クリックすると降順に並べ替えられます。
- ダイアログボックスと列の幅のサイズを調整します。
- ロック。XML ドキュメント内でエンティティが使用されると、このエンティティはロックされ、エンティティの定義 ダイアログ内で編集することはできません。ロックされたエンティティは、最初の列内でロックシンボルにより表示されます。エンティティをロックすることにより、エンティティに対してXML ドキュメントが有効であることを保証することができます（エンティティが参照されており、定義されていない場合は、ドキュメントは無効になります）。
- エンティティの複製はフラグされます。

エンティティの制限

- 他のエンティティ内に含まれるエンティティはダイアログ Authentic View、またはXSLT 出力内で解決されません。このようなエンティティのアンサード文字は&などのエスケープ文字と共に表示されます。
- イメージファイルまたは外部の解析されていないエンティティは Authentic View 内で解決されません。デザイン内のイメージは外部の解析されていないエンティティを読み取るように定義され、URI がエンティティ名（例: 'logo'）になるように設定されています。エンティティの定義 ダイアログ内でエンティティ名を（上のスクリーンショット内のロゴエンティティのために設定されているように）イメージファイルのURI に対して解決する値を持つ外部の解析されていないエンティティとして定義することができます（上のスクリーンショットを参照してください）。


5.7 XML 署名

Authentic View のために構成された XML 署名を使用して SPS をデザインすることができます。XML 署名が SPS 内で有効化された場合、Authentic View ユーザーは有効化された署名を使用して Authentic XML ファイルを電子的に署名することができます。ドキュメントの署名後、変更を加えると署名の検証は失敗します。署名された Authentic XML ドキュメントが Altova 製品内の Authentic View で開かれると、ドキュメントの検証プロセスが実行され、検証の結果はウィンドウ内に表示されます。

メモ 次の Altova 製品の Enterprise と Professional の Authentic View 内で XML 署名を使用し、検証することができます: Authentic Desktop、Authentic Browser、XMLSpy、と StyleVision。

XML 署名アクション

次の署名のための Authentic View ユーザー アクションを使用することができます:

- **証明書/パスワードの選択:** 署名は、証明書、または、パスワードで認証されます。署名が作成され、検証される際に認証オブジェクト (証明書、または、パスワード) が必要とされます。Authentic XML ドキュメントが署名が有効化されている SPS を割り当てられているかにより、デフォルトの証明書、または、署名のパスワードを指定する場合があります。デフォルトの証明書、または、パスワードが指定されているかにより、Authentic View ユーザーが自身の証明書とパスワードを選択できるかを決定する署名を構成することができます。XML 署名ダイアログ内でこの操作を行うことができます (下のスクリーンショット)。自身の証明書及びパスワードの選択はデフォルトの証明書とパスワードをオーバーライドします。自身の証明書とパスワードはメモリ内に保管され、現在のセッションで使用することができます。自身の証明書とパスワードの選択の後、ファイルまたはアプリケーションを閉じます。SPS は証明書とパスワードのためのデフォルトの設定に戻ります。
- **ドキュメントの署名:** Authentic XML ドキュメントを自動的に、または、手動で署名することができます。自動署名は、SPS デザイナーにより署名の構成内で指定することができます。保存されると Authentic XML ドキュメントが自動的に署名されます。自動署名が有効化されていない場合、ドキュメントを手動で署名することができます。XML 署名 ツールバーアイコン  すべての、または、「Authentic | XML 署名 コマンドをクリックすると、XML 署名 ダイアログがポップアップされるので、(上のスクリーンショット) ドキュメントの署名 ボタンをクリックします。埋め込まれた署名を使用したドキュメントの署名はスキーマコレクト (ドキュメント) 要素の最後の子要素として署名要素を許可するように要求します。ドキュメントを署名する場合、認証オブジェクトと署名の配置は署名の構成に従って決定されます。認証情報にアクセスできることを確認してください。詳細に関しては、SPS デザイナーに確認してください。
- **Authentic XML ドキュメントの検証:** SPS のために XML 署名が有効化されていると、検証プロセスは Authentic View XML ドキュメントがロードされる都度、各署名に対して行われます。パスワード、または、証明書 キー情報が SPS と署名と保存されていない場合、Authentic View ユーザーはパスワードを入力、および、検証のための証明書を選択するようにプロンプトされます。XML ファイルと共に保存しない場合、埋め込まれた署名が生成されると、XML ファイルが保存される際に XML ファイルと共に保存されます。(XML 署名ダイアログの「署名を削除」ボタンを使用して生成された署名は明示的に削除される必要はありません。同様に、detachされた署名が生成されると、必要であれば内場合、明示的に削除されます。

5.8 Authentic View 内のイメージ

Authentic View を使用すると、最終出力ドキュメント (HTML、RTF、PDF および Word 2007) 内で使用することのできるイメージを指定することができます。イメージの書式の一部は一部の書式、またはアプリケーションでサポートされない場合があることに注意してください。例えば SVG 書式は PDF 内でサポートされていますが、RTF ではサポートされておらず、HTML 内で表示するためにはブラウザアドオンが必要になります。イメージ書式を選択する場合、ドキュメントの出力書式でサポートされている書式を選択してください。イメージの書式の多数は、多くの出力書式でサポートされています (下のリストを参照してください)。

Authentic View は Internet Explorer をベースとしており、Internet Explorer が表示することのできるイメージの書式の大部分を表示することができます。次の一般的なイメージの書式がサポートされています:

- GIF
- JPG
- PNG
- BMP
- WMF (Microsoft ウィンドウメタファイル)
- EMF (拡張されたメタファイル)
- SVG (PDF 出力のためのみ)

相対的なパス

相対的なパスは SPS ファイルに相対して解決されます。

5.9 Authentic View 内のキーストロール

Enter キー

Authentic View 内では、「Enter」キーは、特定のカーソルの場所で追加の要素を追加する際に使用されます。例えば（スキーマに従い）本のチャプターには複数の段落が含まれており、段落のテキスト内で「Enter」を押すと、新規の段落が現在の段落の後に追加されます。チャプターがタイトルと複数の段落を含むことができる場合（タイトル要素を含む）、段落要素外のチャプター内で「Enter」を押すと（スキーマにより複数のチャプターが許可されていると想定し）現在のチャプターの後に新規のチャプターが追加されます。

メモ「Enter」キーは新規のラインを入力しません。これは、段落などのテキストノード内にカーソルがある場合でも同じです。

キーボードの使用

入力とナビゲートのためにキーボードを通常の方法で使用することができます。

- 「タブ」キーはカーソルを前に進め、ノードの前後で停止し、ノードコンテンツをハイライトし、静的なコンテンツをステップオーバーし、静的なコンテンツをステップオーバーします。
- `add...` と `add Node` ハイパーリンクは、ノードコンテンツとして考えられ、タブ付けされるとハイライトされます。スペースキーまたは「Enter」キーを押すと有効化することができます。

6 Authentic スクリプト

Authentic スクリプト機能により、SPS デザインに対して更なる柔軟性と双方向性を追加することができます。これらデザインは StyleVision Enterprise ならびに Professional Edition で作成または編集することができ、Altova 製品の Enterprise ならびに Professional Edition に搭載された Authentic ビューにて閲覧することができます。

以下のテーブルでは Altova 製品ごとサポートされる機能のリストを示します。Authentic Browser Plugin Trusted バージョンはセキュリティ上の観点から内部スクリプトが無効になっていることにご注意してください。

Altova 製品	Authentic スクリプトの作成	Authentic スクリプトの有効化
StyleVision Enterprise	はい	はい
StyleVision Professional	はい	はい
StyleVision Basic *	いいえ	いいえ
XMLSpy Enterprise	いいえ	はい
XMLSpy Professional	いいえ	はい
AuthenticDesktop Enterprise	いいえ	はい
Authentic Browser Plug-in Enterprise Trusted **	いいえ	はい
Authentic Browser Plug-in Enterprise Untrusted	いいえ	はい

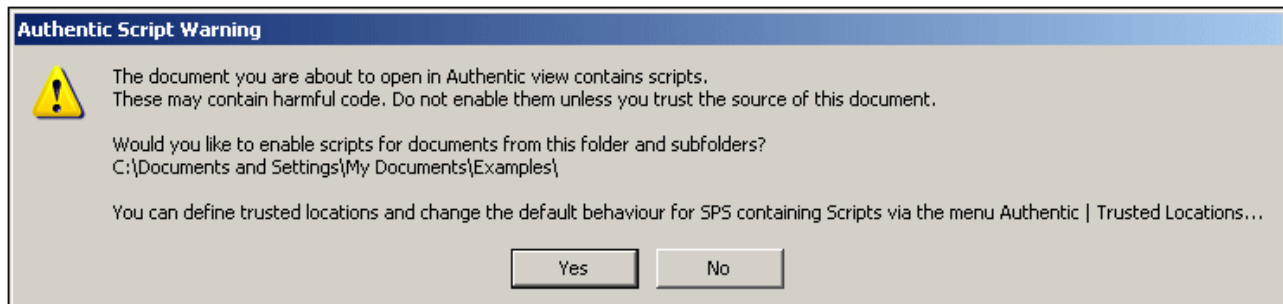
* Authentic ビュー無し

** スクリプトありのデザインは表示可能。内部マクロの実行やイベントハンドリングは無し。外部イベントの実行。

Authentic スクリプトの動作は全 Altova 製品において同一のものとなり、ある製品に特化したコードや設定は必要ありません。

Authentic スクリプト 警告ダイアログ

PXF ファイルまたは SPS へリンクされた XML ファイルにスクリプトが含まれており、そのファイルが開かれた状態で Authentic ビューへの切り替えが行われると、警告ダイアログが表示されます(以下のスクリーンショットを参照)。



以下のオプションから選択を行うことができます:

- はいをクリックすると、ファイルを含んでいるフォルダーが Authentic スクリプトの信頼された場所リストへ追加されます。今後、信頼されたフォルダー内にあるファイルを開いても、警告ダイアログは表示されなくなります。信頼された場所のリストへは [Authentic | 信頼された場所](#) メニューコマンドからアクセスならびに編集することができます。

- いまをクリックすることで、ファイルが含まれているフォルダーを信頼された場所のリストへ追加することなく処理が継続されます。ファイルはスクリプトが無効になった状態で開かれます。ファイルを Authentic ビューで開くたびに Authentic スクリプト 警告ウインドウが表示されます。ファイルが含まれているフォルダーを信頼された場所のリストへ追加するには「[Authentic | 信頼された場所](#)」メニューコマンドから信頼された場所のダイアログを開き、フォルダーを追加するたびに必要に応じて変更することができます。

信頼された場所ダイアログに関する詳細については、ユーザーリファレンスの「[Authentic | 信頼された場所](#)」メニューコマンドのセクションを参照ください。

メモ: COM インターフェイスを介した XMLSpy へのアクセスを行う場合、セキュリティのチェックは行われず、「Authentic スクリプト 警告」ダイアログは表示されません (COM インターフェイスに関する情報は [プログラマーのリファレンス](#) を参照ください)。

Authentic スクリプトの使用法

SPS デザインを作成したデザイナーは以下にある2つの方法により、Authentic ドキュメントをインタラクティブなものにすることができます:

- スクリプトをユーザー定義のアクション (マクロ) に割り当てることで、要素、ツールバーボタン、そしてコンテキストメニューのアイテムをデザイン
- Authentic ビューイベントに反応するデザインイベントハンドラーを追加

Authentic ドキュメントをインタラクティブなものにするために必要な全てのスクリプトは、StyleVision GUI (Enterprise ならびに Professional Edition) 内で作成することができます。フォーム、マクロ、イベントハンドラーは StyleVision に搭載されたスクリプトエディター内で作成され、これらスクリプトは SPS 内に保存されます。その後、StyleVision のデザインビューにて、保存されたスクリプトは要素、ツールバーボタン、コンテキストメニューに割り当てられることとなります。SPS をベースとした XML ドキュメントが Authentic スクリプトをサポートする Altova 製品 (上のテーブルを参照) により開かれると、ドキュメントには様々な柔軟性や双方向性が追加されます。

Authentic スクリプトのドキュメンテーション

Authentic スクリプトに関する情報は StyleVision のドキュメンテーションでご利用いただけます。[Altova ウェブサイトの製品ドキュメンテーションページ](#)から参照することができます。

7 ブラウザービュー

ブラウザービューは、通常以下を閲覧するために使用されます:

- 関連したXSLT ファイルをもつXML ファイル。ブラウザービューに切り替えると、XML ファイルは、関連したXSLT スタイルシートを使用して素早く変換され、結果は直接ブラウザービューに表示されます。
- HTML ファイルは、HTML として直接作成されるか、またはXML ファイルのXSLT 変換を介して作成されます。

ブラウザービュー内のXML およびHTML ファイルを閲覧するには、ブラウザー タブをクリックします。

Microsoft Internet Explorer およびXSLT に関する注意点

ブラウザービューは、ま Microsoft Internet Explorer 5.0 以下以降を必要とします。XSLT スタイルシートにより変換されたXML ファイルを閲覧するためにブラウザービューを使用する場合は、XSLT 1.0 標準を完全にサポートするXML パーサーであるMSXML 3.0 をサポートするInternet Explorer 6.0 以下以降を使用することが奨励されます。MSXML 4.0. のインストールを行う場合は、(IE 5 内のXSLT へのサポートは、公式のXSLT 勧告と100% 互換性があるわけではありません。)IE 5 を使用して、ブラウザービュー内で問題が発生した場合、IE 6 以下以降にアップグレードしてください。使用中のInternet Explorer のバージョンのXSLT に対するサポートを確認してください。

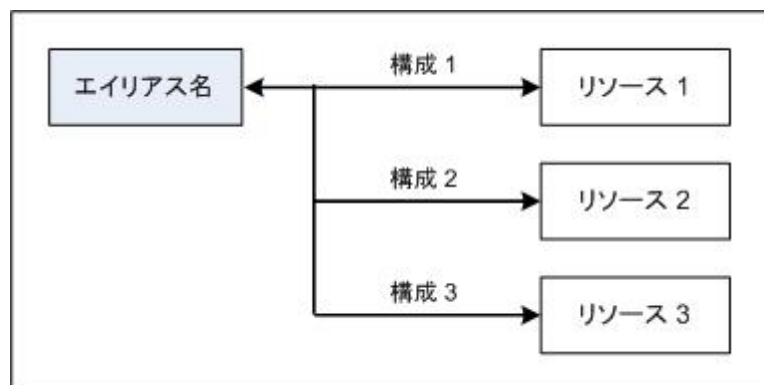
ブラウザービュー機能

ブラウザービューで以下の機能を使用することができます。[ブラウザー](#) メニュー、[ファイル](#) メニュー、および[編集](#) を使用してアクセスすることができます。

- **個別のウィンドウで開く:** ブラウザービューが個別のウィンドウである場合、同じドキュメントの編集ビューで横に並べて表示することができます。これを行うにはメニューコマンド [\[ブラウザー | 個別のウィンドウ\]](#) をクリックします。これは2つのウィンドウ間のブラウザービューを切り替えるグループです: (i) 個別のウィンドウと(ii) メインウィンドウ内のタブ付けだけビュー。これらのコマンドは(メインウィンドウの下の [ブラウザービュー](#) ボタンのドロップダウンメニュー)で使用することができます。
- **「前へ」と「戻る」:** コマンド [ブラウザー](#) コマンド は、ブラウザービュー内でロードされたページをナビゲートするために使用されます。これらのコマンドは[ブラウザー](#) メニュー内にあります。
- **フォントのサイズ:** [ブラウザー](#) メニューを介して、調節することができます。
- **停止、更新、印刷:** 標準 [ブラウザー](#) コマンドは、[ブラウザー](#) と[ファイル](#) メニュー内にあります。
- **検索:** テキスト文字列のために検索を有効化します。これらのコマンドは[編集](#) メニュー内にあります。

8 Altova グローバルリソース

Altova グローバルリソースはファイル、フォルダー、そしてデータベースリソースのエイリアス集合のことです。各エイリアスは複数の構成を持ち、各構成を1つのリソースにマッピングすることができます（下のスクリーンショット参照）。そのため、グローバルリソースを入力として使用することで、グローバルリソースを構成とともに切り替えることができます。この操作は、アクティブな構成を選択することのできる GUI 上で簡単にを行うことができます。



Altova グローバルリソースの使用には、2つのステップがあります：

- [グローバルリソースの定義](#)：XML ファイルによりリソースが定義されます。これらのリソースは複数の Altova 製品間で共有することができます。
- [グローバルリソースの使用](#)：Authentic Desktop 内部ではファイルパスの代わりにグローバルリソースによりファイルを指定することができます。グローバルリソースを使用すると、Authentic Desktop にてアクティブな構成を切り替えることで、使用するリソースを簡単に変更することができるようになります。

その他の Altova 製品におけるグローバルリソース

現在のところ、以下の Altova 製品でグローバルリソースを使用することができます：XMLSpy、StyleVision、MapForce、Authentic Desktop、MobileTogether Designer および DatabaseSpy。

8.1 グローバル リソースの定義

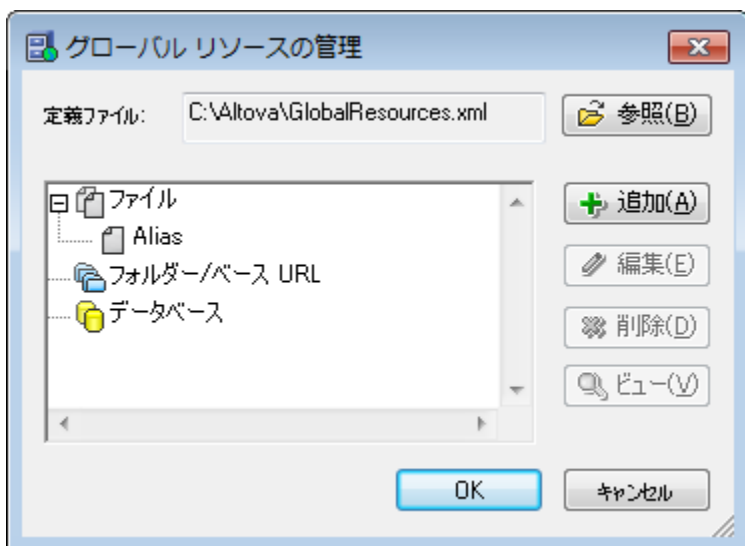
Altova グローバルリソースは、以下の方法によりアクセスすることのできるグローバルリソースの管理ダイアログで定義されます:

- メニューコマンドの「ツール | グローバルリソース」を選択する。
- グローバルリソース ツールバーにある「グローバルリソースの管理」アイコンをクリックする(下のスクリーンショット)。



グローバルリソース定義ファイル

グローバルリソースに関する情報はグローバルリソース定義ファイルと呼ばれる XML ファイルに収められています。このファイルは、グローバルリソースの管理ダイアログにて初めてグローバルリソースが定義ならびに保存された際に作成されます(下のスクリーンショットを参照)。



グローバルリソースの管理ダイアログを初めて開くと、グローバルリソース定義ファイルのデフォルトの場所と名前が、定義ファイルテキストボックスにて指定されます(上のスクリーンショットを参照):

```
C:\Users\\My Documents\Altova\GlobalResources.xml
```

このファイルは、すべての Altova アプリケーションに対するデフォルトのグローバルリソース定義ファイルとしてセットされます。従って、任意の Altova アプリケーションからこのファイルに対して保存されたグローバルリソースは、その他すべての Altova アプリケーションにて直ちグローバルリソースとして使用することができることとなります。グローバルリソースを定義して、グローバルリソース定義ファイルに保存するときは、グローバルリソースの管理ダイアログにてグローバルリソースを追加し、「OK」ボタンをクリックしてください。

既に存在するグローバルリソース定義ファイルを、実行中の Altova アプリケーションにおけるアクティブな定義ファイルとして選択するには、定義ファイルテキストボックスの隣にある「参照」ボタンから、そのファイルを選択してください。(上のスクリーンショットを参照)

メモ: グローバルリソース定義ファイルの名前は自由に変更することができ、Altova アプリケーションからアクセスできる任意の場所へ配置することができます。この場合、各アプリケーションにて(定義ファイルテキストボックスから)このファイルをアプリケーションのグローバルリソース定義ファイルとして指定する必要があります。複数の Altova 製品にて同一の定義ファイルを指定することで、これらの Altova 製品にてリソースが利用できるようになります。

メモ: 複数のグローバルリソース定義ファイルを作成することもできます。この場合、各 Altova アプリケーションにて同時に選択できるアクティブな定義ファイルは1つだけで、そのファイルの中に含まれている定義はアプリケーションでは利用できないことに注意してください。そのため、必要に応じて同じリソースを複数の定義ファイルで記述する必要が生じることもあります。

グローバルリソースの管理: 追加、編集、削除、保存

グローバルリソースの管理ダイアログでは、選択されたグローバルリソース定義ファイルへグローバルリソースを追加することができるほか、選択されたグローバルリソースを編集または削除することができます。ファイル、フォルダー、そしてデータベースのグループを追加されたグローバルリソースは、グローバルリソース定義ファイルにより管理されます(上のスクリーンショットを参照)。

グローバルリソースを追加するには、「追加」ボタンをクリックして、表示される「グローバルリソース」ダイアログにてグローバルリソースを定義します(このセクション以下のサブセクションにある「[ファイル](#)」、「[フォルダー](#)」、「[データベース](#)」の記述を参照ください)。グローバルリソースを定義して(グローバルリソースの管理ダイアログにある「OK」ボタンをクリックすることで)保存すると、選択されたグローバルリソース定義ファイルのグローバルリソース定義ライブラリに、そのグローバルリソースが追加されます。グローバルリソースはエイリアス名で識別されます。

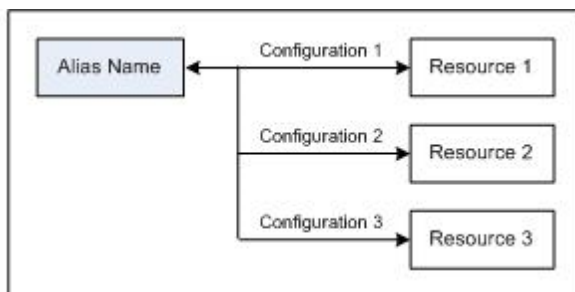
グローバルリソースを編集するには、「編集」ボタンをクリックします。関連するグローバルリソースダイアログが表示され、必要な変更を行うことができます(「[ファイル](#)」、「[フォルダー](#)」、「[データベース](#)」サブセクションの記述を参照ください)。

グローバルリソースを削除するには、「削除」ボタンをクリックします。

グローバルリソースの追加、編集、または削除を終えた後は、グローバルリソースの管理ダイアログにある「OK」ボタンをクリックして、グローバルリソース定義ファイルに変更点を保存するようにしてください。

構成によりグローバルリソースをエイリアス名へ関連付ける

グローバルリソースを定義すると、エイリアス名をリソース(ファイル、フォルダー、データベース)へマッピングすることになります。1つのエイリアス名から複数のリソースへのマッピングを行うことができ、各マッピングは構成と呼ばれます。そのため、単一のエイリアス名から、複数の構成を介して複数のリソースへの関連付けを行うことができます(以下の図を参照)。



Altova アプリケーションではファイルの代わりにエイリアスを割り当てることができます。アプリケーションのアクティブなグローバルリソース構成を切り替えることで、各エイリアスに割り当てられたリソースを切り替えることができます。例えば Altova XMLSpy アプリケーションにて、MyXML.xml という名前の XML ドキュメントに対して XSLT 変換を行う場合、MyXSLT という名前のエイリアスを使用される XSLT としてグローバルリソースに割り当てることができます。XMLSpy では、アクティブな構成を変更することで、異なる XSLT ファイルを使用することができます。構成-1 にて First.xslt が MyXSLT へマッピングされており、構成-1 がアクティブな構成として選択されている場合、First.xslt が変換に使用されます。これにより、複数の構成を使用することで、単一のエイリアスから複数のリソースへアクセスできるようになります。このメカニズムはリソースのテストや比較を行う際に便利な機能となります。更に、グローバルリソースは複数の Altova 製品間で使用することができるため、複数の Altova 製品にてリソースのテストや比較を行うことができます。

8.1.1 ファイル

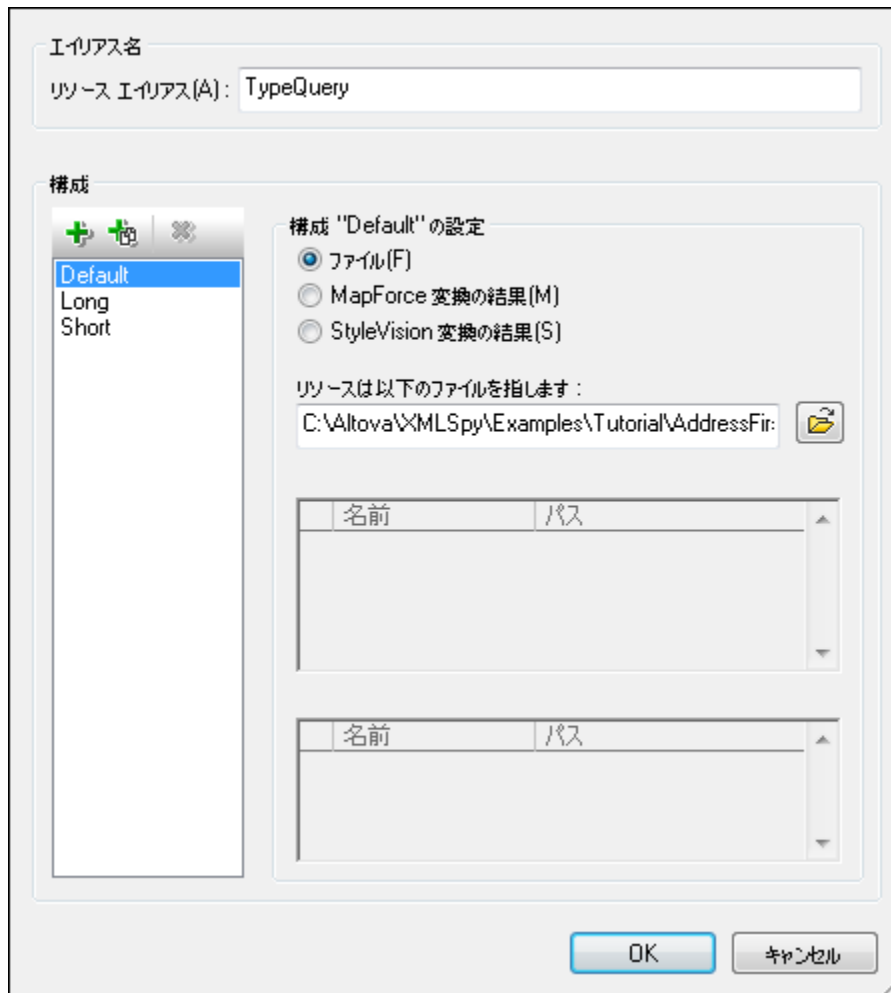
「[グローバルリソースの管理](#)」ダイアログにある「追加 | ファイル」コマンドにより、ファイルのグローバルリソースダイアログへアクセスすることができます(以下のスクリーンショットを参照)。グローバルリソースダイアログでは、リソースエイリアステキストボックスにて入力されたエイリアス

の構成を定義することができます。以下に記される方法により構成のプロパティを指定して、「OK」をクリックしてエイリアスの定義を保存してください。




エイリアスの定義を保存した後は上記の操作を（「[グローバルリソースの管理](#)」ダイアログにある「追加 | ファイル」コマンドを選択するところから）再度行うことで、別のエイリアスを追加することができます。

グローバルリソース ダイアログ

エイリアスはグローバルリソースダイアログにて定義されます（以下のスクリーンショットを参照）。



グローバルリソース ダイアログのアイコン

-  構成の追加: 追加する構成の名前を入力するための、構成の追加ダイアログが表示されます。
-  構成をコピーして追加: 選択された構成のコピーとして追加する構成の名前を入力するための、構成の追加ダイアログが表示されます。
-  削除: 選択された構成を削除します。

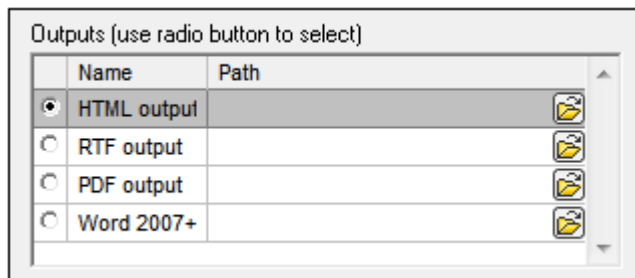



開く: グローバルリソースとして作成されるファイルを参照します。

エイリアスの定義

以下の操作により、エイリアスの名前と構成を定義することができます。

1. **エイリアスに名前を付ける** リソースエイリアステキスト ボックスにエイリアスの名前を入力します。
2. **構成を追加する** 構成ペインには、デフォルトで(名前変更や削除を行うことができない) Default という名前の構成が表示されます。(i) 構成の追加または構成のコピーを追加アイコンをクリックして、(ii) 表示されるダイアログにて構成名を入力することで、必要な数の構成を追加することができます。追加された各構成は構成リストに表示されます。上のスクリーンショットでは Long と Short という名前の構成が構成リストに追加されています。構成のコピーを追加コマンドにより、選択されている構成をコピーして、修正することもできます。
3. **各構成に対してリソースの種類を選択する** 構成のリストから目的の構成を選択し、構成の設定ペインにて、(i) ファイル、(ii) MapForce 変換の結果、または (iii) StyleVision 変換の結果から、構成のリソースを指定します。対応するラジオボタンを選択してください。MapForce または StyleVision 変換の結果が選択された場合、.mfd または .sps ファイルと対応する入力ファイルを使って MapForce または StyleVision により変換が行われ、その変換結果がリソースとなります。
4. **リソースの種類に対するファイルを選択する** リソースの種類(ファイル)が選択された場合、リソースのファイルテキスト ボックスにてファイルを選択してください。リソースの種類に変換の結果が選択された場合、ファイルの選択テキスト ボックスにて (MapForce 変換の場合は) .mfd ファイルまたは (StyleVision 変換の場合は) .sps ファイルを選択してください。変換に対して複数の入力または出力が可能な場合、選択するためのオプションが表示されます。例えば StyleVision 変換の出力オプションが、インストールされている StyleVision のエディションに合わせて表示されます (以下のスクリーンショットでは Enterprise Edition の出力が表示されています)。



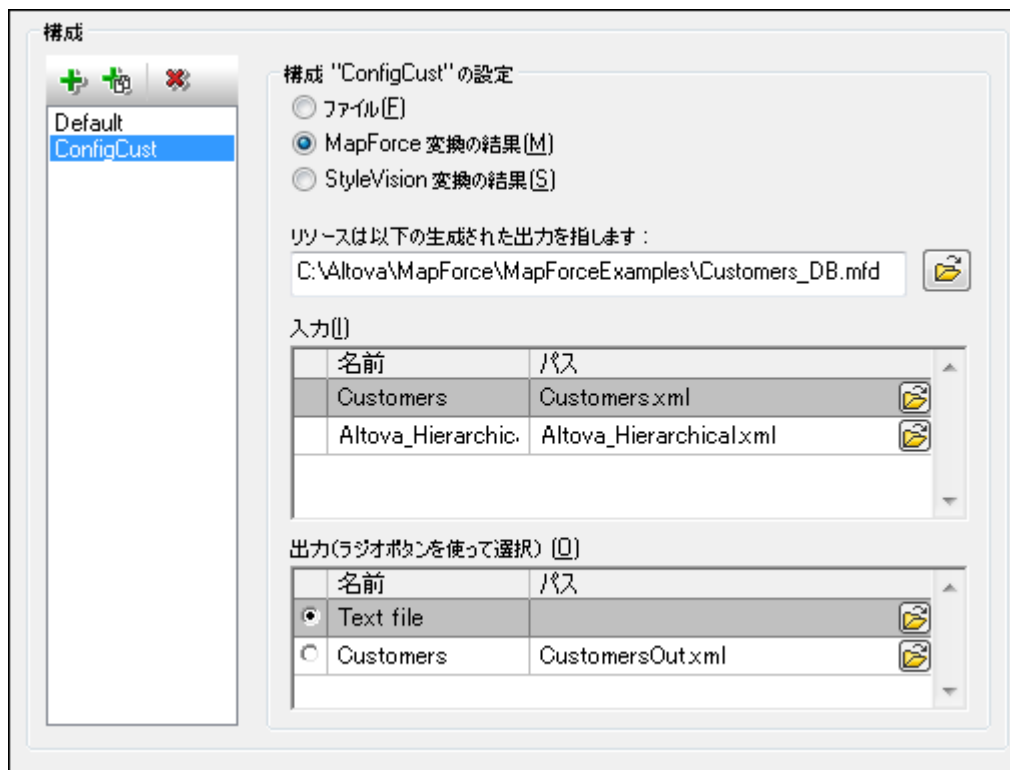
目的のオプションに対応するラジオボタンを選択します(上のスクリーンショットでは「HTML 出力」が選択されています)。変換の結果をリソースとする場合、出力をファイルまたはグローバルリソースとして保存することができます。 アイコンをクリックして、「他のグローバルリソースを選択」または「参照」を選択してください。どちらのオプションも選択されていない場合、グローバルリソースが使用された際に変換の結果が一時ファイルとしてロードされます。

5. **必要に応じて複数の構成を定義する** その他の構成を必要な数が追加して、各構成に対してリソースを指定します。各構成に対して上記のステップ 3 と 4 を実行してください。エイリアスに対する新たな構成はいつでも追加することができます。
6. **エイリアスの定義を保存する** 「OK」をクリックしてエイリアスと、すべての構成をグローバルリソースとして保存します。グローバルリソースが「[グローバルリソースの管理](#)」ダイアログにあるファイル以下に表示されます。

MapForce 変換の結果

Altova MapForce では(既存の)入力ドキュメントスキーマから出力ドキュメントスキーマへのマッピングが行われます。MapForce ユーザーにより作成されたこのマッピングは、MapForce Design (MFD) と呼ばれます。入力スキーマに関連付けられた XML ファイル、テキストファイル、データベースは、データソースとして使用することができ、MapForce により出力ドキュメントスキーマに対応した出力データファイルが生成されます。「MapForce 変換の結果」が出力ドキュメントで、グローバルリソースとなります。

MapForce により生成されたデータファイルをグローバルリソースとしてセットするには、グローバルリソースダイアログにて、以下の設定を行う必要があります(以下のスクリーンショットを参照)。



- .mfd (MapForce Design) ファイル: 「リソースは以下の生成された出力を指します」テキストボックスにて、.mfd ファイルを指定する必要があります。
- 入力データファイル: MFD ファイルが指定されると、そのファイルが解析され、中に含まれる入力スキーマ情報をベースにデフォルトのデータファイルが入力ペインに入力されます(上のスクリーンショットを参照)。他のファイルを指定することで、各入力スキーマに対するデフォルトファイルの選択を変更することができます。
- 出力ファイル: MFD ドキュメントに複数の出力スキーマが含まれる場合、出力ペインにそれらの出力すべてが表示されます(上のスクリーンショットを参照)。個々の出力スキーマに関連付けられた出力ファイルの場所がMFD ドキュメントにて指定されている場合、その出力スキーマに対するファイルの位置が出力ペインに表示されます。上のスクリーンショットからは、Customers 出力スキーマにデフォルトのXML データファイル(CustomersOut.xml) が与えられており、Text ファイル出力スキーマはそのような割り当てが行われていないことが理解できます。出力ペインに表示あるデフォルトのファイルの場所を使用することができるほか、自分で指定することができます。MapForce 変換の結果が、選択された出力スキーマがある場所へ保存され、そのファイルがグローバルリソースとして使用されます。

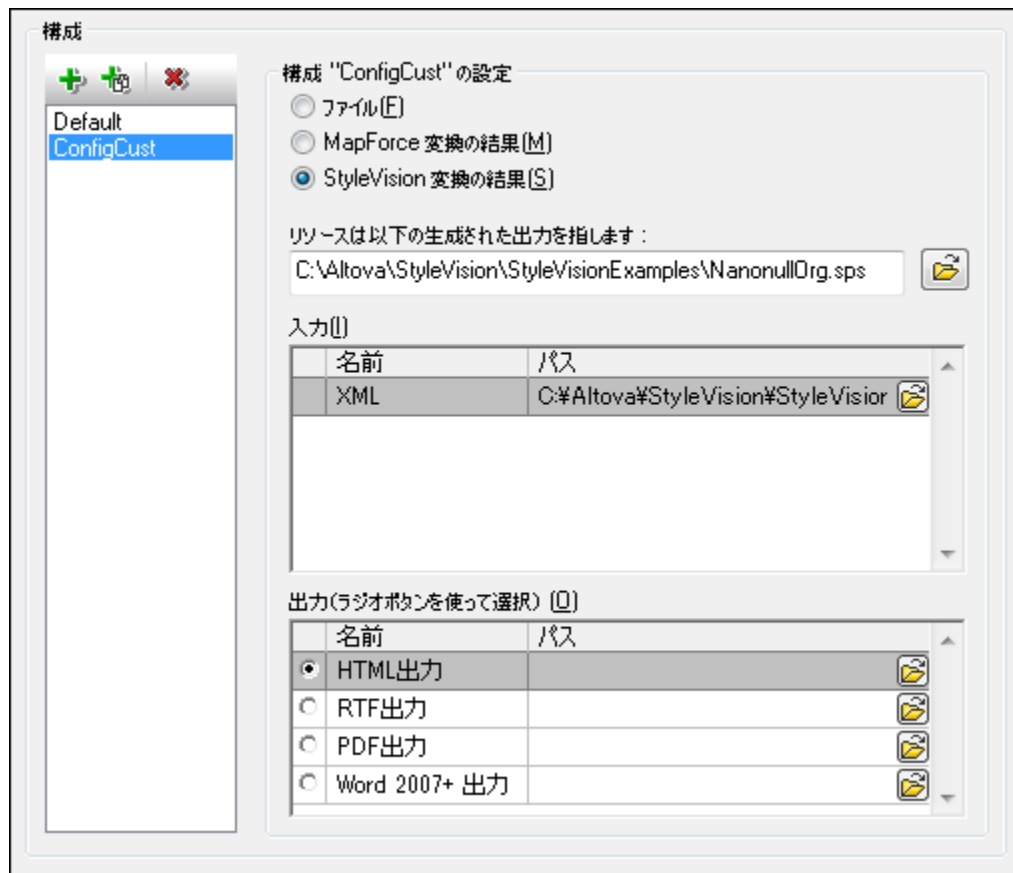
メモ このオプション (MapForce の結果) により、グローバルリソースが呼び出された時に変換が行われるようになります。これで(入力ファイルから得られた) 最新のデータがグローバルリソースに含まれることが保証されます。

メモ 変換を行うためには MapForce が使用されるため、この機能を使用するには Altova MapForce をインストールする必要があります。

StyleVision 変換の結果

Altova StyleVision は StyleVision Power Stylesheet (SPS) ファイルを作成するために使用され、これら SPS ファイルにより、XML ドキュメントを様々なフォーマット (HTML、PDF、RTF、Word 2007+ など) の出力ドキュメントへ変換するための XSLT スタイルシートが生成されます。「StyleVision 変換の結果」オプションを選択すると、StyleVision により作成された出力ドキュメントが、選択された構成に関連付けられたグローバルリソースとなります。

グローバルリソースダイアログにて「StyleVision 変換の結果」オプションを選択すると、以下のファイルを指定する必要があります。



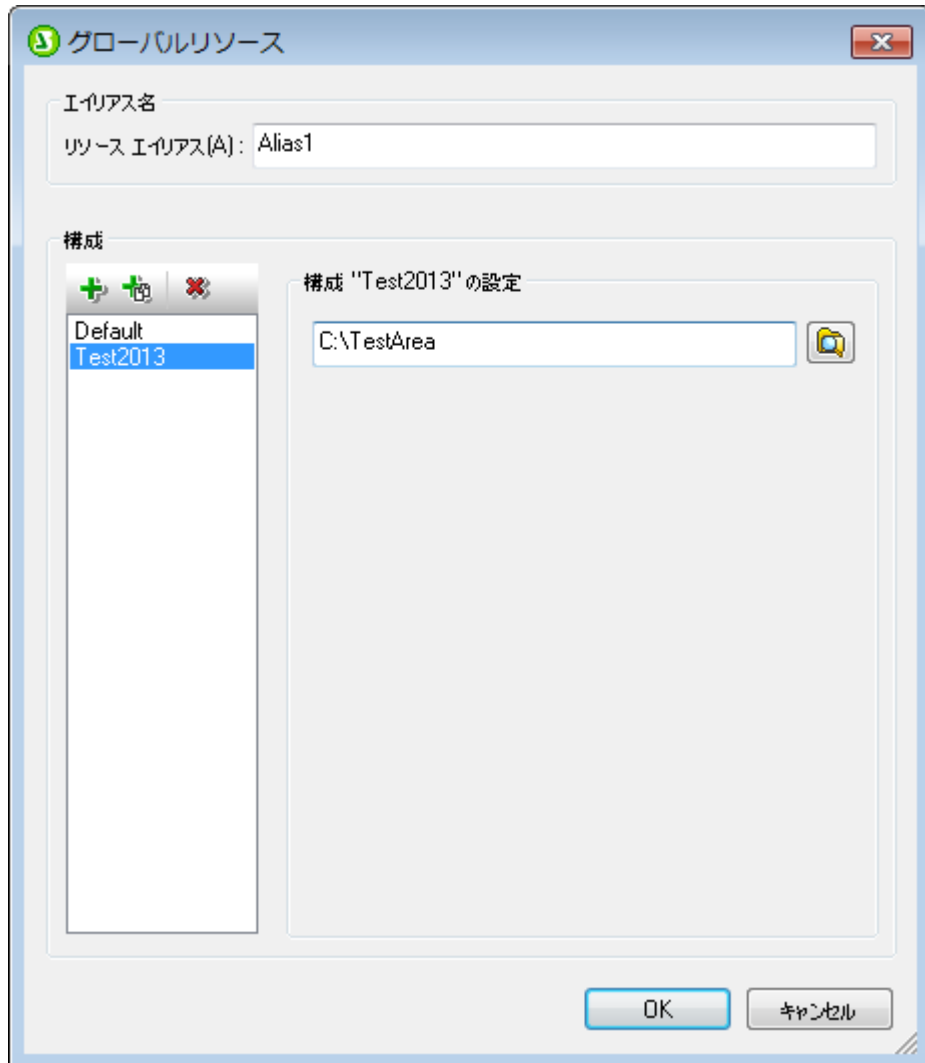
- **.sps (SPS) ファイル:** 「リソースは以下の生成された出力を指します」テキスト ボックスにて、ファイルを指定する必要があります(上のスクリーンショットを参照)。
- **入力ファイル:** 入力ファイルがSPS ファイル内部で既に指定されている場合、SPS ファイルが選択された段階で、そのファイルが入力ペインに表示されます。このエントリは変更することができるほか、エントリが指定されていない場合、追加する必要があります。
- **出力ファイル:** 出力ペインにて出力フォーマットを選択し、そのフォーマットに対する出力ファイルの場所を指定します。

メモ このオプション (StyleVision の結果) により、グローバルリソースが呼び出された時に変換が行われるようになります。これで、(入力ファイルから得られた) 最新のデータがグローバルリソースに含まれることが保証されます。





メモ 変換を行うためには StyleVision が使用されるため、この機能を使用するには Altova StyleVision をインストールする必要があります。

8.1.2 フォルダー

フォルダーに対するグローバルリソースダイアログでは、以下の操作によりフォルダーリソースを追加することができます(以下のスクリーンショットを参照)。



グローバルリソース ダイアログのアイコン

-  構成の追加: 追加する構成の名前を入力するための 構成の追加ダイアログが表示されます。
-  構成をコピーして追加: 選択された構成のコピーとして追加する構成の名前を入力するための 構成の追加ダイアログが表示されます。
-  削除: 選択された構成を削除します。
-  開く: グローバルリソースとして作成されるファイルを参照します。

エイリアスの定義

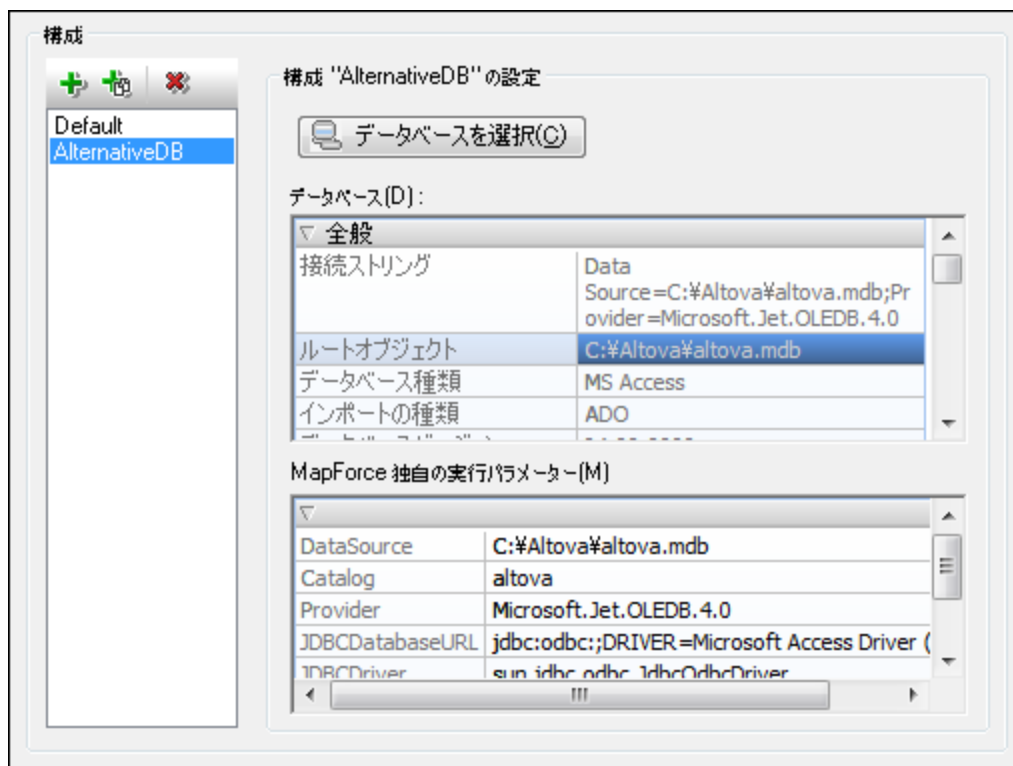
以下の操作によりエイリアス(名前と構成)を定義することができます:

1. エイリアスに名前を付ける リソースエイリアス テキスト ボックスにエイリアスの名前を入力します。

2. 構成を追加する 構成ペインには、デフォルトで(名前変更や削除を行うことができない) Default という名前の構成が表示されます。構成の追加または構成のコピーを追加アイコンをクリックして、表示されるダイアログにて構成名を入力することで、必要な数の構成を追加することができます。構成の追加または構成のコピーを追加アイコンをクリックして、表示されるダイアログで構成の名前を入力することで、構成が追加されます。「OK」をクリックすることで新たな構成が構成ペインに表示されます。同じ作業を必要な数が繰り返してください。
3. 構成のソースとなるフォルダーを選択する 構成ペインに表示された構成を選択し、グローバルリソースとして作成するフォルダーを参照します。セキュリティ資格情報がフォルダーにアクセスするために必要とされる場合、ユーザー名とパスワードフィールドでこれらを指定してください。フォルダーにアクセスするためにセキュリティの資格条件が必要な場合、ユーザー名とパスワードフィールドを指定してください。
4. 必要に応じて複数の構成を定義する その他の構成を必要な数が追加して、各構成に対してソースを指定します。各構成に対して上記のステップ 2 と 3 を実行してください。エイリアスに対する新たな構成はいつでも追加することができます。
5. エイリアスの定義を保存する 「OK」をクリックしてエイリアスと、すべての構成をグローバルリソースとして保存します。グローバルリソースが、「[グローバルリソースの管理](#)」ダイアログにあるファイル以下に表示されます。

8.1.3 データベース

データベースに対するグローバルリソースダイアログでは、以下の操作によりデータベースのソースを追加することができます。




グローバルリソース ダイアログのアイコン



構成の追加: 追加する構成の名前を入力するための 構成の追加ダイアログが表示されます。



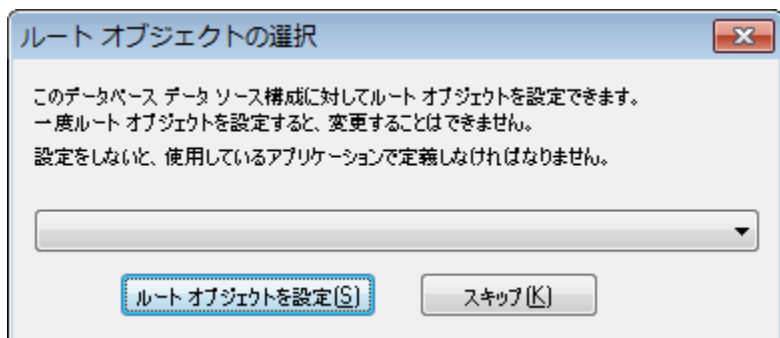
構成をコピーして追加: 選択された構成のコピーとして追加する構成の名前を入力するための 構成の追加ダイアログが表示されます。

 削除: 選択された構成を削除します。

エイリアスの定義

以下の操作によりエイリアス(名前と構成)を定義することができます。

1. **エイリアスに名前を付ける** リソースエイリアステキスト ボックスにエイリアスの名前を入力します。
2. **構成を追加する** 構成ペインには、デフォルトで(名前変更や削除を行うことができない) Default という名前の構成が表示されます。構成の追加または構成のピーを追加アイコンをクリックして、表示されるダイアログにて構成名を入力することで、必要なだけの構成を追加することができます。構成の追加または構成のピーを追加アイコンをクリックして、表示されるダイアログで構成の名前を入力することで、構成が追加されます。「OK」をクリックすることで新たな構成が構成ペインに表示されます。同じ作業を必要な数だけ繰り返してください。
3. **構成のリソースとなるデータベースを選択する** 構成ペインに表示された構成を選択し、「データベースを選択」ボタンをクリックします。グローバルリソース接続を作成ダイアログが表示されます。フォルダーにアクセスするためのセキュリティの資格条件が必要な場合、ユーザー名とパスワードフィールドを指定してください。
4. **データベースへ接続する** 接続ウィザード、既存の接続、ADO 接続、ODBC 接続、または JDBC 接続を使用して、データベースへの接続を作成します。
5. **ルート オブジェクトの選択** ルート オブジェクトを選択することのできるデータベースへ接続した場合、サーバー上にあるルート オブジェクトを選択するためのダイアログが表示されます(以下のスクリーンショットを参照)。ルート オブジェクトを選択して、「ルート オブジェクトを設定」をクリックしてください。選択されたルート オブジェクトが、この構成が使用された時ロードされるルート オブジェクトとなります。



ルート オブジェクトを選択しなかった場合(「スキップ」ボタンをクリックした場合)、グローバルリソースがロードされた際にルート オブジェクトを選択することとなります。

6. **複数の構成を必要に応じて定義する** 作成されたその他の構成に対してデータベースリソースを指定します(作成された構成に対して上記のステップ3 から5 までを繰り返します)。新たな構成をエイリアス定義へ追加することもできます。
7. **エイリアスの定義の保存** 「OK」をグローバルリソースダイアログ内でクリックし、エイリアスとその構成をグローバルリソースとして保存します。グローバルリソースは、グローバルリソースの管理ダイアログ内でデータベースの下にリストされています。

8.2 グローバルリソースを使用する

グローバルリソースには、ファイル型、フォルダー型、そしてデータベース型と、複数の種類が存在します。Authentic Desktop で使用することのできる様々なシナリオについて以下で紹介されています: [ファイルとフォルダー](#)。

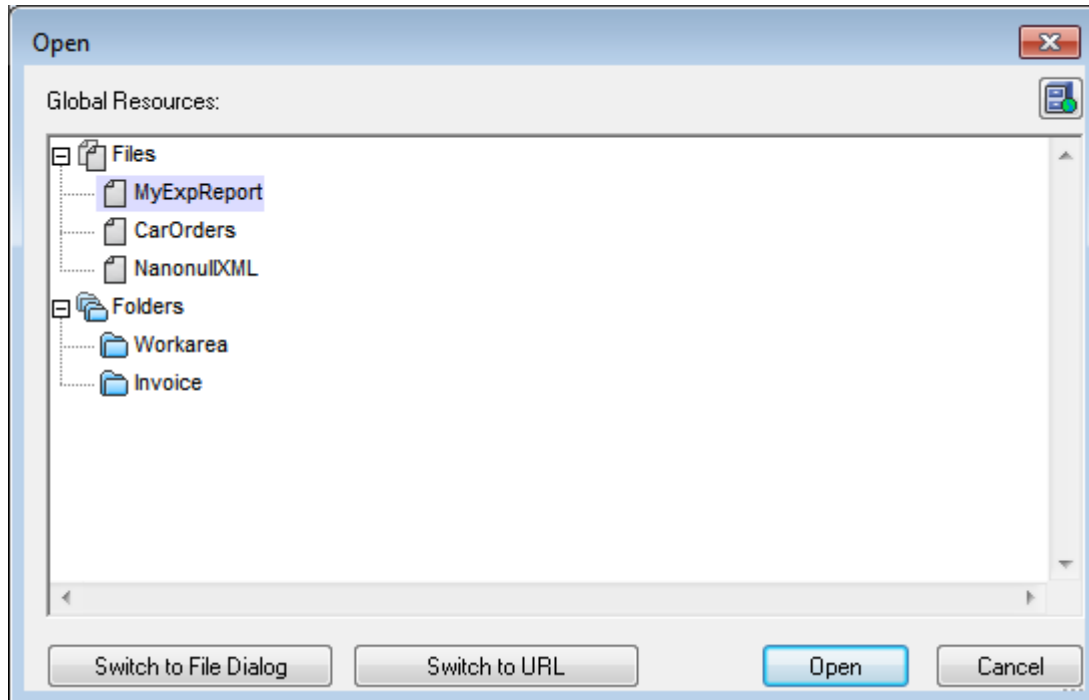
使用するリソースの選択

アプリケーション全体で使用することのできるグローバルリソースの選択、ならびに実際に使用されるグローバルリソースの選択を行うには方法が2つあります:

- **「[グローバルリソース](#)」ダイアログ**にてアクティブなグローバルリソースXML ファイルを選択します。アクティブなグローバルリソースXML ファイル内に収められている利用可能なグローバルリソースの定義は、アプリケーション内の開かれているファイル全てで使用することができます。使用することができるのは、アクティブなグローバルリソースXML ファイルにある定義のみとなります。アクティブなグローバルリソースXML ファイルはいつでも変更することができ、新しいファイルに収められたグローバルリソース定義が、それまで使用されていたアクティブなファイルの内容に置き換えられます。従って、アクティブなグローバルリソースXML ファイルにより、(i) 割り当て可能なグローバルリソースと、(ii) 検索可能なグローバルリソースが決定されます(例えば、あるグローバルリソースXML ファイル内のグローバルリソースに割り当てが行われているが、現在アクティブなグローバルリソースXML ファイルにて、対応する名前のリソースがない場合、割り当てられたグローバルリソースの検索を行うことはできなくなります)。
- **「[ツール](#)」アクティブな構成**。またはグローバルリソースツールバーからアクティブな構成 を選択します。メニューコマンド(またはツールバーのドロップダウンリスト)をクリックすることで、全てのエイリアスに対する構成のリストが表示されます。構成を選択することで、その構成がアプリケーションでアクティブな構成となります。この操作により、グローバルリソース(またはエイリアス)が使用される際には、各エイリアスに対して対応するアクティブな構成が適用されます。アクティブな構成は使用される全てのエイリアスに対して適用されます。アクティブな構成と同じ名前の構成をエイリアスが持たない場合、そのエイリアスに対してはデフォルトの構成が適用されます。アクティブな構成はリソースの割り当てに影響を与えません。アクティブな構成は、実際にリソースが使用される時に重要となります。

8.2.1 ファイルとフォルダーの割り当て

このセクションでは、ファイル型ならびにフォルダー型のグローバルリソース割り当て方法について記述します。ファイル型とフォルダー型のグローバルリソースは異なる方法で割り当てられます。以下にある使用シナリオでは、「グローバルリソースに切り替え」ボタンにてグローバルリソースを開くダイアログが表示されます(以下のスクリーンショットを参照)。



グローバルリソースの管理: [「グローバルリソースの管理」](#) ダイアログが表示されます。

ファイル型のグローバルリソースを選択することで、ファイルが割り当てられます。フォルダ型のグローバルリソースを選択することで、ファイルを開くダイアログが表示され、目的のファイルを選択することができます。選択されたファイルへのパスは、フォルダリソースからの相対パスとなります。フォルダ型のグローバルリソースは2つの構成が含まれ、各構成が異なるフォルダを指す場合（そして、それらのフォルダ内にあるファイルの名前が同一の場合）、構成切り替えることで2つの異なるフォルダをターゲットにすることができます。これは試験などを行う際に便利な機能となります。

ダイアログ下部にあるアイコンをクリックすることで、ファイルダイアログまたはURLダイアログに切り替えることもできます。「グローバルリソースの管理」アイコンをクリックすることで、[「グローバルリソースの管理」](#) ダイアログが表示されます。

使用できるシナリオ

ファイル型やフォルダ型のグローバルリソースに対して以下のような操作を行うことができます:

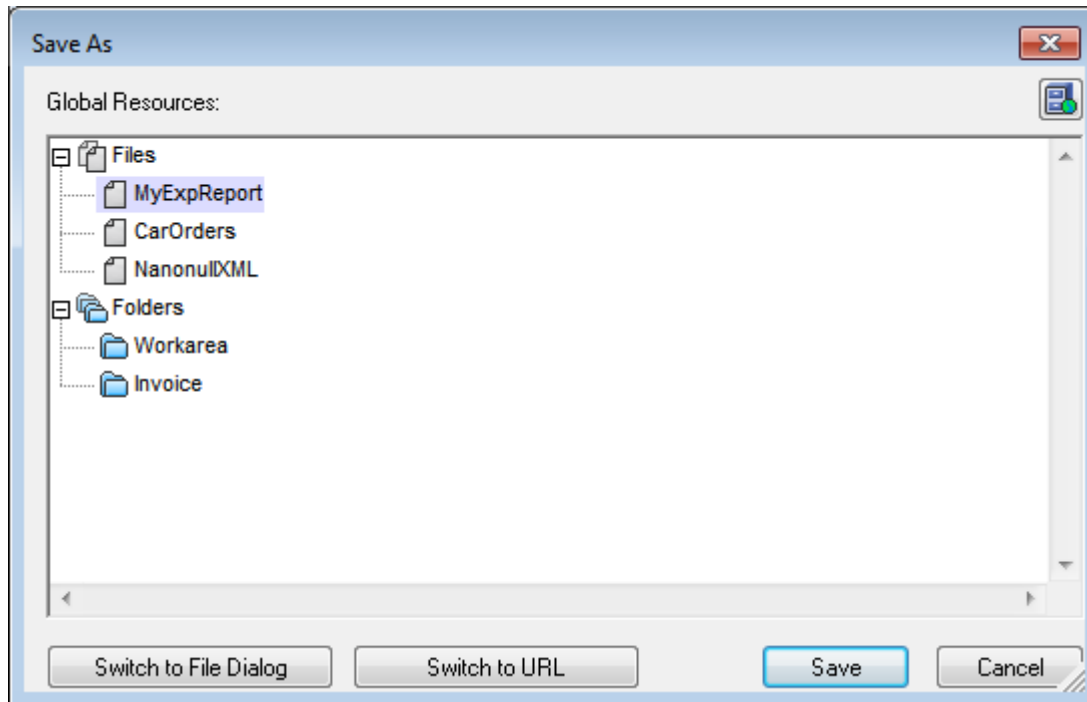
- [グローバルリソースを開く](#)
- [グローバルリソースとして保存](#)
- [XSLT 変換](#)

グローバルリソースを開く

グローバルリソースは Authentic Desktop の [「ファイル | 開く\(グローバルリソースに切り替え\)」](#) コマンドから開き、編集することができます。ファイル型グローバルリソースの場合、ファイルが直接開かれます。フォルダ型グローバルリソースの場合、割り当てられたフォルダが選択された状態でファイルを開くダイアログが表示され、フォルダ以下にある目的のファイルを選択することとなります。グローバルリソースを通してファイルを選択し、編集する利点は、構成を切り替えることで、同一のグローバルリソース内にある(別構成の)ファイルへアクセスできることです。構成を変更する前に、編集がなされたファイルを保存を行う必要があります。

グローバルリソースとして保存

新たに作成されたファイルもグローバルリソースとして保存することができます。同様に、既存のファイルを開いて、グローバルリソースとして保存することもできます。「ファイル | 保存」または「ファイル | 名前を付けて保存」コマンドをクリックすると、保存ダイアログが表示されます。「グローバルリソースに切り替え」ボタンをクリックして、現在使用されているグローバルリソース XML ファイルにて定義されたエイリアスであるグローバルリソースへアクセスします(以下のスクリーンショットを参照)。



エイリアスを選択し、「保存」をクリックします。エイリアスが[ファイルエイリアス](#)の場合、そのファイルに直接保存されます。エイリアスが[フォルダーエイリアス](#)の場合、ダイアログが表示され、保存するファイルの名前を入力するよう促されます。どちらの場合でも、ファイルは[現在アクティブな構成](#)により定義された位置に保存されます。

メモ 各構成ではファイルの位置が構成の定義により指定されます。グローバルリソースとして保存しようとするファイルが、構成で定義された位置にあるファイルと同一の拡張子を持たない場合、このグローバルリソースが開かれたときに、Authentic Desktop にて編集エラーや検証エラーが発生する可能性があります。これは構成の定義により指定されたファイルの種類を想定して Authentic Desktop によりファイルが開かれるためです。

XSLT 変換

コマンド「[XSL/XQuery | XSL 変換](#)」または「[XSL/XQuery | XSL:FO 変換](#)」をクリックすると、または XML ファイルを指定するためのダイアログが表示されるので、XSLT、XQuery、または XML ファイルを選択します。「参照」ボタンをクリックして、「グローバルリソースに切り替え」ボタンをクリックすることで、グローバルリソースを開くダイアログが表示されます([セクション最上部のスクリーンショットを参照](#))。選択されたグローバルリソースの現在アクティブな構成に関連付けられているファイルが変換で使用されます。

8.2.2 構成の切り替え

同時にアクティブにすることができる構成は1つだけ、その設定はアプリケーション全体に対して適用されます。これは、アクティブな構成が、現在開いている全てのファイルやデータベース接続のエイリアスに対して有効であることを意味します。エイリアスがアクティブな構成と同じ名前の構成が割り当てられていない場合、デフォルトの構成がそのエイリアスに対して使用されます。構成を変更する例として、XSLT ファイルが複数の構成を持つグローバルリソースを介してXMLドキュメントに割り当てられると仮定しましょう。各構成は異なるファイルにマップされます。ですから、ファイルの選択はアプリケーションのアクティブな構成としてどの構成が選択されているかによります。

構成の切り替えるには2つの方法があります：

- メニューコマンド「ツール | アクティブな構成」を選択すると、グローバルリソースXMLファイル内にある全ての構成がサブメニューに表示されます。目的の構成を選択します。
- グローバルリソースコンボボックスのツールバーから、目的の構成を選択します(以下のスクリーンショットを参照)。



アクティブな構成を切り替えるという方法により、グローバルリソースを介して割り当てられたソースファイルを変更することができます。

9 Source Control

The source control support in Authentic Desktop is available through the Microsoft Source Control Plug-in API (formerly known as the MSSCCI API), versions 1.1, 1.2 and 1.3. This enables you to run source control commands such as "Check in" or "Check out" directly from Authentic Desktop to virtually any source control system that lets native or third-party clients connect to it through the Microsoft Source Control Plug-in API.

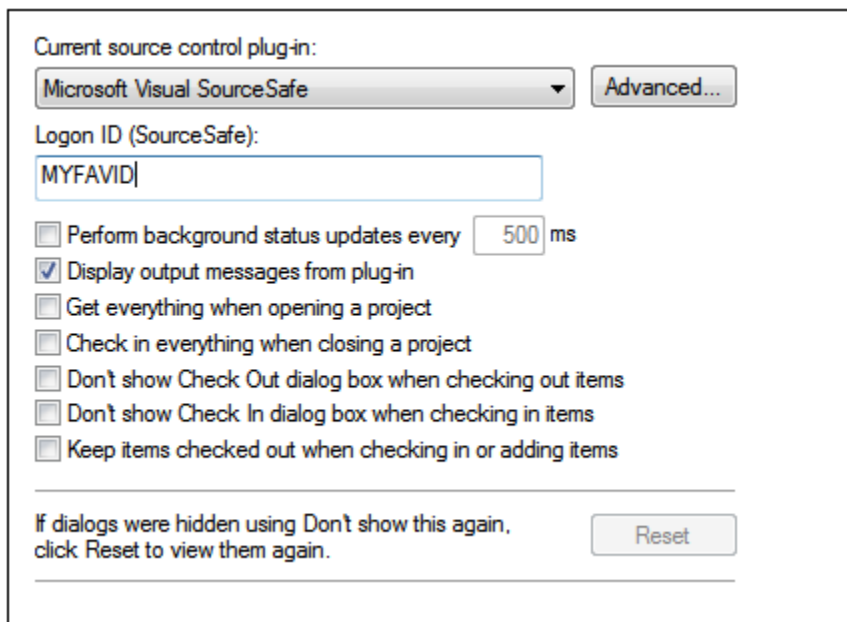
You can use as your source control provider any commercial or non-commercial plug-in that supports the Microsoft Source Control Plug-in API, and can connect to a compatible version control system. For the list of source control systems and plug-ins tested by Altova, see [Supported Source Control Systems](#).

Installing and configuring the source control provider

To view the source control providers available on your system, do the following:

1. On the **Tools** menu, click **Options**.
2. Click the **Source Control** tab.

Any source control plug-ins compatible with the Microsoft Source Code Control Plug-in API are displayed in the **Current source control plug-in** drop-down list.



If a compatible plug-in cannot be found on your system, the following message is displayed:

```
"Registration of installed source control providers could not be found or is incomplete."
```

Some source control systems might not install the source control plug-in automatically, in which case you will need to install it separately. For further instructions, refer to the documentation of the respective source control system. A plug-in (provider) compatible with the Microsoft Source Code Control Plug-in API is expected to be registered under the following registry entry on your operating system:

```
HKEY_LOCAL_MACHINE\SOFTWARE\SourceCodeControlProvider\InstalledSCCProviders
```

Upon correct installation, the plug-in becomes available automatically in the list of plug-ins available to Authentic Desktop.

Accessing the source control commands

The commands related to source control are available in the **Project | Source Control** menu.

Resource / Speed issues

Very large source control databases might be introducing a speed/resource penalty when automatically performing background status updates.

You might be able to speed up your system by disabling (or increasing the interval of) the **Perform background status updates every ... seconds** option in the **Source Control** tab accessed through **Tools | Options**.

Note: The **64-bit** version of your Altova application automatically supports any of the supported 32-bit source control programs listed in this documentation. When using a 64-bit Altova application with a 32-bit source control program, the **Perform background status updates every ... seconds** option is automatically grayed-out and cannot be selected.

Differencing with Altova DiffDog

You can configure many source control systems (including Git and TortoiseSVN) so that they use Altova DiffDog as their differencing tool. For more information about DiffDog, see <https://www.altova.com/diffdog>. For DiffDog documentation, see <https://www.altova.com/documentation.html>.

9.1 Setting Up Source Control

The mechanism for setting up source control and placing files in a Authentic Desktop project under source control is as follows:

1. If this hasn't been done already, install the source control system (see [Supported Source Control Systems](#)) and set up the source control database (repository) to which you wish to save your work.
2. Create a local workspace folder that will contain the working files that you wish to place under source control. The folder that contains all your workspace folders and files is called the local folder, and the path to the local folder is referred to as the local path. This local folder will be bound to a particular folder in the repository.
3. In your Altova application, create an application project folder to which you must add the files you wish to place under source control. This organization of files in an application project is abstract. The files in a project reference physical files saved locally, preferably in one folder (with sub-folders if required) for each project.
4. In the source control system's database (also referred to as source control or repository), a folder is created that is bound to the local folder. This folder (called the bound folder) will replicate the structure of the local folder so that all files to be placed under source control are correctly located hierarchically within the bound folder. The bound folder is usually created when you add a file or an application project to source control for the first time. See the section, [Application Project](#), for information about the repository's folder structure.
5. Project files are added to source control using the command **Project | Source Control | Add to Source Control**. When you add a project or a file in a project for the first time to source control, the correct bindings and folder structure will be created in the repository.
6. Source control actions, such as the checking in and out of files, and the removing of files from source control, can be carried out via commands in the **Project | Source Control** submenu. These commands are described in the [Project menu subsection](#) of the User Reference.

Note: If you wish to change the current source control provider, this can be done in one of two ways: (i) via the Source Control options ([Tools | Options | Source Control](#)), or (ii) in the Change Source Control dialog (**Project | Source Control | Change Source Control**).

9.2 Supported Source Control Systems

The list below shows the Source Control Servers (SCSs) supported by Authentic Desktop, together with their respective Source Control Clients (SCCs). The list is organized alphabetically by SCS. Note the following:

- Altova has implemented the Microsoft Source Control Plug-in API (versions 1.1, 1.2, and 1.3) in Authentic Desktop, and has tested support for the listed drivers and revision control systems. It is expected that Authentic Desktop will continue to support these products if, and when, they are updated.
- Source Code Control clients not listed below, but which implement the Microsoft Source Control Plug-in API, should also work with Authentic Desktop.

Source Control System	Source Code Control Clients
AccuRev 4.7.0 Windows	AccuBridge for Microsoft SCC 2008.2
Bazaar 1.9 Windows	Aigenta Unified SCC 1.0.6
Borland StarTeam 2008	Borland StarTeam Cross-Platform Client 2008 R2
Codice Software Plastic SCM Professional 2.7.127.10 (Server)	Codice Software Plastic SCM Professional 2.7.127.10 (SCC Plugin)
Collabnet Subversion 1.5.4	<ul style="list-style-type: none"> • Aigenta Unified SCC 1.0.6 • PushOK SVN SCC 1.5.1.1 • PushOK SVN SCC x64 version 1.6.3.1 • TamTam SVN SCC 1.2.24
ComponentSoftware CS-RCS (PRO) 5.1	ComponentSoftware CS-RCS (PRO) 5.1
Dynamsoft SourceAnywhere for VSS 5.3.2 Standard/Professional Server	Dynamsoft SourceAnywhere for VSS 5.3.2 Client
Dynamsoft SourceAnywhere Hosted	Dynamsoft SourceAnywhere Hosted Client (22252)
Dynamsoft SourceAnywhere Standalone 2.2 Server	Dynamsoft SourceAnywhere Standalone 2.2 Client
Git	PushOK GIT SCC plug-in (see Source Control with Git)
IBM Rational ClearCase 7.0.1 (LT)	IBM Rational ClearCase 7.0.1 (LT)
March-Hare CVSNT 2.5 (2.5.03.2382)	Aigenta Unified SCC 1.0.6
March-Hare CVS Suite 2008	<ul style="list-style-type: none"> • Jalindi Igloo 1.0.3 • March-Hare CVS Suite Client 2008 (3321) • PushOK CVS SCC NT 2.1.2.5 • PushOK CVS SCC x64 version 2.2.0.4 • TamTam CVS SCC 1.2.40
Mercurial 1.0.2 for Windows	Sergey Antonov HgSCC 1.0.1
Microsoft SourceSafe 2005 with CTP	Microsoft SourceSafe 2005 with CTP

Source Control System	Source Code Control Clients
Microsoft Visual Studio Team System 2008/2010 Team Foundation Server	Microsoft Team Foundation Server 2008/2010 MSSCCI Provider
Perforce 2008 P4S 2008.1	Perforce P4V 2008.1
PureCM Server 2008/3a	PureCM Client 2008/3a
QSC Team Coherence Server 7.2.1.35	QSC Team Coherence Client 7.2.1.35
Reliable Software Code Co-Op 5.1a	Reliable Software Code Co-Op 5.1a
Seapine Surround SCM Client/Server for Windows 2009.0.0	Seapine Surround SCM Client 2009.0.0
Serena Dimensions Express/CM 10.1.3 for Win32 Server	Serena Dimensions 10.1.3 for Win32 Client
Softimage Alienbrain Server 8.1.0.7300	Softimage Alienbrain Essentials/Advanced Client 8.1.0.7300
SourceGear Fortress 1.1.4 Server	SourceGear Fortress 1.1.4 Client
SourceGear SourceOffsite Server 4.2.0	SourceGear SourceOffsite Client 4.2.0 (Windows)
SourceGear Vault 4.1.4 Server	SourceGear Vault 4.1.4 Client
VisualSVN Server 1.6	<ul style="list-style-type: none"> • Aigenta Unified SCC 1.0.6 • PushOK SVN SCC 1.5.1.1 • PushOK SVN SCC x64 version 1.6.3.1 • TamTam SVN SCC 1.2.24

9.3 Local Workspace Folder

The files you will be working with should be saved in a hierarchy inside a local workspace folder (see *diagram below*).

Local Workspace Folder

```
|
|-- MyProject.spp
|-- QuickStart
|   |-- QuickStart.css
|   |-- QuickStart.xml
|   |-- QuickStart.xsd
|-- Grouping
|   |-- Persons
|       |-- Persons.xml
```

The application project file (.spp file) typically will be located directly inside the local workspace folder (see *diagram above*).

When one or more files in this (workspace) folder are placed under source control, the local workspace folder's structure is partly or wholly reproduced in the repository. For example, if the file `Persons.xml` from the local folder shown above is placed under source control, then the path to it in the repository will be:

```
[RepositoryFolder]/MyProject/Grouping/Persons/Persons.xml
```

The `MyProject` folder in the repository folder is bound to the local folder. Typically it would be the name of the project, but you could give it any name.

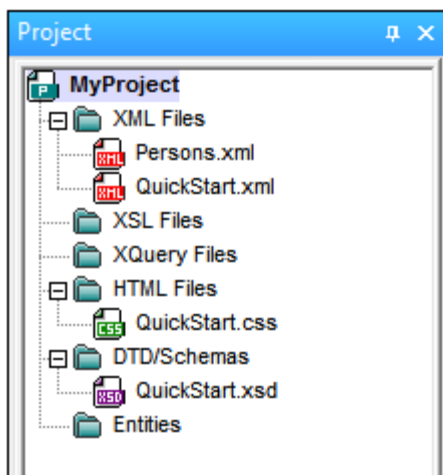
If the entire application project is placed under source control (by selecting the project name in the Projects window and placing it under source control), then the entire local folder structure is recreated in the repository.

Note: Files from outside the local workspace folder can be added to the application project. But whether you can place such a file under source control depends upon the source control system you are using. Some source control systems could have a problem placing a file from outside the local folder into the repository. We therefore recommend that all project files you wish to place under source control be located in the local workspace folder.

9.4 Application Project

Create or load the Altova application project you wish to place under source control. If you wish to place a single file under source control, this file must be included in a project—since source control can only be accessed via a project.

For example, consider a project in Altova's XMLSpy application. The project's properties are saved in a `.spp` file. In the application, the project is displayed in the application's Project window (see *screenshot below*). The project in the screenshot below is named `MyProject` and the project's properties are saved in the file `MyProject.spp`.



You can place the entire project (all files in the project) or only some project files under source control. **Only files that are in the project can be placed under source control.** So you will need to add files to the project before you can place them under source control. The project file (`.spp` file) will automatically be placed under source control as soon as a file from within the project is placed under source control.

The entire project, or one or more project files, is placed under source control via the command **Project | Source Control | Add to Source Control** (see *next section below*).

Note, however, that the folder structure of the repository corresponds not to the project's folder structure (*screenshot above*) but to the structure of the [local workspace folder](#) (see *folder diagram below*). In the diagram below, notice that the `MyProject` folder in the repository has a folder structure corresponding to that of the local workspace folder. Note that the bound folder occurs within the repository folder.

Local Workspace Folder

```
|
|-- MyProject.spp
|-- QuickStart
|   |-- QuickStart.css
|   |-- QuickStart.xml
|   |-- QuickStart.xsd
|-- Grouping
|   |-- Persons
|       |-- Persons.xml
```

Repository

```
|
|-- MyProject (bound to Local Workspace)
|   |-- MyProject.spp
|   |-- QuickStart
|       |-- QuickStart.css
|       |-- QuickStart.xml
|       |-- QuickStart.xsd
|   |-- Grouping
|       |-- Persons
|           |-- Persons.xml
```

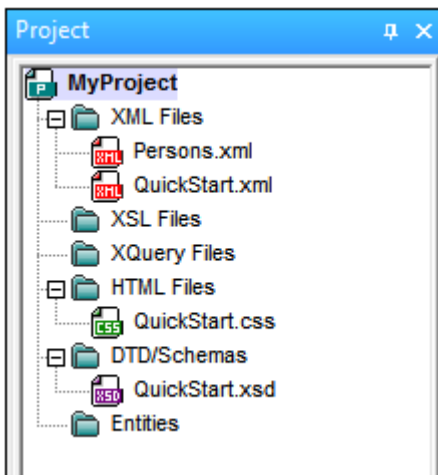
Note: An application project can contain project folders (green) and external folders (yellow). Only files in (green) project folders can be placed under source control. Files in (yellow) external folders cannot be placed under source control.

Note: Files from outside the local workspace folder can be added to the application project. But whether you can place such a file under source control depends upon the source control system you are using. Some source control systems could have a problem placing a file from outside the local folder into the repository. We therefore recommend that all project files you wish to place under source control be located in the local workspace folder.

9.5 Add to Source Control

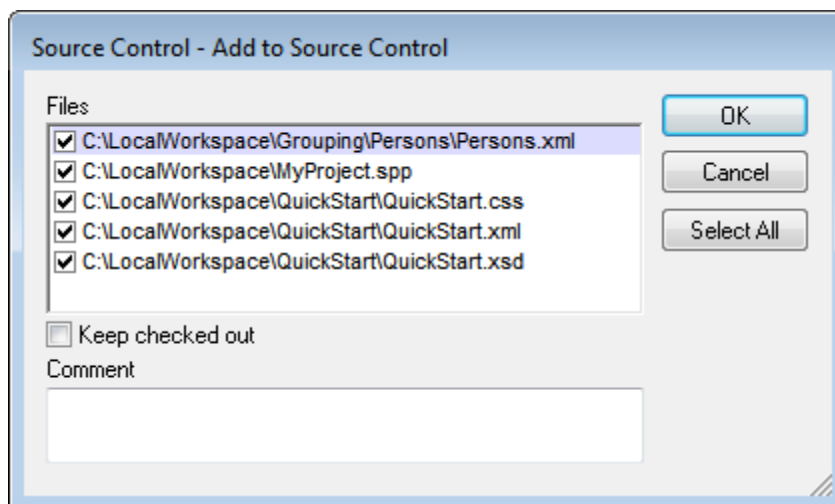
Adding the project to source control will automatically create the correct bindings and repository structure before adding the project file (.spp file) or individual files to source control. Add the project to source control as follows.

Select the project in the Project window (MyProject in the screenshot below) so that it is highlighted (as in the screenshot below). Alternatively select a single file, or select multiple files by clicking them with the **Ctrl** key pressed. Adding a single file to source control will automatically add the project file (.spp file) to source control as well.



Next, select the menu command **Project | Source Control | Add to Source Control**. This pops up the connection and configuration dialogs of the currently selected source control system. (You can change the source control system via the Change Source Control dialog (**Project | Source Control | Change Source Control**).)

Follow the source control system's instructions to make the connection and configuration. After this has been completed, all the files selected for addition plus the project file (.spp file) are displayed in an Add to Source Control dialog (screenshot below). Select the files you wish to add and click **OK**.



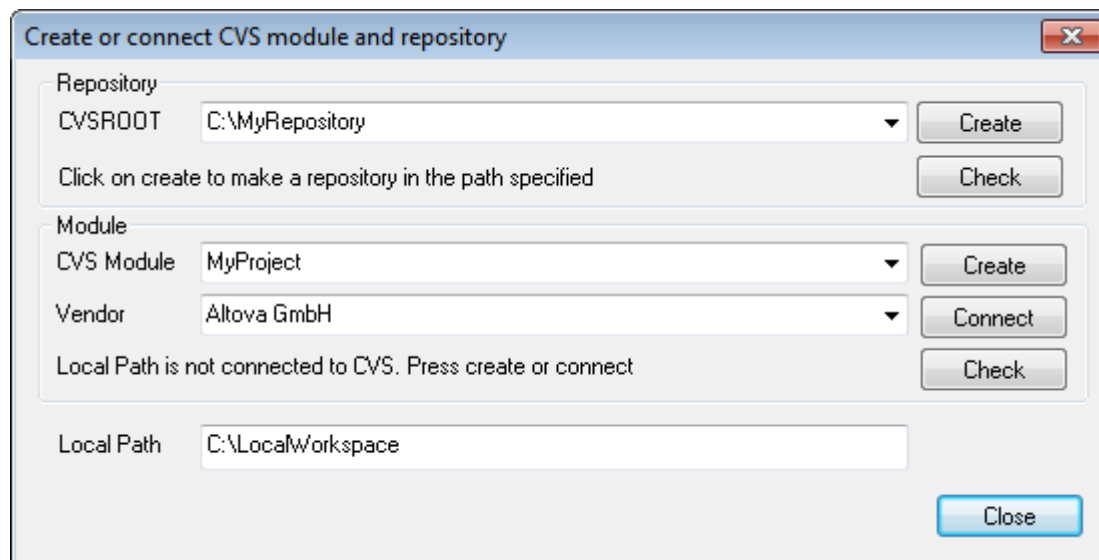
The files will be added to the repository and be either [checked in or checked out](#) depending on whether the *Keep Checked Out* check box has been checked or not.

Configuration notes

You might be prompted to create a folder in the repository for the project if it has not already been created. If you are, go ahead and create it. The [local workspace folder](#) will be bound to this folder created in the repository (see *diagrams below*).

<u>Local Workspace Folder</u>	<u>Repository</u>
-- MyProject.spp	-- <u>MyProject (bound to Local Workspace)</u>
-- QuickStart	-- MyProject.spp
-- QuickStart.css	-- QuickStart
-- QuickStart.xml	-- QuickStart.css
-- QuickStart.xsd	-- QuickStart.xml
-- Grouping	-- QuickStart.xsd
-- Persons	-- Grouping
-- Persons.xml	-- Persons
	-- Persons.xml

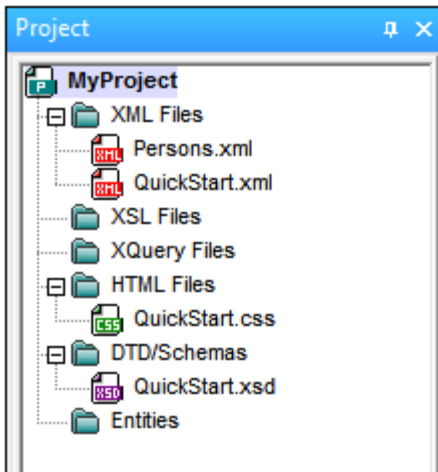
The configuration dialog of Jalindi Igloo is show below. The CVSROOT field is the path to the repository folder.



In the screenshot above, the local path locates the local workspace folder, which corresponds to the CVS module, `MyProject`, and is bound to it.

9.6 Working with Source Control

To work with source control, select the project, a project folder, or a project file in the Project window (*screenshot below*) and then select the command you want in the **Project | Source Control** menu. The **Check In** and **Check Out** commands are available as context menu commands of Project window items.



In this section, we describe the main source control features in detail:

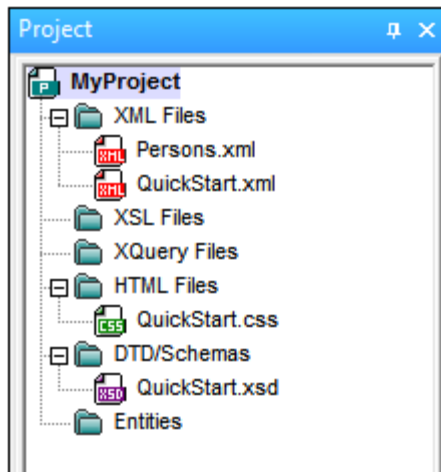
- [Add to, Remove from Source Control](#)
- [Check Out, Check In](#)
- [Getting Files as Read-Only](#)
- [Copying and Sharing from Source Control](#)
- [Changing Source Control](#)

Additional commands in the **Project | Source Control** menu are described in the [User Reference section](#) of the manual. For information specific to a particular source control system, please see the user documentation of that system.

9.6.1 Add to, Remove from Source Control

Adding

After a project has been added to source control, you can place files either singly or in groups under source control. This is also known as adding the files to source control. Select the file in the Project window and then click the command **Project | Source Control | Add to Source Control**. To select multiple files, keep the **Ctrl** key pressed while clicking on the files you wish to add. Running the command on a (green) project folder (see *screenshot below*) adds all files in the folder and its sub-folders to source control.






When files are added to source control, the [local folder hierarchy is replicated in the repository](#) (it is not the project folder hierarchy that is replicated). So, if a file is in a sub-folder X levels deep in the local folder, then the file's parent folder and all other ancestor folders are automatically created in the repository.

When the first file from a project is added to source control, the correct bindings are created in the repository and the project file (.spp file) is added automatically. For more details, see the section [Add to Source Control](#).

Source control symbols

Files and the project folder display certain symbols, the meanings of which are given below.

	Checked in. Available for check-out.
	Checked out by another user. Not available for check-out.
	Checked out locally. Can be edited and checked-in.

Removing




To remove a file from source control, select the file and click the command **Project | Source Control | Remove from Source Control**. You can also remove: (i) files in a project folder by executing the command on the folder, and (ii) the entire project by executing the command on the project.

9.6.2 Check Out, Check In

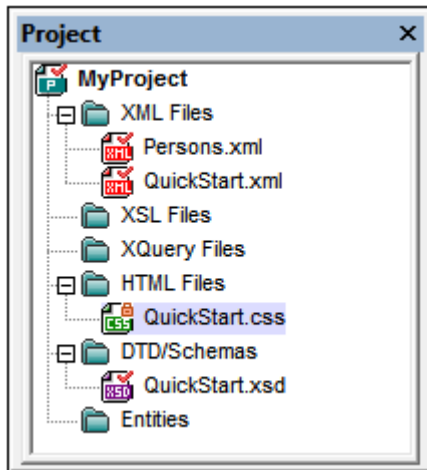
After a project file has been placed under source control, it can be checked out or checked in by selecting the file (in the Project window) and clicking the respective command in the **Project | Source Control** menu: **Check Out** and **Check In**.

When a file is checked out, a copy from the repository is placed in the local folder. A file that is checked out can be edited. If a file that is under source control is not checked out, it cannot be edited. After a file has been edited, the changes can be saved to the repository by checking in the file. Even if the file is not saved in the application, checking it in will save the changes to the repository. Whether a file is checked out or not is indicated with a tick or lock symbol in its Project window icon.

Files and the project folder display certain symbols, the meanings of which are given below.

	Checked in. Available for check-out.
	Checked out by another user. Not available for check-out.
	Checked out locally. Can be edited and checked-in.

Selecting the project or a folder within the project selects all files in the selected object. To select multiple objects (files and folders), press the **Ctrl** key while clicking the objects. The screenshot below shows a project that has been checked out. The file `QuickStart.css` has subsequently been checked in.



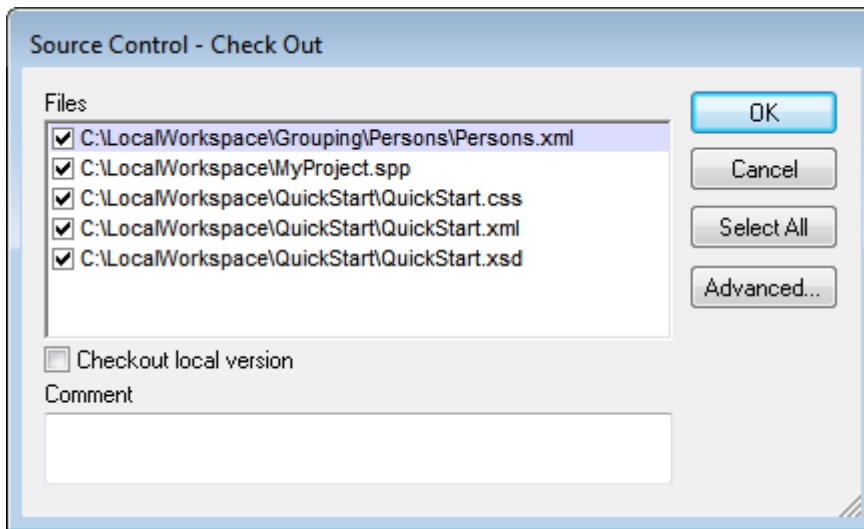
Saving and rejecting editing changes

Note that, when checking in a file, you can choose to leave the file checked out. What this does is save editing changes to the repository while continuing to keep the file checked out, which is useful if you wish to periodically save editing changes to the repository and then continue editing.

If you have checked out a file and made editing changes, and then wish to reject these changes, you can revert to the document version saved in the repository by selecting the command **Project | Source Control | Undo Check Out**.

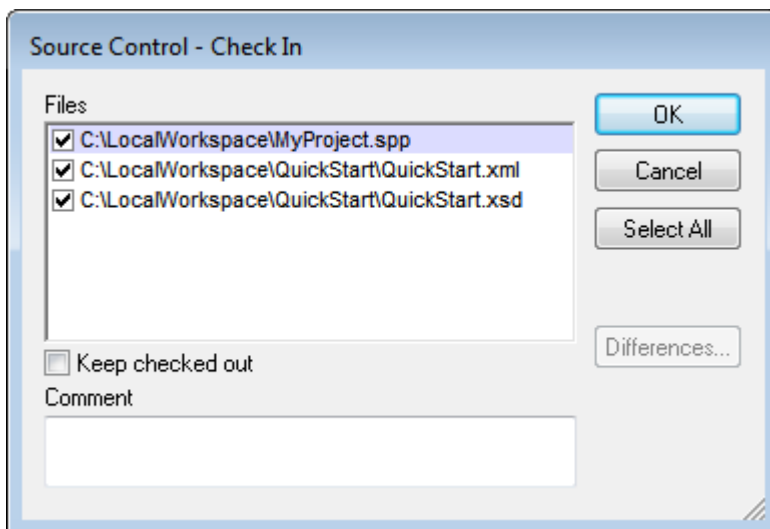
Checking out

The Check Out dialog (*screenshot below*) allows you: (i) to select the files to check out, and (ii) to select whether the repository version or the local version should be checked out.



Checking in

The Check In dialog (*screenshot below*) allows you: (i) to select the files to check in, and (ii) if you wish, to keep the file checked out.

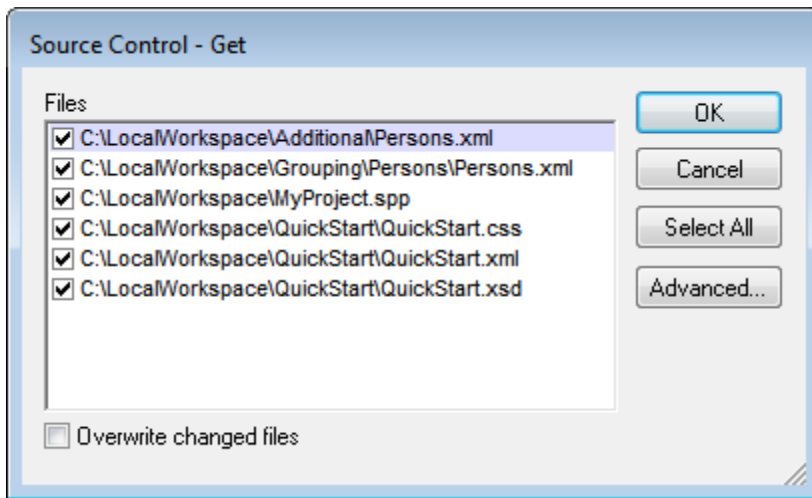


Note: In both dialogs (Check Out and Check In), multiple files appear if the selected object (project or project folder/s) contain multiple files.

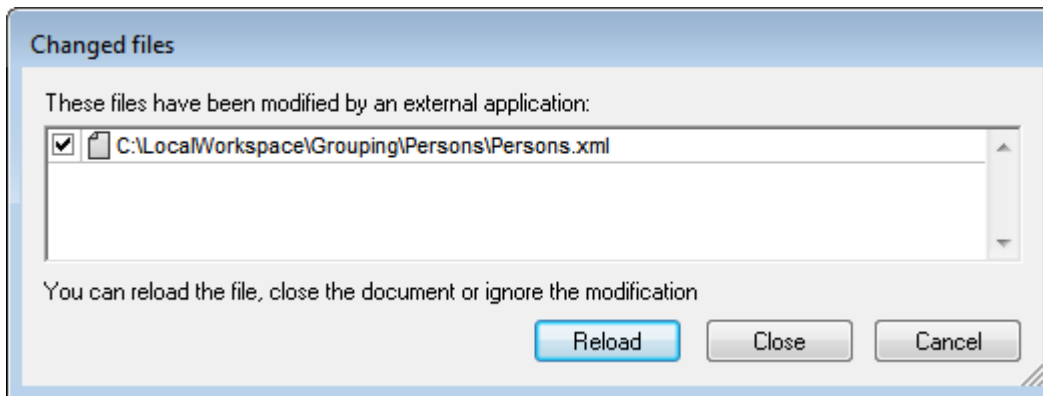
9.6.3 Getting Files as Read-Only

The **Get** command (in the **Project | Source Control** menu) retrieves files from the repository as read-only files. (To be able to edit a file, you must [check it out](#).) The Get dialog lists the files in the object (project or folder) on which the **Get** command was executed (*see screenshot below*). You can select the files to retrieve by checking them in the Get dialog list.

Note: The **Get Folders** command allows you to select individual sub-folders in the repository if this is allowed by your source control system, .

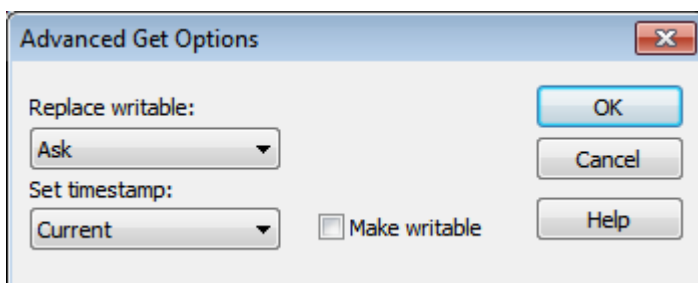


You can choose to overwrite changed checked-out files by checking this option at the bottom of the Get dialog. On clicking **OK**, the files will be overwritten. If any of the overwritten files is currently open, a dialog pops up (*screenshot below*) asking whether you wish to reload the file/s (**Reload** button), close the file/s (**Close**), or retain the current view of the file (**Cancel**).



Advanced Get Options

The Advanced Get Options dialog (*screenshot below*) is accessed via the **Advanced** button in the Get dialog (see first screenshot in this section).



Here you can set options for (i) replacing writable files that are checked out, (ii) the timestamp, and (iii) whether the read-only property of the retrieved file should be changed so that it will be writable.

Get latest version

The **Get Latest Version** command (in the **Project | Source Control** menu) retrieves and places the latest source control version of the selected file(s) in the working directory. The files are retrieved as read-only and are not checked out. This command works like the **Get** command (see *above*), but does not display the Get dialog.

If the selected files are currently checked out, then the action taken will depend on how your source control system handles such a situation. Typically, the source control system will ask whether you wish to replace, merge with, or leave the checked-out file as it is.

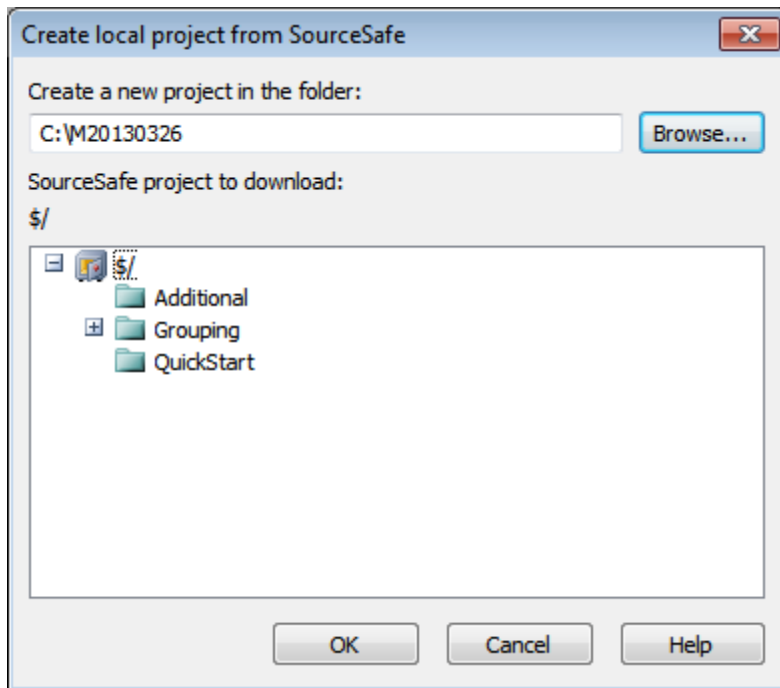
Note: This command is recursive when performed on a folder, that is, it affects all files below the current one in the folder hierarchy.

9.6.4 Copying and Sharing from Source Control

The **Open from Source Control** command creates a new application project from a project under source control.

Create the new project as follows:

1. Depending on the source control system used, it might be necessary, before you create a new project from source control, to make sure that no file from the source-controlled project is checked out.
2. No project need be open in the application, but can be.
3. Select the command **Project | Source Control | Open from Source Control**.
4. The source control system that is currently set will pop up its verification and connection dialogs. Make the connection to the [bound folder in the repository](#) that you want to copy.
5. In the dialog that pops up (*screenshot below*), browse for the local folder to which the contents of the bound folder in the repository (that you have just connected to) must be copied. In the screenshot below the bound folder is called `MyProject` and is represented by the `$` sign; the local folder is `c:\M20130326`.

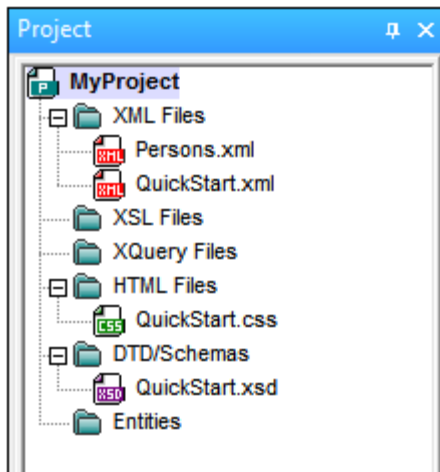


6. Click **OK**. The contents of the bound folder (`MyProject`) will be copied to the local folder `C:\M20130326.`, and a dialog pops up asking you to select the project file (`.spp` file) that is to be created as the new project.
7. Select the `.spp` file that will have been copied to the local folder. In our example, this will be `MyProject.spp` located in the `C:\M20130326` folder. A new project named `MyProject` will be created in the application and will be displayed in the Project window. The project's files will be in the folder `C:\M20130326.`

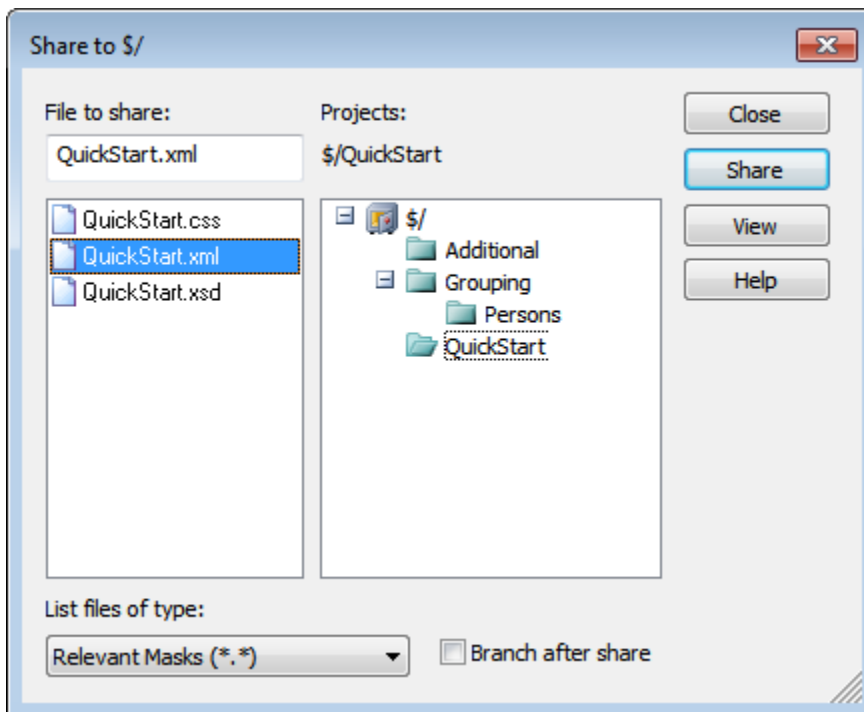
Sharing from source control

The **Share from Source Control** command is supported when the source control system being used supports shares. You can share a file, so that it is available at multiple local locations. A change made to one of these local files will be reflected in all the other "shared" versions.

In the application's Project window first select the project (*highlighted in the screenshot below*). Then click the **Share from Source Control**.

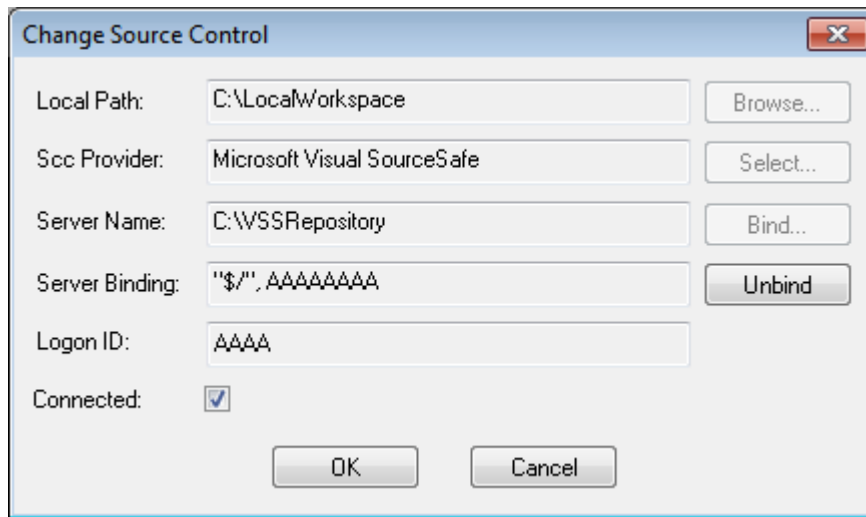


The Share To [Folder] dialog (*screenshot below*) pops up.



To select the files to share, first choose, in the project tree in the right-hand pane of the dialog (see *screenshot above*), the folder in which the files are. The files in the chosen folder are displayed in the left-hand pane. Select the file you wish to share (multiple files by pressing the **Ctrl** key and clicking the files you want to share). The selected file/s will be displayed in the *Files to Share* text box (*at top left*). The files disappear from the left hand pane. Click **Share** and then **Close** to copy the selected file/s to the local share folder. When you click **Close**, the files to share will be copied to the selected local location.

The share folder is noted in the name of the Share to [Folder] dialog. In the screenshot above it is the local folder (since the $\$$ sign is the folder in the repository to which the local folder is bound). You can see and set the share folder in the Change Source Control dialog (*screenshot below*, **Change Source Control**) by changing the local path and server binding.



For more details about sharing using your source control system, see the source control system's user documentation.

9.6.5 Changing Source Control

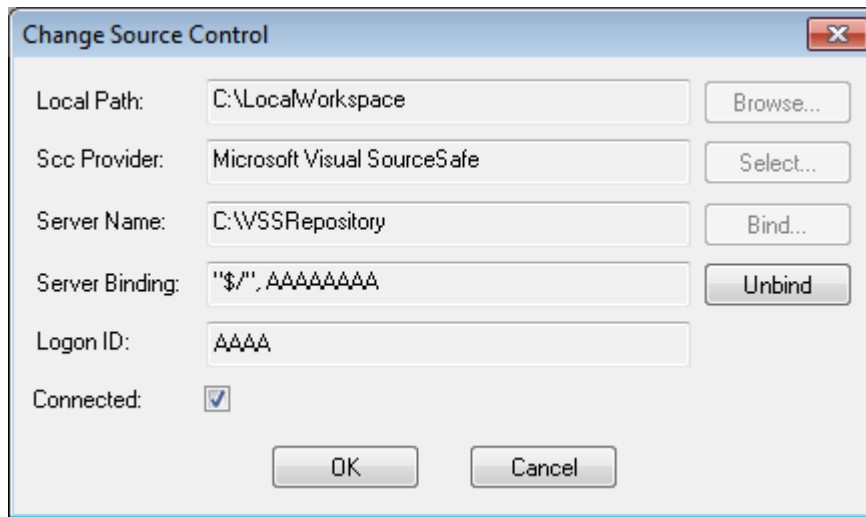
Source control settings can be changed via two commands in the **Project | Source Control** menu:

- **Source Control Manager**, which opens the source control system application and allows you to set up databases and configure bindings.
- **Change Source Control**, which pops up the Change Source Control dialog, in which you can change the source control system being used by the Altova application and the current binding. This dialog is described below.

The current binding is what the active application project will use to connect to the source control database. The current binding is correct when the application project file (.spp file) is in the local folder and the bound folder in the repository is where this project's files are stored. Typically the bound folder and its sub-structure will correspond with the local workspace folder and its sub-structure.

In the Change Source Control dialog (*screenshot below*), you can change the source control system (SCC Provider), the local folder (*Local Path*), and the repository binding (*Server Name* and *Server Binding*).

Only after undoing the current binding can the settings be changed. Undo the current binding with the **Unbind** button. All the settings are now editable.



Change source control settings as follows:

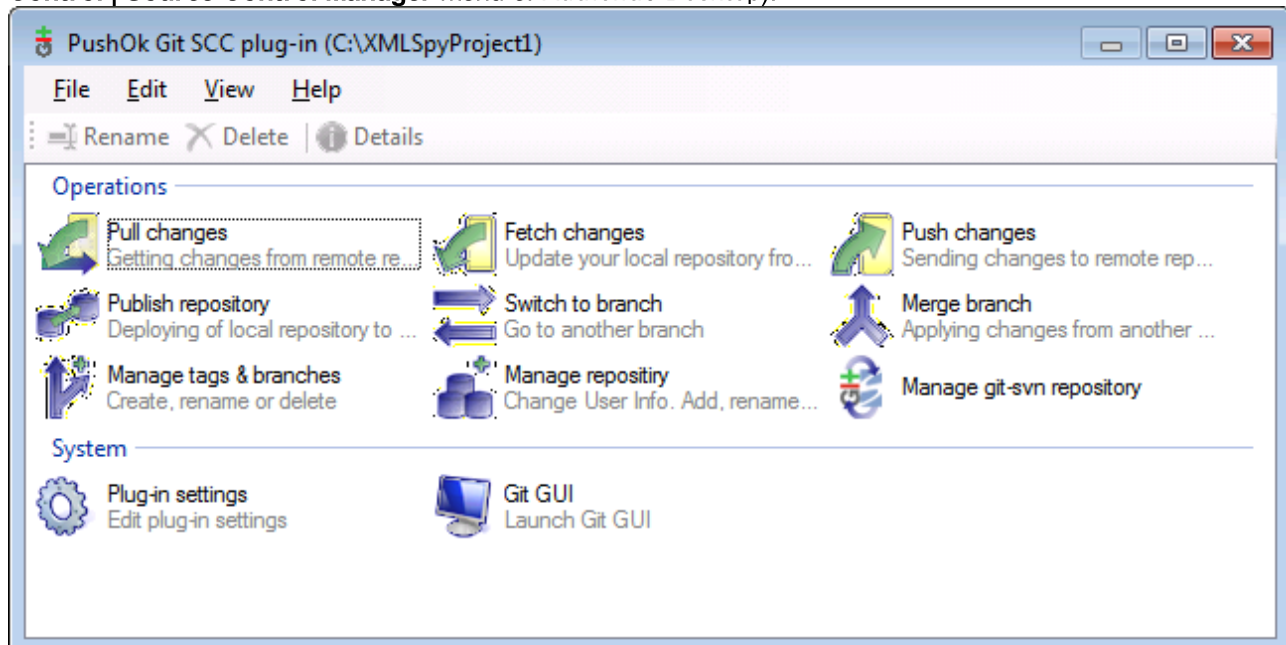
1. Use the **Browse** button to browse for the local folder and the **Select** button to select from among the installed source control systems.
2. After doing this you can bind the local folder to a repository database. Click the **Bind** button to do this. This pops up the connection dialog of your source control system.
3. If you have entered a *Logon ID*, this will be passed to the source control system; otherwise you might have to enter your logon details in the connection dialog.
4. Select the database in the repository that you wish to bind to this local folder. This setting might be spread over more than one dialog.
5. After the setting has been created, click **OK** in the Change Source Control dialog.

9.7 Source Control with Git

Support for Git as a source control system in Authentic Desktop is available through a third-party plug-in called **GIT SCC plug-in** (<http://www.pushok.com/software/git.html>).

At the time when this documentation is written, the **GIT SCC plug-in** is available for experimental use. Registration with the plug-in publisher is required in order to use the plug-in.

The GIT SCC plug-in enables you to work with a Git repository using the commands available in the **Project | Source Control** menu of Authentic Desktop. Note that the commands in the **Project | Source Control** menu of Authentic Desktop are provided by the Microsoft Source Control Plug-in API (MSSCCI API), which uses a design philosophy different from Git. As a result, the plug-in essentially mediates between "Visual Source Safe"-like functionality and Git functionality. On one hand, this means that a command such as **Get latest version** may not be applicable with Git. On the other hand, there are new Git-specific actions, which are available in the "Source Control Manager" dialog box provided by the plug-in (under the **Project | Source Control | Source Control Manager** menu of Authentic Desktop).



The Source Control Manager dialog box

Other commands that you will likely need to use frequently are available directly under the **Project | Source Control** menu.

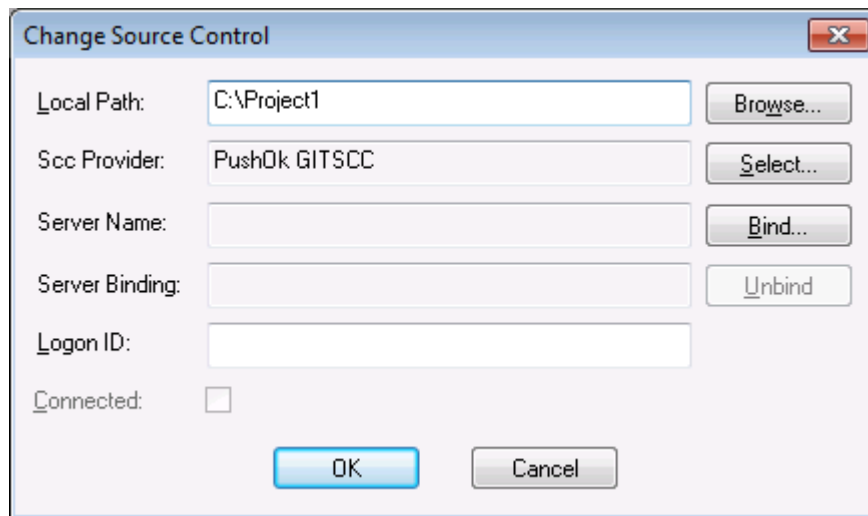
The following sections describe the initial configuration of the plug-in, as well as the basic workflow:

- [Enabling Git Source Control with GIT SCC Plug-in](#)
- [Adding a Project to Git Source Control](#)
- [Cloning a Project from Git Source Control](#)

9.7.1 Enabling Git Source Control with GIT SCC Plug-in

To enable Git source control with Authentic Desktop, the third-party **PushOK GIT SCC plug-in** must be installed, registered, and selected as source control provider, as follows:

1. Download the plug-in installation file from the publisher's website (<http://www.pushok.com>), run it, and follow the installation steps.
2. On the **Project** menu of Authentic Desktop, click **Change Source Control**, and make sure **PushOk GITSCC** is selected as source control provider. If you do not see **Push Ok GITSCC** in the list of providers, it is likely that the installation of the plug-in was not successful. In this case, check the publisher's documentation for a solution.



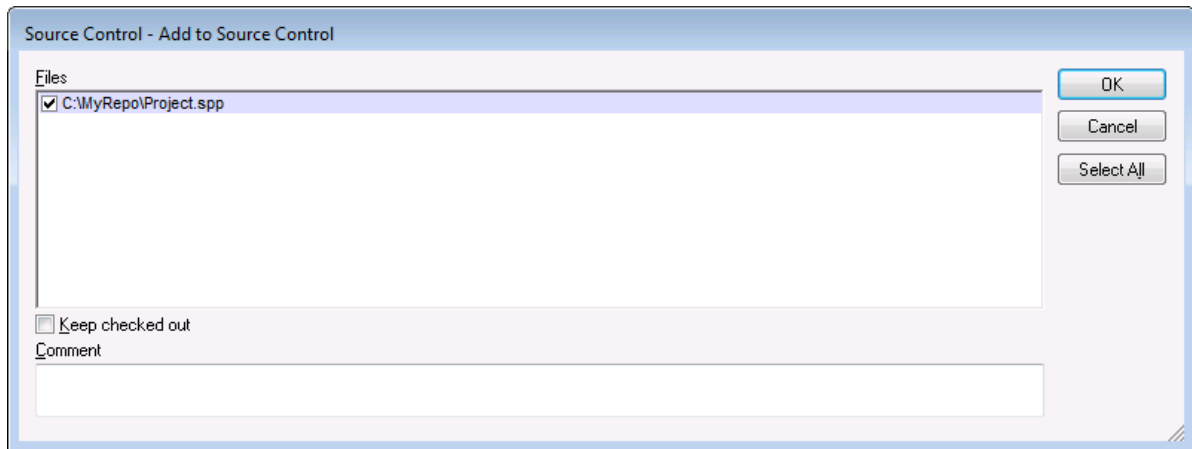
3. When a dialog box prompts you to register the plug-in, click **Registration** and follow the wizard steps to complete the registration process.

9.7.2 Adding a Project to Git Source Control

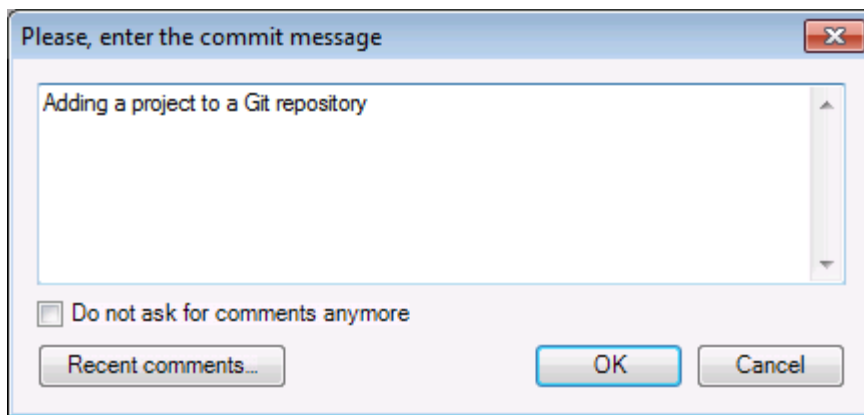
You can save Authentic Desktop projects as Git repositories. The structure of files or folders that you add to the project would then correspond to the structure of the Git repository.

To add a project to Git source control:

1. Make sure that **PushOK GIT SCC Plug-in** is set as source control provider (see [Enabling Git Source Control with GIT SCC Plug-in](#)).
2. Create a new project using the menu command **Project | Create Project**.
3. Save the project to a local folder, for example `C:\MyRepo\Project.spp`
4. On the **Project** menu, under **Source Control**, click **Add to Source Control**.

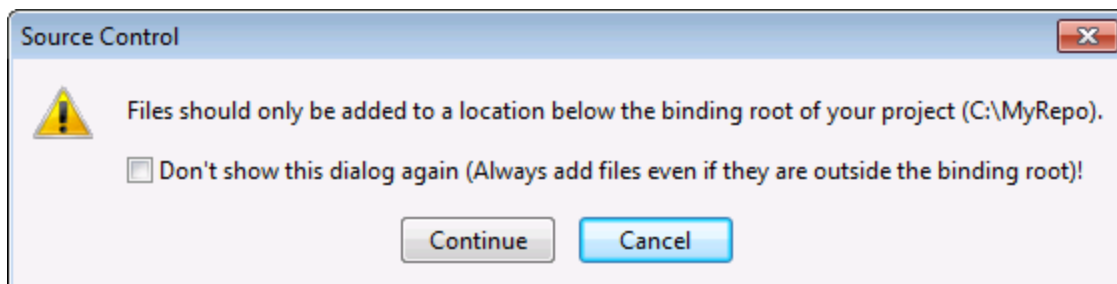


5. Click **OK**.



6. Enter the text of your commit message, and click **OK**.

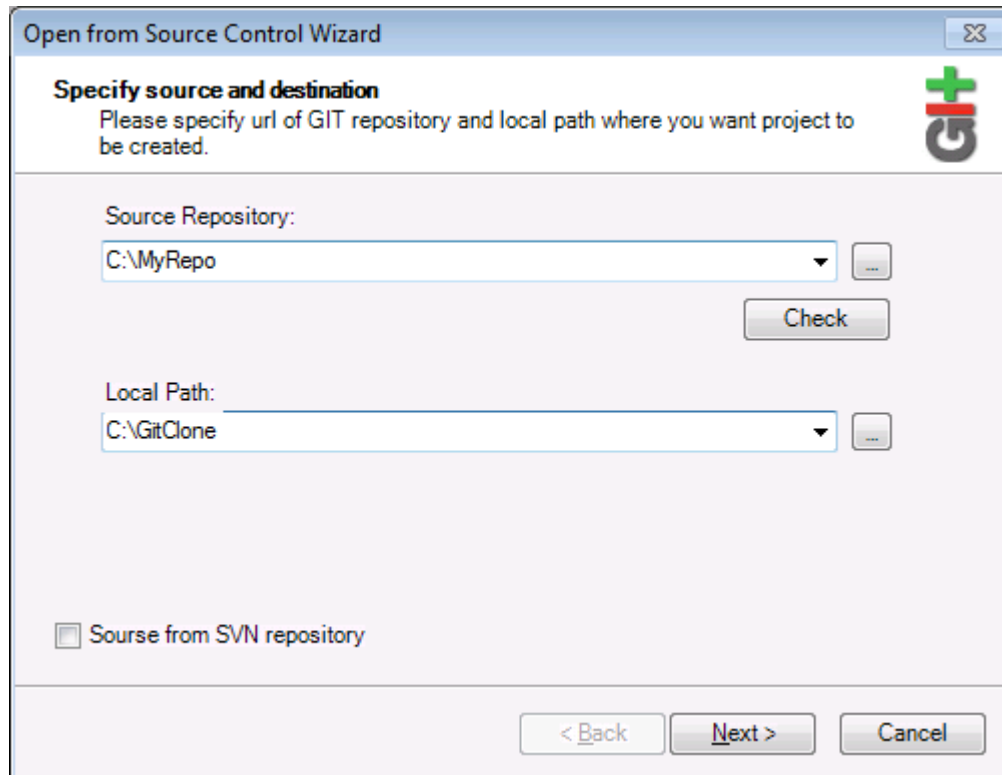
You can now start adding files and folders to your project. Note that all project files and folders must be under the root folder of the project. For example, if the project was created in the `C:\MyRepo` folder, then only files under `C:\MyRepo` should be added to the project. Otherwise, if you attempt to add to your project files that are outside the project root folder, a warning message is displayed:



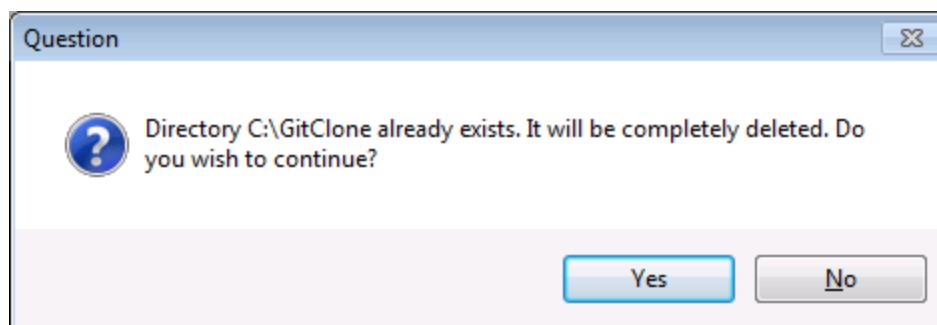
9.7.3 Cloning a Project from Git Source Control

Projects that have been previously added to Git source control (see [Adding a Project to Git Source Control](#)) can be opened from the Git repository as follows:

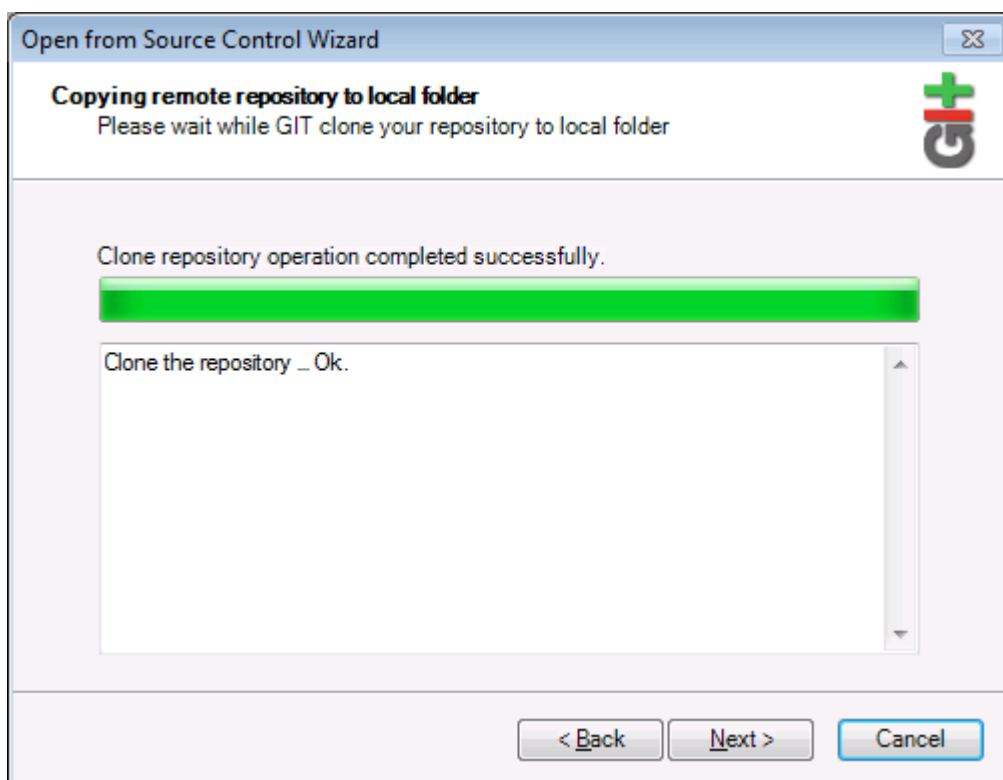
1. Make sure that **PushOK GIT SCC Plug-in** is set as source control provider (see [Enabling Git Source Control with GIT SCC Plug-in](#)).
2. On the **Project** menu, click **Source Control | Open from Source Control**.
3. Enter the path or the URL of the source repository. Click **Check** to verify the validity of the path or URL.



4. Under **Local Path**, enter the path to local folder where you want the project to be created, and click **Next**. If the local folder exists (even if it is empty), the following dialog box opens:



5. Click **Yes** to confirm, and then click **Next**.



6. Follow the remaining wizard steps, as required by your specific case.
7. When the wizard completes, a Browse dialog box appears, asking you to open the Authentic Desktop Project (*.spp) file. Select the project file to load the project contents into Authentic Desktop.

10 メニューコマンド

ユーザーファレンスは、Authentic Desktop のメニューコマンドに関する全てが記述されており、それらの一般的な使用方法について説明します。網羅的な内容になるよう努力はしておりますが、ユーザーファレンスで扱われていない事柄に関する質問などありましたら、Altova Web サイトのFAQ またはディスカッションフォーラムを参照ください。適切な答えが見つからない場合、[Altova サポートセンター](#)にお問い合わせください。

準的な Windows コマンド(開く、保存、切り取り、コピー、貼り付けなど)は、[ファイル](#)と[編集](#)メニューに収められています。これらのメニューにはXML やインターネットに関するコマンドも収められています。

10.1 ファイルメニュー


ファイルメニューには、一般的な Windows ソフトウェア製品で使われるような [新規作成](#)、[開く](#)、[保存](#)、[印刷](#)、[印刷設定](#)、と [終了](#) などのファイルの操作を行うためのコマンドが収められています。Authentic Desktop には、XML 固有およびアプリケーション固有のコマンドが含まれています。

10.1.1 新規作成

このセクション:

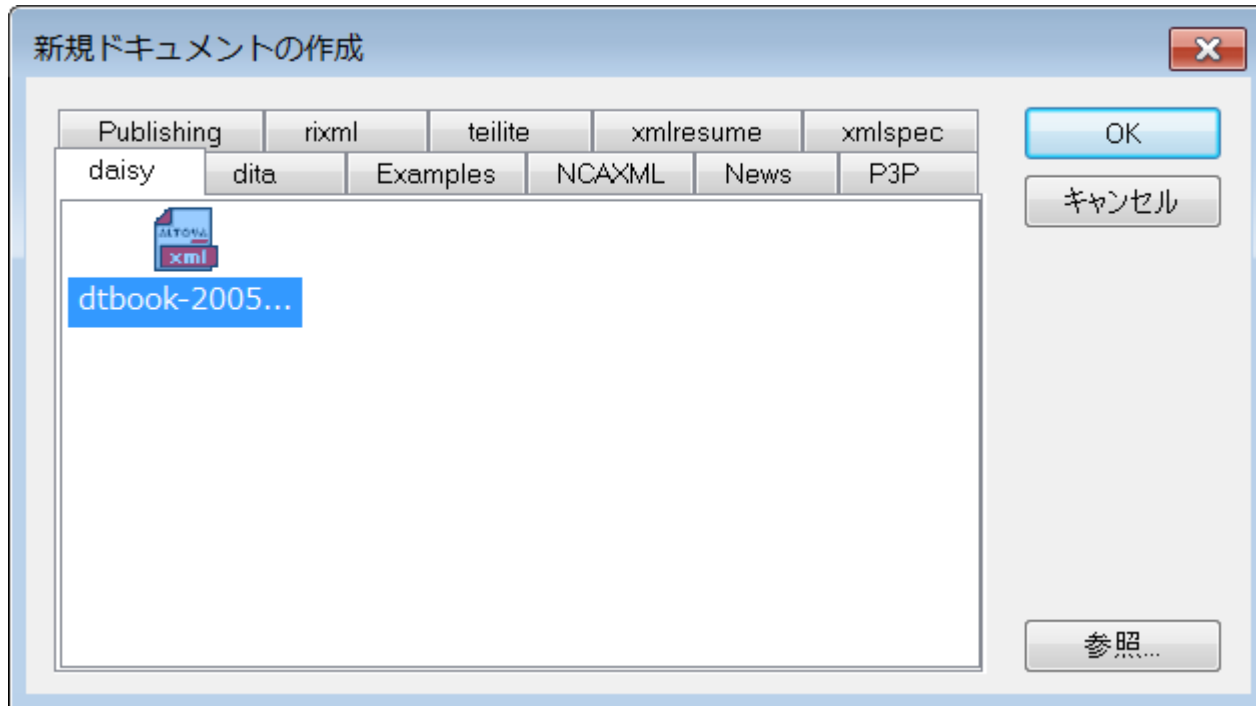
- [アイコンとショートカット](#)
- [説明](#)

アイコンとショートカット

アイコン:	
ショートカット:	Ctrl+N

説明

このコマンドにより Authentic View 内で新規の XML ドキュメントテンプレートを開くことができます。XML ドキュメントテンプレートは、StyleVision Power Stylesheet (.sps ファイル) をベースとしています。新規のドキュメントの作成 ダイアログ内で StyleVision Power Stylesheet (SPS ファイル) を選択する事により開かれます (下のスクリーンショット)。SPS を選択して、「OK」をクリックすると、SPS ファイルのために定義された XML ドキュメントテンプレートが Authentic View 内で開かれます。



新規のドキュメントの作成 ダイアログは、一般的に使用される DTD またはスキーマをベースとした XML ドキュメントテンプレートの選択を提供します。または、テンプレート XML ファイルが割り当てられているカスタムメイドの SPS ファイルを参照することができます。SPS ファイルは、Altova StyleVision を使用して作成されます。アプリケーションにより、DTD または XML スキーマをベースとした XML ドキュメントテンプレートをデザインすることができます。StyleVision 内で必要な SPS をデザインすると、XML ファイルは (StyleVision 内で SPS へのテンプレート XML ファイルとして割り当てられます。XML ファイル内のデータは、Authentic Desktop の Authentic View 内で開かれた新規のドキュメントテンプレートの開始データを与えます。

新規の XML ドキュメントテンプレートは、ですから、テンプレート XML ファイルとして選択された SPS と XML ファイルのデータ内で定義されたドキュメントプレゼンテーションプロパティを持つようになります。Authentic View ユーザーは、XML ドキュメントテンプレートをグラフィカルな WYSIWYG インターフェイスで編集することができ、XML ドキュメントとして保存することができます。

10.1.2 開く

アイコンとショートカット

アイコン:	
ショートカット:	Ctrl+O

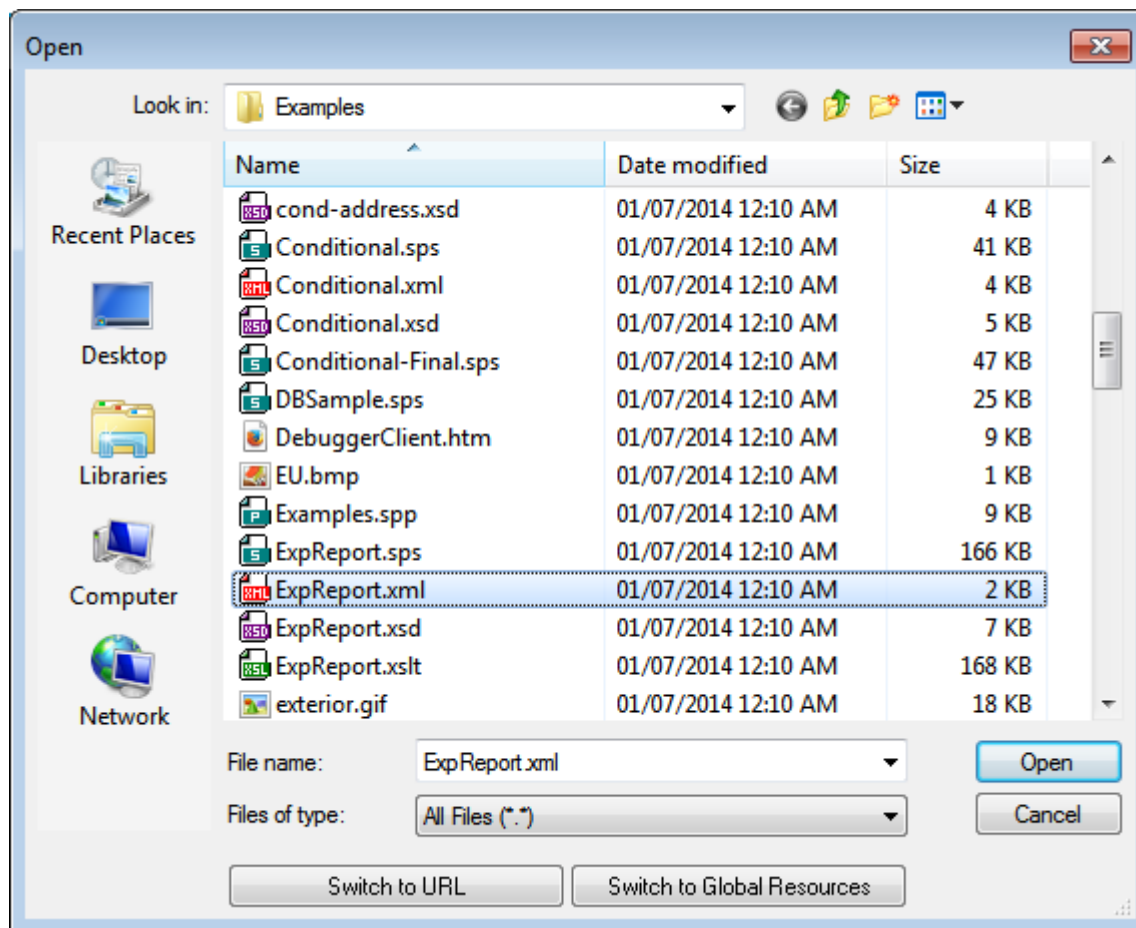
説明

開くコマンドにより、慣れ親しんだ Windows の開くダイアログが表示され、XML に関するドキュメントや、テキストドキュメントを開くことができます。開くダイアログでは、2つ以上のファイルを開くことができます。ファイルの種類コンボボックスを使用することで、ダイアログボックスに表示されるファイルの型セグメントを制限することができます (表示されるファイルの種類一覧は、オプションダイアログ ([ツール | オプション](#)) にて設定することができます)。XML ファイルが開かれると、整形形式のチェックが行われます。ファイルが整形形式で無い場合、整形形式に関するエラーが表

示されます。エラー箇所を修正して、メニューコマンドから [XML | 整形形式のチェック \(F7\)](#) を選択します。 [ファイルを開く再自動検証](#) を行うよう選択しており、ファイルが妥当でない場合、エラーメッセージが表示されます。エラー箇所を修正して、メニューコマンドから [XML | XML の検証 \(F8\)](#) を選択して、再度検証を行います。

▼ URL とグローバルリソースを使用してファイルを選択または保存する

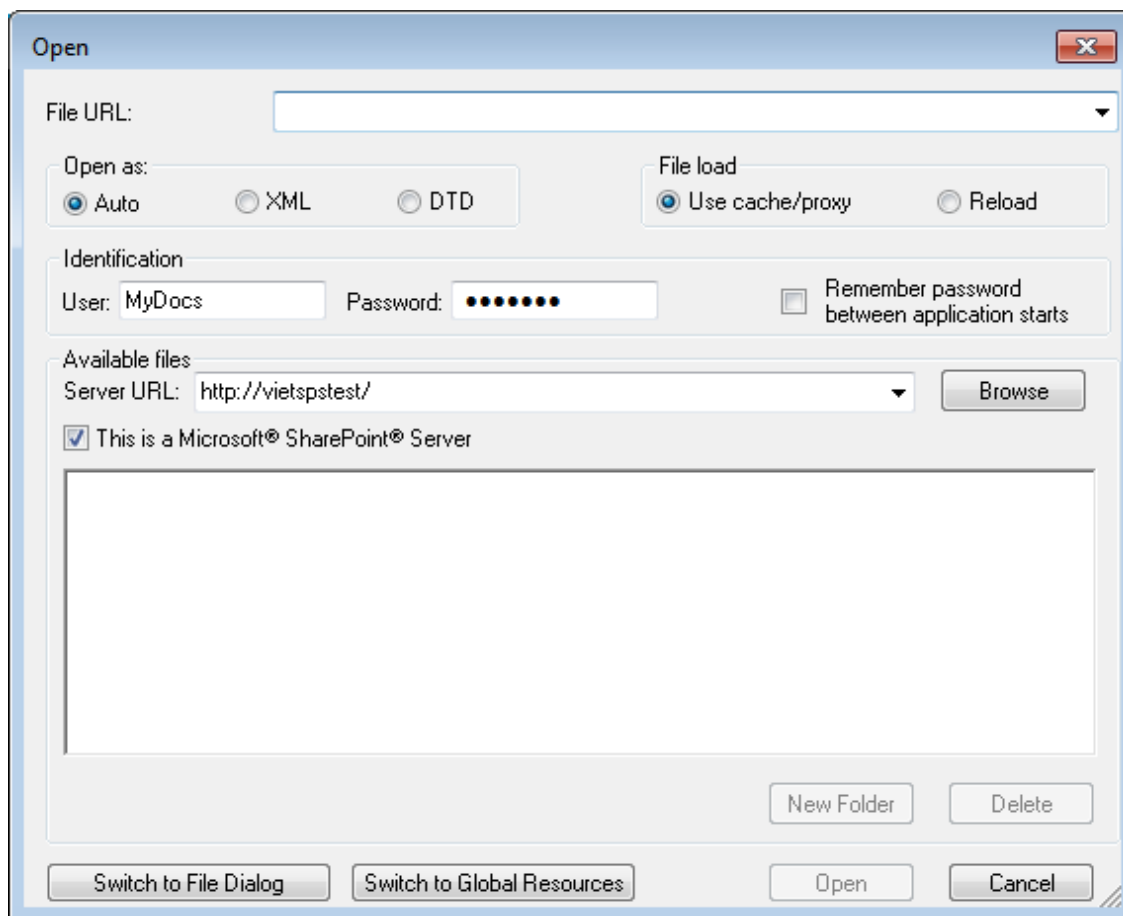
「ファイルを開く」および「ファイルの保存」ダイアログでは、URL またはグローバルリソースを使用して必要なファイルを選択またはファイルを保存することができます (下のスクリーンショット参照)。選択プロセスを移動するために「URL に切り替える」または「グローバルリソース」をクリックしてください。



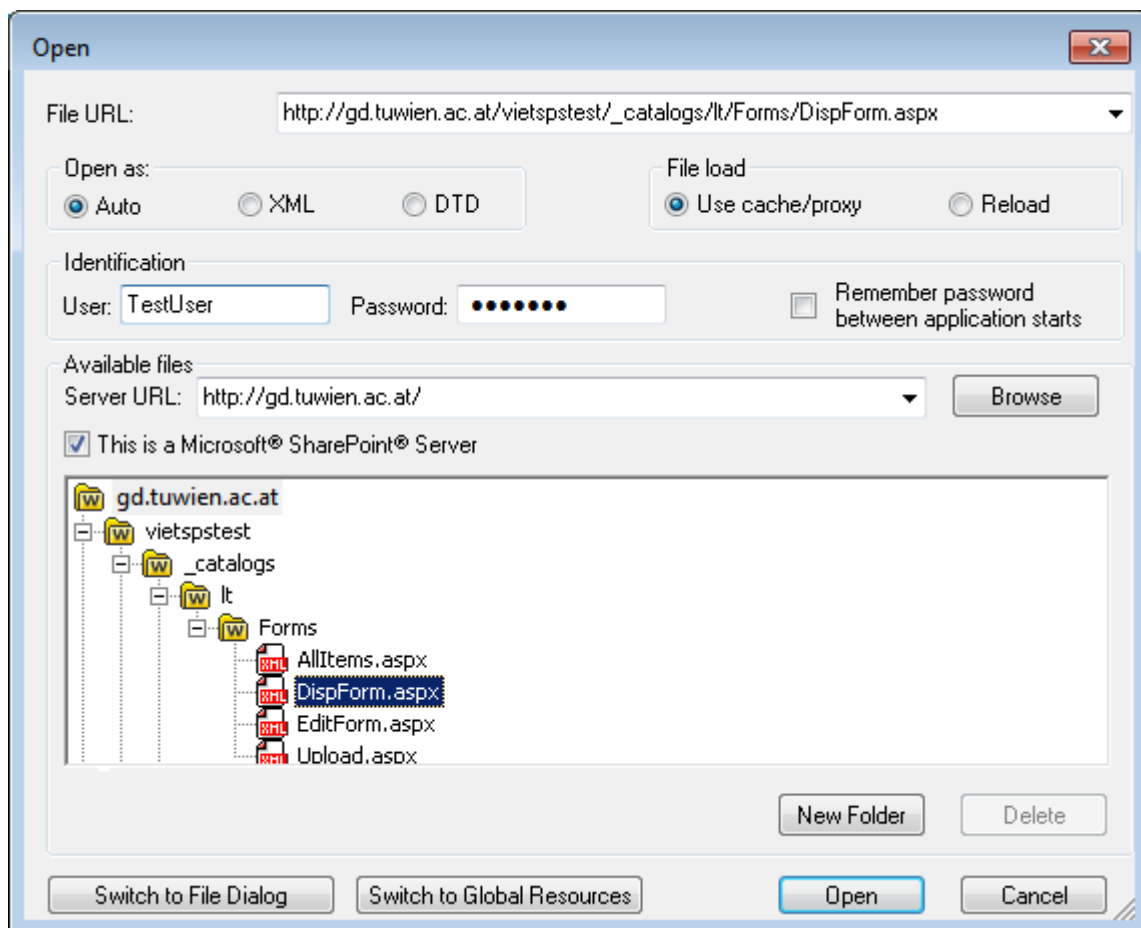
URL を使用してファイルを切り替える

URL から開くまたは保存するファイルを選択するご以下を行ってください

1. 「URL に切り替える」コマンドをクリックします。これにより、開くまたは保存ダイアログの URL モードにスイッチされます。(下のスクリーンショットは開くダイアログを表示していません)。



2. アクセスする URL をサーバー URL フィールドに入力します（上のスクリーンショット）。サーバーが Microsoft® SharePoint® Server の場合 Microsoft® SharePoint® Server チェックボックスをチェックしてください。この種類のサーバー上にあるファイルと作業する場合、下の Microsoft® SharePoint® Server メモを参照してください。
3. サーバーがパスワードにより保護されている場合、ユーザーとパスワードフィールドに入力してください。
4. 「参照」をクリックして、サーバーのディレクトリ構造をナビゲートします。
5. フォルダーツリー内でロードするファイルを参照してクリックします。



ファイルURL がファイルURL フィールドに表示されます(上のスクリーンショット参照)。「開く」または「保存」ボタンを使用できるようになります。

6. 「開く」をクリックしてファイルをロードするか、「保存」をクリックして保存します。

以下の点に注意してください!

- WebDAV をサポートするサーバーとMicrosoft SharePoint Servers のみで参照機能を使用することができます。サポートされるプロトコールはFTP、HTTP およびHTTPS です。
- ファイルを開く際のロード処理を更に管理するには、ローカルのキャッシュまたはファイルが以前ロードされている場合スピードを向上するプロキシのサーバーからファイルをロードします。(ファイルが事前にロードされることにより更に処理スピードを向上することができます)。また、電子出版またはデータベースシステムなどの作業中のファイルを再ロードして、「再ロード」オプションを選択します

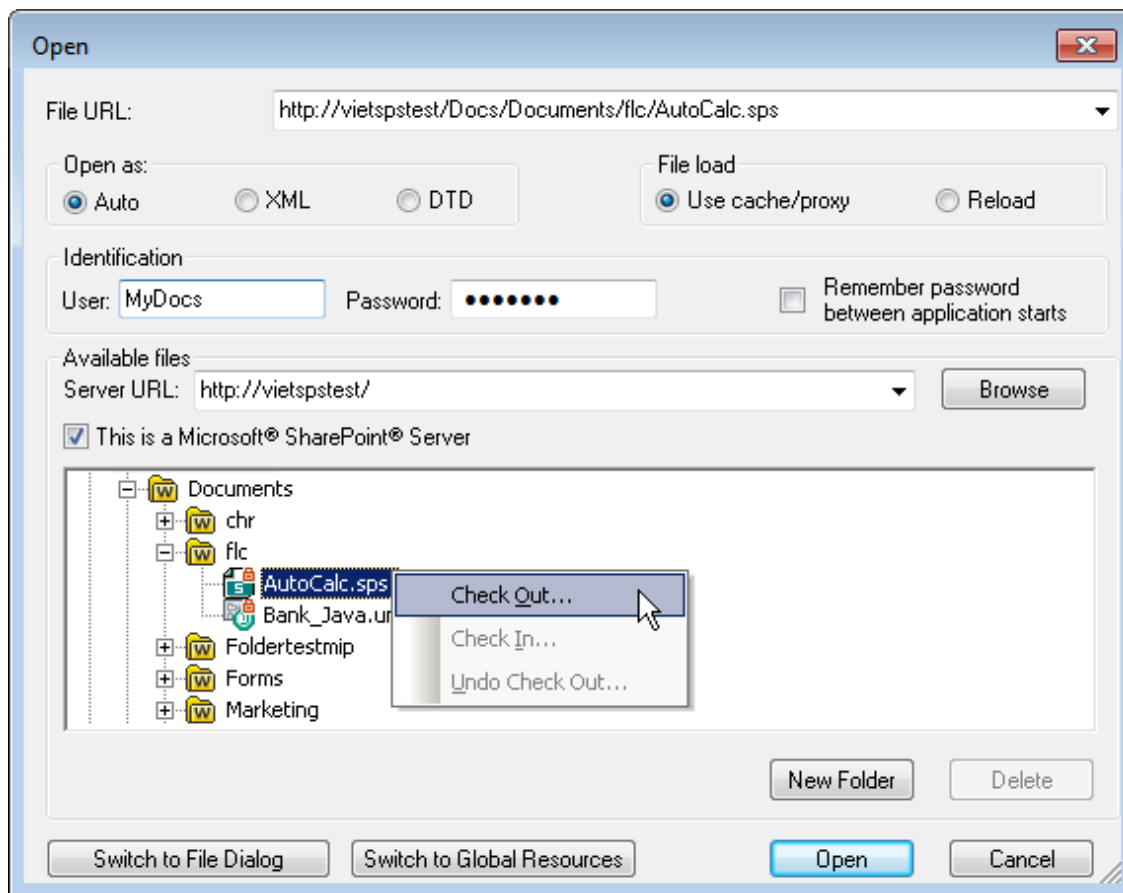
。

▼ Microsoft® SharePoint® Server への

Microsoft® SharePoint® Servers に関する以下の点に注意してください!

- 「利用可能なファイル」ペインに表示されるディレクトリ構造では、ファイルアイコンはファイルのチェックイン/チェックアウトの状

態を表示するシンボルが異なります(下のスクリーンショット)。

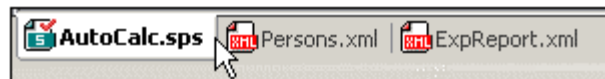


ファイルを右クリックすると、ファイルで使用することのできるコンテキストメニューがポップアップします(上のスクリーンショット)。

- 異なるファイルアイコンが以下に表示されます:

	チェックイン状態。
	チェックアウト可能。他のユーザーによりチェックアウト状態。
	チェックアウト不可能。ローカルに対してチェックアウト。

- ファイルをチェックアウトした後、Altova アプリケーションを使用して編集することができ、「ファイル|保存 (Ctrl+S)」を使用して保存することができます。
- 編集されたファイルを「URLを開く」ダイアログのコンテキストメニューからチェックインすることができます(上のスクリーンショット参照) または、アプリケーションのメインウィンドウのファイルタブを右クリックすることによりポップアップするコンテキストメニューを使用して(下のスクリーンショット)。



- ファイルが他のユーザーによりチェックアウトされている場合、チェックアウトすることができません。
- 他のユーザーによりファイルがチェックアウトされている場合、チェックアウトを行うことはできません。これによりファイルを変更す

ることなくサーバーに戻すことができます。

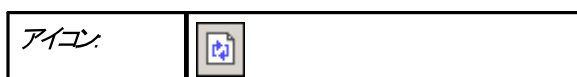
- 変更することなしにファイルがサーバーへ戻されます。ある Altova 製品でファイルをチェックアウトした場合、他の Altova 製品で同じファイルをチェックアウトすることできません。この時点で使用できる Microsoft® SharePoint® Server をサポートする Altova アプリケーションのコマンドは、以下の通りです: 「チェックイン」 および 「チェックアウトを元に戻す」 です。

▼ グローバルリソースを使用してファイルを選択または保存する

グローバルリソースを使用してファイルを開くまたは保存する場合、「グローバルリソース」をクリックします。グローバルリソースを選択するためのダイアログが表示されます。これらのダイアログは以下のセクションで説明されます: [グローバルリソースの使用](#)。グローバルリソースの説明に関してはこのドキュメントの [グローバルリソース](#) セクションを参照してください。

10.1.3 再ロード

アイコン

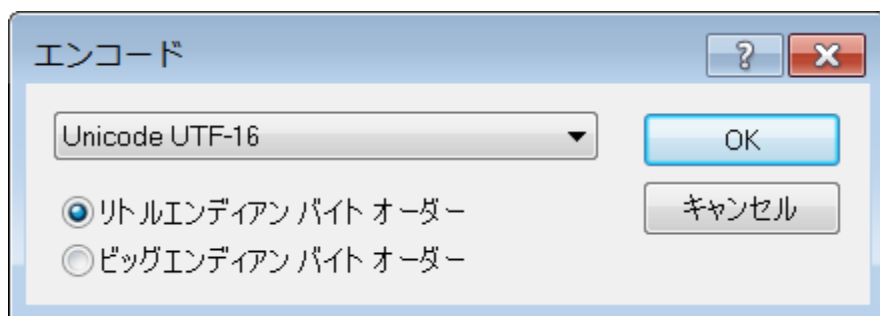


説明

Authentic Desktop 外で変更され開かれているドキュメントを再ロードします。Authentic Desktop 外でドキュメントが変更されると変更されたドキュメントの再ロードを問うプロンプトが表示されます。再ロードを選択すると、ファイルの最後に保存された時点からの変更が失われます。

10.1.4 エンコード

エンコードコマンドにより、(XML 以外も含む) 現在アクティブなドキュメントにて使用されているエンコーディングを確認し、他のエンコーディングを選択して、次の保存時に反映させることができます。



XML ドキュメントにおいて、それまで使われていたものと異なるエンコーディングを指定すると、XML 宣言内のエンコーディング指定もそれに合わせて変更されます。2 バイト文字ならびに 4 バイト文字エンコーディング (UTF-16、UCS-2、UCS-4) では、ファイルで使用される バイト オーダーを指定することができます。XML ドキュメントのエンコーディングを変更するもう 1 つの方法は、ドキュメントの XML 宣言で使用されて

いる encoding 属性を直接編集することです。既存ならびに新規 XML ドキュメント、または非 XML ドキュメントのデフォルト エンコーディングは、[オプションダイアログのエンコードセクション](#)にて指定することができます。

メモ: ドキュメントを保存する際、Authentic Desktop はエンコーディングの指定を自動的にチェックして、ユーザーにより入力されたエンコーディングが認識できない場合は、ダイアログを表示します。更に、選択されたエンコーディングにより表現することができない文字がドキュメントに含まれている場合、ファイルを保存する際に警告メッセージが表示されます。

10.1.5 閉じる、全て閉じる、非アクティブを全て閉じる

閉じる

閉じるコマンドにより、アクティブなドキュメントウィンドウが閉じられます。ファイルが変更されている場合(タイトルバーに表示されるファイル名の後にアスタリスク * が表示されます)、まずファイルを保存するか尋ねられます。

全て閉じる

全て閉じるコマンドにより、開かれているドキュメントウィンドウが全て閉じられます。ドキュメントのどれかが変更されている場合、まずファイルを保存するか尋ねられます。

非アクティブを全て閉じる

非アクティブを全て閉じるコマンドにより、アクティブなドキュメントウィンドウ以外の、全ウィンドウが閉じられます。ドキュメントのどれかが変更されている場合、まずファイルを保存するか尋ねられます。

10.1.6 保存、名前を付けて保存、全て保存

アイコンとショートカット

コマンド	アイコン	ショートカット
保存		Ctrl+S
名前を付けて保存		

保存

保存 (Ctrl+S) コマンドにより、アクティブなドキュメントのコンテンツを開かれた元のファイルに保存されます。ドキュメントを保存する際、[整形形式が自動的にチェック](#)されます。オプションダイアログ([ツール | オプション](#)) のファイルセクションにてオプションが指定されているか、ファイルの検証も同時に実行されます。XML 宣言の [エンコード](#) 指定もチェックされ、ファイルの保存時にエンコーディングがドキュメントに対して適用されます。

名前をつけて保存

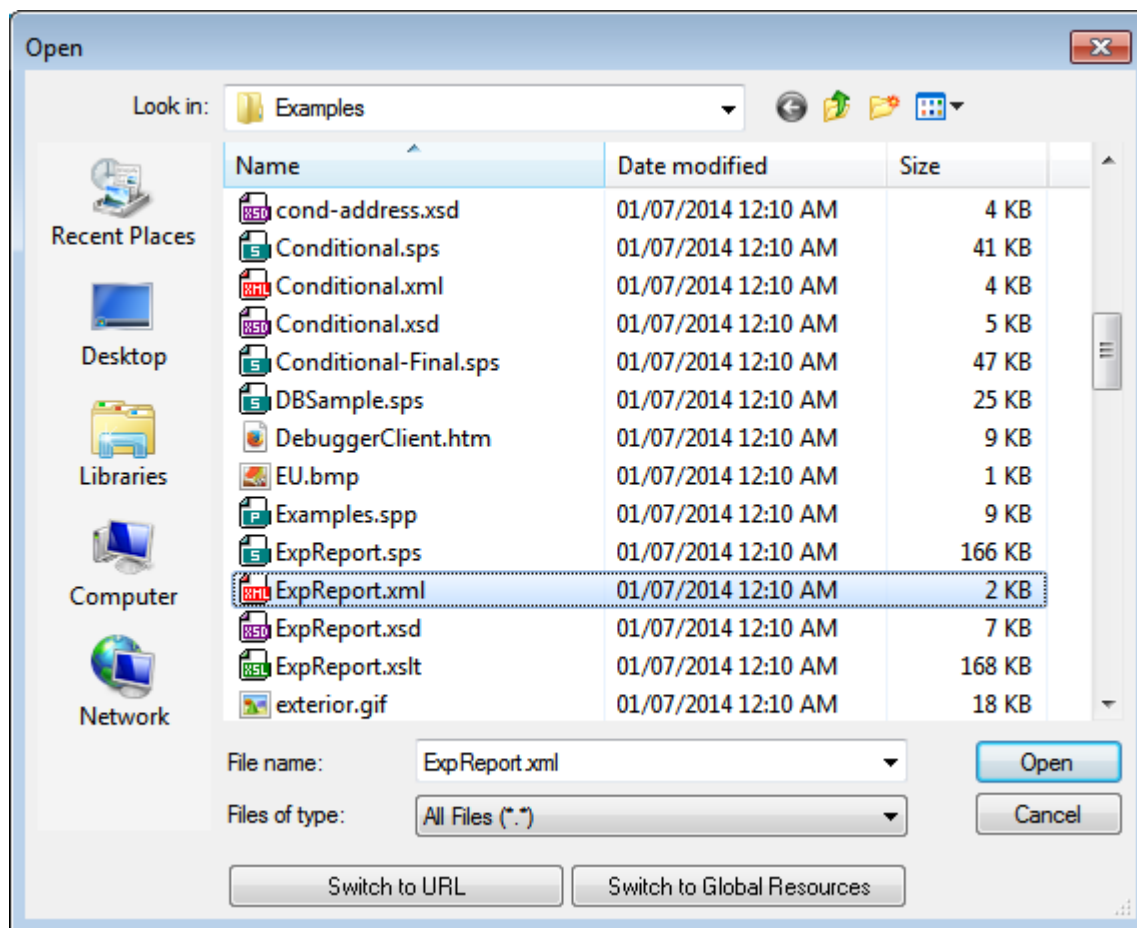
名前をつけて保存コマンドを選択すると、馴染みのある Windows 名前をつけて保存ダイアログボックスが表示され、アクティブなドキュメントの内容を保存するファイルの場所と名前を入力することになります。保存コマンドに対して行われるものと同じチェックがここでも行われます。

全て保存

全て保存 コマンドでは、開かれているドキュメントに対して行われた全ての変更を保存します。このコマンドは、複数のドキュメントを同時に保存する際に便利な機能となります。まだドキュメントが保存されていない場合（例えば、ファイルが新たに作成された後、まだ保存されていない場合）、そのドキュメントに対して名前を付けて保存ダイアログボックスが表示されます。

▼ URL とグローバルリソースを使用してファイルを選択または保存する

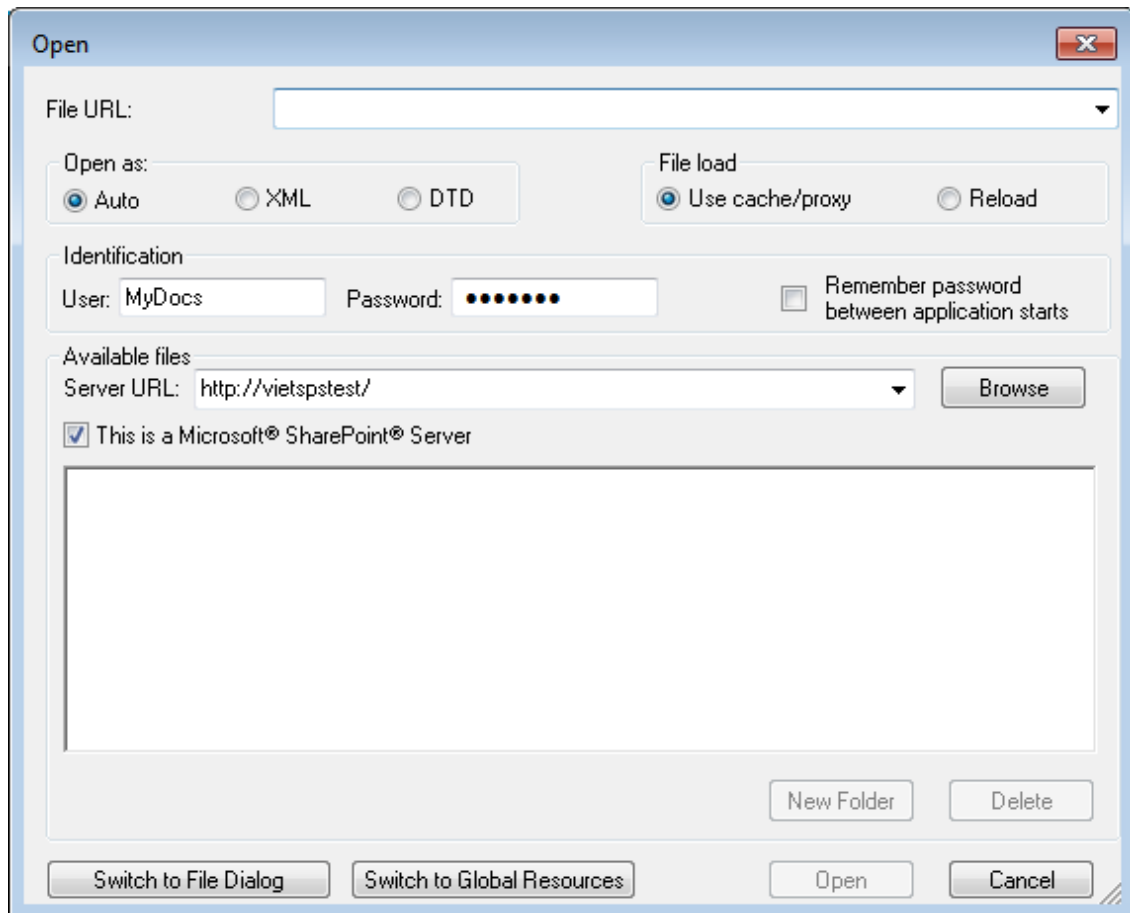
「ファイルを開く」および「ファイルの保存」ダイアログでは、URL またはグローバルリソースを使用して必要なファイルを選択またはファイルを保存することができます（下のスクリーンショット参照）。選択プロセスを移動するために「URL に切り替える」または「グローバルリソース」をクリックしてください。



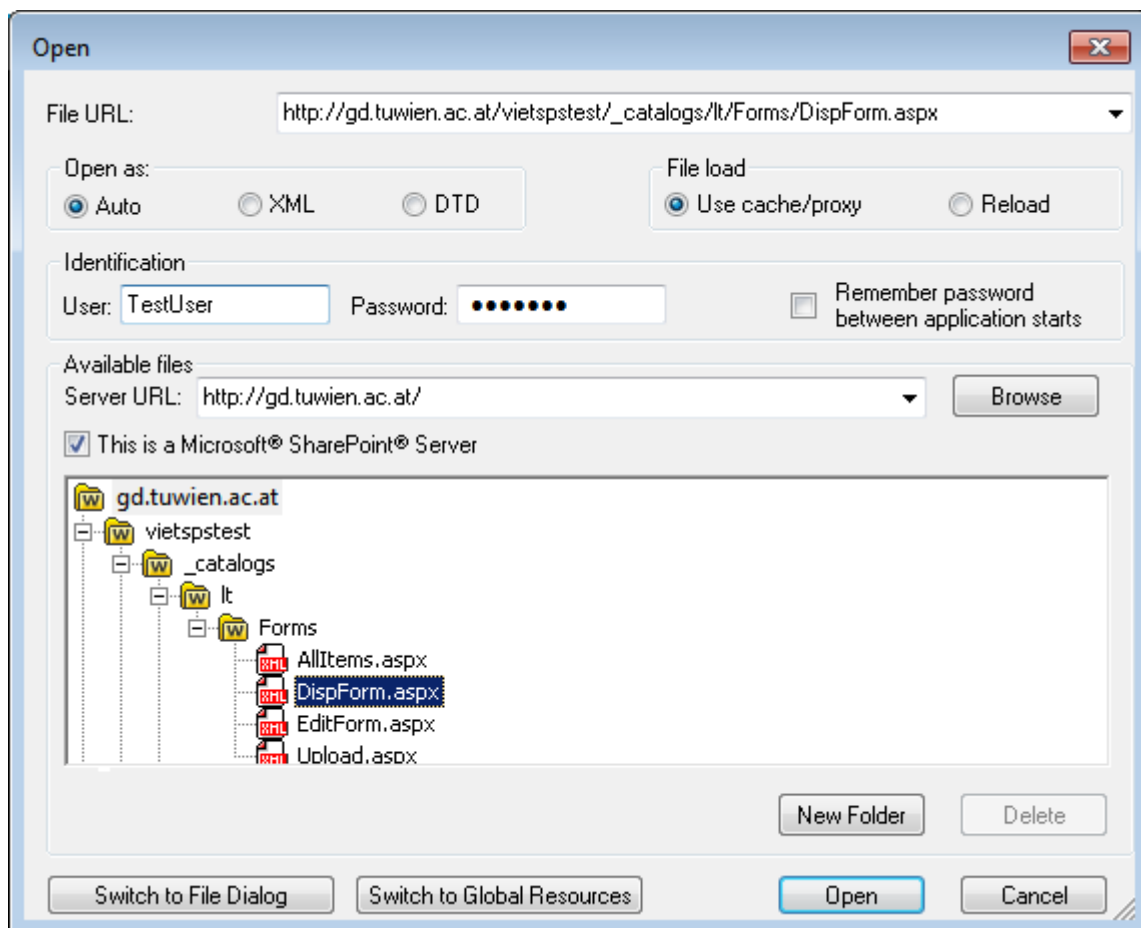
URL を使用してファイルを切り替える

URL から開くまたは保存するファイルを選択するご以下を行ってください！

1. 「URL に切り替える」コマンドをクリックします。これにより、開くまたは保存ダイアログの URL モードにスイッチされます。（下のスクリーンショットは開くダイアログを表示していません）。



2. アクセスする URL をサーバー URL フィールドに入力します（上のスクリーンショット）。サーバーが Microsoft® SharePoint® Server の場合 Microsoft® SharePoint® Server チェックボックスをチェックしてください。この種類のサーバー上にあるファイルと作業する場合、下の Microsoft® SharePoint® Server メモを参照してください。
3. サーバーがパスワードにより保護されている場合、ユーザーとパスワードフィールドに入力してください。
4. 「参照」をクリックして、サーバーのディレクトリ構造をナビゲートします。
5. フォルダーツリー内でロードするファイルを参照してクリックします。



ファイルURL がファイルURL フィールドに表示されます(上のスクリーンショット参照)。「開く」または「保存」ボタンを使用できるようになります。

6. 「開く」をクリックしてファイルをロードするか、「保存」をクリックして保存します。

以下の点に注意してください!

- WebDAV をサポートするサーバーとMicrosoft SharePoint Servers のみで参照機能を使用することができます。サポートされるプロトコルはFTP、HTTP およびHTTPS です。
- ファイルを開く際のロード処理を更に管理するには、ローカルのキャッシュ またはファイルが以前ロードされている場合スピードを向上するプロキシのサーバーからファイルをロードします。(ファイルが事前にロードされることにより更に処理スピードを向上することができます)。また、電子出版またはデータベースシステムなどの作業中のファイルを再ロードして、「再ロード」オプションを選択します

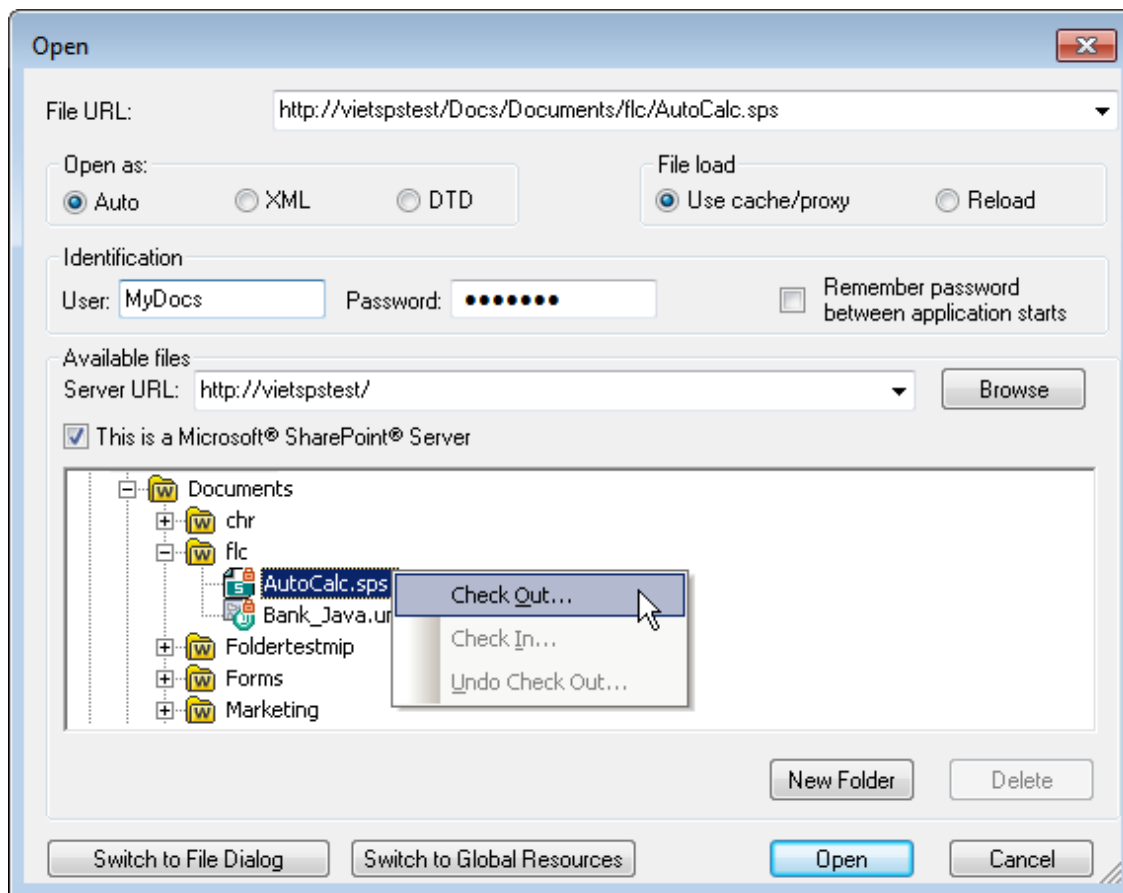
。

▼ Microsoft® SharePoint® Server への

Microsoft® SharePoint® Servers に関する以下の点に注意してください!




- 「利用可能なファイル」ペインに表示されるディレクトリ構造では、ファイルアイコンはファイルのチェックイン/チェックアウトの状

態を表示するシンボルが異なります(下のスクリーンショット)。

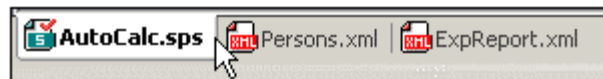


ファイルを右クリックすると、ファイルで使用することのできるコンテキストメニューがポップアップします(上のスクリーンショット)。

- 異なるファイルアイコンが以下に表示されています:

	チェックイン状態。
	チェックアウト可能。他のユーザーによりチェックアウト状態。
	チェックアウト不可能。ローカルに対してチェックアウト。

- ファイルをチェックアウトした後、Altova アプリケーションを使用して編集することができ、「ファイル|保存 (Ctrl+S)」を使用して保存することができます。
- 編集されたファイルを「URLを開く」ダイアログのコンテキストメニューからチェックインすることができます(上のスクリーンショット参照) または、アプリケーションのメインウィンドウのファイルタブを右クリックすることによりポップアップするコンテキストメニューを使用して(下のスクリーンショット)。



- ファイルが他のユーザーによりチェックアウトされている場合、チェックアウトすることができません。
- 他のユーザーによりファイルがチェックアウトされている場合、チェックアウトを行うことはできません。これによりファイルを変更す

ることなくサーバーに戻すことができます。

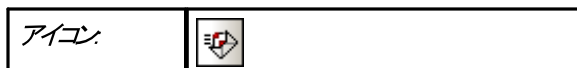
- 変更することなしにファイルがサーバーへ戻されます。ある Altova 製品でファイルをチェックアウトした場合、他の Altova 製品で同じファイルをチェックアウトすることできません。この時点で使用できる Microsoft® SharePoint® Server をサポートする Altova アプリケーションのコマンドは、以下の通りです: 「チェックイン」 および 「チェックアウトを元に戻す」 です。

▼ グローバルリソースを使用してファイルを選択または保存する

グローバルリソースを使用してファイルを開くまたは保存する場合、「グローバルリソース」をクリックします。グローバルリソースを選択するためのダイアログが表示されます。これらのダイアログは以下のセクションで説明されています: [グローバルリソースの使用](#)。グローバルリソースの説明に関してはこのドキュメントの [グローバルリソース](#) セクションを参照してください。

10.1.7 メールで送信

アイコン

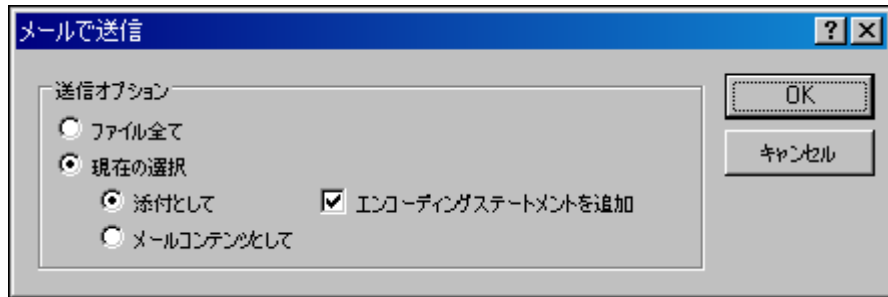


説明

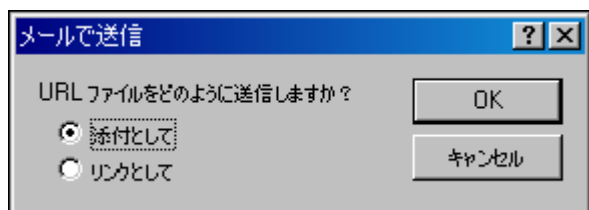
「電子メールにて送信」コマンドにより XML ドキュメント、または XML ドキュメントからの選択を電子メールにて送信することができます。ドキュメント、またはドキュメントの選択の種類により、ドキュメントを添付、コンテンツ、またはリンクとして送信することができます。詳細に関しては以下を参照してください。

送信できる内容	送信の方法
アクティブな XML ドキュメント	電子メールの添付
アクティブな XML ドキュメント内の選択	電子メールの添付または電子メールの内容として送信
プロジェクトウィンドウ内の 1 つまたは複数のファイル	電子メールの添付
プロジェクトウィンドウ内の 1 つまたは複数の URL	電子メールの添付またはリンクとして送信

「電子メールにて送信」コマンドがアクティブな XML ドキュメント内で呼び出されると、「電子メールにて送信」ダイアログがポップアップし、スクリーンショット内に送信のオプションが表示されます (下のスクリーンショット)。アクティブなファイル内でテキストが選択されておらず、「電子メールにて送信」コマンドが呼び出されると、ファイル全てがラジオボタンのみ有効化され、他のオプションは無効化されます (下のスクリーンショットを参照してください)。



プロジェクトウィンドウから送信されるファイルは、電子メールの添付のみで送信されるため、電子メールで送信ダイアログはスキップされ、選択されたファイルが添付された電子メールが開かれます。プロジェクトウィンドウ内の URL は、添付またはリンクとして送信することができます（下のスクリーンショットを参照）。URL を送信する方法を選択し、「OK」をクリックします。



10.1.8 印刷

アイコンとショートカット

アイコン:	
ショートカット:	Ctrl+P

説明

印刷コマンドにより、印刷ダイアログボックスが表示され、プリンターのオプションを指定することができます。現在アクティブなドキュメントを、表示されているまま印刷することができます。

10.1.9 印刷プレビュー、印刷設定

印刷プレビュー

テキストビューとブラウザビュー内の印刷プレビューコマンドをクリックすると、ボタンをクリックすると、現在アクティブなドキュメントの印刷プレビューが表示されます。

拡大や縮小ボタンを使用することで、プレビューの表示を最適化することができます。ページ全体が表示されている状態であれば、1ページ / 2ページボタンにより、同時に表示されるページの数を切り替えることができます。次のページならびに前のページボタンにより、ページ間の移動を行うことができます。ツールバーには前ページの印刷やプレビューウィンドウを閉じるオプションも用意されています。

メモ: 印刷プレビューにて背景色やイメージを有効にするには、(i) Internet Explorer のツールメニューにて、インターネットオプションを選択し、詳細設定タブをクリックし、(ii) 設定以下にある印刷にて、背景の色とイメージを印刷する チェックボックスにチェックを入れ、(iii) 「OK」をクリックします。

印刷設定

印刷設定コマンドでは、プリンターに関する印刷設定ダイアログボックスが表示され、用紙のフォーマットや印刷の向きと、設定を行うことができます。これらの設定は、その後行われる印刷ジョブにも引き継がれます。



10.1.10 最近使用されたファイル、終了

最近使用されたファイル

「ファイル」メニューの一番下に最近使用した順に9つのファイルのリストが表示されます。名前をクリックしてファイルを開くことができます。キーボードを使用してファイルを開く場合、「Alt+F」を押して「ファイル」メニューを開き、開く番号を押します。

終了

Authentic Desktop を終了します。保存されていない変更のある開かれているファイルがある場合、これらの変更を保存するようプロンプトされます。Authentic Desktop はプログラム設定への変更と最近使用されたファイルの情報も保存します。

10.2 編集メニュー

編集メニューには、Authentic Desktop にて開かれているドキュメントの編集を行うためのコマンドが用意されています。標準的な [元に戻す](#)、[やり直し](#)、[切り取り](#)、[コピー](#)、[貼り付け](#)、[削除](#)、[全て選択](#)、[検索](#)、[次を検索](#) および [置換](#) コマンドなどが用意されています。

10.2.1 元に戻す、やり直し

アイコンとショートカット

コマンド	アイコン	ショートカット
元に戻す		Ctrl+Z
やり直し		Ctrl+Y

元に戻す





元に戻すコマンドでは、回数に制限の無いアンドウ(作業の取り消し)がサポートされます。この機能を使って行われた全ての操作を取り消すことができます。元に戻すコマンドの記録は、ドキュメントが保存されても蓄えられ、変更を保存する前の状態まで操作を取り消すことができます。元に戻すとやり直しコマンドを使用して履歴に戻り、また、先に進むことができます(やり直しコマンドを参照してください)。

やり直し

やり直しコマンドでは、それまで取り消されたコマンドをやり直し、それまで行われた作業の記録を辿ることができます。元に戻すや、やり直しコマンドを使用することで、これまでの記録を行き来することができます。

10.2.2 切り取り、コピー、貼り付け、削除

アイコンとショートカット

コマンド	アイコン	ショートカット
切り取り		Ctrl+X または Shift+Del
コピー		Ctrl+C
貼り付け		Ctrl+V
削除		Del

切り取り

切り取りコマンドは、選択されたテキストまたはイメージをクリップボードにコピーして、オリジナルの場所にあるオブジェクトを削除します。

コピー

コピーコマンドでは、選択されたテキストやアイテムをクリップボードにコピーします。このコマンドを使用することで、Authentic Desktop 内部でデータの複製を行い、他のアプリケーションへデータを移動することができます。

貼り付け

貼り付けコマンドにより、クリップボードにコピーされたコンテンツを、現在カーソルがある位置に挿入することができます。

削除

削除コマンドを使うことで、現在選択されているテキストやアイテムを、クリップボードにコピーすること無く削除することができます。

10.2.3 全て選択

全て選択 (Ctrl+A) コマンドにより、ドキュメント全体のコンテンツが選択されます。

10.2.4 検索、次を検索

アイコンとショートカット

コマンド	アイコン	ショートカット
検索		Ctrl+F
次を検索		F3

検索

検索コマンドにより、検索ダイアログが表示され、検索を行なうための文字列の入力や、検索に伴う各種オプションを指定することができます。テキストの検索を行うには、検索対象テキストボックスにその文字列を入力するか、エディタボックスを使用して、最後に使用された10個ある検索文字列から選択を行い、検索に必要なオプションを指定します。


検索と次を検索コマンドをプロジェクトウィンドウ内でプロジェクトが選択されるとファイルとフォルダーを検索するために使用することができます。

次を検索

次を検索コマンドは、最後に実行された検索コマンドを再度実行し、目的のテキスト内で次に出現する検索結果を表示します。

10.2.5 置換

アイコンとショートカット

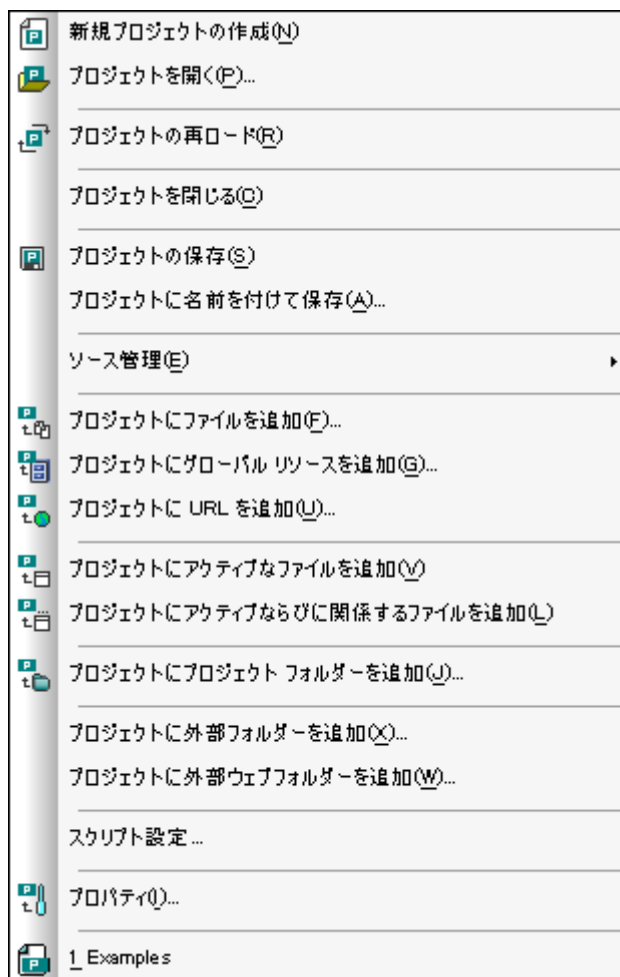
コマンド	アイコン	ショートカット
置換		Ctrl+H

説明

置換コマンドでは、テキスト内のある文字列を別の文字列に置き換えることができます。検索コマンドにて使用されるオプションをここで使用することができます。各アイテムを個々に置き換えるか、全て置換ボタンを使用することで、検索と置換処理をグローバルに行うことができます。

10.3 プロジェクトメニュー

Authentic Desktop では、広く使用されているツリービューに表示されるXML プロジェクトを使用することで、複数のファイルやURLを管理することができます。[ファイル](#)と[URL](#)は拡張子などで[フォルダー](#)内にグループ化することができ、バッチ処理などを簡単にこなすことができます。



メモ: プロジェクトに関連したコマンドは、プロジェクト ウィンドウにあるアイテムを右クリックすることで表示されるコンテキストメニューからアクセスすることができます。

絶対パスと相対パス

各プロジェクトは、.spp という拡張子の単一プロジェクトファイルとして保存されます。これらのファイルの内容はXMLドキュメント形式で保存されており、通常のXMLファイルとしても編集することができます。プロジェクトファイルでは、同じ位置より高い場所に対しては絶対パスが、現在のフォルダーやサブフォルダーに対しては相対パスが使用されます。例えば、ディレクトリ構造が以下の様な状況で、

```
| -Folder1
|   |
|   | -Folder2
|   |
|   | -Folder3
```


プロジェクトの更新

外部フォルダーに変更が加えられた場合、この変更は、プロジェクトが更新されるまで、プロジェクトウィンドウ内で反映されません。

コンテキストメニュー内のグローバルリソース

プロジェクトウィンドウのフォルダーを右クリックすると、コンテキストメニューが表示され、その中にあるグローバルリソースファイルを追加メニューアイテムから、[グローバルリソース](#)を追加することができます。メニューコマンドにより、グローバルリソースファイルを追加ダイアログが表示され、現在アクティブなグローバルリソースXMLファイルにある全てのファイル型ならびにフォルダー型のグローバルリソースが表示されます。目的のグローバルリソースを選択すると、選択されたプロジェクトフォルダーに追加されます。

プロジェクトならびにソース管理プロバイダー

Authentic Desktop プロジェクトをソース管理リポジトリに追加する場合、プロジェクト内にあるファイルを、プロジェクトファイルが収められている場所以下に配置するようにしてください。

プロジェクトファイルが位置している場所が、ソース管理リポジトリ内部にあるプロジェクトのルートディレクトリとなります。これ(プロジェクトルートディレクトリ)より上の階層にあるファイルで Authentic Desktop プロジェクトに追加されたファイルは、(仮にファイルの配置ができたとしても)リポジトリ内部にて予期しない場所に配置される可能性があります。

例えば、上に示したディレクトリ構造で、プロジェクトファイルが Folder3 に保存され、ソース管理にも配置されると:

- Folder1 に追加されたファイルはソース管理にて配置されない場合があります。
- Folder2 に追加されたファイルは、(プロジェクトフォルダーではなく)リポジトリのルートディレクトリに追加されますが、それでもソース管理以下に配置されます。
- Folder3 ならびに Folder4 に配置されているファイルは、ソース管理以下においても期待されている通りの場所に配置されます。

10.3.1 新規プロジェクト



「新規プロジェクトの作成」コマンドを選択すると、Authentic Desktop にて新たなプロジェクトが作成されます。既に他のプロジェクトを開いている場合、そのプロジェクトに属している全てのドキュメントを閉じるかどうかを選択することができます。

10.3.2 プロジェクトを開く



「プロジェクトを開く...」コマンドでは、Authentic Desktop における既存のプロジェクトを開きます。既に別のプロジェクトを開いている場合、まずはそのプロジェクトが閉じられます。

10.3.3 プロジェクトの再ロード



プロジェクトの再ロードコマンドは、現在のプロジェクトをディスクから再度ロードします。複数のユーザーが作業をしているような環境では、他のユーザーがプロジェクトに加えられた変更を反映するために、プロジェクトの再ロードが必要になる場合があります。


メモ: プロジェクトファイル(.spp ファイル)の内容はXMLドキュメントで、その他の一般的なXMLファイルと同様の方法で編集することができます。

10.3.4 プロジェクトを閉じる

プロジェクトを閉じるコマンドにより、現在アクティブなプロジェクトが閉じられます。プロジェクトに変更が加えられている場合、閉じる前にプロジェクトを保存するか選択することができます。プロジェクトに変更が加えられている場合は、プロジェクトウィンドウに表示されているプロジェクト名に、アスタリスクが表示されます。

10.3.5 プロジェクトの保存、名前を付けて保存



プロジェクトの保存コマンドにより、現在のプロジェクトが保存されます。プロジェクトウィンドウをアクティブにして、保存  アイコンをクリックすることでも、プロジェクトの保存を行うことができます。

プロジェクトに名前を付けて保存コマンドは、現在のプロジェクトを新たに指定した別の名前で保存します。

10.3.6 ソース管理

Altova アプリケーションでは、Microsoft SourceSafe や、それに準拠した監視機能がサポートされます。これらシステムのインストール方法については、[サポートされているソース管理システム](#)を参照ください。このセクションでは、XMLSpy 内部からソース管理システムへのアクセスを行う「プロジェクト | ソース管理」のサブメニューコマンドについて説明します。

ソース管理機能の概要

アプリケーションプロジェクト内部にあるファイルをソース管理へ配置するメカニズムは以下のようになっています:

1. Authentic Desktop にて、ソース管理へ配置するファイルが含まれているアプリケーションプロジェクトフォルダーを作成します。通常アプリケーションフォルダーは、プロジェクトファイルが含まれているローカルのフォルダーに対応しています。ローカルフォルダーへのパスは、ローカルパスと呼ばれます。
2. (ソース管理 または監視として参照される) ソース管理 システムのデータベース内では、ソース管理下に置かれるファイルを含むフォルダーが作成されます。

3. 「[プロジェクト | ソース管理 | ソース管理に追加](#)」コマンドにより、アプリケーションプロジェクトファイルがソース管理へ追加されます。
4. チェックインやチェックアウト、ソース管理からファイルを削除といった、ソース管理に関するアクションは、「[プロジェクト | ソース管理 サブメニュー](#)」コマンドを使用することで行うことができます。このセクションのサブセクションでは、このサブメニューにあるコマンドに関する記述を行います。

メモ: 現在使用しているソース管理プロバイダーを変更するには、(i) ソース管理オプション ([ツール | オプション | ソース管理](#)) から行う方法と、(ii) ソース管理の変更ダイアログ ([プロジェクト | ソース管理 | ソース管理の変更](#)) に行う方法の2つが用意されています。

メモ: ソース管理プロジェクトは、アプリケーションプロジェクトとは異なるものです。ソース管理プロジェクトはディレクトリ構造に依存しているのに対して、Authentic Desktop プロジェクトはディレクトリ構造から独立した論理的な構造となっています。

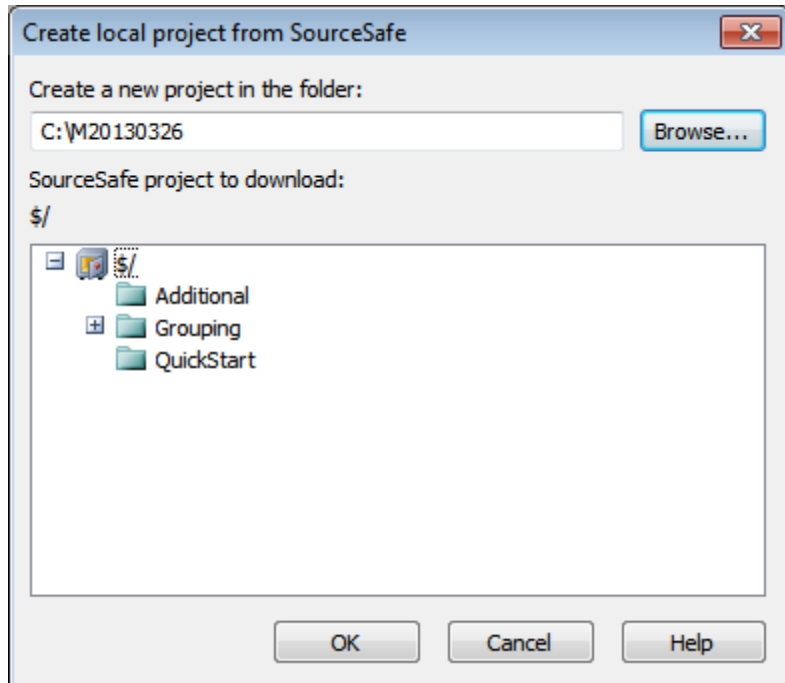
詳細に関しては次のセクションを参照してください! [ソース管理](#)。

10.3.6.1 ソース管理から開く

ソース管理から開く コマンドにより新規アプリケーションプロジェクトがソース管理下のプロジェクトとして作成されます。

新規プロジェクトは以下のとおり作成することができます:

1. 使用されるソース管理システムにより、場合によっては、ソース管理から新規のプロジェクトが作成される前に、プロジェクトからファイルがチェックアウトされないようご注意ください。
2. 全てのプロジェクトはアプリケーション内に存在する必要はありません。
3. コマンド「[プロジェクト | ソース管理 | ソース管理から開く](#)」を選択します。
4. 現在設定されているソース管理システムは、検証と接続ダイアログをポップアップします。ローカルフォルダーに対応するレポジトリ内のバインドされたフォルダーであるレポジトリに接続を作成します。
5. ポップアップするダイアログ内で、(接続が作成された)レポジトリ内のバインドされたフォルダーのコンテンツがコピーされるローカルフォルダーを参照します(下のスクリーンショット)。下のスクリーンショットでは、バインドされたフォルダーは MyProject と呼ばれ、\$ サインにより表示されています。ローカルフォルダーは C:\M20130326 です。



6. 「OK」をクリックします。バインドされたフォルダー(MyProject)のコンテンツは、次のローカルフォルダーにコピーされ、C:\M20130326.、新規のプロジェクトとして作成される。プロジェクトファイル(.spp ファイル)を選択するようプロンプトするダイアログがポップアップします。
7. ローカルフォルダーにコピーされる .spp ファイルを選択します。サンプルでは、C:\M20130326 フォルダーにある MyProject.spp です。アプリケーション内に MyProject という名前の新規のプロジェクト が作成され、プロジェクトウィンドウ内に表示されます。プロジェクトのファイルは以下のフォルダー内にあります: C:\M20130326。

ソース管理のシンボル

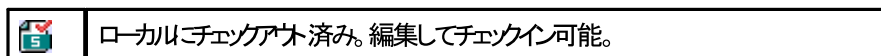
フォルダーやファイルのアイコンが表示されており、それぞれ以下に類似した意味を持ちます。

	チェックイン状態。チェックアウト可能。
	他のユーザーによりチェックアウトされた状態。チェックアウト不可能。
	ローカルにチェックアウト済み。編集してチェックイン可能。

10.3.6.2 ソース管理を有効にする

ソース管理を有効にするコマンドにより、アプリケーションプロジェクトのソース管理を有効化、または無効化することができます。このオプションを選択することで、プロジェクト全体のソース管理を有効化/無効化することができます。ソース管理が有効化されると、さまざまなファイルのチェックインとアウトの状況が抽出され、プロジェクトウィンドウ内に表示されます。

	チェックイン状態。チェックアウト可能。
	他のユーザーによりチェックアウトされた状態。チェックアウト不可能。



10.3.6.3 最新バージョンの取得

(プロジェクト | ソース管理 メニュー内の最新バージョンの取得コマンドにより、現在作業中のディレクトリ内で選択されたファイルの最新版をソース管理から取得します。ファイルは読み取り専用として取得され、チェックアウトはされません。このコマンドは **Get コマンド** と同じように作動しますが、Get ダイアログは表示しません。

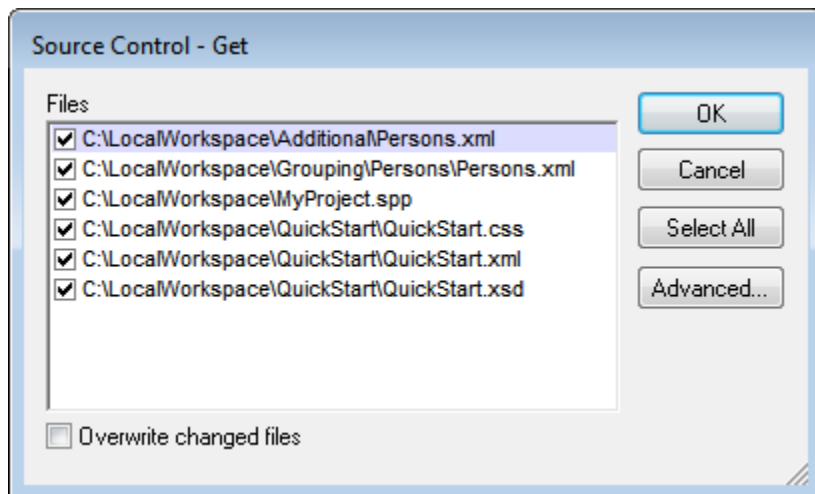
選択されたファイルが現在チェックアウトされている場合、ソース管理システムがどのようにこのような状況を扱うかにより取られるアクションは異なります。通常、ソース管理システムは、チェックアウトされたファイルを置き換える、マージする、またはそのままにするかを問います。

メモ: このコマンドは、フォルダーに対して使用されると、再帰的です。これは、フォルダー階層構造内の下の全てのファイルに影響を与えます。

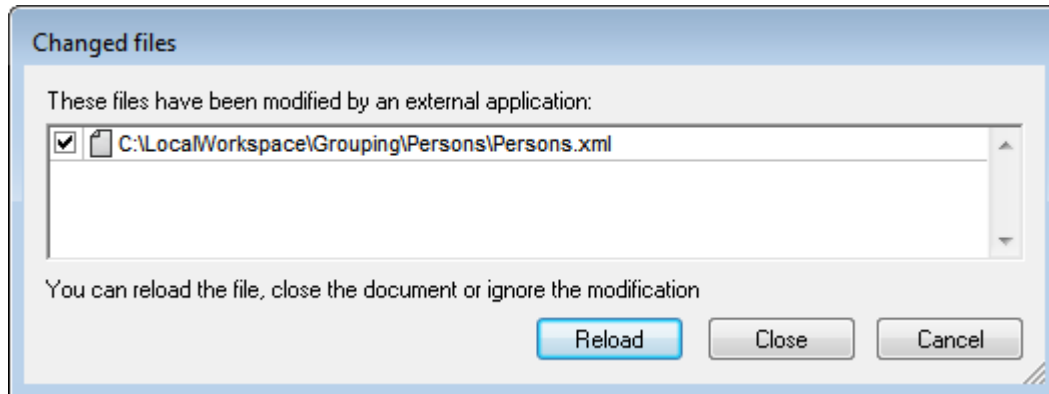
10.3.6.4 フォルダーの取得

(プロジェクト | ソース管理 メニュー内の取得 コマンドはレポジトリからのファイルを読み取り専用 ファイル として取得します (編集するにはファイルをチェックアウトする必要があります)。取得ダイアログは、「取得」コマンドが実行されるオブジェクト内のファイル(プロジェクトまたはフォルダー)をリストします (下のスクリーンショットを参照)。チェックして取得するファイルを選択することができます。

メモ: ソース管理システムが許可している場合、「フォルダーの取得」コマンドによりレポジトリ内の個別のサブフォルダーを選択することができます。

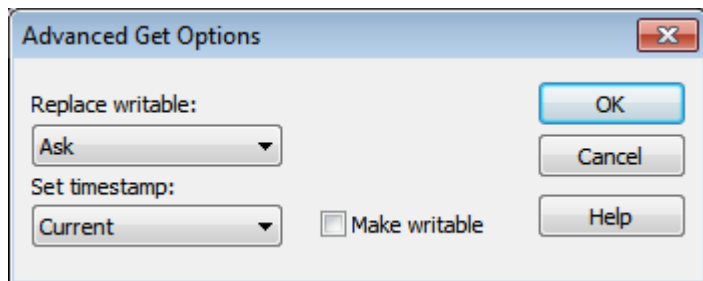


取得ダイアログの下にあるこのオプションをチェックして、変更された、または、チェックアウトされたファイルを上書きすることを選択することができます。「OK」をクリックすると、ファイルが上書きされます。上書きされたファイルが開かれていると、ファイルを再ロード(再ロード ボタン)、ファイルを閉める(閉じる)、または、ファイルの現在表示を保持する(キャンセル)オプションを問うダイアログがポップアップします (下のスクリーンショット)。



高度な取得オプション

高度な取得オプションは取得ダイアログ内の「高度な」ボタンを使用してアクセスすることができますダイアログ（下のスクリーンショット）（このセクションの最初のスクリーンショットを参照）。



ここでは以下に関する設定をおこなうことができます: (i) チェックアウトされた書き込みことのできるファイルを置き換える。(ii) タイムスタンプ。(iii) 取得されたファイルの読み取り専用 プロパティを変更して書き込み可能にする。

10.3.6.5 チェックアウト、チェックイン

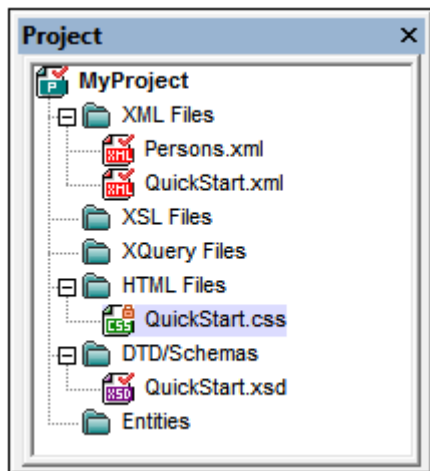
プロジェクト ファイルがソース管理下に置かれると、(プロジェクト ウィンドウ内で) ファイルを選択して、「プロジェクト | ソース管理」メニュー「チェックアウト」と「チェックイン」内の対応するコマンドをチェックして、チェックアウトするか、チェックインするかを選択することができます。

ファイルがチェックアウトされると、ローカルフォルダー内にレポジトリからのコピーが置かれます。チェックアウトされたファイルは編集することができます。ソース管理下にあるファイルがチェックアウトされていない場合、このファイルを編集することはできません。ファイルの編集後、レポジトリに編集を保存することができます。ファイルが保存された場合でも、レポジトリ変更をチェックインすると変更は保存されます。ファイルがチェックアウトされているかは、アイコン内のチェック、または、ロック済みのシンボルに表示されます。

ファイルとプロジェクト フォルダーには、下に示されるような特定のシンボルが表示されます。

	チェックイン状態。チェックアウト可能。
	他のユーザーによりチェックアウトされた状態。チェックアウト不可能。
	ローカルにチェックアウト済み。編集してチェックイン可能。

プロジェクト、または、フォルダーをプロジェクト内から選択するときは、選択内のファイルを選択します。複数のオブジェクト（ファイルとフォルダー）を選択するときは、オブジェクトをクリックして、Ctrl キーを押します。下のスクリーンショットはチェックアウトされたプロジェクトを示しています。ファイル QuickStart.css が次にチェックインされています。



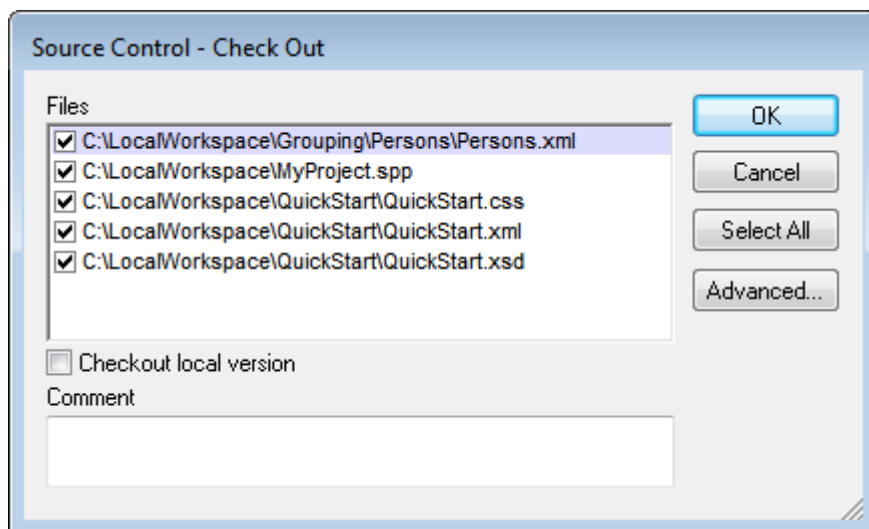
編集の変更を保存、または、拒否する

ファイルをチェックインする場合、ファイルをチェックインしたままにすることができます。ファイルをチェックアウトしたままにして、レポートリに編集の変更を保存することを意味します。この機能は、定期的レポートリに編集の変更を保存して、編集を続ける場合とても役に立ちます。

ファイルをチェックアウトして、編集を行い、ファイルが加えられた変更を拒否する場合、コマンド「プロジェクト | ソース管理 | チェックアウトの取り消し」を選択して、レポートリ内に保存されたドキュメントにバージョンを戻すことができます。

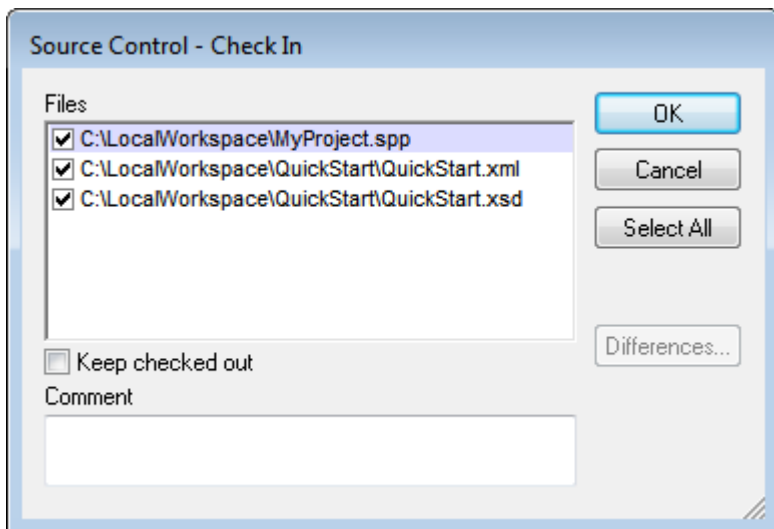
チェックアウト

チェックアウト ダイアログ（下のスクリーンショット）にて以下を行うことができます: (i) ファイルをチェックアウトする、および、(ii) レポートリバージョンを選択する、または、ローカルバージョンがチェックアウトされるかを選択する。



チェックイン

チェックインダイアログ(下のスクリーンショット)にて以下を行うことができます: (i) ファイルをチェックインする、および、(ii) 希望する場合、ファイルをチェックアウトしたままにすることができます。






メモ: 両方のダイアログ(チェックアウトとチェックイン)内で、選択されたオブジェクト(プロジェクトまたはプロジェクトフォルダー)が複数のファイルを含む場合、複数のファイルが表示されます。

10.3.6.6 チェックアウトの取り消し

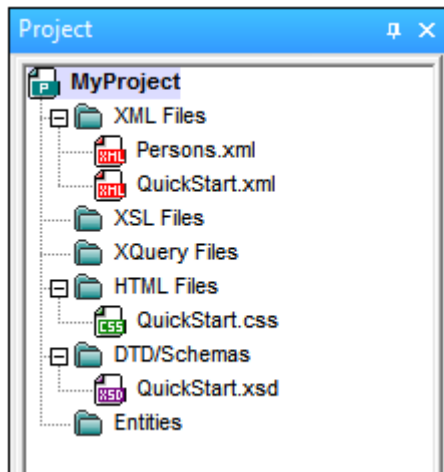
チェックアウトされたファイルが存在し、編集の変更が加えられており、変更を拒否する場合、コマンド「プロジェクト | ソース管理 | チェックアウトの取り消し」を使用してポストアウトで保存されているドキュメントのバージョンに戻ります。

ファイルとプロジェクトフォルダーには、下に示されるような特定のシンボルが表示されます。

	チェックイン状態。チェックアウト可能。
	他のユーザーによりチェックアウトされた状態。チェックアウト不可能。
	ローカルにチェックアウト済み。編集してチェックイン可能。

10.3.6.7 ソース管理へ追加

ソース管理にプロジェクトが追加されると、ファイルを単一に、または、グループとしてソース管理に追加することができます。プロジェクトウィンドウ内でファイルを選択し、プロジェクト | ソース管理 | ソース管理に追加 をクリックします。複数のファイルを選択するには、追加するファイルをクリック中に **Ctrl** キーを押します。(緑) プロジェクトフォルダー上でコマンドを実行する場合は、全てのファイルをソース管理のフォルダーとサブフォルダー内に追加します(下のスクリーンショットを参照)。






ソース管理にファイルが追加されると、(プロジェクトフォルダ階層構造ではなく)ローカルフォルダ階層構造がレポジトリ内でシミュレートされます。ファイルがサブフォルダ内のローカルフォルダ内で X レベル下にあると、ファイルの親フォルダ他の祖先フォルダは自動的にレポジトリに追加されます。

ソース管理にプロジェクトから最初のファイルが追加されると、正確なレポジトリ内に作成され、プロジェクトファイル(.spp ファイル)が自動的に追加されます。詳細に関しては [ソース管理へ追加](#) のセクションを参照してください。

ソース管理シンボル

ファイルとプロジェクトフォルダには、下に示されるような特定のシンボルが表示されます。

	チェックイン状態。チェックアウト可能。
	他のユーザーによりチェックアウトされた状態。チェックアウト不可能。
	ローカルにチェックアウト済み。編集してチェックイン可能。

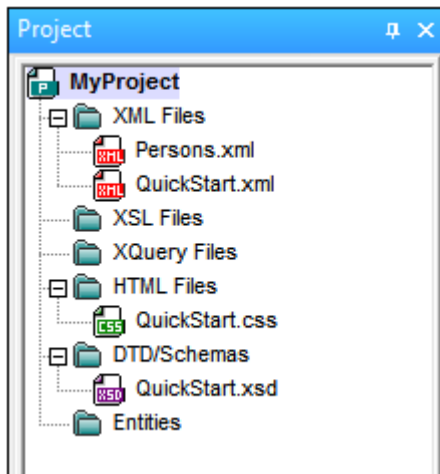
10.3.6.8 ソース管理から削除

ソース管理ファイルからファイルを削除するためには、ファイルを選択して、コマンド「プロジェクト | ソース管理 | ソース管理から削除する」をクリックします。以下も削除することができます: (i) フォルダ上の実行コマンドによるプロジェクトフォルダ内のファイルの削除、(ii) **Ctrl** キーを押して複数のファイルを選択しての削除 (iii) プロジェクト上の実行コマンドによるプロジェクト全体の削除。

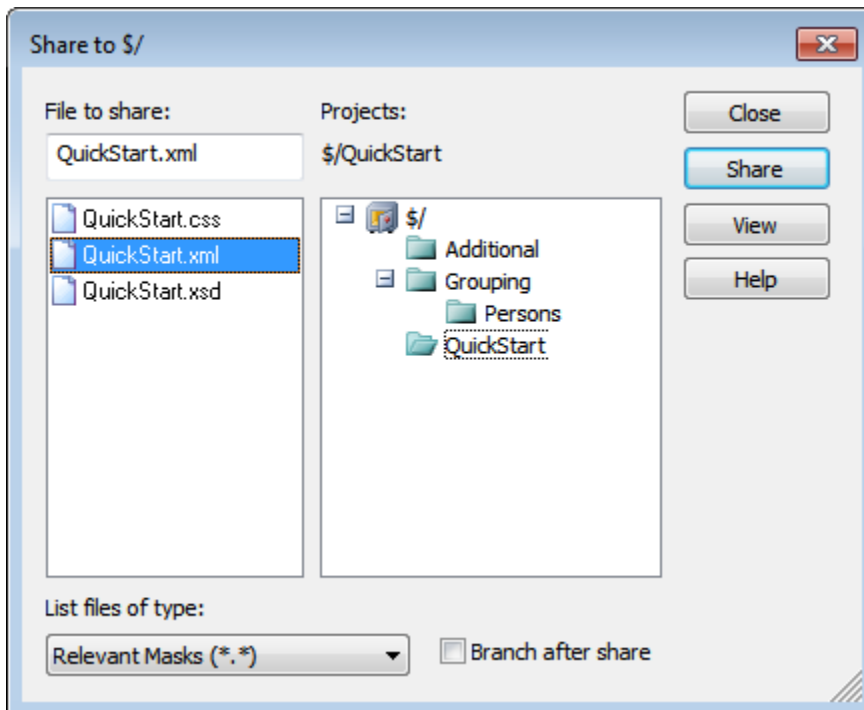
10.3.6.9 ソース管理から共有

ソース管理システムが共有するために使用されている場合、「ソース管理から共有」コマンドがサポートされます。複数のローカルの場所でファイルを共有することができます。これらのローカルファイルに加えられ変更は、他の「共有された」バージョンで反映されます。

アプリケーションのプロジェクトウィンドウから最初にプロジェクトを選択します (下のスクリーンショット内でハイライトされています)。「ソース管理から共有」をクリックします。

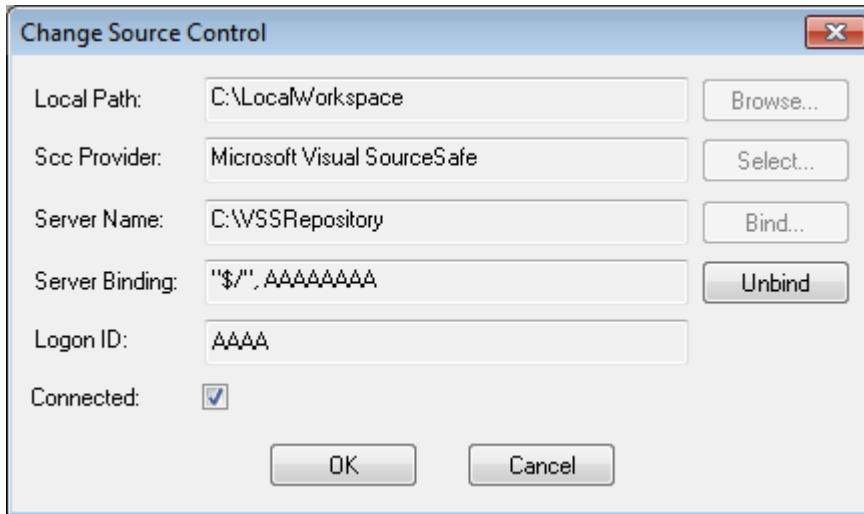


共有するフォルダーダイアログ (下のスクリーンショット) がポップアップします。



共有するファイルを選択するには、ファイルが存在するフォルダーが存在する右側のペイン内のプロジェクトツリーを選択します。フォルダー内のファイルは、左側のペインに表示されます。共有するファイルを選択します (複数のファイルは **Ctrl** キーを押して共有するファイルをクリックします)。選択されたファイルは共有するファイルテキストボックス内 (左上) に表示されます。「共有」をクリックして、選択されたファイルをローカルの共有フォルダーにコピーするために「閉じる」をクリックします。

共有フォルダーは、共有するフォルダーダイアログの名前内に記録されます。上のスクリーンショット内では (\$ サイン) ローカルフォルダーがバウンスされるレボネトリ内のフォルダーであるため) ローカルフォルダーです。ローカル IP とサーバーの IP アドレスを変更し、ソース管理の変更 ダイアログ内で共有フォルダーを確認し設定することができます (下のスクリーンショットを参照してください)。

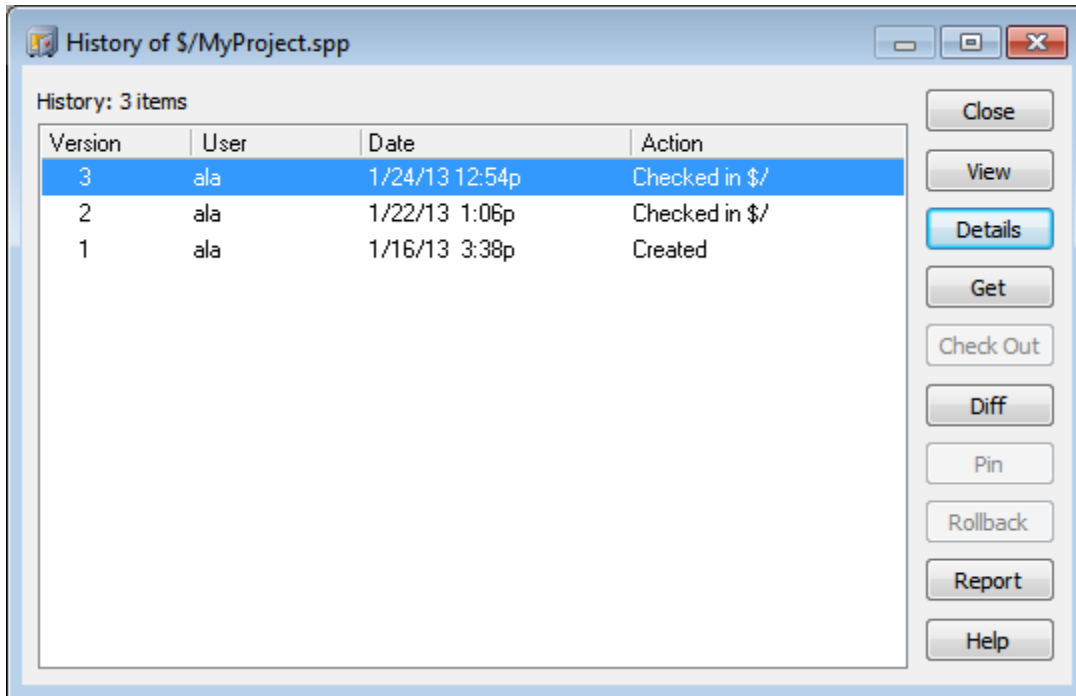


ソース管理システムの詳細に関しては、次を参照してください：ソース管理システムのユーザードキュメント。

10.3.6.10 履歴の表示

履歴の表示コマンドでは、ソース管理以下にあるファイルの履歴を表示して、プロジェクトファイル(.spp ファイル) に対して施されたそれまでの変更情報を確認し、差分の表示や、過去バージョンのファイルを取得することができます。以下の操作を行って、ファイルの履歴を確認することができます：

下のスクリーンショットは、Visual SourceSafe ソース管理のシステムの履歴ダイアログを表示しています。MyProject.spp ファイルの異なるバージョンを表示しています。



このダイアログボックスでは、目的のファイルと比較したり、特定のバージョンを取得するためのオプションが提供されます。リストにあるエントリーをダブルクリックすることで、そのファイルに関する履歴の詳細情報ダイアログボックスが表示されます。ダイアログ内に表示されるボタンでは、以下の機能が提供されます：

閉じる：このダイアログボックスを閉じます。

表示：ファイルを表示するためのアプリケーションを選択するためのダイアログボックスが表示されます。

詳細情報：現在アクティブになっているファイルの**プロパティ**を確認するためのダイアログボックスが表示されます。

取得：バージョンリストに表示されているファイルのバージョンを取得し、現在作業中のディレクトリに配置します。

チェックアウト：前バージョンのファイルをチェックアウトすることができます。

差分：**差分表示オプション**ダイアログボックスを開き、ファイルバージョンにおける差分を表示する際に使用される表示オプションを設定することができます。

ピン設定：ファイルのバージョンにピンを打つことで、特定のファイルバージョンを差分に使用することができます。

ロールバック：選択されたバージョンのファイルへロールバックします。

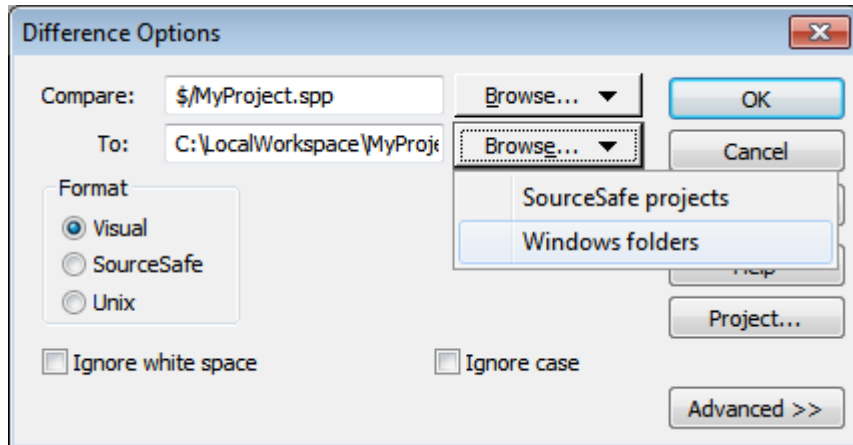
レポート：プリンター、ファイル、またはクリップボードへ送ることができる履歴レポートを作成します。

ヘルプ：ソース管理プロバイダープラグインのオンラインヘルプを開きます。

10.3.6.11 差分の表示

「差異の表示」コマンドは、プロジェクトウィンドウ内でファイルが選択されると有効化されます。プロジェクトファイル(.spp ファイル)を選択する場合は、プロジェクトウィンドウ内でプロジェクトのタイトルを選択します。「差異の表示」コマンドは、ソース管理システムの差異ツールを開始し、Altova アプリケーションからファイル間の差異が直接チェックできるようになります。

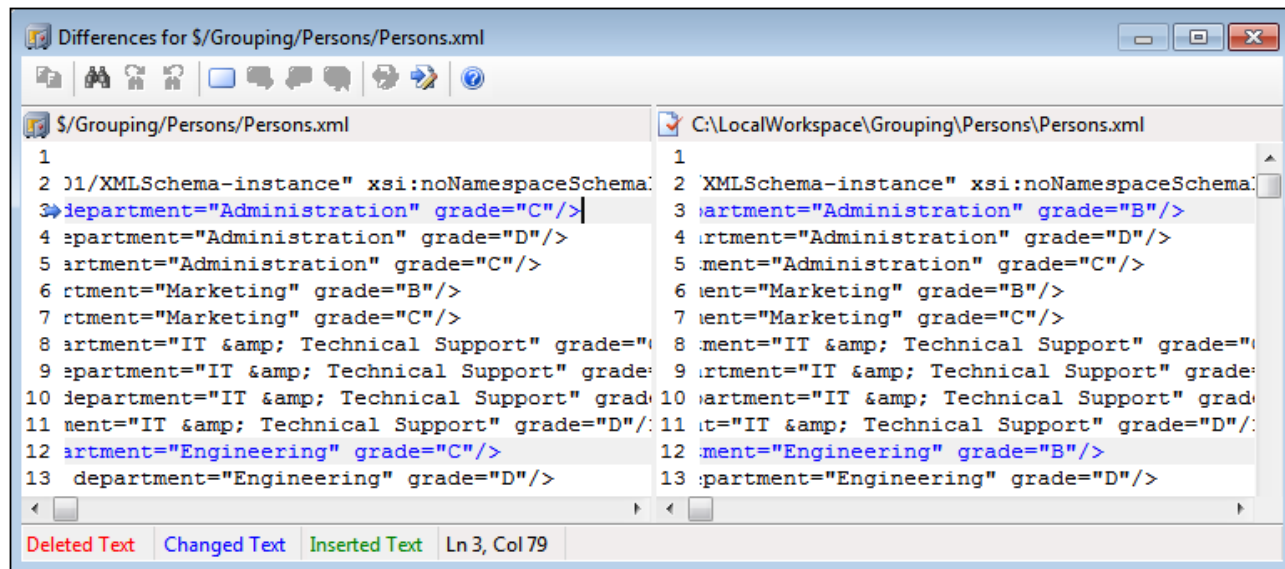
下のスクリーンショットは、Visual SourceSafe ソース管理システムの差異ツールを表示しています。



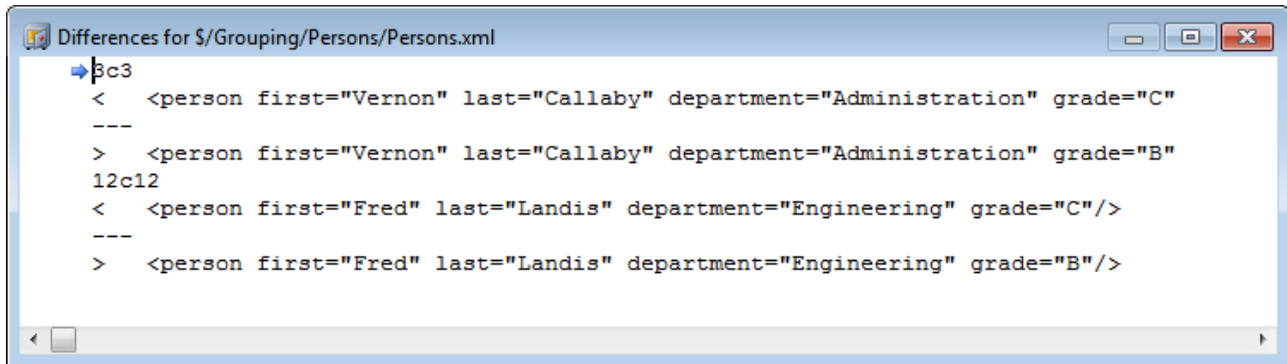
レボネリとローカルバージョンがそれぞれ比較とテキストフィールドにデフォルトで表示されます。他のファイルの参照の方法は、以下のとおりです。

1. 「参照」ボタンとドロップダウンリストから、(レボネリフィールドを参照するため) ソースセーフプロジェクト または (ローカルフォルダーを選択するため) Windows フォルダーを選択します。
2. ファイルを参照し、選択します。

適切なエントリーを選択し、「OK」により確定します。差異の結果が個別のウィンドウに表示されます。下のスクリーンショットは、2つのフォーマットのチェックインの結果を表示しています。



上のスクリーンショットは、Visual SourceSafe 差異の結果を Visual フォーマットで表示して折、したスクリーンショットは、Unix フォーマットでの結果を表示しています (上のオプションダイアログを参照してください)。2つの差異の種類が存在し、それぞれが C から B. へのグレードの変更を表示しています。



```

Differences for $/Grouping/Persons/Persons.xml
  3c3
  < <person first="Vernon" last="Callaby" department="Administration" grade="C"
  ---
  > <person first="Vernon" last="Callaby" department="Administration" grade="B"
12c12
  < <person first="Fred" last="Landis" department="Engineering" grade="C"/>
  ---
  > <person first="Fred" last="Landis" department="Engineering" grade="B"/>

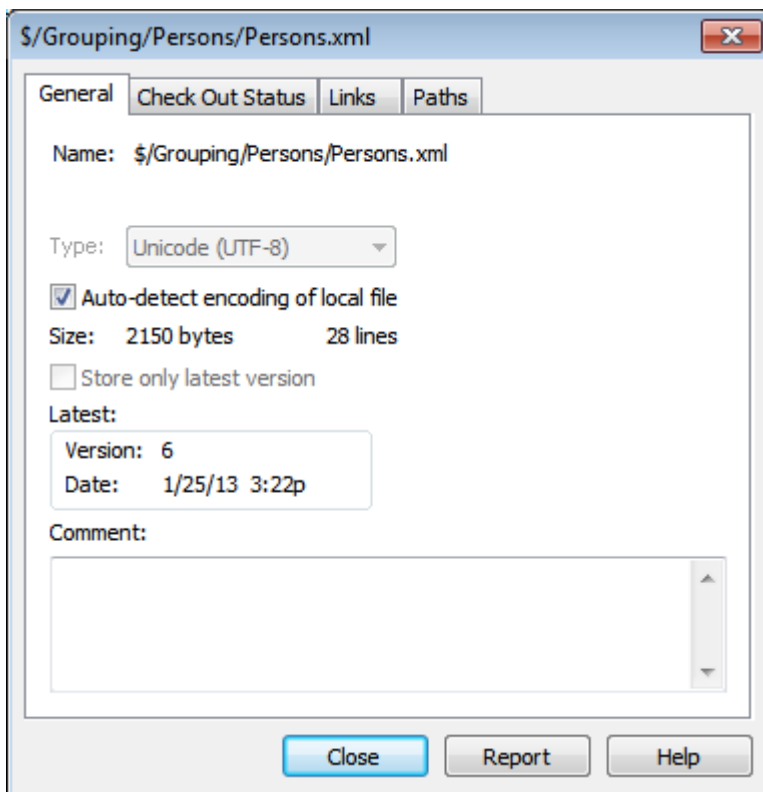
```

ソース管理システムの差異の処理に関しては、次を参照してください：製品のユーザードキュメント。

10.3.6.12 プロパティの表示

プロパティの表示コマンドにより、現在選択されているファイルのプロパティが表示されます(以下のスクリーンショットを参照)。表示されるプロパティは、使用しているソース管理プロバイダーにより変化します。下のスクリーンショットは、Visual SourceSafe がアクティブなソース管理システムであることを示しています。

このコマンドは、単一のファイルのために有効化されていることご注意ください。



このコマンドは単一のファイルに対してしか使用できないことご注意ください。

10.3.6.13 ステータスを最新の状態に更新

「ステータスを最新の状態に更新」コマンドにより、プロジェクトファイル全てのステータスが、現在のステータスに関係なく最新の状態に更新されます。

10.3.6.14 ソース管理マネジャー

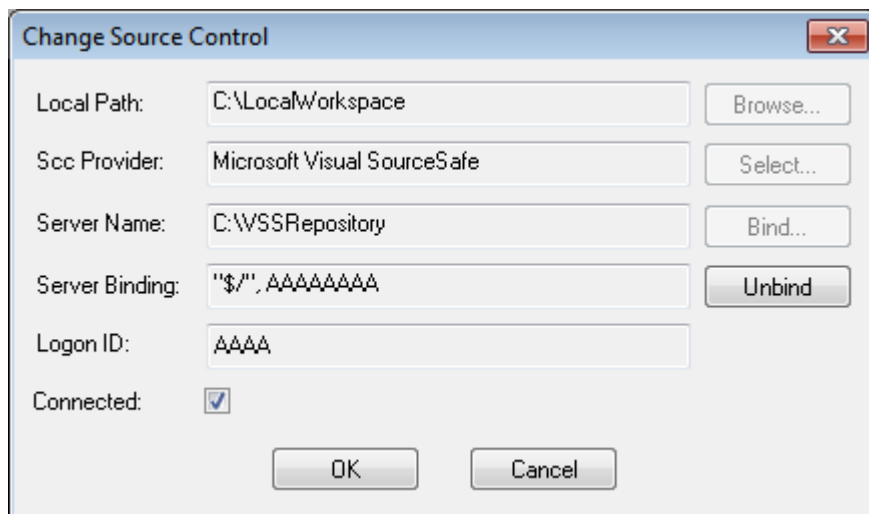
「ソース管理」コマンドにより、ソース管理ソフトウェアがネイティブのユーザーインターフェースにて起動されます。

10.3.6.15 ソース管理の変更

アクティブなアプリケーションプロジェクトがソース管理データベースに接続するため現在のバインドが使用されるため、現在のバインドは正確である必要があります。アプリケーションプロジェクトファイル(.spp ファイル)がローカルパスフォルダー内に存在し、レポジトリ上のバインドされたフォルダーがプロジェクトのファイルが保管されているデータベースである必要があることを意味します。通常、バインドされたフォルダーとそのサブ構造は、ローカルワークスペースとそのサブ構造に対応しています。

ソース管理の変更 ダイアログ内でソース内で管理システム(SCC プロバイダー)、ローカルフォルダー(ローカルパス)、およびレポジトリバインド(サーバー名とサーバーのバインド)を変更することができます(下のスクリーンショット)。

現在のバインドのバインドを解除した後のみに設定を変更することができます。「バインドの解除」ボタンを使用して現在のバインドを解除します。その後、全ての設定を編集することができます。



ソース管理の変更設定は、以下のとおりです:

1. 「参照」ボタンを使用してローカルフォルダーを参照し、インストールされたソース管理システムから「選択」ボタンを選択します。
2. レポジトリデータベース(ローカルフォルダー)をバインドすることができます。「バインド」ボタンをクリックしてこれを行います。ソース管理システムの接続ダイアログが表示されます。
3. ログインIDを入力すると、ソース管理システムに接続されます。それ以外の場合、接続ダイアログ内に詳細を入力します。

4. ローカルフォルダーにコネクトするレポネトリ内のデータベースを選択します。この設定は、1つ以上のダイアログを完成する必要がある可能性があります。
5. 設定が作成された後、「OK」ボタンをクリックしてください。

10.3.7 プロジェクトにファイルを追加



「プロジェクト | プロジェクトにファイルを追加」コマンドにより、ファイルを現在のプロジェクトへ追加することができます。このコマンドを使用し、ファイルをプロジェクト内部にある任意のフォルダーに追加することができます。開くダイアログでは、単一のファイルを選択したり、**Ctrl** を押下したままクリックすることで複数のファイルを選択することができます。プロジェクトにファイルを追加している場合は、[プロジェクトプロパティ](#) ダイアログボックスで定義されているファイル拡張子をベースに、適切なフォルダーにファイルが配置されます。

10.3.8 プロジェクトにグローバルリソースを追加

「プロジェクト | プロジェクトにグローバルリソースを追加」コマンドにより、グローバルリソースを追加ダイアログが表示され、プロジェクトに追加するファイルまたはフォルダーのグローバルリソースを選択することができます。ファイル型のグローバルリソースが選択されると、[プロジェクトプロパティ](#) ダイアログボックスで定義されているファイル拡張子をベースに、ファイルが適切なフォルダーに追加されます。フォルダー型のグローバルリソースが選択されると、そのフォルダーがファイルを開くダイアログにて開かれ、ファイルを選択するよう促されます。選択されたファイルは、[プロジェクトプロパティ](#) ダイアログボックスで定義されているファイル拡張子をベースに、適切なフォルダーに追加されます。グローバルリソースに関する詳細は、ドキュメンテーションのグローバルリソースのセクションを参照ください。

10.3.9 プロジェクトに URL を追加



「プロジェクト | プロジェクトに URL を追加」コマンドでは、現在のプロジェクトに URL を追加することができます。プロジェクトに URL を追加することで、URL のターゲットオブジェクトをプロジェクトに追加することができます。URL や URL オブジェクトが含まれるフォルダーに対して、バッチ処理が行われると、Authentic Desktop により URL からドキュメントが取得され、指定されたオペレーションが行われます。

10.3.10 プロジェクトにアクティブなファイルを追加



「プロジェクト | プロジェクトにアクティブなファイルを追加」コマンドにより、アクティブなファイルを現在のプロジェクトに追加することができます。ハードディスクや URL からファイルを開くと、このコマンドによりそのファイルを現在のプロジェクトに追加することができます。

10.3.11 プロジェクトにアクティブならびに関係するファイルを追加



「プロジェクト | プロジェクトにアクティブならびに関係するファイルを追加」コマンドを選択すると、現在アクティブな XML ドキュメントと、それに関係する全てのファイルがプロジェクトに追加されます。DTD やスキーマをベースとした XML ドキュメントを処理している場合、このコマンドを選択することで XML ドキュメントだけでなく、関連する全てのファイル(例えば DTD ならびに DTD が参照する全ての外部エンティティ)が現在のプロジェクトに追加されます。

メモ: : 処理命令により参照されるファイル(例えば XSLT ファイル)は、関係するファイルとして認識されません。

10.3.12 プロジェクトにアクティブならびに関係するフォルダーを追加



プロジェクト | プロジェクトにプロジェクト フォルダーを追加するコマンドは、新規のフォルダーを現在のプロジェクトに追加します。このコマンドを使用して、新規のフォルダーをプロジェクト フォルダーの現在のプロジェクト またはサブフォルダーに追加することができます。プロジェクト ウィンドウ内のフォルダーを右クリックして、コンテキストメニューからこのコマンドにアクセスすることができます。

メモ: プロジェクト フォルダーをドラッグして、プロジェクト フォルダー、または、プロジェクト内の他の場所へドロップすることができます。更に、フォルダーを Windows (ファイル) Explorer からドロップして、プロジェクト フォルダーへドロップすることもできます。

メモ: プロジェクト フォルダーは緑で示され、[外部フォルダー](#)は黄色で示されています。

10.3.13 プロジェクトに外部フォルダーを追加

「プロジェクト | プロジェクトに外部フォルダーを追加」コマンドにより、外部フォルダーを現在のプロジェクトへ追加することができます。このコマンドによりローカルまたはネットワークフォルダーを現在のプロジェクトへ追加することができます。プロジェクト ウィンドウにあるフォルダーを右クリックすることで表示されるコンテキストメニューからも、このコマンドへアクセスすることができます。

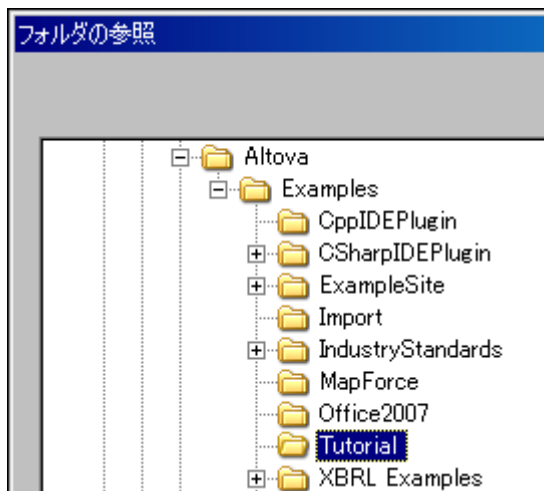
メモ: 外部フォルダーは黄色で、[プロジェクトフォルダー](#)は緑で表示されます。

メモ: 外部フォルダー内部に収められているファイルを、ソース管理内に収めることはできません。

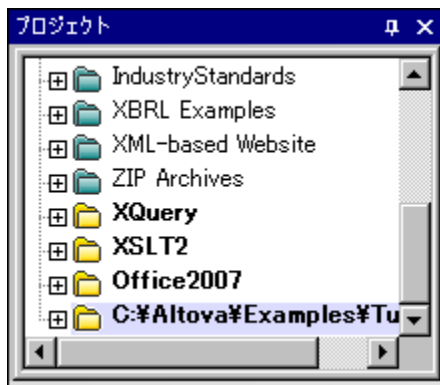
外部フォルダーをプロジェクト追加する

外部フォルダーをプロジェクトへ追加するには、以下の操作を行ってください:

1. メニューオプションから「プロジェクト | プロジェクトに外部フォルダーを追加」を選択します。
2. フォルダーの参照ダイアログボックスにて追加するフォルダーを選択し、「OK」により確定します。



選択されたフォルダーがプロジェクトウィンドウに表示されます。

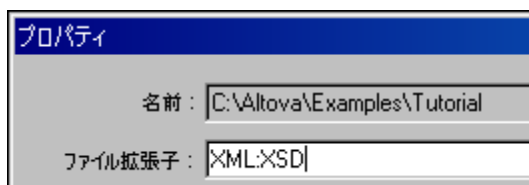


3. プラスアイコン(+)をクリックして、フォルダーのコンテンツを確認します。

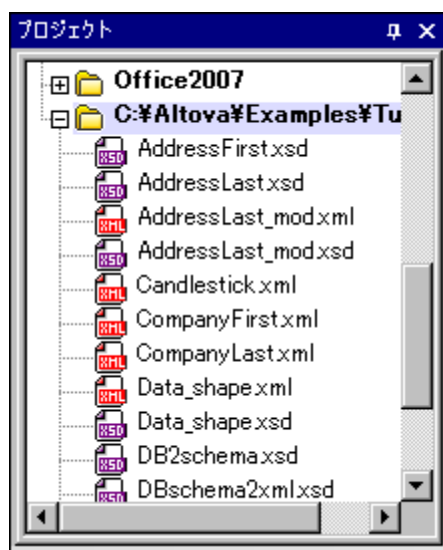
フォルダーのコンテンツをフィルタリング

フォルダー内部にあるコンテンツをフィルタリングするには、以下の操作を行ってください。

1. ローカルフォルダーを右クリックして、表示されるメニューオプションから「プロパティ」を選択します。プロパティダイアログボックスが表示されます。



2. ファイル拡張子フィールドをクリックし、表示したファイル拡張子を入力します。セミコロンにより、各ファイルタイプを分けることができます(例えば、この例ではXMLとスキーマXSDが分けられています)。
3. 「OK」をクリックして確認します。



プロジェクトウィンドウにはXMLとXSDファイルが表示されるようになります。

外部フォルダーの検証

外部フォルダーの整形形式チェックや検証を行うには、以下の操作を行ってください。

1. 外部フォルダーからチェックを行うファイルを選択します。
2. フォルダーを選択して、メニューコマンド[XML | 整形形式をチェック]または[XML | XMLの検証]をクリックします(それぞれホットキー[F7]または[F8]が対応します)。フォルダーの下に可視化されているファイルがチェックされます。ファイルが形式に誤りがある、または無効な場合このファイルはメインウィンドウで開かれ編集することが可能になります。
3. エラーを正して再チェックのために検証プロセスを実行します。

プロジェクトフォルダーの更新

ローカルやネットワーク上にファイルが追加されたり、削除されることもあります。フォルダーの表示を更新するには、外部フォルダーを右クリックして、ポップアップメニューオプションから「最新の状態に更新」を選択します。

外部フォルダーやファイルの削除

外部フォルダーを選択して「削除」キーを押下することで、プロジェクトウィンドウからフォルダーを削除することができます。ほかにも、外部フォルダーを右クリックして「削除」コマンドを選択することもできます。これらの操作によりプロジェクトウィンドウからフォルダーが削除されますが、ハードディスクやネットワーク上にあるフォルダーが削除されることはありません。

外部フォルダーにあるファイルを削除するにはハードディスクやネットワークから物理的に削除する必要があります。プロジェクトになされた変更を確認するには、外部フォルダーを右クリックして、「最新の状態に更新」を選択してください。

メモ: ドラッグアンドドロップにより、外部フォルダーを他のプロジェクトフォルダーや、プロジェクト内の別の場所へ移動することができます(但しほかの外部フォルダーへは不可)。更に、Windows (ファイル) エクスプローラーから、外部フォルダーを(外部フォルダー以外の)プロジェクトフォルダーへドラッグすることで、フォルダーを追加することができます。

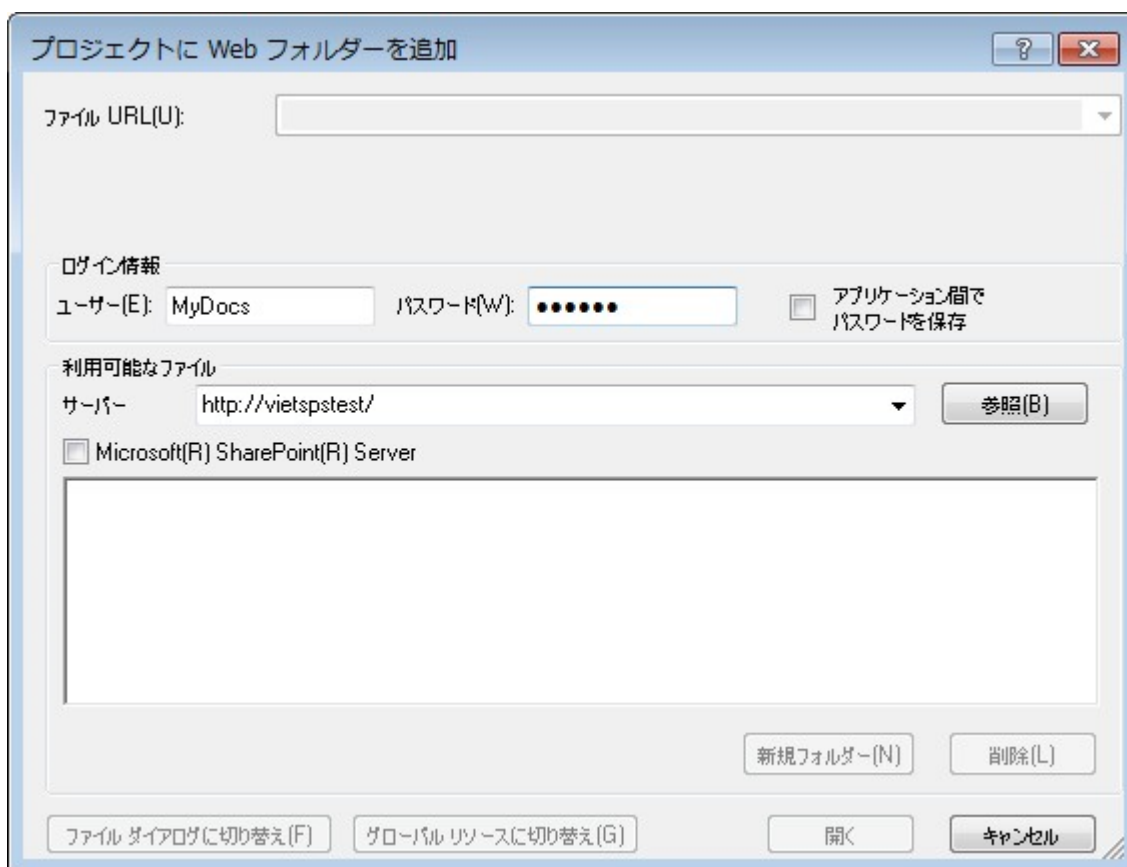
10.3.14 プロジェクトに外部ウェブフォルダーを追加

このコマンドは現在のプロジェクトに新規の外部 Web フォルダーを追加します。プロジェクトウィンドウ内のフォルダーを右クリックすることによりコンテキストメニューからこのコマンドにアクセスすることができます。外部フォルダーに含まれるファイルは、ソース管理の下に置くことはできません。

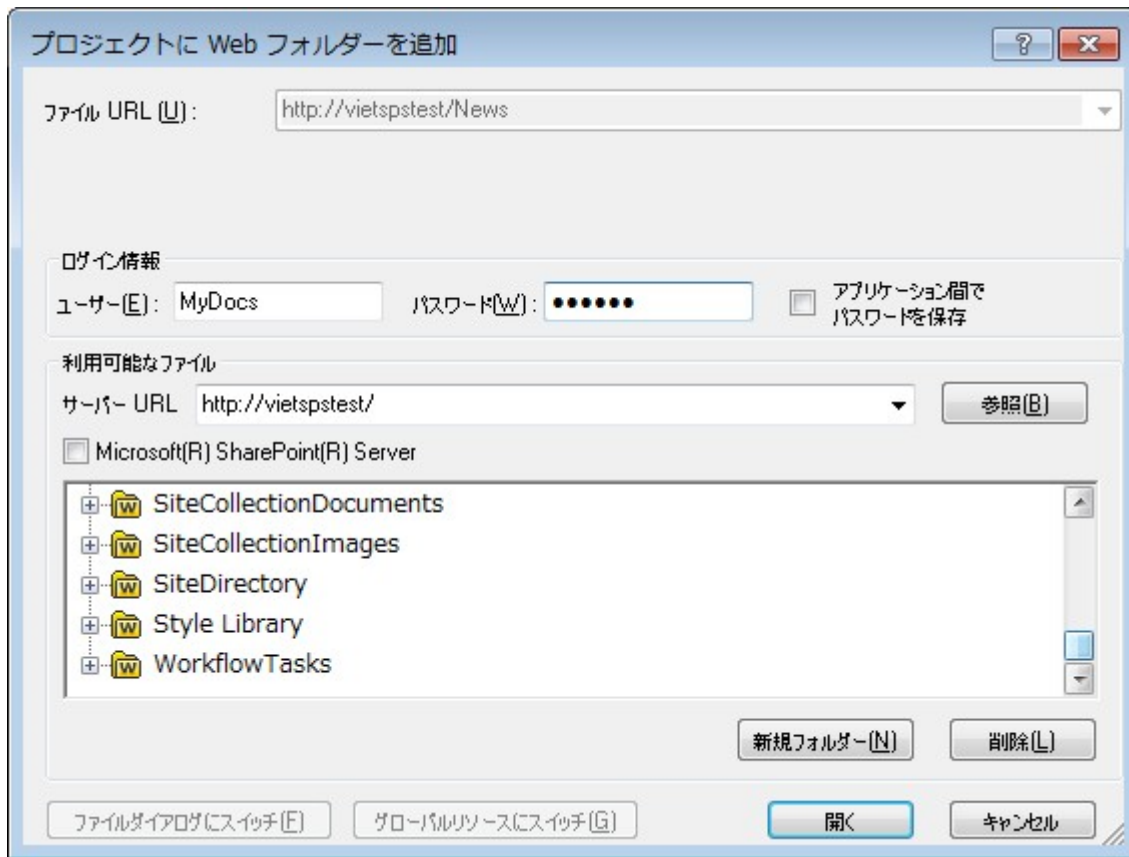
外部 Web フォルダーをプロジェクトに追加する

外部 Web フォルダーをプロジェクトに追加するには、以下を行います：

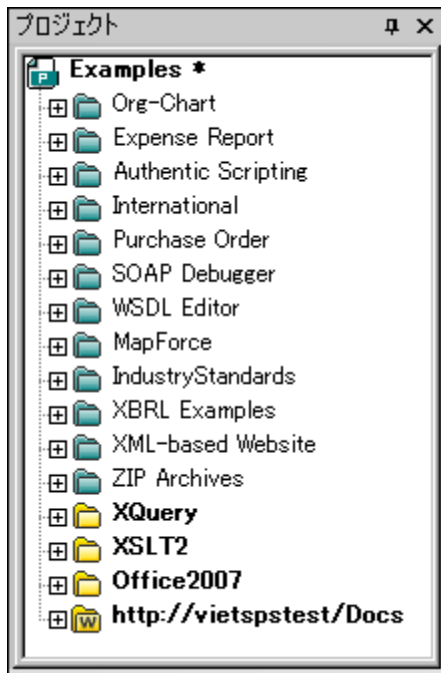
1. メニューオプション「プロジェクト | 外部 Web フォルダーをプロジェクトに追加する」を選択します。プロジェクトに Web フォルダーを追加するダイアログボックスが開かれます（下のスクリーンショット）。



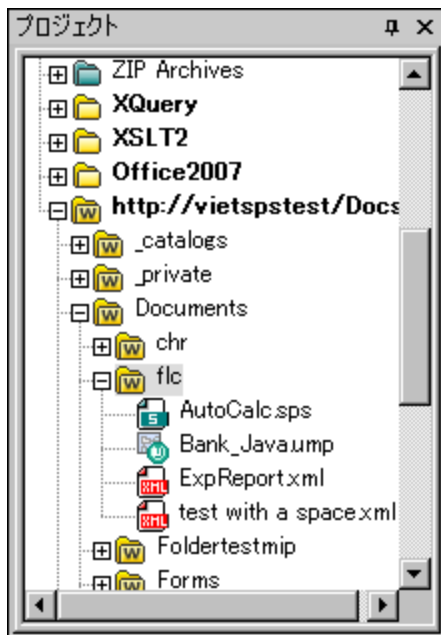
2. サーバー URL フィールドをクリックして、サーバー URL を入力します。サーバーが Microsoft® SharePoint® Server の場合、このオプションをチェックします。サーバーのこの型のファイルと作業するための詳細に関しては、Microsoft® SharePoint® Server 下のセクションのフォルダーを参照してください。
3. サーバーがパスワードにより保護されている場合、ユーザー ID とパスワードをユーザーとパスワードフィールドに入力してください。
4. 参照 をクリックして、サーバーに接続し、使用することのできるフォルダーを確認してください。



5. プロジェクトウィンドウに追加したいフォルダーをクリックします。この操作により、「開く」ボタンが有効になります。フォルダーの URL がファイル URL フィールドに表示されます。
6. 「開く」ボタンをクリックして、プロジェクトにフォルダーを追加します。



7. プラスアイコンをクリックして、フォルダー内部にあるコンテンツを確認することができます。



フォルダーのコンテンツをフィルタリング

フォルダーのコンテンツをフィルタリングするには、フォルダーを右クリックして、コンテキストメニューから「プロ/ 検索」を選択します。表示されるプロ/ 検索ダイアログにて、ファイル拡張子 フィールドをクリックし、表示させたいファイル種類の拡張子(例えば XML や XSD ファイル)を入力し

ます。セミコロンによりファイルの種類を分けることができます(例: xml; xsd; sps)。プロジェクトウィンドウでは、指定された拡張子を持つファイルだけが表示されるようになります。

フォルダーの検証と整形形式のチェック

フォルダーに収められているファイルの整形形式をチェック、または検証を行うには、フォルダーを選択して、[XML | 整形形式をチェック]または [XML | XML の検証] アイコンをクリックします(それぞれホットキー[F7] または [F8] が対応します)。フォルダーに表示されている全てのファイルがチェックされます。ファイルに不備がある場合、そのファイルがメインウィンドウで開かれ、編集を行うことができます。エラーの修正を行い、再度チェックを行ってください。Ctrl キーを押下しながらファイルをクリックすることで、個別にファイルを選択することができ、「F7」または「F8」を押下することで、選択されたファイルがチェックされます。

プロジェクトフォルダーのコンテンツを更新

ファイルの追加など、ウェブフォルダー内のファイルは常に更新される可能性があります。フォルダーのビューを更新するには、外部フォルダーを右クリックして、コンテキストメニューから最新の状態に更新オプションを選択します。

フォルダーとファイルを削除

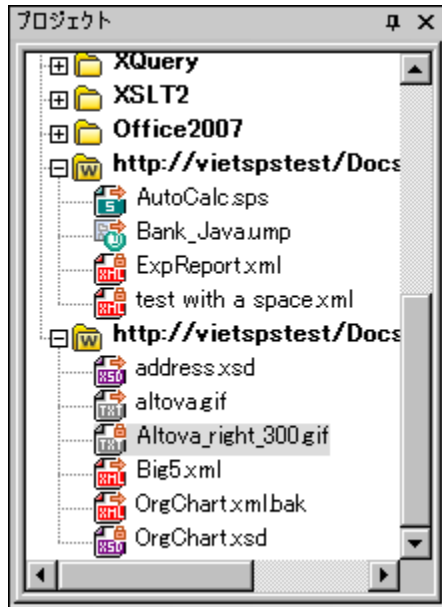
プロジェクトに追加されたウェブフォルダーは、プロジェクトから削除されるのもウェブフォルダーになります(つまりその中にあるファイルは削除されません)。ウェブフォルダーを削除するには (i) フォルダーを右クリックして、「削除」を選択するか、(ii) フォルダーを選択して、「削除」キーを押下します。これでプロジェクトビューからフォルダーが削除されますが、ウェブサーバーにおけるファイル(やフォルダー)の削除は行われません。

メモ: 単一のファイルを右クリックして、「削除」キーを押しても、プロジェクトウィンドウからファイルを削除することはできません。ファイルを削除するにはサーバーから物理的にファイルを削除し、外部フォルダーのコンテンツを更新して確認する必要があります。




Microsoft® SharePoint® Server 上のフォルダー

Microsoft(R) SharePoint(R) Server 上にあるフォルダーがプロジェクトに追加されると、プロジェクトウィンドウ内に表示されているファイルのコンテキストメニューから、フォルダー内のファイルをチェックインまたはチェックアウトすることができるようになります。これらのコマンドにアクセスするには、目的のファイルを右クリックし、目的のコマンド(チェックアウト、チェックイン、チェックアウトの取り消し)を選択します。

ユーザーID ならびにパスワードは [プロジェクト内部にある各フォルダーのプロパティ](#) に保存することができ、サーバーにアクセスするたびに、認証を行う必要がなくなります。

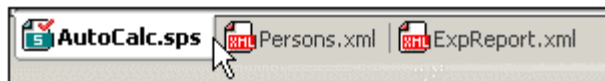


プロジェクトウィンドウ(上のスクリーンショット)では、ファイルのチェックイン/チェックアウトステータスを表すシンボルがファイルアイコンに表示されます。ファイルアイコンの説明を以下に示します:

	チェックイン状態。チェックアウト可能。
	他のユーザーによりチェックアウトされた状態。チェックアウト不可能。
	ローカルにチェックアウト済み。編集してチェックイン可能。

以下の点に注意してください

- ファイルのチェックアウトを行うと、Altova アプリケーション上でファイルの編集を行い、「ファイル | 保存 (Ctrl+S)」により保存を行うことができます。
- 編集されたファイルのチェックインを行うには、プロジェクトウィンドウからアクセスすることができるコンテキストメニューか、アプリケーションのメインウィンドウにあるファイルのタブを右クリックすることで表示されるコンテキストメニューから、チェックインコマンドを選択します。

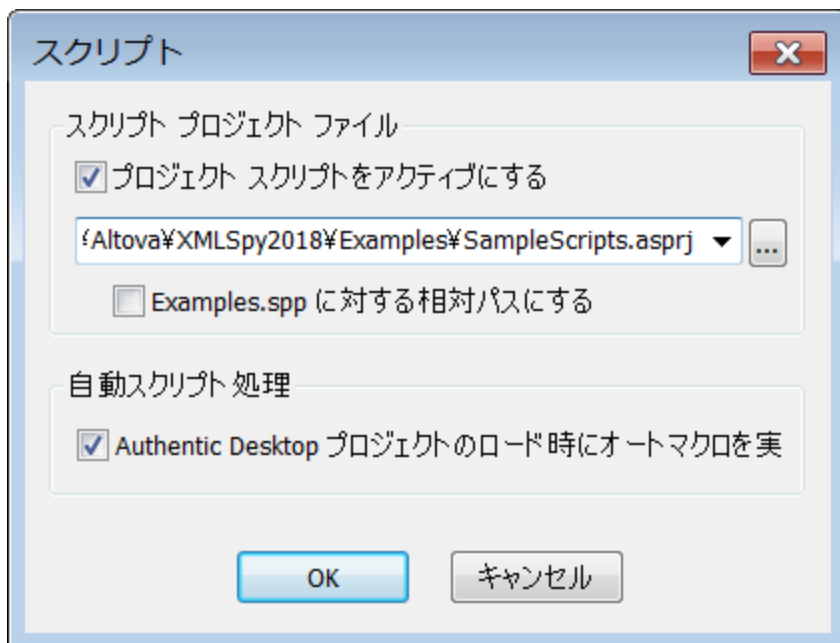


- 他のユーザーによりファイルがチェックアウトされている場合、そのファイルをチェックアウトすることはできません。
- (あなた自身により)ファイルがローカルにチェックアウトされている場合、コンテキストメニューからチェックアウトの取り消しを選択することで、チェックアウトを取り消すことができます。これでファイルが変更されないままサーバー側に返されることとなります。
- ある Altova アプリケーションでファイルをチェックアウトしている場合、他の Altova アプリケーションにてそのファイルをチェックアウトすることはできません。ファイルは既にチェックアウトされている状態であると認識されます。Microsoft(R) SharePoint(R) Server をサポートしている Altova 製品が、この状態で行えるコマンドは、「チェックインとチェックアウトの取り消し」だけです。

10.3.15 スクリプトの設定

スクリプトプロジェクトは、Authentic Desktop プロジェクトに以下のおお割り当てられます:

1. Authentic Desktop GUI 内で、必要なアプリケーションプロジェクトを開きます。
2. メニューコマンド「プロジェクト | スクリプト設定」を選択すると、スクリプト ダイアログが開かれます(下のスクリーンショット)。



3. プロジェクト スクリプトの有効化 チェックボックスをチェックし、必要なスクリプトプロジェクト (.asprj ファイル)を選択します。Authentic Desktop プロジェクトがロードされる時に、自動マクロを実行する場合、自動マクロを実行するチェックボックスをチェックします。
4. 「OK」をクリックして完了します。

メモ: Authentic Desktop プロジェクトのスクリプトプロジェクトを無効化(割り当てを解除)するには、プロジェクト スクリプトを有効化するチェックボックスのチェックを解除します。

10.3.16 プロパティ



「プロジェクト | プロジェクトプロパティ」コマンドは、アクティブなプロジェクトのプロパティダイアログを開きます(下のスクリーンショット)。(プロジェクトフォルダー自身をクリックすることとは異なり)プロジェクトウィンドウ内のフォルダーを右クリックし、「プロパティ」を選択すると、そのフォルダーのプロパティダイアログが開かれます。ダイアログ設定については下で説明されています。

メモ: プロジェクトファイルがソース管理下にある場合、プロパティが表示され、プロジェクト (.spp) ファイルをチェックアウトするか問われます。設定を編集し、保存するには、「OK」をクリックします。

プロパティ

名前: Invoices-EU

ファイル拡張子: xml.svg

OK

キャンセル

検証

使用するファイル: [] 参照... ウインドウ...

XML ファイルの XSL 変換

使用する XSL: C:\Invoices\reports.xslt 参照... ウインドウ...

XML ファイルの XSLFO 変換

使用する XSL: C:\Invoices\reportsFO.xslt 参照... ウインドウ...

XML ファイルの XQuery/Update 変換

この XQuery を使用: [] 参照... ウインドウ...

XML/XQuery/Update 変換のための入力 XML

使用する XML: [] 参照... ウインドウ...

XSL/XQuery/Update 変換のファイル

保存先フォルダー: C:\Invoices\Reports 参照...

ファイル拡張子: .html

Authentic ビュー

設定を使用: [] 参照... ウインドウ...

設定

ファイル拡張子

ファイル拡張子設定は、個別のフォルダーのために有効化されており、プロジェクトフォルダーのためには有効化されていません。ファイルがプロジェクトに追加されると、定義されているファイル拡張子を持つフォルダーに追加されます。MyReport.xml という名前のファイルがプロジェクトに追加されると仮定します。(上のスクリーンショット内で示されるとおり)、.xml ファイル拡張子が Invoices-EU フォルダーで設定されている場合、MyReport.xml が Invoices-EU フォルダーに追加されます。XML ファイルを追加するフォルダーが一つ以上存在する場合、個別の XML ファイルをプロジェクトではなく直接フォルダーに追加します。

外部フォルダーのためのユーザー ID とパスワード

(外部 Web フォルダーを含む) 外部フォルダーでは、サーバーへのアクセスに必要な可能性のあるユーザー ID とパスワードを保存することができます。

検証

現在のフォルダー(または、プロジェクトの場合はプロジェクト全体)、DTD、XML スキーマ、または JSON スキーマ内で使用される [検証](#) ファイル。

XML ファイルの XSL 変換

XSLT スタイルシートはフォルダー内の XML ファイルの [XSLT 変換](#) のために使用されます。

XML ファイルの XSL-FO 変換

XSLT スタイルシートは、フォルダー内の XSL-FO での XML ファイル変換のために使用されます。

XML ファイルの XQuery/Update 変換

フォルダー内の XML ファイルの XQuery または XQuery Update 実行のために使用される XQuery または XQuery Update ファイル。

XML ファイルの XSL/XQuery/Update 変換のための XML 入力

XSLT 変換のために入力として使用される XML ファイル、またはフォルダー内で対応する XSLT、XQuery または XQuery Update ファイルを持つ XQuery/XQuery Update 実行

XSL/XQuery/Update 変換のための出力ファイル

変換の保存先ディレクトリと任意の結果ドキュメントのファイル拡張子。

XULE 実行

XMLSpy アプリケーションウィンドウ内でアクティブな XULE ドキュメントを処理するための XBRL インスタンスファイル

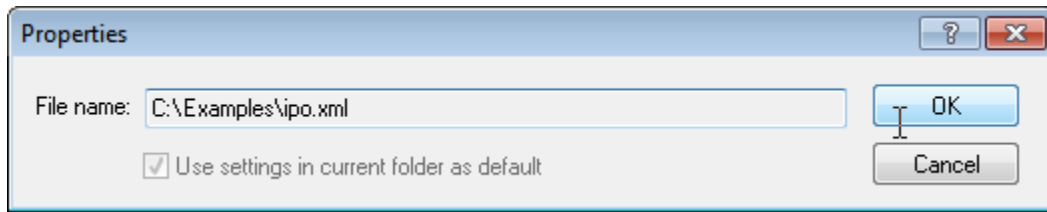
Authentic ビュー

構成の使用は、フォルダー内の XML ファイルの Authentic View 表示のために使用される StyleVision Power Stylesheet (SPS ファイル) を指定します。XML ファイルは、SPS のために使用される同じスキーマに対して有効である必要があります。

プロジェクトプロパティの注意点

以下の点と優先事項に注意してください！

- 変換、または、XSLT/XQuery 変換がプロジェクトフォルダーコンテキストメニューにより実行される場合、このダイアログで指定される検証 または 変換 ファイルに XML ファイル内の割り当てによる優先順位が与えられます。更に、個別のプロジェクトフォルダーのために指定されている設定に、祖先フォルダーのために指定されている設定による優先順位が与えられます。
- プロジェクトの複数フォルダー内で 1 つのファイルが存在する場合、異なるフォルダー内の異なる検証または変換 ファイルは、異なるフォルダーに割り当てられ、プロジェクトの外部でファイルが処理される場合、割り当てを設定することができます。これらの設定を指定する方法は以下のとおりです：割り当てを希望するプロジェクトフォルダー内のファイルを検索します。プロジェクトフォルダー内のプロジェクトを右クリックして、「プロパティ」を選択します。(下のスクリーンショット)表示されるダイアログ内で「現在のフォルダー内の設定をデフォルトとして使用」を選択します。(現在のフォルダーはファイルが存在するプロジェクトフォルダーです。)オプションが無効化された場合、現在のフォルダーの設定は、既にデフォルトの設定として使用されることを意味します。デフォルトではないプロジェクトフォルダー内のファイルインスタンスを選択すると、オプションが有効化され、デフォルトの設定をこのフォルダーの設定に切り替えることができます。ファイルローカルの割り当て(ファイル自身内の割り当て)が存在する場合、ローカルの割り当てが使用され、デフォルトのフォルダー設定は無視されます。



10.3.17 最近使用されたプロジェクト

このコマンドは最近使用されたプロジェクトを9つ表示し、それぞれのファイルへの素早いアクセスを提供します。

Authentic Desktop を起動すると、[最後に使用されたプロジェクト](#) を自動的に開くこともできます(「ツール | オプション | ファイル」セクションにて、プログラム起動時に最後のプロジェクトを開く)。

10.4 XML メニュー

XML メニューには XML ドキュメントの処理を行う際に使用されるコマンドが収められています。

 整形形式のチェック(W)	F7
 XML 検証(V)	F8

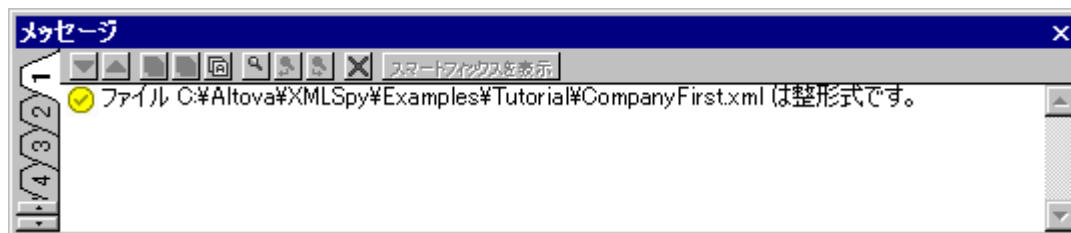
最も頻繁に使用される XML 処理には、ドキュメントの [整形形式チェック](#) や、XML ドキュメントの [検証](#) も含まれます。これらの処理に対応したコマンドはこのメニューに含まれます。

10.4.1 整形形式のチェック

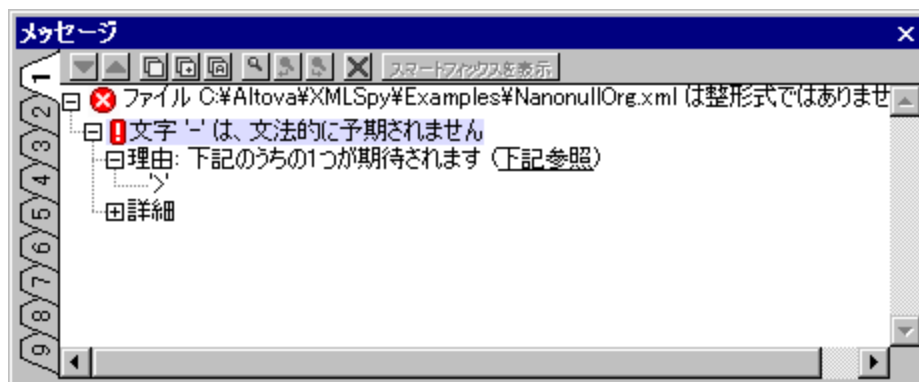


「XML | 整形形式のチェック(F7)」コマンドにより、XML 1.0 仕様の定義に従い、アクティブなドキュメントの整形形式をチェックすることができます。全ての XML ドキュメントは整形形式でなければなりません。Authentic Desktop では、ドキュメントが開かれたときと保存します。

整形形式のチェックに成功すると、その旨のメッセージがメッセージウィンドウに表示されます(以下のスクリーンショットを参照)。



整形形式チェックの結果、エラーが見つかった場合、エラーメッセージが表示されます(以下のスクリーンショットを参照)：



メモ メッセージウィンドウには9つのタブがあります。整形形式チェックの結果はアクティブなタブに表示されます。そのため、タブ1にて整形形式のチェックを行った後、タブ2に切り替えて、整形形式チェックの結果を保持した

まま、別のドキュメント検証を行うといったことができます(タブが切り替えられない場合、最初のタブに出力された情報が上書きされます)。

プロジェクトウィンドウから検証を行う

[検証] コマンドをアクティブなプロジェクト内のファイル、フォルダー、ファイルのグループに適用することができます。(クリックしてプロジェクトウィンドウ内で必要とされるファイルまたはフォルダーを選択します。[XML | Validate XMLの検証]をクリックします。プロジェクト内の無効なファイルが開かれ、メインウィンドウ内でアクティブ化され、ファイルが無効ではないことを知らせるエラーメッセージが表示されます。

メモ メッセージウィンドウには9つのタブがあります。整形形式チェックの結果はアクティブなタブに表示されます。そのため、タブ1にて整形形式のチェックを行った後、タブ2に切り替えて、整形形式チェックの結果を保持したまま、別のドキュメント検証を行うといったことができます(タブが切り替えられない場合、最初のタブに出力された情報が上書きされます)。

通常問題のあるXMLドキュメントを保存することは許可されていませんが、Authentic Desktop ではそれでも保存を行うことができます。作業を行なっているドキュメントの保存を(整形形式で無い状態で)行い、後で作業を再開するような状況で、この機能を利用することができます。

メモ 整形形式のチェックコマンドは、プロジェクトウィンドウでアクティブなファイルやフォルダー、ファイルのグループに対して使用することができます。目的のアイテムをクリックし、整形形式のチェックアイコンをクリックしてください。

10.4.2 XML の検証

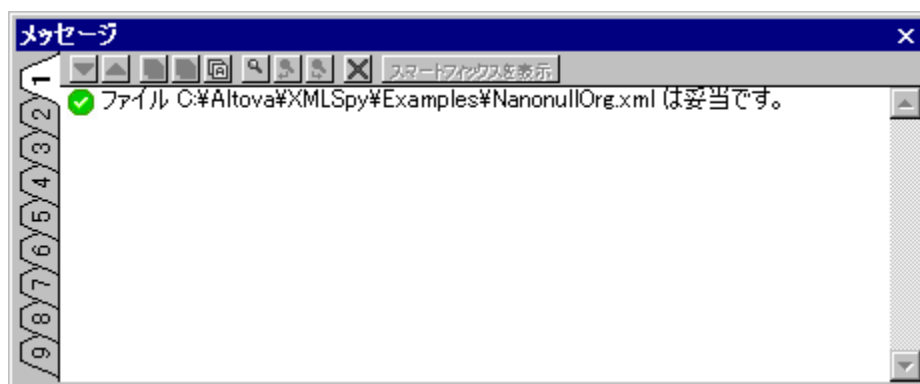


F8

「XML 検証(F8)」コマンドにより、DTD やXML スキーマ、またはその他のスキーマに対してXMLドキュメントの検証を行うことができます。ファイルが開かれたり保存されるときに、自動的にドキュメントの検証が行われるよう指定することもできます(「ツール | オプション | ファイル」)。「検証」コマンドでは、検証が行われる前に整形形式のチェックも行われます。そのため、「検証」コマンドを使用する前に整形形式のチェックを行う必要はありません。

メモ データを編集中に検証するために編集中に検証 コマンドをオン/切り替えることができます。

ドキュメントが無効な場合、メッセージウィンドウ内に検証の成功メッセージが表示されます。



そうでない場合、エラーの内容がメッセージウィンドウに表示されます(以下のスクリーンショットを参照)。リンクをクリックすることで、XML ファイル内のエラーが発見された箇所へジャンプすることができます。

メモ : 出力ウィンドウは9つのタブがあります。検証の結果はアクティブなタブに表示されます。そのため、最初のタブで整形形式のチェックを行った後、2つ目のタブに切り替えて、別のドキュメントを検証を行うことができます。2番目のドキュメントを検証するために、チェックを実行する前に、タブ2(または、タブ3)に切り替えます。タブを切り替えない場合、タブ1(またはアクティブなタブ)が最新の検証のフォーミュラにより書き込まれます。

プロジェクトウィンドウからの検証

検証コマンドはアクティブなプロジェクト内部にあるファイル、フォルダー、またはファイルのグループに対しても適用することができます。目的のファイルやフォルダーをプロジェクトウィンドウで(クリックにより)選択し、メニューオプションからXML | XML 検証」を選択するか、「F8」を押下してください。プロジェクト内の妥当ではないファイルがメインウィンドウに表示され、ファイルが妥当ではない旨のメッセージが表示されます。

RaptorXML 2021 を使用して検証を自動化する

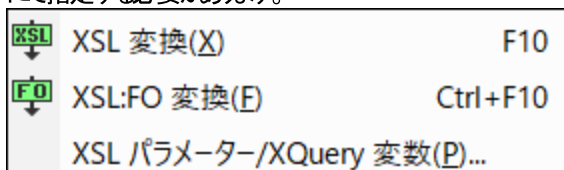
RaptorXML は、XML 検証、XSLT 変換、とXQuery 変換のためのAltova のスタンドアロンアプリケーションです。Java プログラムと.NET 内のCOM インターフェイスを使用して、アプリケーションコマンドラインから使用することができます。RaptorXML を使用すると、検証のタスクを自動化することもできます。例えば、ドキュメントのセット上で検証を実行するためにRaptorXML を呼び出すバッチファイルを作成し、テキストファイルに出力を送ることができます。詳細に関しては、[RaptorXMLドキュメンテーション](#)を参照ください。

10.4.3 編集集中に検証

編集集中に検証 コマンドは Authentic ビューで入力中に検証を行う **編集集中に検証** モードをオンとオフに切り替えます。コマンドのツールバーボタンまたは [オプションダイアログのファイルの検証](#) > **編集集中 オプション**によりモードを切り替えることができます。

10.5 XSL/XQuery メニュー

XSL 変換言語を使えば、どのようにXMLドキュメントが他のXMLドキュメントやテキストファイルへ変換されるかを指定することができます。XSLTドキュメントにより生成されるXMLドキュメントの1つにPDFの生成処理を行うために使用されるFOドキュメントがあります。Authentic Desktop には(XSLT 1.0, XSLT 2.0, and XSLT 3.0 のための) XSLT プロセッサが内蔵されており、使用中のシステムにあるFOプロセッサと連携することで、XMLファイルから様々な形式の出力を生成することができます。Authentic Desktop インターフェースから、直接FOプロセッサを使用するには、FOプロセッサの場所をオプションダイアログ([「ツール」 オプション](#))のXSLセクションにて指定する必要があります。



10.5.1 XSL 変換



F10

「XSL/XQuery | XSL 変換」コマンドにより、割り当てられたXSLTスタイルシートを使ってXMLドキュメントの変換を行うことができます。変換は、内蔵されている適切なAltova XSLTエンジン(XSLT 1.0スタイルシートに対してはAltova XSLT 1.0エンジンを、XSLT 2.0スタイルシートに対してはAltova XSLT 2.0エンジンを使用)、Microsoftから提供されているMSXMLモジュール、または外部のXSLTプロセッサを使用することで行うことができます。このコマンドにより使用されるプロセッサは、オプションダイアログ([「ツール」 オプション](#))の[XSL](#)セクションにて指定することができます。

XMLドキュメントにXSLTスタイルシートへの参照が含まれている場合、このスタイルシートが変換に使用されます(XMLドキュメントがプロジェクトに含まれている場合[プロジェクトプロパティ](#)ダイアログにて、フォルダごとXSLTスタイルシートを指定することができます。変換を行うプロジェクトフォルダやファイルを右クリックして「XSL 変換」を選択してください)。XSLTスタイルシートがXMLファイルに割り当てられていない場合、使用するXSLTスタイルシートを指定するよう促されます。ファイルの選択は([参照](#) ボタンをクリックして)グローバルリソースやURL、(ウィンドウボタンをクリックして)XMLSpyにて既に開かれているファイルからも行うことができます。

RaptorXML 2021 を使用した自動検証

RaptorXMLは、XML検証、XSLT変換、とXQuery変換のためのAltovaのスタジオアノアプリケーションです。Javaプログラムと.NET内のCOMインターフェイスを使用して、アプリケーションコマンドラインから使用することができます。このため、RaptorXMLを使用してXSLT変換タスクを実行することができます。例えば、ドキュメントのセット上でXSLT変換を実行するためにRaptorXMLを呼び出すバッチファイルを作成し、テキストファイルに出力を送ることができます。詳細に関しては、[RaptorXMLドキュメンテーション](#)を参照ください。

ZIPファイルへの変換

変換の結果を.docx拡張子を持つOpen Office XML(OOXML)やZIPファイルに収める場合は、出力ファイルのファイルパスに以下のようなZIPプロトコルを指定する必要があります:

```
filename.zip|zip/filename.xxx
```

```
filename.docx|zip/filename.xxx
```

メモ: 変換を行う前に、ディレクトリ構造の作成を行わなければならない場合もあります。Open Office XML アーカイブ形式で変換を行う場合、(例えば .docx のような)トップレベルの OOXML ファイルを作成するために、アーカイブファイルの ZIP 化を行う必要があります。

10.5.2 XSL-FO 変換



Ctrl+F10

FO は、印刷することを前提にしたドキュメントを記述するための XML フォーマットです。Apache XML プロジェクトの FOP のような FO プロセッサでは FO ファイルを入力することで PDF を出力として生成することができます。そのため XML ドキュメントから PDF を生成するにはお2つのステップを踏む必要があります。

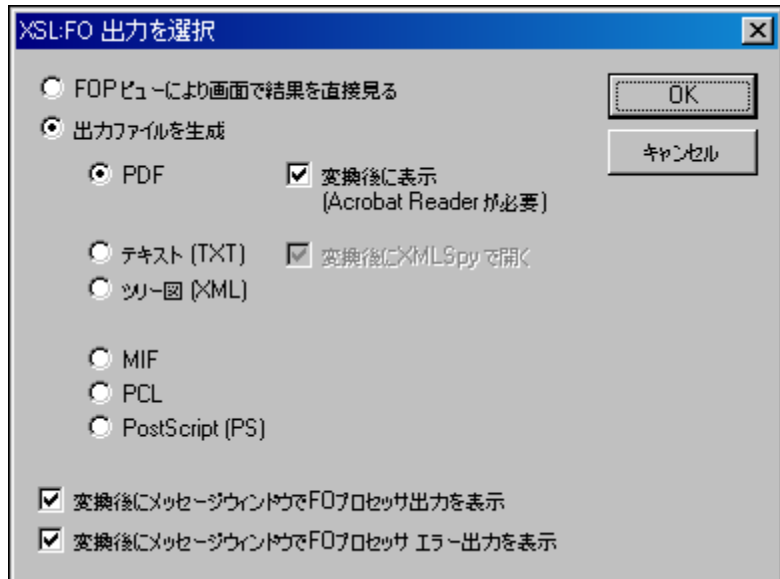
1. XSLT (または XSL-FO) スタイルシートを使うことで XML ドキュメントを FO ドキュメントへ変換します。
2. FO プロセッサにより FO ドキュメントを処理することで、PDF (またはその他の出力) を生成します。

「XSLT/XQuery | XSL:FO 変換」コマンドにより XML ドキュメントや FO ドキュメントを PDF に変換します。

- XSL:FO 変換コマンドをソース XML ドキュメントに対して実行すると、上にある両方のステップが順に処理されます。FO への変換に必要な XSLT (または XSL-FO) スタイルシートが XML ドキュメントから参照されていない場合、変換に使用するスタイルシートを指定する必要があります (以下のスクリーンショットを参照)。ファイルの選択は (参照 ボタンをクリックして) グローバルリソースや URL、(ウインドウボタンをクリックして) XMLSpy (にて既に開かれているファイルから) 行うことができます。XML から XSL-FO への変換が、オプションダイアログ (「ツール | オプション」) の [XSL タブ](#) にて指定された XSLT プロセッサにより行われます。デフォルトでは Authentic Desktop XSLT プロセッサが選択されます。結果として生成される FO ドキュメントが、オプションダイアログ (「ツール | オプション」) の [XSL タブ](#) にて指定された FO プロセッサにより変換されます。
- XSL:FO 変換コマンドが FO ドキュメントに対して実行されると、ドキュメントがオプションダイアログ (「ツール | オプション」) の [XSL タブ](#) にて指定された FO プロセッサにて処理されます。

XSL:FO 変換出力

XSL:FO 変換コマンドを選択すると、XSL:FO 出力を選択ダイアログが表示されます (以下のスクリーンショットを参照)。アクティブなドキュメントが XSLT 割り当てを持たない XML ドキュメントの場合、XSLT ファイルの選択を促されます。



FO プロセッサの出力はFOP ビューアを使用することで、画面上にて直接確認する他にも、以下のフォーマットにより出力ファイルを生成することができます:PDF、テキスト、XML エリアソー、MIF PCL、またはPostScript。FO プロセッサから得られたメッセージ出力するオプションにより、(i) プロセッサの標準出力メッセージや(ii) プロセッサのエラーメッセージをメッセージウィンドウにて表示することができます。これら2つのオプションを有効にするには、ダイアログ下部にある対応するチェックボックスにチェックを入れてください。

メモ:

- [Apache XML プロジェクト](#) のFOP プロセッサをインストールするオプションを解除しては限り、次のフォルダーにインストールされます:C:\ProgramData\Altova\SharedBetweenVersions。インストールされると、それは自動的にオプションダイアログ(ツール| オプション)内の [XSL タブ](#) に使用するFO プロセッサとして入力されます。使用するFO プロセッサアイコンを設定することができます。
- XSL: 変換コマンドは、メインウィンドウにてアクティブとなっているファイルだけでなく、アクティブなプロジェクトにて選択することができるファイルやフォルダーに対しても使用することができます。右クリックで表示されるコンテキストメニューからXSL:FO 変換を選択してください。選択されたプロジェクトフォルダーに割り当てられたXSLT スタイルシートが使用されます。

10.5.3 XSL パラメーター / XQuery 実行

「XSL/XQuery | XSL パラメーター/XQuery 変数」コマンドにより、XSLT 入力パラメーター/XQuery 外部変数ダイアログが表示されます(スクリーンショットを参照)。XSLT スタイルシートに属するパラメーターの名前や、XQuery ドキュメントへ属するXQuery 変数の名前と、対応する値を入力します。Authentic Desktop においてこれらのパラメーターは以下のように使用されます:

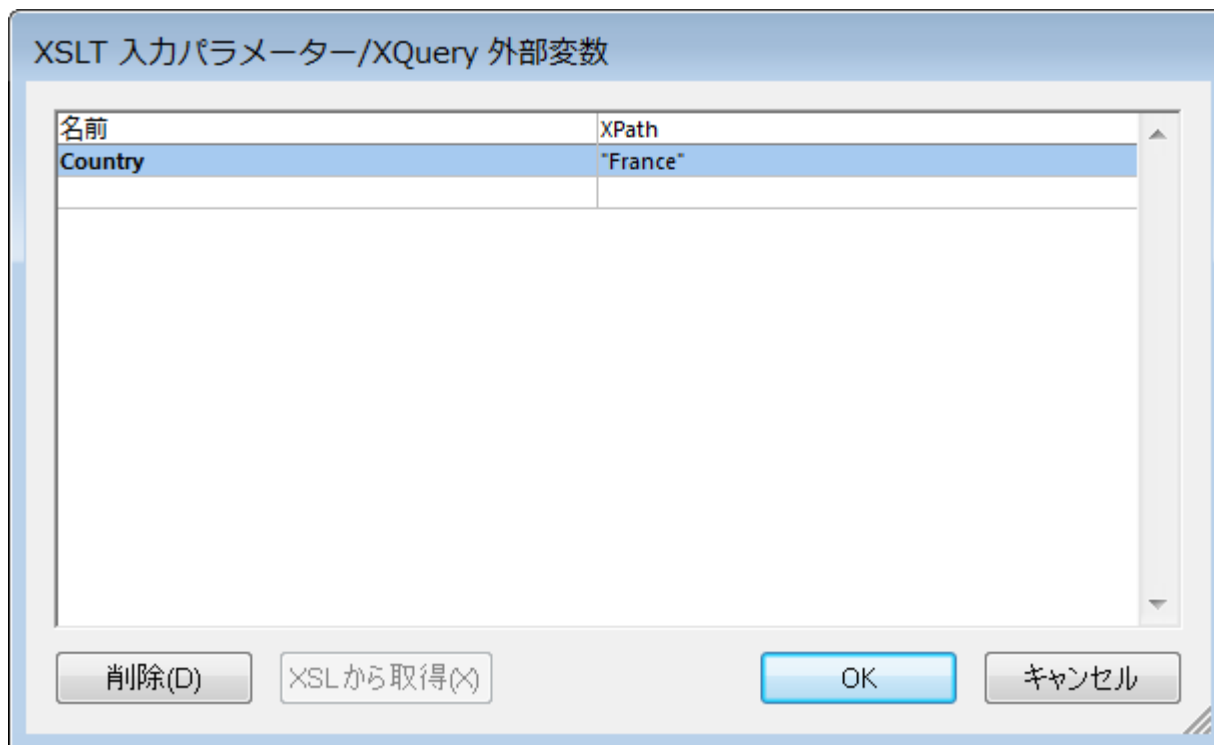
- XSL/XQuery メニューのXSL 変換コマンドがXMLドキュメントの変換に使用される際、ダイアログ内にて保存されたパラメーター値が選択されたXSLTドキュメントへ渡され、変換に使用されます。
- XSL/XQuery メニューのXQuery 実行コマンドがXQueryドキュメントの処理に使用される際、ダイアログにて現在保存されているXQuery 外部変数の値がXQueryドキュメントに渡され、実行に使用されます。

メモ XSLT 入力パラメーター/XQuery 外部変数ダイアログにて入力されたパラメーターや変数は、内蔵のAltova XSLT エンジンに対してのみ使用されます。このため、MSXML、または構成済みの外部エンジンを使用する場合、これらの値はエンジンに渡されません。

メモ XSLT 入力パラメーター/XQuery 外部変数 ダイアログ内でXSLT /パラメーター、または、外部 XQuery 変数 を定義することはエラーではありませんが、XSLT/XQuery ドキュメント、または、変換内では使用されません。

XSLT /パラメーターの使用

パラメーターに対して入力した値は引用符無し XPath 条件式や、引用符無しテキスト文字列となります。アクティブドキュメントが XSLT ドキュメントの場合、XSL から取得ボタンが有効になります。このボタンをクリックすることで、XSLT にて宣言されているパラメーターが、デフォルト値とともにダイアログに入力されます。この操作により、宣言されたパラメーターを素早く入力し、デフォルト値を必要に応じて変更することができます。



メモ パラメーター値がダイアログに入力されると、明示的に削除、または、アプリケーションが再起動されるまですべての変換で使用されます。ダイアログ内に入力されるパラメーターはそのセッションのアプリケーションレベルで指定され、この時点からIDE を使用して実行される各変換のためのそれぞれのXSLT ドキュメントのためにコピされます。これは以下を意味します:

- パラメーターは特定のドキュメントに割り当てられていたわけではありません。
- ダイアログにて入力されたパラメーターは、Authentic Desktop の終了時に削除されます。

XSLT /パラメーターの使用例

国名とそれぞれの首都を含むXML ドキュメントは以下のとおりです:

```
<document>
  <countries>
    <country name="USA" capital="Washington DC"/>
    <country name="UK" capital="London"/>
    <country name="France" capital="Paris"/>
    <country name="Russia" capital="Moscow"/>
    <country name="China" capital="Beijing"/>
  
```

```
</countries>
</document>
```

次のXSLTドキュメントは、XMLファイルから国名と首都を表示するXMLドキュメントを生成します。country という名前を持つパラメータの値として名前を入力することにより国名は選択されます(以下では黄色のハイライトにより表示されています)。

```
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:param name="country" select="'USA'"/>
  <xsl:template match="countries">
    <xsl:for-each select="country[@name=$country]">
      <country>
        <name><xsl:value-of select="$country"/></name>
        <capital><xsl:value-of select="@capital"/></capital>
      </country>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

このXSLTドキュメントが上記のXMLドキュメントで実行されると、結果は以下のようになります。

```
<country><name>USA</name><capital>Washington DC</capital></country>
```

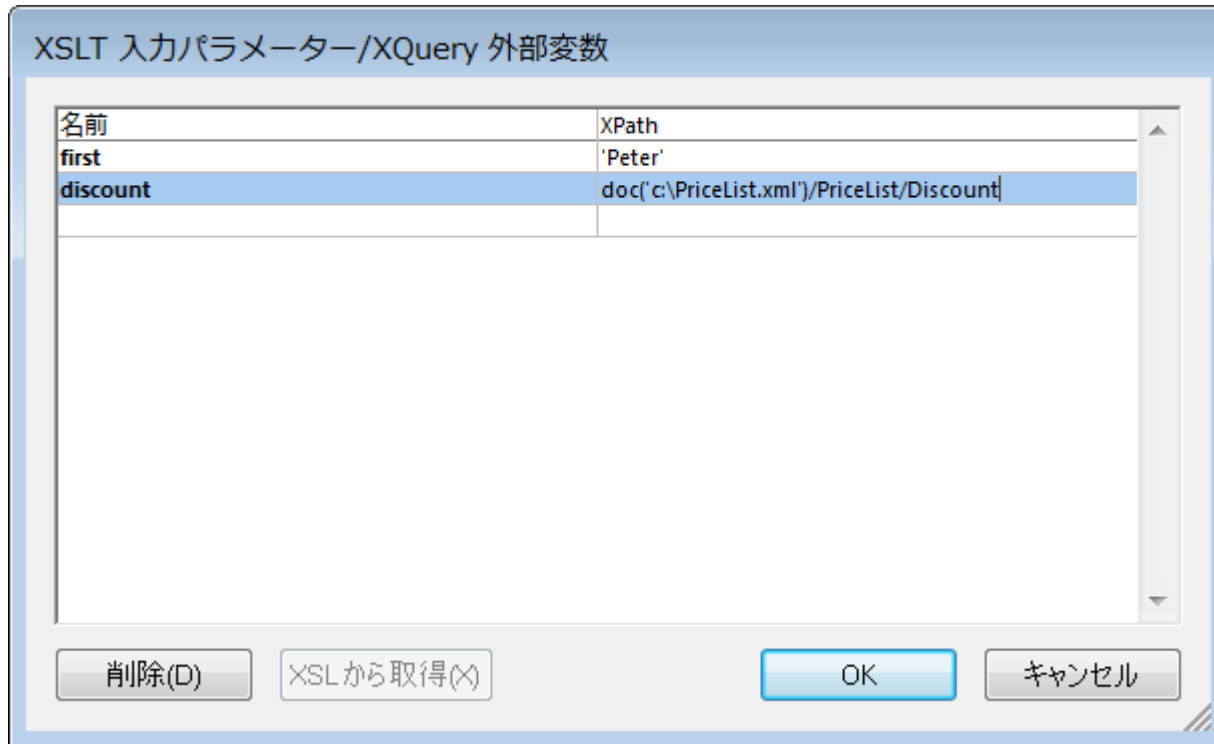
XSLT 入力パラメータ/XQuery 外部変数 ダイアログ内で country という名前のパラメータが作成され、値が与えられると、新規のパラメータの値は、XSLT 変換内のパラメータにコピーされます(上のスクリーンショット参照)。この値は変換のためにXSLT スタイルシート内のパラメータ country にコピーされます。このようにして、異なるパラメータへ異なる値をランタイムにコピーすることができます。

メモ

- XSL:FO 変換コマンド(「XSL/XQuery | XSL:FO 変換」)を使用する場合、ダイアログにて入力された値はスタイルシートへ渡されません。これらのパラメータをPDF出力にて使用するには、まずXSLT変換コマンド(「XSL/XQuery | XSL 変換」)を使用してXMLからFOドキュメントへの変換を行い、その後「XSL:FO 変換」コマンド(XSL/XQuery | XSL:FO T変換)によりFOからPDFへの変換を行う必要があります。
- 内蔵されているAltova XSLTエンジン以外のXSLTプロセッサを使用する場合、ダイアログにて入力されたパラメータは外部プロセッサへ渡されません。

外部 XQuery 変数の使用

外部 XQuery 変数に対して入力する値は、引用符無し XPath 条件式か、引用句により分離されたテキスト文字列となります。外部変数のデータ型は、XQuery ドキュメントの変数宣言にて指定されます。



メモ 外部 XQuery 変数がダイアログにて一度入力されると、そのエントリが明示的に削除されるかアプリケーションが再起動されるまで、その後の実行でもそれらの変数が使用され続けます。ダイアログにて入力された変数はアプリケーションレベルで指定されており、その後行われる実行のたびに関連する XQuery ドキュメントへ渡されます。これは以下を意味します：

- 変数は特定のドキュメントに割り当てられているわけではありません。
- ダイアログにて入力された変数は、アプリケーション (Authentic Desktop) の終了時に削除されます。

外部 XQuery 変数の使用例

以下の例では、変数 `$first` が XQuery ドキュメントにて宣言され、FLW OR ステートメントの `return` 句にて使用されます：

```
xquery version "1.0";
declare variable $first as xs:string external;
let $last := "Jones"
return concat($first, " ", $last )
```

(XSLT 入力パラメーター/XQuery 外部変数ダイアログにて入力された) 外部変数に `Peter` がセットされている場合、XQuery により `Peter Jones` という値が返されます。以下の点に注意してください！

- XQuery ドキュメントの変数宣言にある `external` キーワードにより、この変数が外部変数であるということが認識されます。
- 静的な変数の型定義はオプションです。変数の宣言時に変数のデータ型が定義されていない場合、変数の値は `xs:untypedAtomic` に割り当てられます。
- XQuery ドキュメントにて外部変数が宣言されているが、その変数名に対して外部変数が与えられない場合、エラーとなります。
- 外部変数が宣言され、XSLT 入力パラメーター/XQuery 外部変数ダイアログにて入力された場合、実行されるのは XQuery ドキュメントの範囲内となります。XQuery ドキュメントにて新たな変数が同じ名前でも宣言されている場合、その変数により範囲内の外部変数が一時的にオーバーライドされます。例えば、以下にある XQuery ドキュメントでは、(外部変数により `$first` に対して `Peter` という値が渡されているに関わらず) `Paul Jones` という値が返されます。

```
xquery version "1.0";
```

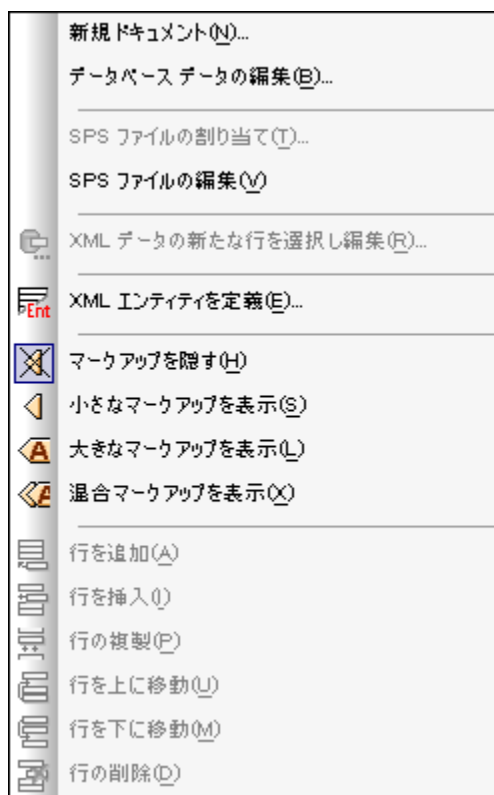
```
declare variable $first as xs:string external;  
let $first := "Paul"  
let $last := "Jones"  
return concat($first, " ", $last )
```


10.6 Authentic メニュー

Authentic View を使用することで、**Altova StyleVision** により作成された **StyleVision Power Stylesheet** (.sps ファイル) をベースにしたドキュメントの編集を行うことができます！スタイルシートにはXML ファイルを Authentic View にて視覚的に表示するための情報が取られています。表示するための情報に加えて、StyleVision Power Stylesheet によりXML ファイルヘッダーを書き込むこともできます。データはXSLT スタイルシートに用意されている機能により動的に処理され、その結果は Authentic View にて直ちに表示されます。

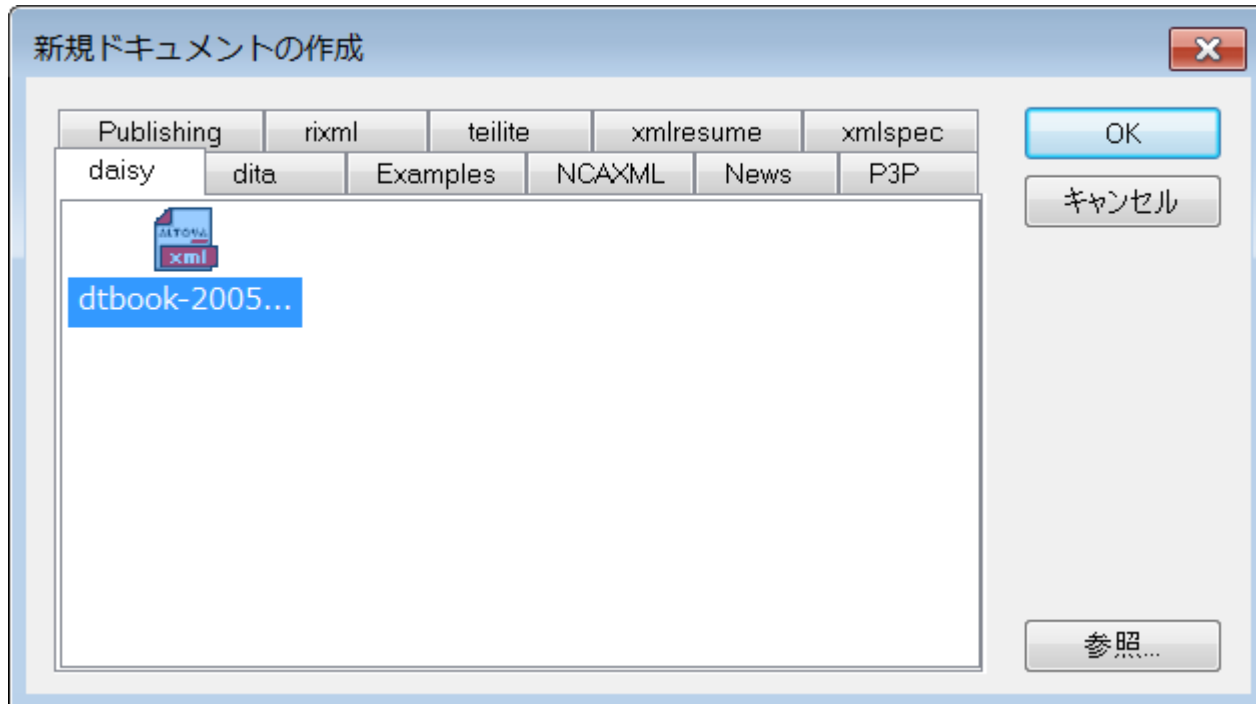
他にも StyleVision Power Stylesheet を使うことでデータベースにあるデータの編集を行うこともできます StyleVision Power Stylesheet にはデータベースへ接続するための情報が含まれており、データベースから得られたデータを Authentic View にて表示し、編集後のデータをデータベースへ書き込むことができます。

Authentic メニューコマンドにはXML ドキュメントを Authentic View にて編集するためのコマンドが取られています。のチュートリアルは、[Authentic View チュートリアル](#) セクションを参照ください。



10.6.1 新規ドキュメント

このコマンドにより Authentic View 内で新規のXML ドキュメントテンプレートを開く事ができます。XML ドキュメントテンプレートは、StyleVision Power Stylesheet (.sps ファイル)をベースとしています。新規のドキュメントの作成 ダイアログ内で StyleVision Power Stylesheet (SPS ファイル) を選択する事により開かれます(下のスクリーンショット)。SPS を選択して、「OK」をクリックすると、SPS ファイルのために定義されたXML ドキュメントテンプレートが Authentic View 内で開かれます。



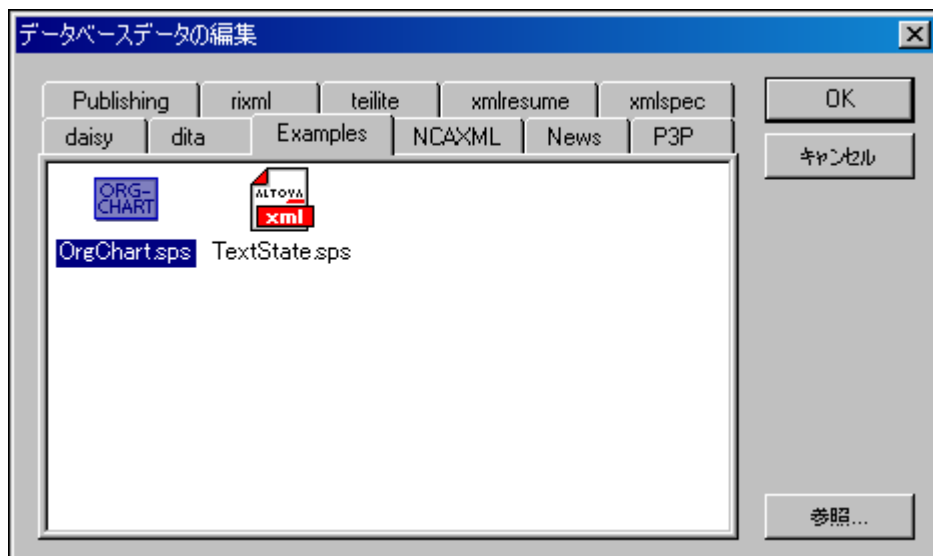
新規のドキュメントの作成 ダイアログは、一般的に使用される DTD またはスキーマをベースとした XML ドキュメントテンプレートの選択を提供します。または、テンプレート XML ファイルが割り当てられているカスタムメイドの SPS ファイルを参照することができます。SPS ファイルは、Altova StyleVision を使用して作成されます。アプリケーションにより、DTD または XML スキーマをベースとした XML ドキュメントテンプレートをデザインすることができます。StyleVision 内で必要な SPS をデザインすると、XML ファイルは (StyleVision 内で) SPS へのテンプレート XML ファイルとして割り当てられます。XML ファイル内のデータは、Authentic Desktop の Authentic View 内で開かれた新規のドキュメントテンプレートの開始データを与えます。

新規の XML ドキュメントテンプレートは、ですから、テンプレート XML ファイルとして選択された SPS と XML ファイルのデータ内で定義されたドキュメントプレゼンテーションプロパティを持つようになります。Authentic View ユーザーは、XML ドキュメントテンプレートをグラフィカルな WYSIWYG インターフェイスで編集することができ、XML ドキュメントとして保存することができます。

10.6.2 データベースの編集

「Authentic | データベースの編集...」コマンドにより、データベースのビューを Authentic View にて開くことができます。DB への接続方法や DB の表示方法、そして Authentic View にて許されている編集方法などは、StyleVision Power Stylesheet に記述されます。データベースの編集コマンドで開かれるのは、そのような DB をベースとした StyleVision Power Stylesheet です。このコマンドにより、DB への接続が確立され、DB のデータが (XML を介して) Authentic View にて表示されます。

データベースの編集... コマンドにより、データベースデータの編集ダイアログが表示されます。



目的のSPS ファイルを選択してください。この操作によりDB への接続が行われ、DB が編集できる状態でAuthentic View に表示されます。Authentic View にて表示されているDB ビューのデザインはStyleVision Power Stylesheet に含まれています。

注意: データベースの編集コマンドを使ってDB をベースとしないStyleVision Power Stylesheet や、StyleVision 2005 よりも前のStyleVision で作成されたDB ベースのStyleVision Power Stylesheet を開こうとした場合、エラーが表示されます。

メモ: StyleVision Power Stylesheet はAltova StyleVision により作成されます。

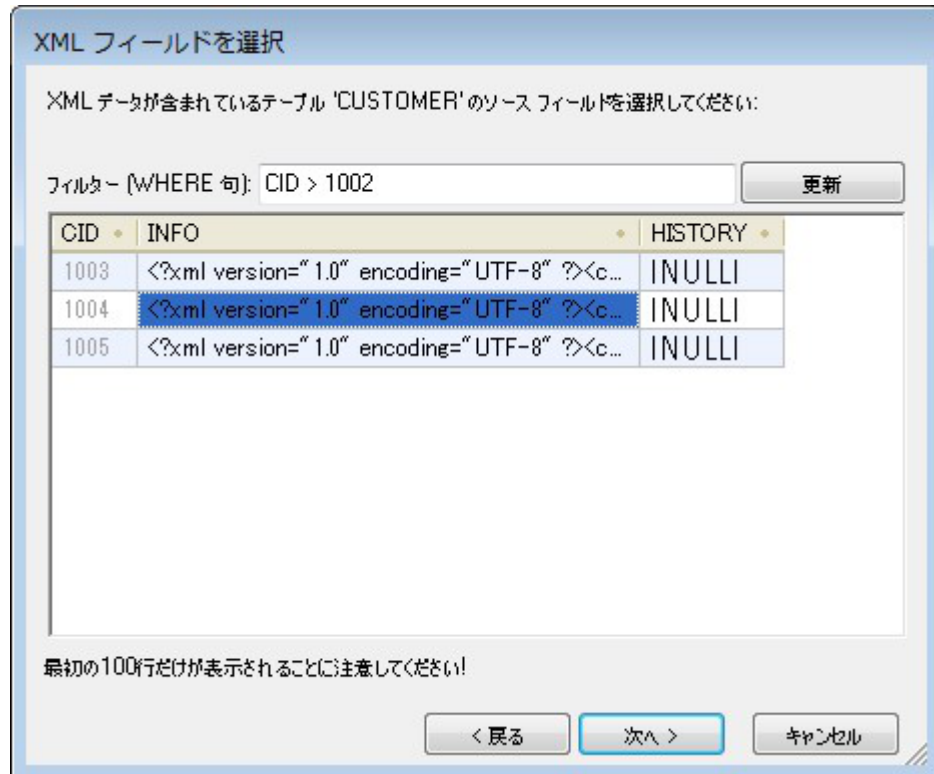
10.6.3 StyleVision スタイルシートの編集

Authentic | 編集 StyleVision スタイルシート コマンドはAuthentic View のみで使用することができ、すなわち StyleVision Power Stylesheet がXMLドキュメントに割り当てられています。StyleVision を開始し StyleVision内ですぐにStyleVision Power Stylesheet を編集することを許可します。

10.6.4 XML データの新たな行を選択し編集


「XML データの新たな行を選択し編集」コマンドにより、IBM DB2 といったXML DB にあるテーブルから新たな行を選択することができます。新たな行はAuthentic View に表示され、編集を行う後にDB への書き込みを行うことができます。

XML DB がXML データソースとして使用される場合、XML データカラムのセルに収められているXMLドキュメントがAuthentic View に表示されます。XML データの新たな行を選択し編集コマンドにより、XML カラム上の他のセル(または行)にあるXMLドキュメントを選択することができます。このコマンドを選択することで、XML フィールドの選択ダイアログが表示され、XML カラムに含まれているテーブルが表示されます。



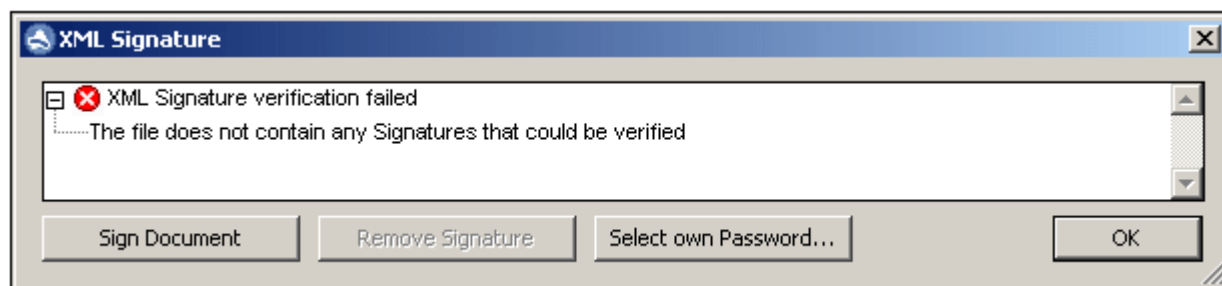
フィルタリングによりこのテーブルに表示される内容を制限することができます。フィルタリングに使用されるのはSQLのWHERE句です(例えば CID>1002 としたWHERE句の中に入る条件になります)。更新ボタンをクリックして、ダイアログの表示内容を更新することができます。上のスクリーンショットではフィルタリングされた結果を見ることができます。目的のXMLドキュメントが表示されているセルを選択してOKをクリックします。選択されたセルのXMLドキュメントがAuthentic Viewにロードされます。

10.6.5 XML 署名

関連付けられているSPSにてXML署名が有効になっている場合、Authenticビューで「XML署名」コマンドを使用することができるようになります。Authentic ツールバーにある「XML署名」ツールバーアイコン  からXML署名コマンドへアクセスすることができます。

検証と独自の証明書/パスワード

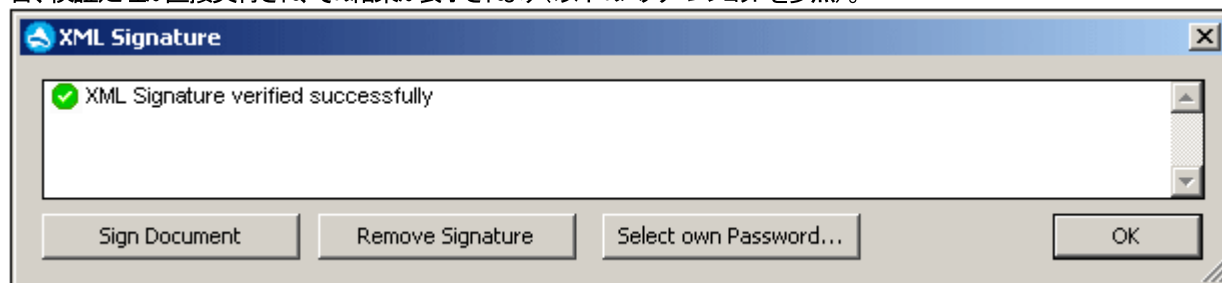
「XML署名」コマンドをクリックすると、署名の検証処理が開始されます。ドキュメント内に署名が存在しない場合、その結果がXML署名ダイアログに表示され、表示されているボタンからAuthentic Viewユーザーがドキュメントの署名を行うことができるようになります(以下のスクリーンショットを参照)。



Authentic View にて独自の証明書/パスワードを選択することができるようオプションが指定されている場合、「独自の署名を選択」または「独自のパスワードを選択」ボタンがダイアログにて表示されます。認証において証明書が使用されるかパスワードが使用されるかは、SPS デザイナーが署名の設定を行う際に指定されます。署名は証明書またはパスワードをベースにしたものとなります。ダイアログにボタンが表示される場合、そのボタンをクリックすることで、Authentic View が証明書を選択、またはパスワードを入力することができるようになります。Authentic View ユーザーの選択はメモリ上に保管されており、そのセッションでのみ有効となります。証明書の選択またはパスワードの入力を行う後、ドキュメントやアプリケーションが閉じられると、証明書/パスワードの設定も、SPS に保存されたオリジナルの設定に戻されます。

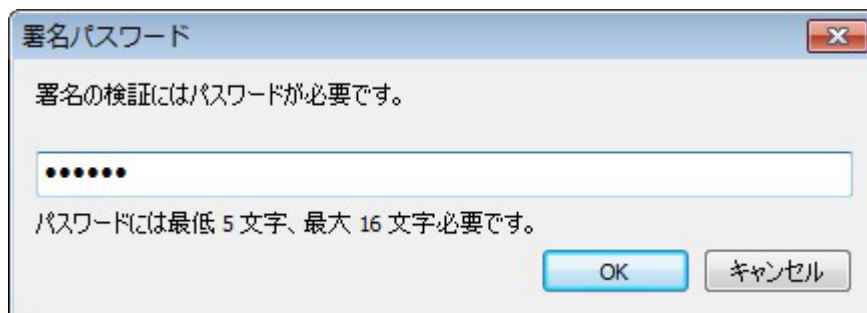
検証ならびに認証情報

単一のドキュメントに対して検証処理が行われた場合、2つのケースが考えられます。認証情報(署名またはSPS にて)が利用できる場合、検証処理が直接実行され、その結果が表示されます(以下のスクリーンショットを参照)。



認証情報は証明書の鍵情報または署名時に使用したパスワードとなります。XML ドキュメントが署名されたときに、証明書の鍵情報が署名の内に収められるか、またはパスワードをベースにした署名の場合、パスワードがSPS の中に収められるかは、SPS デザイナーにより指定されます。どちらの場合でも認証を行うことができ、検証処理はAuthentic View ユーザーからの入力が必要とすることなく直接実行されます。

もう一つのケースは、認証情報が署名の中に存在しない(証明書の場合)、またはSPS ファイルの中に存在しない(パスワードの場合)というものです。この場合、認証情報(パスワードが証明書の場所)を入力するよう求められます(以下のスクリーンショットを参照)。



10.6.6 XML エンティティの定義

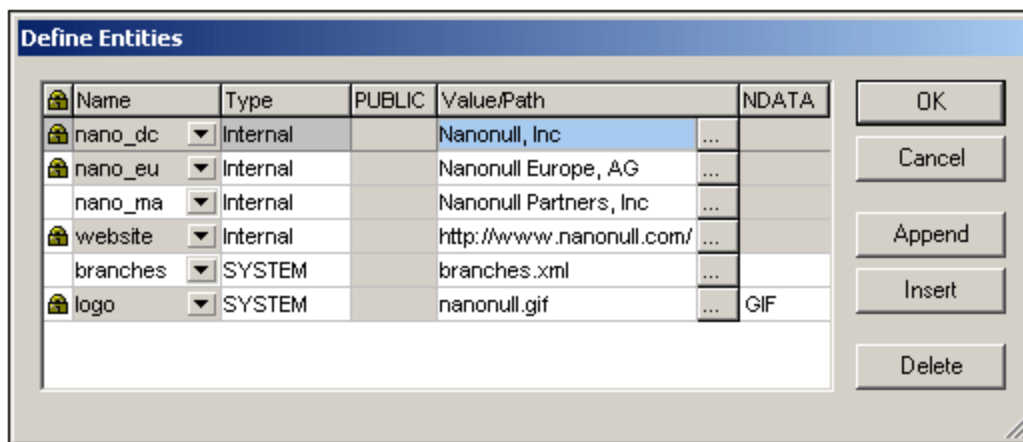
Authentic View 内で使用するエンティティを定義することができます。定義後、これらのエンティティはエンティティの入力ヘルパー内に表示され、コンテキストメニューの「エンティティの挿入」サブメニュー内に表示されます。エンティティの入力ヘルパー内のエンティティをダブルクリックすると、カーソル挿入ポイントにエンティティが挿入されます。

テキスト文字列、XML フラグメント、または他の外部リソースをドキュメント内の複数の場所で使用する場合、エンティティは役立ちます。必要とされるデータの目的の短い名前であるエンティティを定義することは、エンティティの定義ダイアログを使用します。エンティティを一度定義すると、そのエンティティをドキュメントの複数箇所で使用することができるようになります。この機能により、編集時間の短縮や、メンテナンスコストを大幅に軽減することができます。

ドキュメントで使用するのことができるエンティティは、XML データ(テキスト文字列またはXML ドキュメントのフラグメント)の解析対象実体(解析対象エンティティ)と、バイナリファイル(通常はイメージや音楽などのマルチメディアオブジェクト)の様な非 XML データを扱う解析対象外実体(解析対象外エンティティ)と、大きく分けて2つの種類があります。各エンティティは名前と値が与えられています。解析されたエンティティの場合、エンティティはXML データのためのプレースホルダです。エンティティの値はXML データ自身、または、データを含む.xml ファイルをポイントするURI です。解析されていないエンティティの場合、エンティティの値は非XML データファイルをポイントするURI です。

エンティティを定義する方法

1. 「Authentic | XML エンティティの定義」をクリックします。エンティティの定義ダイアログが表示されます。



2. 名前フィールドにエンティティ名を入力します。この値がエンティティ入力ヘルパーに表示される名前となります。
3. エンティティの型を型フィールドのドロップダウンリストから入力します。3つある種類から選択を行います。テキストが使用される **Internal** エンティティがXML ドキュメント自身内に保管されています。 **PUBLIC** または **SYSTEM** では、エンティティのリソースがXML ファイルの外に配置され、それぞれPUBLIC 識別子とSYSTEM 識別子を使用することで、リソースが特定されます。システム識別子でリソースの場所を表すURI が記述されます。公開識別子はプロセッサがリソースを識別するためのローケーションに影響されない識別子です。公開とシステム識別子の両方を指定すると、公開識別子はシステム識別子、システム識別子を使用して解決します。
4. PUBLIC がType として選択されている場合、PUBLIC フィールド内にリソースの公開識別子を入力します。Type にて Internal またはSYSTEM を選択した場合、PUBLIC フィールドが無効になります。
5. **Value/Path** フィールドでは、以下のどれかを入力することができます:
 - エンティティの種類がInternal の場合、エンティティの値となるテキスト文字列を入力してください。エントリに引用符を入力しないでください。入力する引用符はテキスト文字列の一部として扱われます。
 - SYSTEM がエンティティの型である場合、参照ボタンを使用してリソースのURI、またはローカルネットワーク上のリソースを入力します。リソースが参照可能エンティティである場合、参照されるリソースはXML ファイル(つまり.xml 拡張子を持ったファイル)である必要があります。他にも、GIF ファイルのよびバイナリファイルを使用することもできます。
 - エンティティの型がPUBLIC の場合、このフィールドにシステム識別子を追加で入力する必要があります。

6. NDATA エントリはプロセッサへのエンティティを解析せず適切なプロセッサに送信されるように命令します。そのため NDATA フィールドは解析対象外エンティティに対してのみ使用されます。

ダイアログ機能

対応するボタンをクリックすることでエンティティの追加、挿入、そして削除を行うことができます。カラムのヘッダーをクリックすることで、エンティティをアルファベット順に並び替え、1回クリックすることで昇順、2回クリックで降順の並び替えとなります。

ダイアログボックスの大きさならびにカラムの幅を変更することができます。エンティティが一度 XML ドキュメント内で使用されると、そのエンティティはロックされ、エンティティの編集ダイアログにて編集することができなくなります。ロックされたエンティティは、最初の列内でロックシンボルにより表示されます。エンティティをロックすることにより、エンティティに対して XML ドキュメントが有効であることを保証することができます (エンティティが参照されており、定義されていない場合は、ドキュメントは無効となります)。

エンティティの複製はフラグされます。

制約

- 他のエンティティ内に含まれているエンティティは、ダイアログ、Authentic View、XSLT 出力にて解決されず、そのようなエンティティのアンチ・サントはエスケープされた形 (つまり `&`) で表示されます。
- 外部エンティティは、そのエンティティがイメージである場合と ENTITY または ENTITIES 型の属性値として入力された場合を除き Authentic View にて解決されません。このようなエンティティは、SPS から生成された XSLT にてドキュメントが処理されたときに解決されます。

10.6.7 マークアップの表示

Authentic XML ドキュメント内のマークアップをコントロールするオプションを使用するサブメニューがマークアップコマンドに搭載されています。これらのオプションは下で説明されています。



マークアップを隠すコマンドにより、Authentic View に表示されているマークアップシンボルを非表示にします。



小さなマークアップを表示コマンドにより、小さなマークアップシンボルを Authentic View にて表示します。



大きなマークアップを表示コマンドにより、大きなマークアップシンボルを Authentic View にて表示します。



混合マークアップを表示コマンドにより、異なるマークアップシンボルが Authentic View で表示されます。StyleVision Power Stylesheet では、デザイン時ドキュメントの各要素/属性に対して、大きなマークアップ、小さなマークアップ、またはマークアップ無しというオプションを指定することができます。混合マークアップのモードでは、カスタマイズされたマークアップが Authentic View に表示されます。

10.6.8 行の追加/挿入/複製/削除



行の追加コマンドにより、Authentic View の現在アクティブなテーブルに新たな行を追加します。



行の挿入コマンドにより、Authentic View の現在アクティブなテーブルに新たな行を挿入します。



行の複製コマンドにより、Authentic View の現在アクティブなテーブル行を複製します。



行の削除コマンドにより、Authentic View の現在アクティブなテーブル行を削除します。

10.6.9 上に/下に移動



行を上に移動コマンドにより、Authentic View における現在のテーブル行を、1行上に移動します。



行を下に移動コマンドにより、Authentic View における現在のテーブル行を、1行下に移動します。

10.6.10 HTML、RTF、PDF、Word 2007+ ドキュメントの生成

以下にある4つのコマンドにより、PXF ファイル内部に収められた Authentic View XML ドキュメントから出力ドキュメントを生成することができます:

- HTML ドキュメントの生成
- RTF ドキュメントの生成
- PDF ドキュメントの生成
- Word 2007+ ドキュメントの生成

これらコマンドへは Portable XML Form (PXF) ツールバーからもアクセスすることができます(以下のスクリーンショットを参照)。

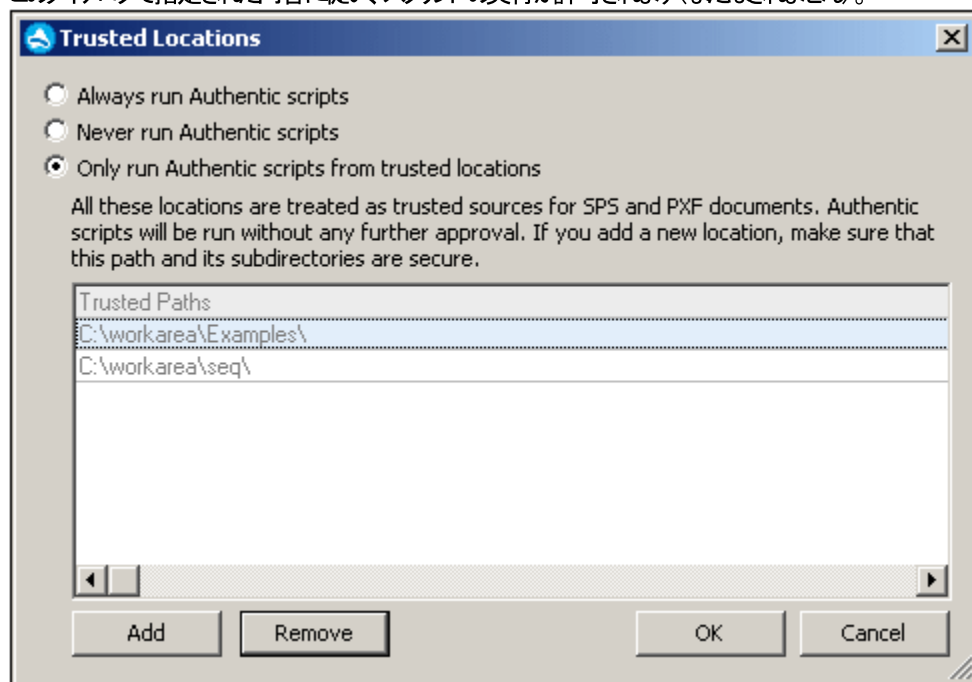


個々のコマンドまたはボタンをクリックすることで、HTML、RTF、PDF、または DocX 形式の出力がそれぞれ生成されます。

これらのボタンは PXF ファイルが Authentic View にて開かれた時に有効になります。PXF ファイル内にそれぞれの出力形式に対応した XSLT ファイルが含まれることで、個々のコマンドならびにボタンが有効になります。例えば、HTML ならびに RTF を出力する XSLT スタイルシートが PXF ファイル内部に含まれている場合、HTML ならびに RTF 出力を行うためのコマンドならびにツールバーボタンが有効になり、PDF と DocX (Word 2007+) 出力を行うコマンドは無効状態となります。

10.6.11 信頼された場所

信頼された場所コマンドを選択すると、(以下のスクリーンショットにある)信頼された場所ダイアログが表示され、SPS 内にあるスクリプトのセキュリティ設定を指定することができます。スクリプトを含む SPS をベースとしている XML ファイルを Authentic ビューで開いた場合、このダイアログで指定された内容に従い、スクリプトの実行が許可されます(またはされません)。



以下にある3つのオプションが指定可能です:

- Authentic ビューにてファイルが開かれた時に、Authentic スクリプトを常に動作させる。
- Authentic ビューにてファイルが開かれても、Authentic スクリプトを動作させない。
- 信頼された場所にある Authentic スクリプトだけを動作させる。信頼された場所(フォルダー)のリストが下部のペインに表示されます。追加ボタンを使用することで、フォルダーの参照とリストへの追加を行うことができます。リストからエンTRIESを削除するには、リストからエンTRIESを選択した後、削除ボタンをクリックしてください。

10.7 表示メニュー

表示メニューでは(下のスクリーンショット)では、アクティブな [メインウィンドウ](#) の表示設定を行い、ドキュメントの表示方法を変更することができます。

このセクションでは、「表示」メニューから使用することのできるコマンドを説明します。

10.7.1 Authentic View



このコマンドは、現在のドキュメントを [Authentic View](#) に切り替えます。

Authentic View を使用すると Altova の StyleVision アプリケーションにより作成された StyleVision Power Stylesheet テンプレートをベースに XML ドキュメントを編集することができます。これらのテンプレート (StyleVision スタイルシート または SPS ファイル) は、XML ドキュメントをグラフィカルなフォーマットで表示し、XML ドキュメントを (マークアップを使用するテキストフォーマット編集するより) 簡単に編集することができます。

10.7.2 ブラウザービュー

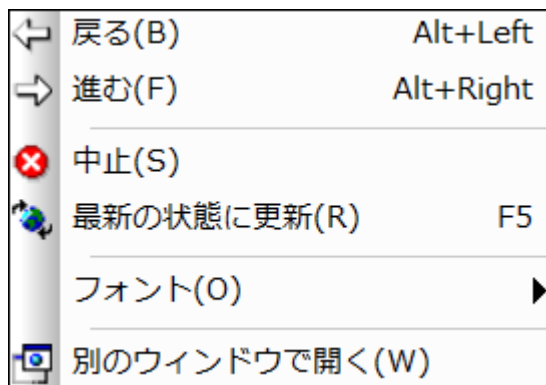


このコマンドにより、現在のドキュメントビューを [ブラウザービュー](#) へ切り替えます。このビューでは XML が利用可能なブラウザを使用して、CSS や XSL スタイルシートにより得られた情報による XML ドキュメントのレンダリングを行います。

[オプションダイアログのファイルタブ](#) にて保存時に検証を行うよう設定している場合、ブラウザービューへの切り替えが行われる際に、ドキュメントの検証が行われます。オプションダイアログは、「ツール | オプション」メニューコマンドにより開くことができます。詳細に関しては、このドキュメントの [ブラウザービュー](#) のセクションを参照してください。

10.8 ブラウザーメニュー

「ブラウザ」メニューのコマンドは、[ブラウザビュー](#)で有効になります。「戻る」と「進む」コマンドはしかしながら、前回使用されたコマンドに移動することのできるスキーマビューでも有効化することができます。



10.8.1 戻る



戻るコマンド (ショートカット: Alt + 左矢印) は、ブラウザビューとスキーマビューで有効化されます。

ブラウザビューにて、「戻る」コマンドにより、その前に表示されたページが表示されます。バックスペースキーを押下することも同様の操作を行えます。戻るコマンドは、XMLドキュメントのリンクをクリックした後に、再度XMLドキュメントへ戻るために使用することができます。

スキーマビューでは、戻るコマンドにより、その前に閲覧していたコンポーネントが表示されます。Alt + 左矢印がショートカットキーになります。戻るコマンドにより、最大500回前の閲覧箇所まで戻ることができます。

10.8.2 進む



進むコマンド (ショートカット: Alt + 右矢印) は、ブラウザビュー内で有効化されます。「進む」コマンドは、「戻る」コマンドを使用した後に有効になります。このコマンドにより、(i) ブラウザビューにて既に閲覧したページや、(ii) スキーマビューにて既に閲覧したスキーマコンポーネントを閲覧することができます。

10.8.3 中止



中止コマンドにより、ブラウザービューで行われているドキュメントの読み取りを中止します。スピードが遅いインターネット回線などにより外部ファイルや画像などがダウンロードされており、その処理を停止した場合にこのコマンドを使用することができます。

10.8.4 最新の状態に更新



最新の状態に更新 (F5) コマンドにより、ブラウザービューのドキュメントや、CSS、XSL スタイルシート、DTD と関連するドキュメントが再ロードされます。

10.8.5 フォント

フォントコマンドにより、XMLドキュメント内にあるテキストのレンダリングに使用されるデフォルトのフォントサイズを選択することができます。殆どのブラウザーにおける文字のサイズコマンドに似た動作をします。

10.8.6 別のウィンドウ



別のウィンドウで開くコマンドにより、ブラウザービューが別のウィンドウにて開かれ、他のビューと隣合わせで表示できるようになります。ブラウ別のウィンドウでは、ブラウザービューはドキュメントの編集ビューと左右に表示することができます。

ブラウザービューが別のウィンドウで表示されていると、編集ビューにて F5 を押下することで、対応するブラウザービューの内容も更新されます。ウィンドウを元のインターフェースに統合するには、アクティブなウィンドウの右上にある最大化ボタンをクリックしてください。

10.9 ツールメニュー

ツールメニューを使用して以下を行うことができます:

- XMLドキュメントの[スペル](#) チェックをチェックします。
- Authentic Desktop の[スクリプト環境](#) にアクセスします。フォーム、マクロ、イベントハンドラーを作成、管理、および保管することができます。
- 現在割り当てられているマクロを[ビュー](#)します。
- 差異をチェックするために2つのファイルを比較する。
- 差異をチェックするために2つのフォルダーを比較する。
- 外部アプリケーションを使用するカスタマイズされたコマンドにアクセスします。これらのコマンドは、[カスタマイズダイアログのツールタブ内で作成することができます](#)。
- [グローバルリソースの定義](#)
- XMLSpy 内のグローバルリソースのために[アクティブな構成を変更](#)します。
- Authentic Desktop の自身のバージョンを[カスタマイズ](#)します: 自身のツールバー、キーボードショートカット、メニューとマクロを定義します。
- グローバルな Authentic Desktop [設定](#)を定義します。

10.9.1 スペリング

内蔵の言語辞書を搭載した Authentic Desktop のスペルチェッカーは、Authentic View.

メモ: Altova ソフトウェアに内蔵されている辞書は、Altova の好みにより選択されたものではありません。[MPL](#)、[LGPL](#)、または [BSD](#) ライセンスという商用ソフトウェアにおける再配布が認められているライセンス下で利用可能かどうかによって依存しております。[GPL](#) ライセンスなどのより強力なライセンス下で配布されています。これらの辞書は <https://www.altova.com/ja/dictionaries> に配置されているインストーラーからご利用いただけます。ライセンスと有用性を基に使用する辞書を選択してください。

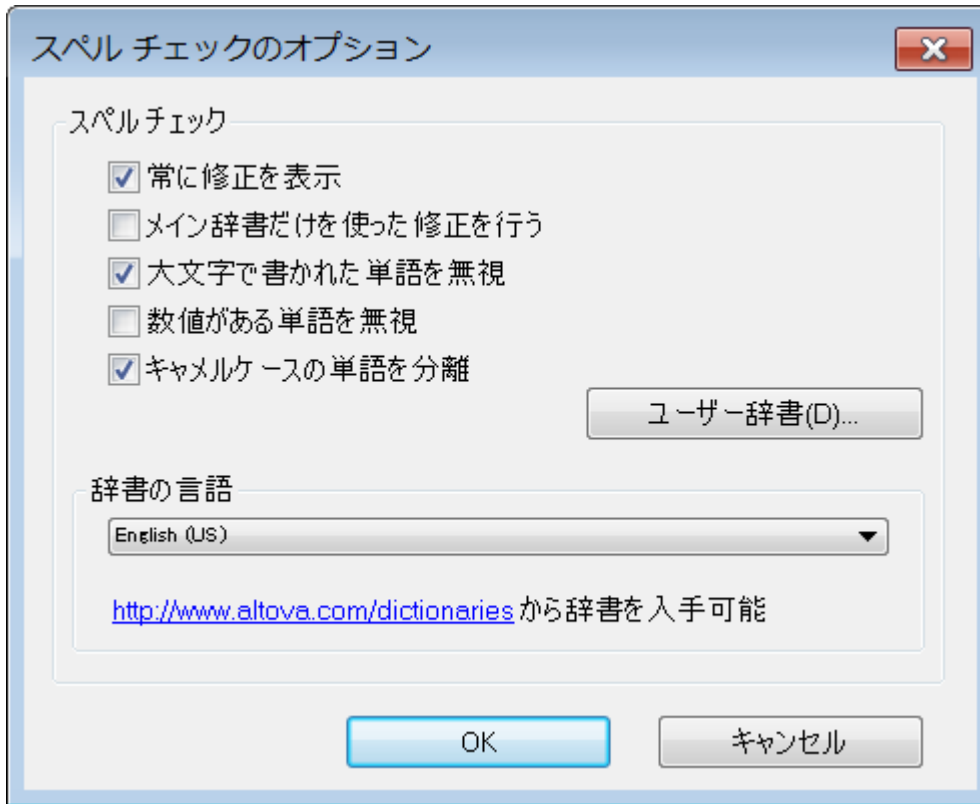
このセクションではスペルチェッカーの使用方法について記述され、以下の3つのサブセクションから構成されます:

- [スペルチェッカーの言語を選択](#)
- [スペルチェッカーを起動](#)する

スペルチェッカーの言語を選択

以下の操作により、スペルチェックを行う言語をセットすることができます:

1. 「ツール | スペルチェックのオプション」メニューコマンドをクリックします。
2. スペルチェックのオプションダイアログにて、辞書の言語コンボボックスのリストからインストールされた辞書を選択します(以下のスクリーンショットを参照)。

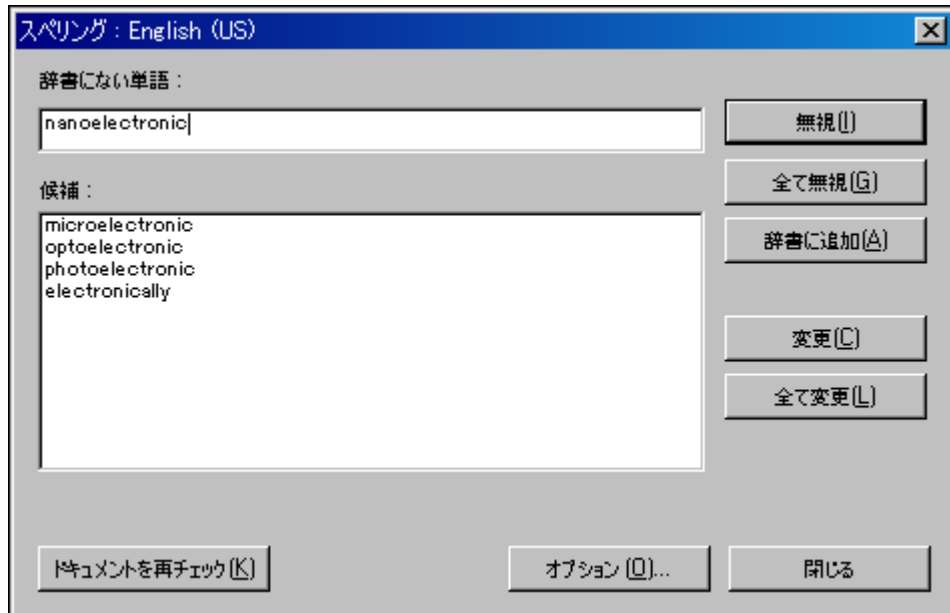


3. **OK** をクリックして完了します。

選択された言語の辞書がスペルチェックに使用されます。スペルチェックを行う言語の辞書がインストールされていない場合、新たな辞書をダウンロードすることができます。ダウンロードの方法については、[スペルチェッカーに辞書を追加のセクション](#)を追加を参照ください。

スペルチェッカーを実行する

「ツール | スペルチェック」コマンド (Shift + F7) により、現在アクティブになっているXMLドキュメント内における、**に対して**チェックが自動的に始まります。未知の単語が発見されると、**スペルング**ダイアログが表示されます(以下のスクリーンショットを参照)。未知の単語が無い場合、スペルチェックがドキュメントの最後まで行われます。



スペリングダイアログで使用するのことができるオプションを以下に記します:

辞書にない単語

このテキストボックスには、選択された言語辞書やユーザー辞書にて発見できなかった単語が表示されます。以下のオプションが利用可能です:

- テキストボックスに表示されている単語をキーボードを使って編集するか、候補リストから選択することができます。その後変更ボタンをクリックすることで、XMLドキュメント内にある単語が、編集された単語に置き換えられます(候補リストにある単語をダブルクリックすることで、XMLドキュメントの内容が直接置き換わります)。単語が辞書にない単語テキストボックスに表示されると、XMLドキュメント内の対応する箇所がハイライトされ、ドキュメント内の単語を直接編集することもできます。全て変更をクリックすることで、置換処理がXMLドキュメント全体に対して行われます。
- 変更を行わず、現在ハイライトされている箇所のみ、またはドキュメント全体においてスペルチェッカーの警告を無視することができます。
- 単語をユーザー辞書に追加することで、それ以降のチェックにおいてその単語が正しいものであると認識させることができます。

候補

このリストボックスには、未知の単語に近い(言語辞書ならびにユーザー辞書から得られ)単語が表示されます。リストにある単語をダブルクリックすることで、その単語がドキュメント内へ挿入され、スペルチェック処理が続けられます。

一度無視する

このコマンドにより、現在選択されている未知の単語を無視して、ドキュメントのチェックを続けることができます。同じ単語がドキュメント内で再度発見された場合、その単語が再度表示されます。

全て無視

このコマンドにより、現在の未知単語をドキュメント全体において無視することができます。

辞書に追加

このコマンドにより、未知の単語をユーザー辞書へ追加します。ユーザー辞書へは [スペルチェックのオプション](#) ダイアログにてアクセス、編集を行うことができます。

変更

このコマンドにより、XMLドキュメント内で現在ハイライトされている単語が、(変更された) *辞書のない単語* テキストボックス内の単語に置き換えられます。

全て変更

このコマンドにより、XMLドキュメント内で現在ハイライトされている単語が、ドキュメント全体で(変更された) *辞書のない単語* テキストボックス内の単語に置き換えられます。

ドキュメントを再チェック

ドキュメントを再チェックボタンにより、ドキュメントの先頭から再度チェックを行います。

オプション

現在使用されているビューにより以下のダイアログボックスが表示されます:

- 現在のビューがAuthentic View の場合、[スペルチェックのオプション](#) ダイアログボックスが表示されます。

ダイアログボックスに関する詳細は[スペルチェックのオプション](#)セクションを参照ください。

閉じる

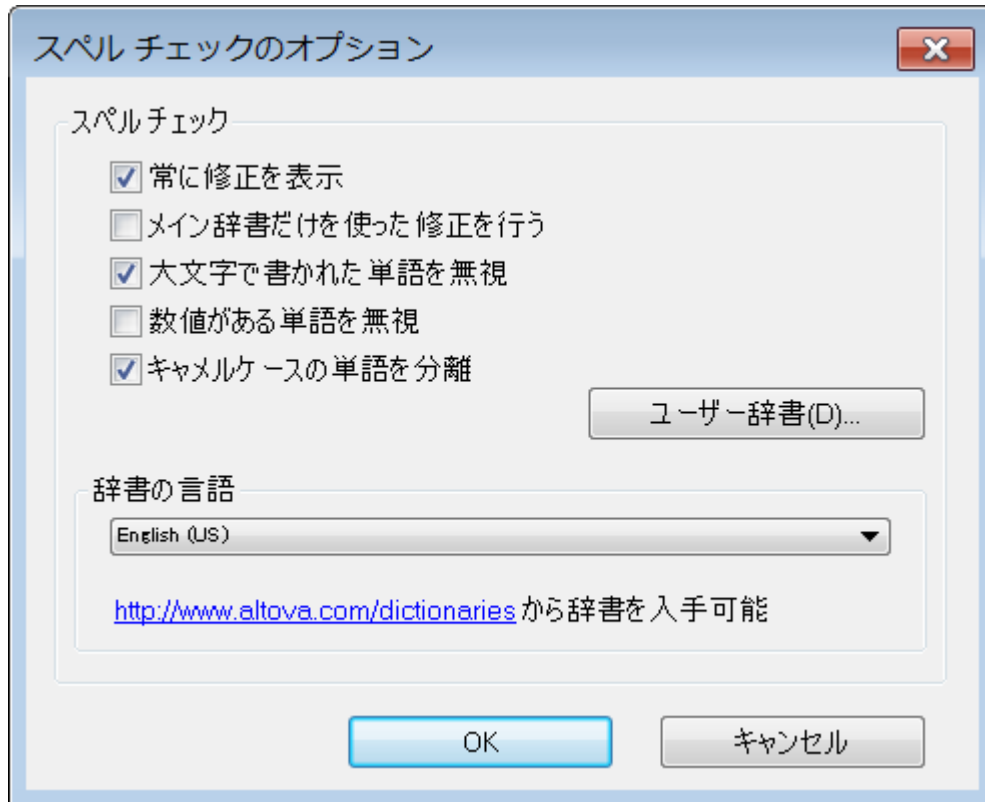
スペルチェックダイアログボックスが閉じられます。

10.9.2 スペルチェックのオプション

「ツール | スペルチェックのオプション」コマンドにより [スペルチェックのオプション](#)

スペルチェックのオプション

スペルチェックのオプションダイアログではグローバルなスペルチェッカーオプションを設定することができます。

**常に修正を表示:**

このオプションを選択することで、ユーザー辞書とそれ以外の辞書から得られた単語の候補が候補リストボックスに表示されます。このオプションを無効にすると、候補が表示されなくなります。

メイン辞書だけを使った修正を行う

このオプションを選択することで、言語辞書（メイン辞書）だけが使用されます。提案にユーザー辞書は使用されません。「ユーザー辞書」ボタンも無効化され、ユーザー辞書の編集を行うことができなくなります。

大文字で書かれた単語を無視:

このオプションを選択することで、すべて大文字で記述された単語がチェックの対象から外されます。

数値を伴う単語を無視:

このオプションを選択することで、数値を含んだすべての単語が無視されます。

キャメルケースの単語を分離:

単語内に大文字の文字が含まれる単語をキャメルケースの単語とします。例えば“CamelCase”という単語では“Case”の“C”が大文字で示されているため、キャメルケースの単語として認識されます。キャメルケースの単語は通常辞書に含まれていないため、スペルチェッカーによりエラーとして認識されます。キャメルケースの単語オプションにより、キャメルケースの単語を大文字からなるパーツに分離して、個々のパーツをチェックすることによりこの問題を回避することができます。このオプションはデフォルトでチェックされています。

辞書の言語:

このエポボックスではスペルチェッカーにて使用される辞書の言語が選択されます。デフォルトの選択はEnglish (US) となります。[Altova Web サイト](http://www.altova.com/dictionaries)からはその他の言語の辞書を無料でダウンロードいただけます。

スペルチェッカーに辞書を追加

辞書の言語は .aff ファイルと .dic ファイルという 2 つの Hunspell 辞書ファイルから構成されます。言語辞書は、以下の場所にある Lexicons フォルダにインストールされます。全ての言語の辞書は次の場所にある Lexicons フォルダ内にインストールされます: C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons.

Lexicons フォルダ内部には、異なる言語に対応した辞書のファイルが、〈言語名〉※ 辞書ファイル という構成で各フォルダに収められます。例えば 2 つの (English (British) と English (US)) 英語辞書が以下のように配置されます:

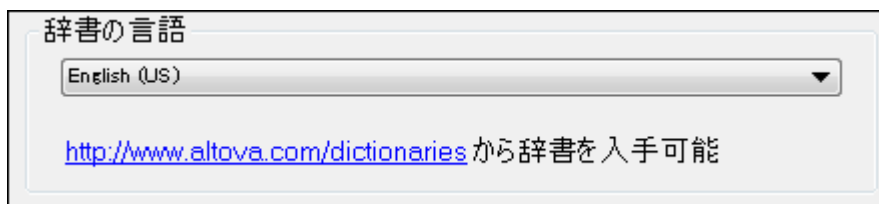
```
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (British)
\en_GB.aff
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (British)
\en_GB.dic
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (US)\en_US.dic
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (US)
\en_US.dic
```

スペルチェッカーオプションダイアログでは「辞書の言語」コンボボックスにあるドロップダウンリストに利用可能な辞書の言語が表示されます。Lexicons フォルダ以下にある言語サブフォルダの名前がリストには表示されます。例えば上にある英語辞書の場合、English (British) と English (US) という名前がリストに表示されます。

インストールされた辞書はコンピュータを使っているすべてのユーザー、そして (32 ビットか 64 ビットがかわらず) 複数の Altova 製品にも共有されます。

スペルチェッカーで使用される辞書は以下に示される 2 種類の方法で追加することができ、どちらの方法でもファイルをシステムに登録する必要はありません

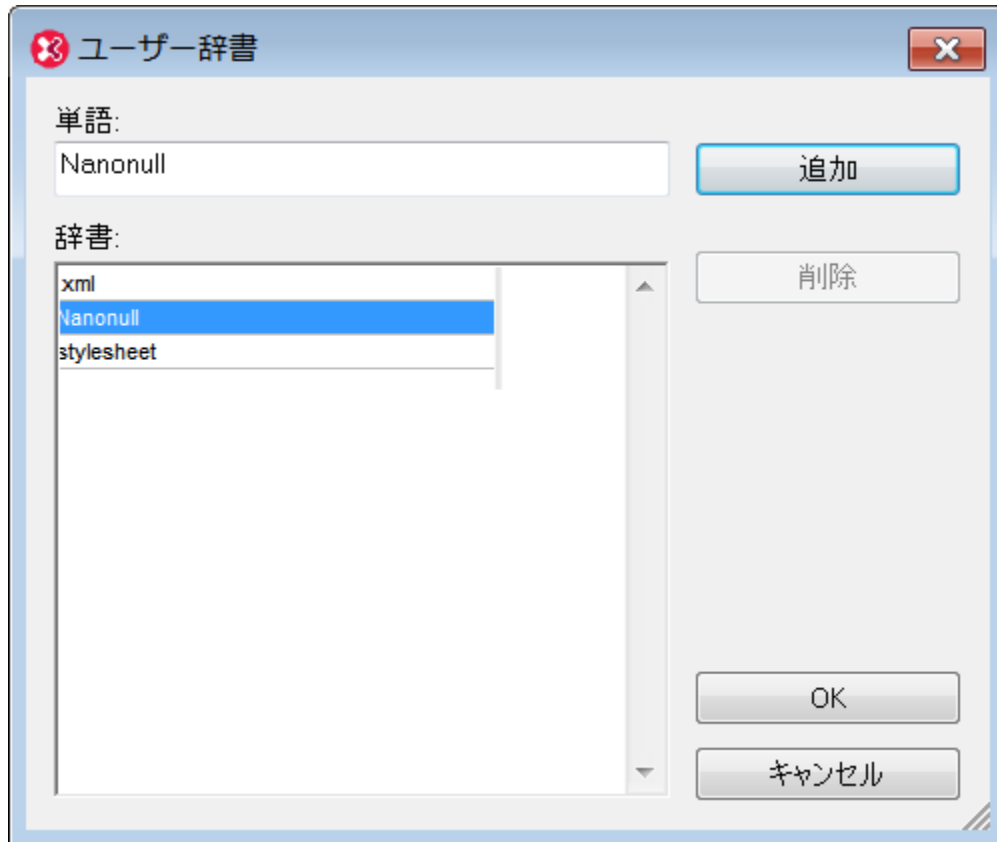
- Hunspell 辞書を Lexicons フォルダの新規サブフォルダへ追加する。Hunspell 辞書は、例えば <http://wiki.services.openoffice.org/wiki/Dictionaries> や <http://extensions.services.openoffice.org/en/dictionaries> という場所からダウンロードすることができます (OpenOffice では OXT という拡張子の ZIP アーカイブが配布されているため、ファイル拡張子を .zip へ変換して、回答した .aff と .dic ファイルをファイルとして Lexicons フォルダ以下にある言語フォルダへコピーしてください)。Hunspell 辞書は Myspell 辞書をベースとしているため、Myspell 辞書を使用することもできます)
- 複数の言語辞書をコンピュータの適切な場所へインストールする **Altova 辞書インストーラ** を使用する。スペルチェックのオプションダイアログにあるリンクをクリックすることで、インストーラをダウンロードすることができます (以下のスクリーンショットを参照)。



メモ 辞書に対して適用されるライセンスの条項に同意するか、そしてお使いのコンピュータにおける辞書の使用が適切なものかの判断は、お客様へ委ねられます。

ユーザー辞書と作業

コンピュータ上の各ユーザーには、ユーザーにより許可された単語を収めるためのユーザー辞書が与えられます。スペルチェックを行うと、言語辞書とユーザー辞書に含まれている単語のリストに対してドキュメント内の単語がチェックされます。ユーザー辞書ダイアログでは、ユーザー辞書に単語を追加、または削除することができます (以下のスクリーンショットを参照)。スペルチェックのオプションダイアログにて「ユーザー辞書」ボタンをクリックすることで、この辞書へアクセスすることができます (このセクションの 2 番目のスクリーンショットを参照してください)。



ユーザー辞書へ単語を追加するには、単語フィールドに単語を入力し、「追加」ボタンをクリックしてください。入力された単語がアルファベット順で辞書ページに追加されます。辞書から単語を削除するには、目的の単語を辞書ページにて選択し、「削除」ボタンをクリックしてください。その単語が辞書ページから削除されます。ユーザー辞書ダイアログにおける編集を終えた後は「OK」をクリックすることで、変更点がユーザー辞書へ保存されます。

スペルチェックの際に単語をユーザー辞書へ追加することもできます。[スペルチェックダイアログ](#)は取るアクションのためのプロンプトをポップアップします。「辞書に追加」ボタンをクリックすると、未知の単語がユーザー辞書へ追加されます。

ユーザー辞書は以下の場所にあります: C:\Users\\Documents\Altova\SpellChecker\Lexicons\user.dic

10.9.3 スクリプトエディター

スクリプトエディターコマンドにより、スクリプトエディターウィンドウが表示されます。スクリプトエディターの使用方法については、ドキュメンテーションの[スクリプト環境](#)セクションを参照ください。

メモ: スクリプトエディターを動作させるには、.NET Framework のバージョン 2.0 以降を、お使いのコンピューターにインストールする必要があります。

10.9.4 マクロ

マクロコマンドにマウスオーバーすることで、Authentic Desktop にて現在アクティブなスクリプトプロジェクトにて定義されているマクロがサブメニューに表示されます(以下のスクリーンショットを参照)。アクティブなスクリプトプロジェクトは、[オプションダイアログのスクリプトのセクション](#)内で指定されています。

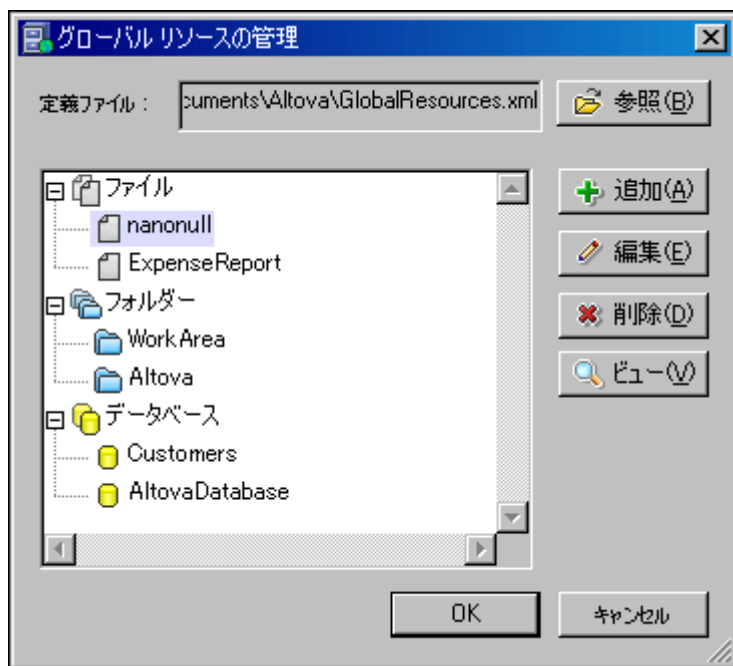


サブメニューにあるマクロをクリックすることで、マクロが開始されます(上のスクリーンショットを参照)。

10.9.5 グローバルリソース

グローバルリソースコマンドにより、以下の操作を行うことができるグローバルリソースダイアログが表示されます(以下のスクリーンショットを参照)：

- グローバルリソースXML ファイルを指定。
- ファイル、フォルダー、データベースのグローバルリソース(またはエイリアス)を追加。
- 各グローバルリソース(エイリアス)に対して、各種構成を指定。各構成により、特定のリソースへのマッピングが行われます。

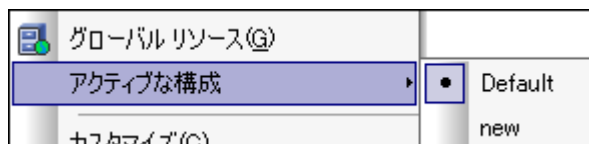


グローバルリソースの定義方法に関する詳細は、[グローバルリソースの定義セクション](#)を参照ください。

メモ Altova グローバルリソースダイアログへは、[グローバルリソースツールバー](#)（「ツール | カスタマイズ | ツールバー | グローバルリソース」）からアクセスすることができます。

10.9.6 アクティブな構成

アクティブな構成メニューアイテムにマウスオーバーすることで、現在アクティブな[グローバルリソースXMLファイル](#)にて定義されている全ての構成が表示されます（以下のスクリーンショットを参照）。



現在アクティブな構成は、中黒(•)により表示されます。上のスクリーンショットでは、現在アクティブな構成が Default となっています。アクティブな構成を変更するには、アクティブにしたい構成を選択してください。

メモ: アクティブな構成は、[グローバルリソースツールバー](#)（「ツール | カスタマイズ | ツールバー | グローバルリソース」）から選択することができます。

10.9.7 カスタマイズ

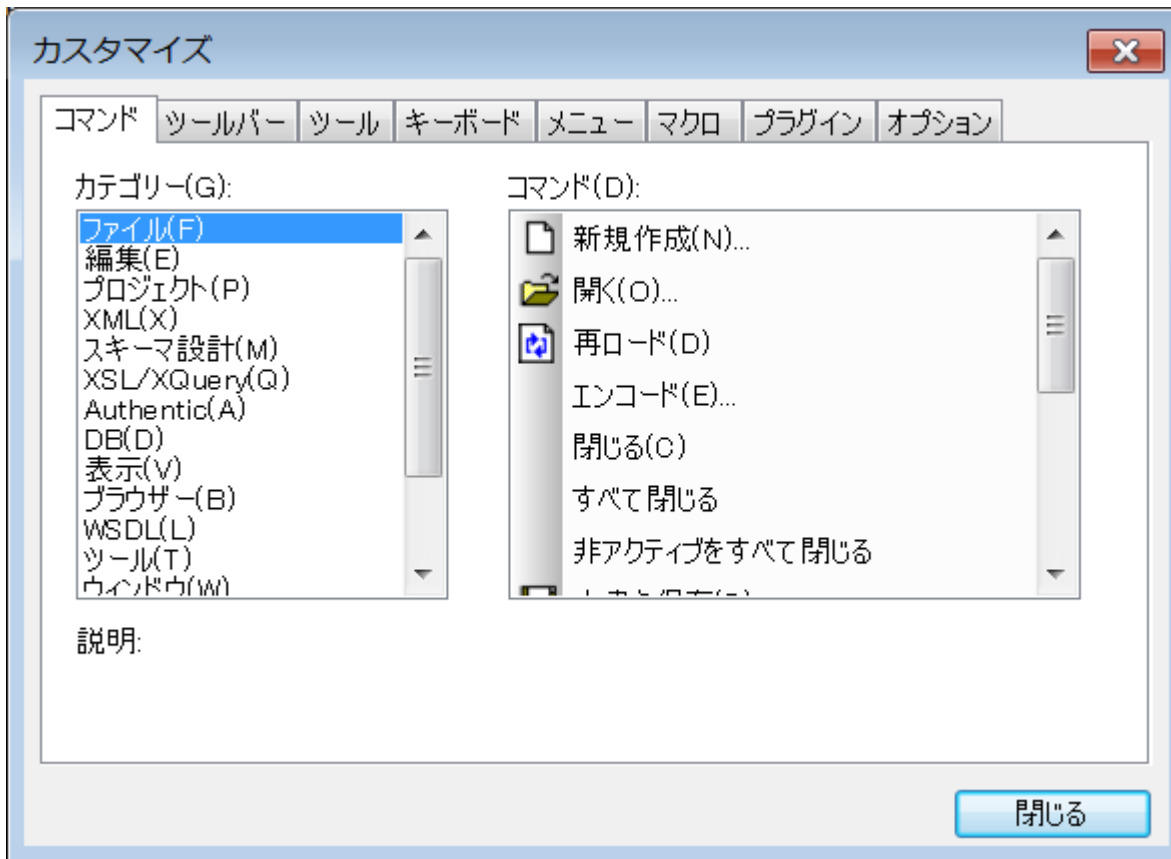
カスタマイズコマンドにより XMLSpy に搭載されたメニューやツールバーをカスタマイズすることができます。以下のタブが用意されています:

- [コマンド](#): 全てのアプリケーションやマクロコマンドをメニューバー、メニュー、ツールバーへドラッグします。
- [ツールバー](#): 個々のツールバーを有効化、無効化、リセットすることができます。
- [ツール](#): 外部プログラムを開くコマンドをインターフェースへ追加することができます。
- [キーボード](#): 個々のアプリケーションならびにマクロに対してキーボードのショートカットを作成することができます。
- [メニュー](#): カスタマイズするメニューバーならびにコンテキストメニューを選択し、有効にすることができます。コマンドタブと一緒に操作します。
- [マクロ](#): コマンドに対してマクロを関連付けることができます。
- [プラグイン](#): プラグインの有効化と統合を行うことができます。
- [オプション](#): ツールバーの表示オプションをセトすることができます。

このセクションではカスタマイズダイアログが開かれ、メニューバー、メニュー、またはツールバーアイテムが右クリックされた時に表示される [コンテキストメニュー](#) についても記述されます。

10.9.7.1 コマンド

コマンドタブでは、メニューやツールバーをカスタマイズすることができ、必要に応じてアプリケーションコマンドをメニューやツールバーに追加することができます。新たなアプリケーションコマンドやメニューを独自に作成することはできない点に注意してください。



以下の操作によりコマンドをツールバーやメニューへ追加することができます:

1. メニューオプションから「ツール | カスタマイズ」を選択します。カスタマイズダイアログが表示されます。
2. カテゴリリストボックスから全てのコマンドカテゴリを選択します。利用可能なコマンドがコマンドリストボックスに表示されます。
3. コマンドリストボックスに表示されたコマンドをクリックして、表示されているメニューやツールバーへドラッグします。許可されている場所へカーソルを移動させると、マウスポインタにI トラアイコンが表示されます。
4. コマンドを挿入する場所でマウスボタンを離してください。

以下の点に留意してください:

- コマンドをドラッグすると、マウスポインタの端の小さなボタが表示されます。これにより、ボタンがドラッグされていることが示されます。
- ポインタ下部に表示された“x”は、現在マウスポインタがある位置にドロップできないことを表しています。
- コマンドをドロップすることができる位置(ツールバーまたはメニュー)にカーソルを移動すると、“x”が消えてI トラアイコンが表示されます。
- コマンドはメニューまたはツールバーに配置することができます。**独自のツールバーを作成した場合**は場合、このカスタマイズ機能を使用することでコマンドを配置することができます。
- カーソルをメニュー近くに移動させると、そのメニューが開かれ、メニュー内部にコマンドを配置することができるようになります。

コンテキストメニューへコマンドを追加

以下の操作により、コンテキストメニューにコマンドを追加することもできます:

1. カスタマイズダイアログにて [メニュータブ](#) をクリックします。

2. コンテキストメニューペインにてコンボボックスから目的のコンテキストメニューを選択します。選択されたコンテキストメニューが表示されます。
3. カスタマイズダイアログにてコマンドタブへ切り替えます。
4. コマンドリストボックスから追加するコマンドを選択し、コンテキストメニューまでドラッグします。

コマンドまたはメニューの削除

以下の操作により、メニューやコンテキストメニュー、またはツールバーからコマンドを削除する、またはメニュー全体を削除することができます：

1. メニューオプションの「ツール | カスタマイズ」を選択してカスタマイズダイアログを開きます。
2. カスタマイズダイアログが表示されている状態で、メニューまたはメニューコマンドを右クリックして、表示されるコンテキストメニューから削除を選択します。また、メニューやメニューコマンドを、“x” がマウスポインタに表示される場所までドラッグアンドドロップすることでもメニューまたはメニューコマンドを削除することができます。

このセクションに記載された方法により削除されたメニューコマンドは、復元することができます。削除されたメニューを復元するには、メニューオプションから「ツール | カスタマイズ | メニュー」を選択し、アプリケーションフレームメニューペインにあるリセットボタンをクリックしてください。その他にも、「ツール | カスタマイズ | ツールバー」を選択し、左側に表示されたリストからツールバーを選択し、リセットボタンをクリックすることもできます。

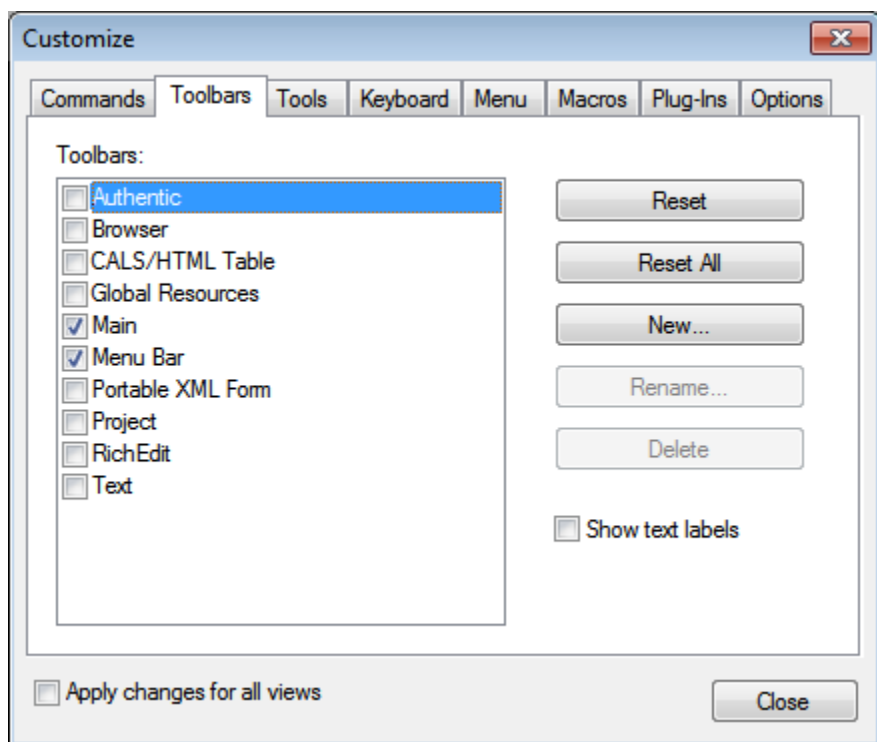
10.9.7.2 ツールバー

ツールバータブでは (i) 特定のツールバーを有効/無効にする(つまりどのツールバーがインターフェイスにて表示されるかを指定する)、(ii) 各ツールバーに表示されるアイコンをセトする、そして (iii) 独自のツールバーを作成することができます。

ツールバーには頻繁に使用されるメニューコマンドが収められています。カーソルをアイコン上部に移動することで、そのアイコンに関する情報がツールチップとステータスバー上に表示されます。ツールバーは、フロート状態のウィンドウとして画面上の任意の場所へ配置することができます。

メモ: ツールバーへコマンドを追加するには、目的のコマンドを [コマンド](#) タブにあるコマンドリストボックスからツールバーへドラッグする必要があります。ツールバーからコマンドを削除するには、カスタマイズダイアログが開かれた状態で、ツールバーからコマンドをドラッグしてください(詳細については [コマンド](#) のセクションを参照ください)。

メモ: 特定のビューに対して定義されたツールバーの設定は、デフォルトでそのビューに対してだけ有効になっています。その設定を全てのビューに対して適用するには、ダイアログ下部にある全てのビューに変更を適用 チェックボックスにチェックを入れてください。

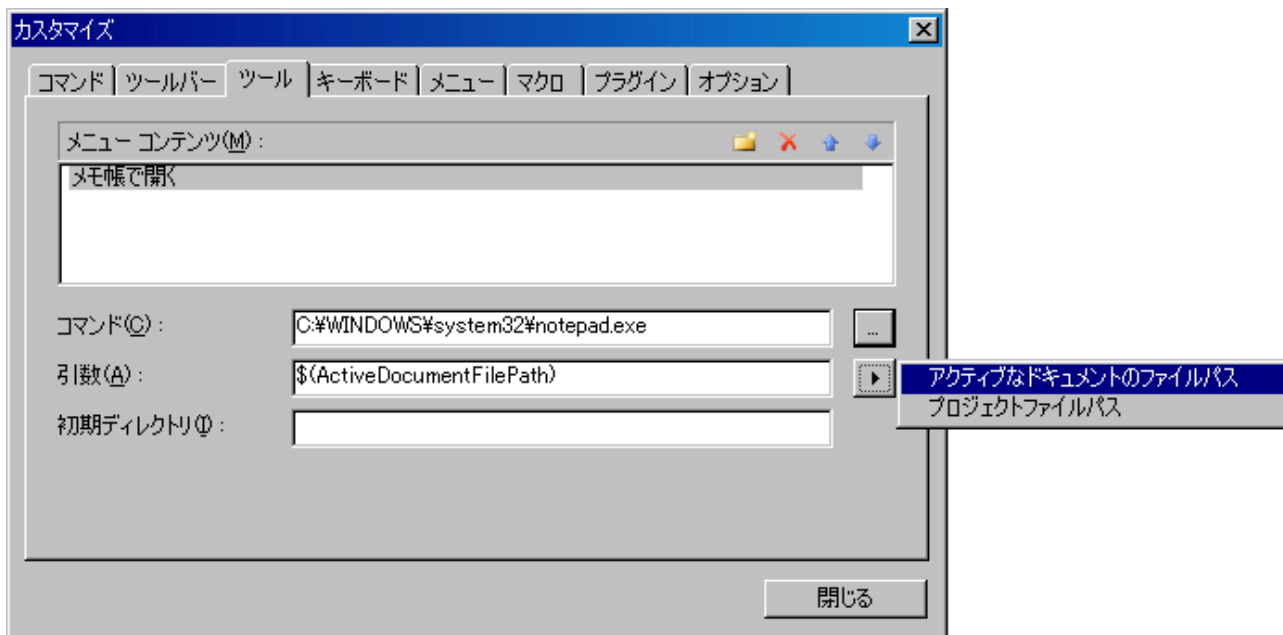


以下の機能を利用することができます:

- ツールバーの有効化/無効化: ツールバーリストボックス内に表示されたチェックボックスをクリックしてください。
- 全てのビューに変更を適用: ダイアログ下部に表示されているチェックボックスにチェックを入れてください。チェックが入っていない場合、アクティブなビューに対してのみ変更が適用されます。全てのビューに変更を適用チェックボックスをクリックした後の変更は、か全てのビューに対して適用される点に注意してください。
- ツールバーを新たに追加: 新規ボタンをクリックして、表示されるツールバーの名前ダイアログにてツールバーの名前を入力します。作成されたツールバーは **コマンド** タブからコマンドをドラッグすることができます。
- 追加されたツールバーの名前を変更: 追加されたツールバーをツールバーペインにて選択し、名前の変更ボタンをクリックします。表示されるツールバーの名前ダイアログにて名前の編集を行います。
- メニューバーのリセット: ツールバーペインにてメニューバーアイテムを選択し、リセットをクリックします。この操作によりアプリケーションがインストールされた初期の状態にメニューバーがリセットされます。
- ツールバーとメニューコマンドのリセット: 全てのリセットボタンをクリックすることで、アプリケーションがインストールされた初期の状態に全てのツールバーならびにメニューがリセットされます。
- ツールバーの削除: ツールバーペインにて削除するツールバーを選択し、削除ボタンをクリックします。
- 特定のツールバーにコマンドのテキストラベルを表示: 目的のツールバーを選択し、テキストラベルを表示チェックボックスにチェックを入れます。テキストラベルは各ビューに対して個別に有効化する必要がある点に注意してください。

10.9.7.3 ツール

ツールタブでは Authentic Desktop 内部から外部アプリケーションを使用するコマンドをセットアップすることができます。これらのコマンドは「ツール | ユーザー定義ツール」メニュー以下に表示されます。例えば、作成されたツールメニューのコマンドを「ツール | ユーザー定義ツール」以下からクリックすることで、Authentic Desktop のメインウィンドウに表示されているアクティブなファイルを、メモ帳のような外部アプリケーションにて開くことができます。



以下の操作により外部アプリケーションを使用するコマンドをセットアップすることができます：

1. メニューコンテンツペインにて、ペインのタイトルバーにある新規作成アイコンをクリックし、作成されたアイテムにメニューコマンドの名前を入力します。上のスクリーンショットでは、メモ帳で開くというメニューコマンドが作成されており、アクティブなドキュメントを外部アプリケーションの(Windows に搭載されている)メモ帳にて開くようなコマンドを作成します。新規作成アイコンをクリックすることで、更にコマンドを追加することもできます。アイテムを上へならびアイコンを下へアイコンを使用することで、コマンドの相対的な位置を変更することができます。コマンドを削除するには、削除アイコンをクリックしてください。
2. 外部アプリケーションとコマンドを関連付けるには、メニューコンテンツペインにあるコマンドを選択し、コマンドフィールドにて外部アプリケーションの実行可能ファイルへのパスを入力するか、(入力フィールドの隣にあるボタンから)参照してください。上のスクリーンショットでは、メモ帳へのパスがコマンドフィールドへ入力されています。
3. 引数フィールドにあるボタンをクリックすることで、外部アプリケーションに対して利用可能なアクションが表示されます(上のスクリーンショットを参照)。これらのアクションについては以下にあるリストを参照ください。アクションを選択すると、そのアクションに対応したコード文字列が引数フィールドに入力されます。
4. 処理を行う際の初期ディレクトリを指定する場合、その値を初期ディレクトリフィールドに入力してください。
5. 閉じるボタンをクリックすることで処理を完了します。

作成されたコマンドは「ツール | ユーザー定義ツール」メニューと、プロジェクトウィンドウにてファイルを右クリックすることで表示されるコンテキストメニューの「ユーザー定義ツール」サブメニュー以下に表示されます。

新規作成されたコマンドをツールメニューにてクリックすると、コマンドに関連付けられたアクションが実行されます。上のスクリーンショットに示されたコマンドの場合、Authentic Desktop のメインウィンドウでアクティブなドキュメントがメモ帳で開かれます。外部アプリケーションコマンドでは、プロジェクトウィンドウのユーザーツールコンテキストメニューからもアクセスすることができます(プロジェクトウィンドウにあるファイルを右クリックすることでコンテキストメニューが表示されます)。プロジェクトウィンドウでは、複数のファイルを選択し、コンテキストメニューからコマンドを選択することで、複数のファイルに対して関連付けられたアクションを実行することができます。

引数

引数フィールドでは外部アプリケーションコマンドにより実行されるアクションが指定されます。以下の引数を利用することができます:

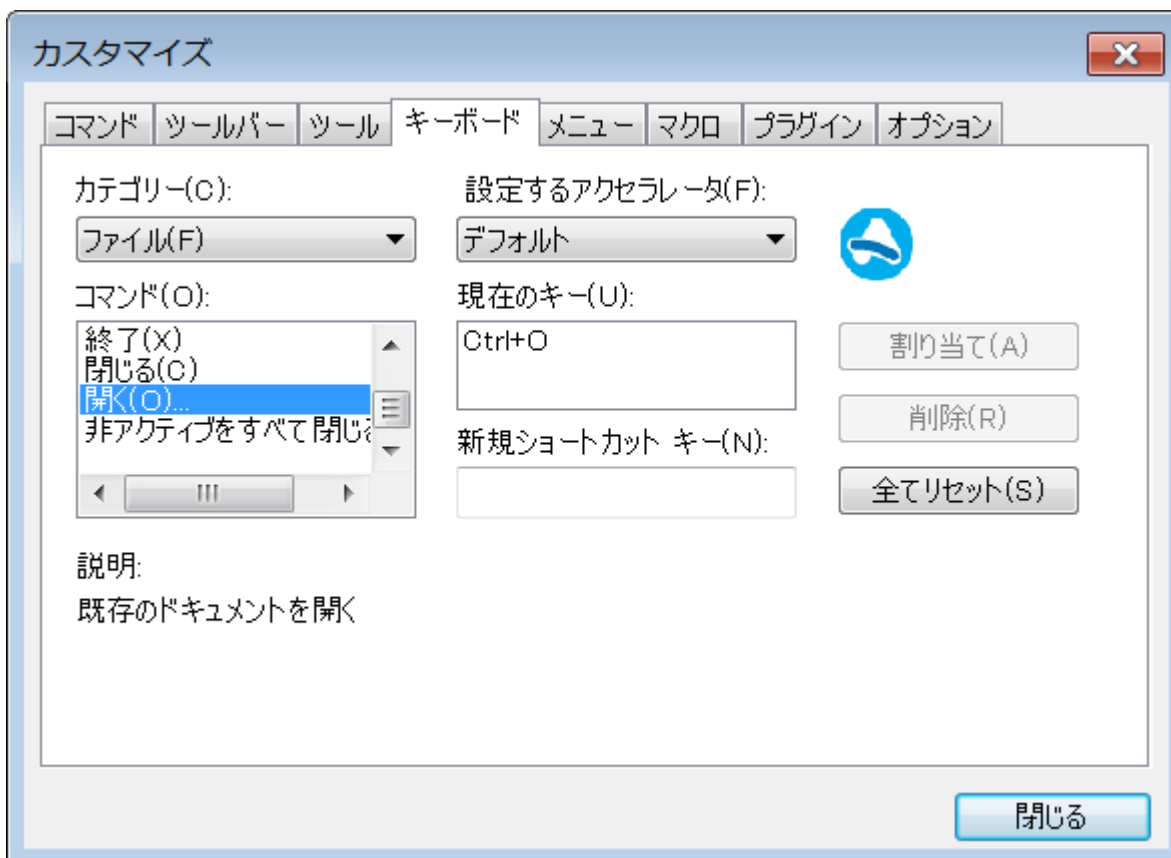
- アクティブなドキュメントのファイル名: 「ユーザー定義ツール」メニューにあるコマンドから、Authentic Desktop にてアクティブになっているドキュメントを外部アプリケーションで開きます。プロジェクトウィンドウのコンテキストメニューからコマンドへアクセスした場合、選択されたファイルが外部アプリケーションで開かれます。
- プロジェクトファイル名: Authentic Desktop のプロジェクトファイル(.spp ファイル)を外部アプリケーションにて開きます。

初期ディレクトリ

オプションとして、外部アプリケーションを実行する際にカレントディレクトリとして使用されるディレクトリへのパスを、初期ディレクトリフィールドへ入力することができます。

10.9.7.4 キーボード

キーボードタブでは、キーボードショートカットを新たに作成したり、既存のショートカットを変更することができます。



以下の操作により、新たなショートカットをコマンドへ割り当てる、または既存のショートカットを変更することができます:

1. カテゴリーエボボックスから全てのコマンドを選択します。[関連付けられたコマンドとしてマクロが選択された場合](#)、カテゴリーエボボックスにマクロをセットすることもできます。

2. コマンドリストボックスにて、ショートカットを割り当てる、またはショートカットを変更するコマンドを選択します。
3. 新規ショートカットキーテキストボックスをクリックし、そのコマンドに対して割り当てるショートカットキーを押下します。入力されたショートカットが新規ショートカットキーテキストボックスに表示されます。そのショートカットが他のコマンドにも割り当てられていない場合、割り当てボタンが選択可能状態になります。ショートカットが他のコマンドに既に割り当てられている場合、そのコマンド名がテキストボックス以下に表示され、割り当てボタンが選択不可能状態になります。新規ショートカットキーテキストボックスに入力された内容をクリアするには、Ctrl、Alt、またはShift のいずれかのキーを押下してください。
4. 割り当てボタンをクリックすることで、ショートカットの割り当てを行うことができます。ショートカットが現在のキーリストボックスに表示されます。複数のショートカットを同じコマンドへ割り当てすることもできます。
5. 閉じるボタンをクリックすることで、変更を確定します。

ショートカットの削除

同じショートカットを複数のコマンドに対して割り当てることはできません。ショートカットを削除するには、現在のキーリストボックスにあるショートカットを選択し、削除ボタンをクリックし、閉じるボタンをクリックすることで削除を確定してください。

設定するアクセラレータ

現在割り当てられている機能はありません。

デフォルトのキーボードショートカット

インストール時に割り当てられているコマンドのリストを(機能ごと)以下に示します: アプリケーションではキーボードマップダイアログ([ヘルプ | キーボードマップ](#))では、ショートカットとその説明とともに、コマンドのリストを確認することができます。

関数キーのショートカット (検証と変換のためのショートカットも含む)機能キーのショートカット

F1	ヘルプメニュー
F1 + Alt	最後に開かれたファイルを開く
F3	次を検索
F4 + CTRL	アクティブなウィンドウを閉じる
F4 + Alt	Authentic Desktopを閉じる
F5	更新
F6 + CTRL	開かれているウィンドウを循環する
F7	整形形式チェック
F8	検証
F10	XSL 変換
F10 + CTRL	XSL:FO 変換

ファイルとアプリケーションコマンド

Alt + F1	最後に開かれたファイルを開く
CTRL + O	ファイルを開く
CTRL + N	新規ファイル
CTRL + P	ファイルの印刷
CTRL + S	ファイルの保存
CTRL + F4	アクティブなウィンドウを閉じる
CTRL + F6	開かれているウィンドウを循環する
CTRL + TAB	開かれているドキュメントを切り替える

Alt + F4	Authentic Desktopを閉じる
----------	-----------------------

☐ その他のキー

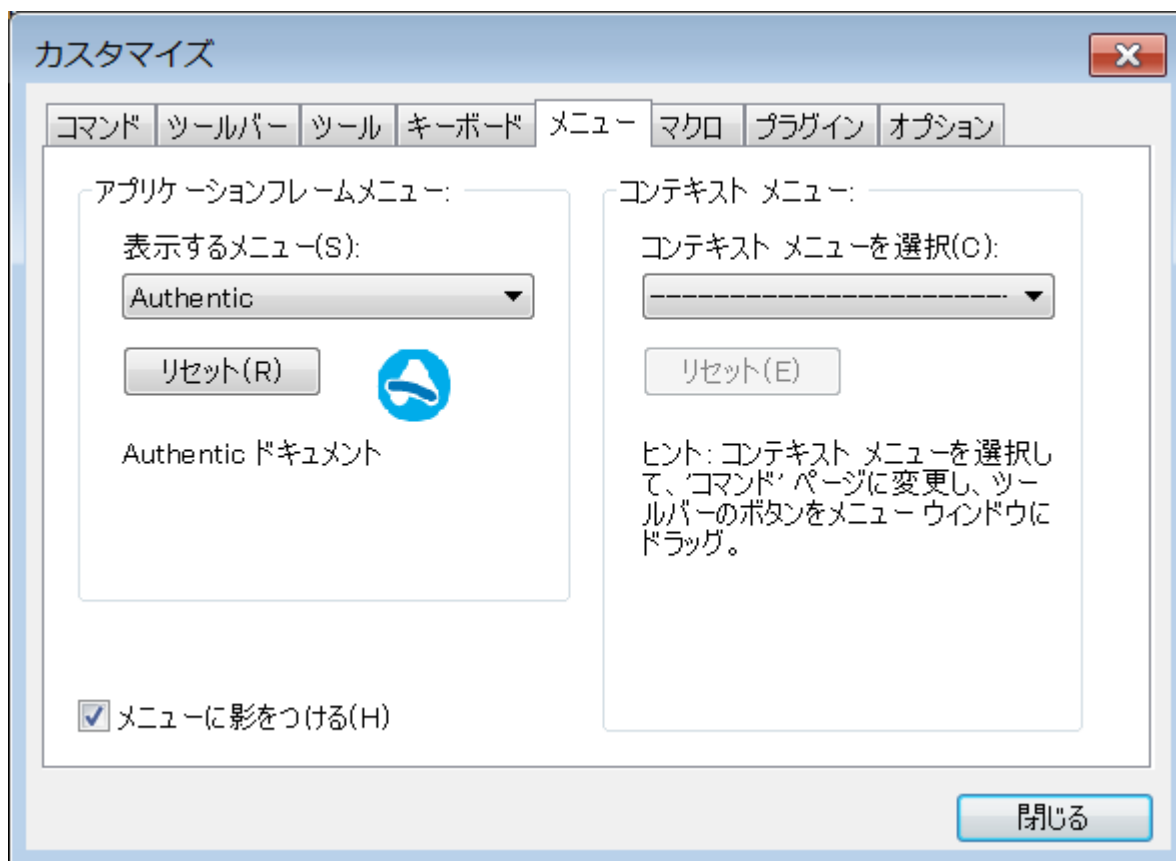
上/下矢印キー	カーソルまたは選択バーを移動する
Esc	編集を破棄する、または、ダイアログボックスを閉じる
リターン	選択を確認する
Del	文字、または、選択範囲を削除する
Shift + Del	切り取り

☐ 編集コマンド

CTRL + A	すべて選択
CTRL + F	検索
CTRL + G	ライン/文字に移動する
CTRL + H	置き換える
CTRL + V	貼り付け
CTRL + X	切り取り
CTRL + Y	やり直し
CTRL + Z	元に戻す

10.9.7.5 メニュー

メニュータブでは、デフォルトならびにアプリケーションメニューバーとら2つのメインメニューバー、そしてアプリケーションのエレキストメニューをカスタマイズすることができます。



デフォルトメニューバーとアプリケーションメニューバーのカスタマイズ

デフォルトのメニューバーとは、メインウィンドウにてドキュメントが開かれていない状態で表示されるメニューバーのことです。アプリケーションメニューバーとは、1つ以上のドキュメントがメインウィンドウにて開かれている状態で表示されるメニューバーのことです。各メニューバーは、お互いに干渉しないため個別にカスタマイズすることができます。

メニューバーをカスタマイズするには、「表示するメニュー」コンボボックスから目的のメニューを選択し、[カスタマイズダイアログのコマンドタブ](#)へ切り替えます。その後コマンドリストボックスに表示されたコマンドを、メニューバーまたはメニューへドラッグします。

コマンドの削除とメニューバーのリセット

メニュー内のメニュー全体またはコマンドを削除するには以下を行います:

1. アプリケーションフレームメニューペイン内で デフォルト (ドキュメントが開かれていない場合) または1つ以上のドキュメントが開かれている場合使用可能なメニューを表示する) *Authentic* を選択します。
2. 開かれているカスタマイズダイアログから (i) アプリケーションのメニューバーから削除するコマンド (ii) これらのメニューから削除するコマンドを選択します。
3. (i) メニューバーからメニューをドラッグ、または、メニューからメニューコマンドをドラッグ、または (ii) メニューまたはメニューコマンドを右クリックして [削除] を選択します。

「表示するメニュー」コンボボックスにてメニューを選択し、「リセット」ボタンをクリックすることで、2つのメニューバー (デフォルトならびにアプリケーションメニューバー) のそれぞれをリセットすることができます。

アプリケーションのコンテキストメニューをカスタマイズする

コンテキストメニューとは、アプリケーションインターフェイスに表示されたオブジェクトを右クリックすることで表示されるメニューのことです。以下の方法により、コンテキストメニューのカスタマイズを行うことができます。

1. コンテキストメニューを選択コンボボックスにて目的のコンテキストメニューを選択します。コンテキストメニューが表示されます。
2. [カスタマイズダイアログのコマンドタブ](#)へ切り替えます。
3. コマンドリストボックスからコンテキストメニューへコマンドをドラッグします。
4. コンテキストメニューからコマンドを削除するには、コンテキストメニューに表示されたコマンドを右クリックして、「削除」を選択します。コンテキストメニューからコマンドをドラッグすることでも、そのコマンドを削除することができます。

コンテキストメニューを選択コンボボックスにてメニューを選択し、「リセット」ボタンをクリックすることで、コンテキストメニューをオリジナルの状態にリセットすることができます。

メニューに影を付ける

「メニューに影を付ける」チェックボックスを有効にすることで、全てのメニューに影が与えられます。

10.9.7.6 マクロ

マクロタブでは、Authentic Desktop のスクリプトエディターにより作成されたマクロに対するアプリケーションコマンドを作成することができます。関連付けられたマクロを実行するこれらのアプリケーションコマンドは、マクロタブから直接、または[カスタマイズダイアログのコマンドタブ](#)を使用することにより、メニューやツールバーにて利用できるようになります。アプリケーションコマンドとして、[カスタマイズダイアログのキーボードタブ](#)にてショートカットを割り当てることもできます。

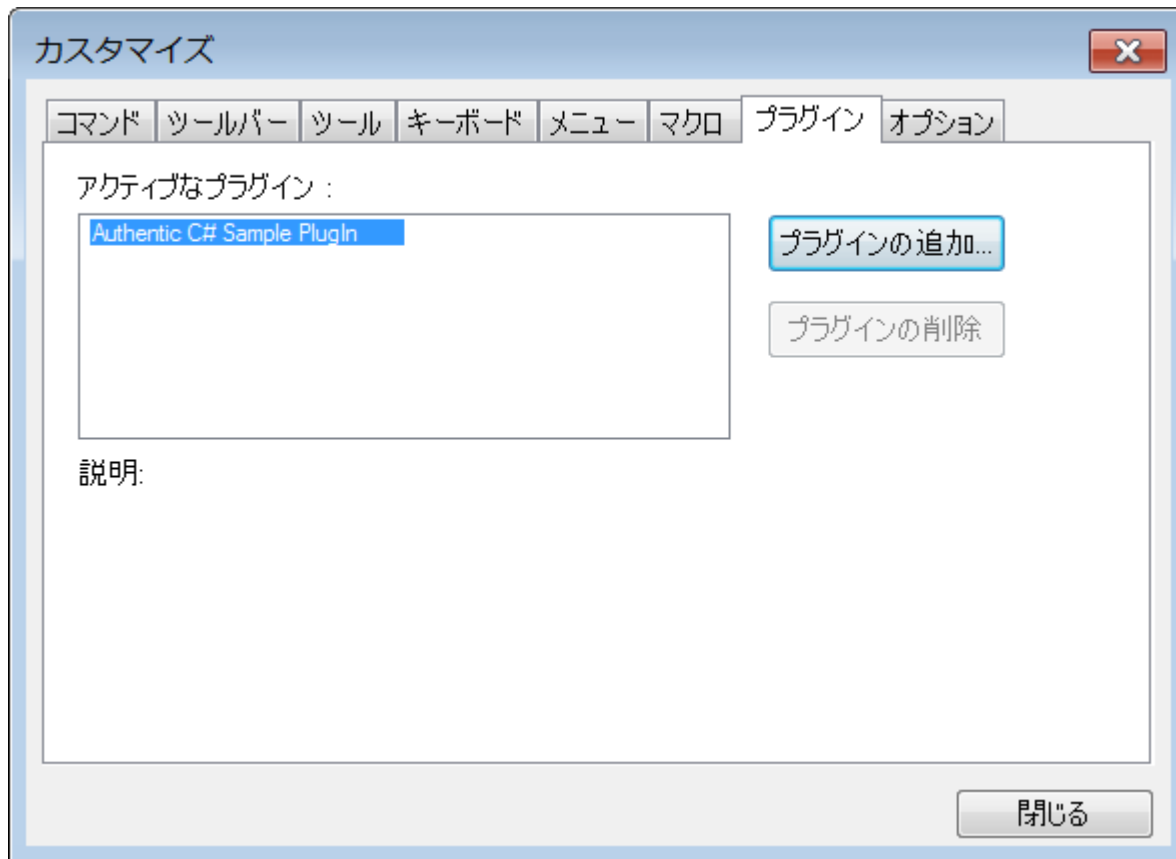
Authentic Desktop におけるマクロの作動

Authentic Desktop におけるマクロは以下のように動作します：

- Authentic Desktop の[スクリプトエディター](#)により、Altova スクリプトプロジェクト (.asprj ファイル) が作成されます。スクリプトプロジェクトには、Authentic Desktop で使用することができるマクロを含めることができます。
- Authentic Desktop では (i) [オプションダイアログのスクリプトタブ](#)にて指定されるアプリケーションプロジェクトと (ii) [スクリプト設定ダイアログ \(プロジェクト | スクリプト設定\)](#)内で指定されるアクティブな

10.9.7.7 プラグイン

「プラグイン」タブでは、アプリケーションやツールバー動作するようプログラムされたプラグインの統合とコマンドの配置を行うことができます。(下のスクリーンショット) プラグインタブでは、「プラグインを追加」をクリックすることで、プラグインの DLL ファイルを参照することができます(下の「プラグインの作成」を参照してください)。「OK」をクリックすることでプラグインが追加されます。アプリケーションには複数のプラグインを追加することができます。



プラグインの追加に成功すると、プラグインの説明がダイアログに表示され、プラグインを削除ボタンが選択可能になります。プラグインコードによりツールバーやメニューが作成されると、それらがアプリケーションインターフェースに表示されます。プラグインを削除するには、プラグインを削除ボタンをクリックしてください。

プラグインの作成

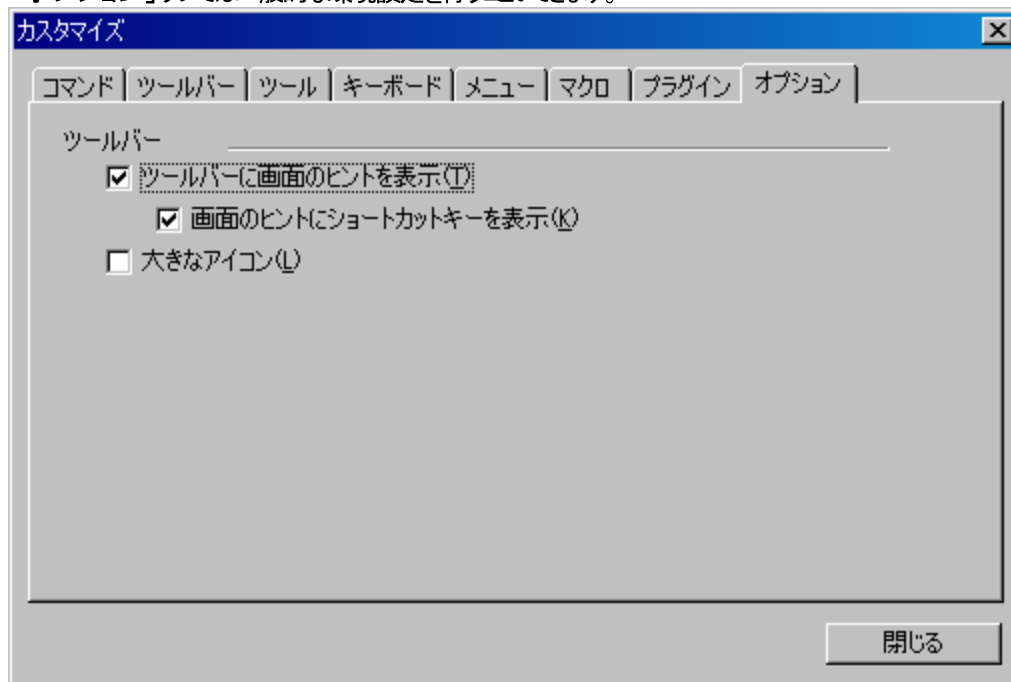
サンプルプラグインのソースコードはアプリケーションの [マイドキュメント フォルダー](#): Examples\IDEPlugin フォルダー内に収められています。ソースコードからプラグインを作成するには以下を行います:

1. Visual Studio 内でプラグインとして作成するソリューションを開きます。
2. ビルドメニュー内のコマンドを使用してプラグインを作成します。
3. プラグインのDLLファイルは Bin または Debug フォルダー内で作成されます。このDLL ファイルは、プラグインとして追加される必要があるファイルです (上記参照)。

プラグインの作成に関する詳細については [IDE プラグイン](#) のセクションを参照ください。

10.9.7.8 オプション

「オプション」タブでは一般的な環境設定を行うことができます。

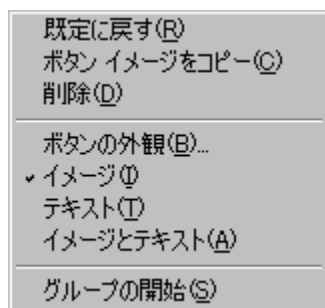


チェックボックスをクリックすることで以下のオプションを有効にすることができます:

- ツールバーに画面のヒントを表示: ツールバーのアイコン上にマウスポインターが配置されると、ポップアップが表示されます。ポップアップにはアイコン機能の説明と、(ショートカットキーを表示オプションが有効な場合) 関連付けられたショートカットが表示されます。
- 画面のヒントにショートカットキーを表示: 画面のヒントにショートカットキーに関する情報を表示するか指定することができます。
- 大きなアイコン: ツールバーアイコンの大きさを通常のサイズと大きめのサイズへ切り替えることができます。

10.9.7.9 コンテキストメニューのカスタマイズ

コンテキストメニューのカスタマイズコマンドは、カスタマイズダイアログが表示されている状態で、メニュー、メニューコマンド、またはツールバーアイコンを右クリックすることにより表示されます。

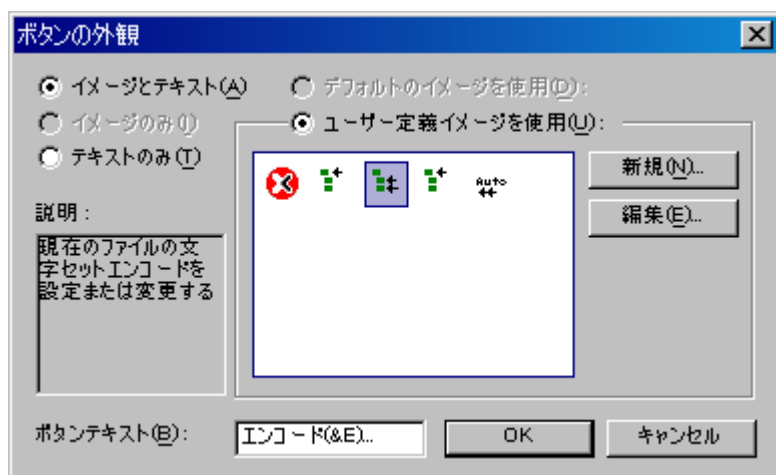


以下の機能を利用することができます:

- 規定に戻す: 現在機能は与えられていません。
- ボタンイメージをコピー: 右クリックしたアイコンをクリップボードにコピーします。
- 削除: 選択されたメニュー、メニューコマンド、またはツールバーアイコンを削除します。削除されたアイコンを再度表示する方法については、下記を参照ください。
- ボタンの外観: ボタンの外観ダイアログが表示され、選択されたツールバーアイコンの外観を定義するためのプロパティをセットすることができます。詳細については以下の記述を参照ください。
- イメージ、テキスト、イメージとテキスト: 相互に排他的なオプションで、選択されたツールバーアイコンをアイコンだけ、テキストだけ、またはアイコンとテキストの両方で表示するのを指定します。いずれかのオプションを選択することで、必要な変更が行われます。
- グループの開始: 選択されたツールバーアイコンの左側にグループを分離する垂直の線を挿入します。これにより選択されたツールバーアイコンがグループ最初のアイコンとなります。

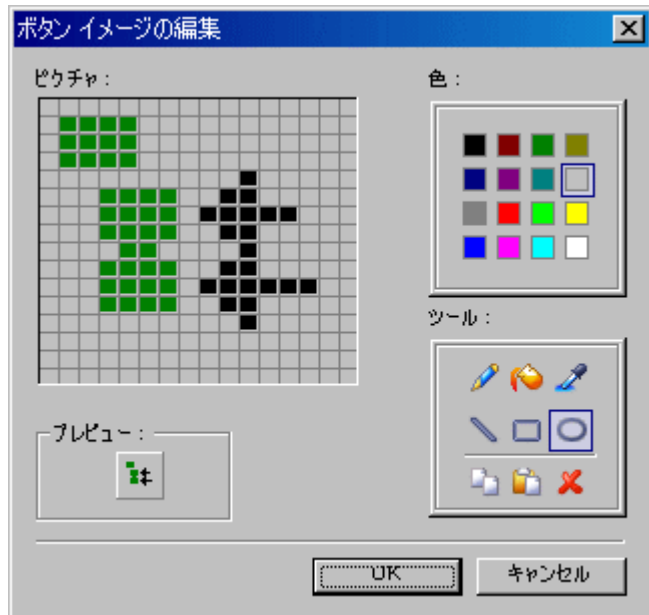
ボタンの外観ダイアログ

ツールバーアイコンを右クリックして、「ボタンの外観」を選択することで、ボタンの外観ダイアログが表示されます(以下のスクリーンショットを参照)。このダイアログでは、ツールバーアイコンのイメージやテキストを選択することができます。現在このダイアログでは、マクロならびにコマンドのツールバーアイコンが編集することができます。



選択されたツールバーのアイコン(コンテキストメニューのカスタマイズにて右クリックされたアイコン)に対して以下の編集機能を使用することができます。

- イメージのみ、テキストのみ、イメージとテキスト: 目的のラジオボタンを選択して、ツールバーのアイコンが表示される形式を指定することができます。
- イメージの編集: イメージのみまたはイメージとテキストが選択された場合、イメージの編集オプションが有効になります。「新規」ボタンをクリックすることでユーザー定義の新たなイメージが作成され、イメージペインに追加されます。イメージを選択して、「編集」ボタンにより編集を行うことができます。



- **イメージ選択:** イメージペインから目的のイメージを選択し、OK をクリックすることにより選択されたイメージをツールバーで使用することができます。
- **テキストの編集とセクター:** テキストのみまたはイメージとテキストが選択された場合、ボタンテキストテキストボックスが選択可能になります。テキストを入力または編集し、OK をクリックすることで、ツールバーアイコンのテキストを変更することができます。

メモ ボタンの外観ダイアログでは、メニューコマンドのテキストを編集することもできます。カスタマイズダイアログが開かれている状態でメニューコマンドを右クリックして、「ボタンの外観」を選択し、「ボタンテキスト」テキストボックスにてメニューコマンドのテキストを編集してください。

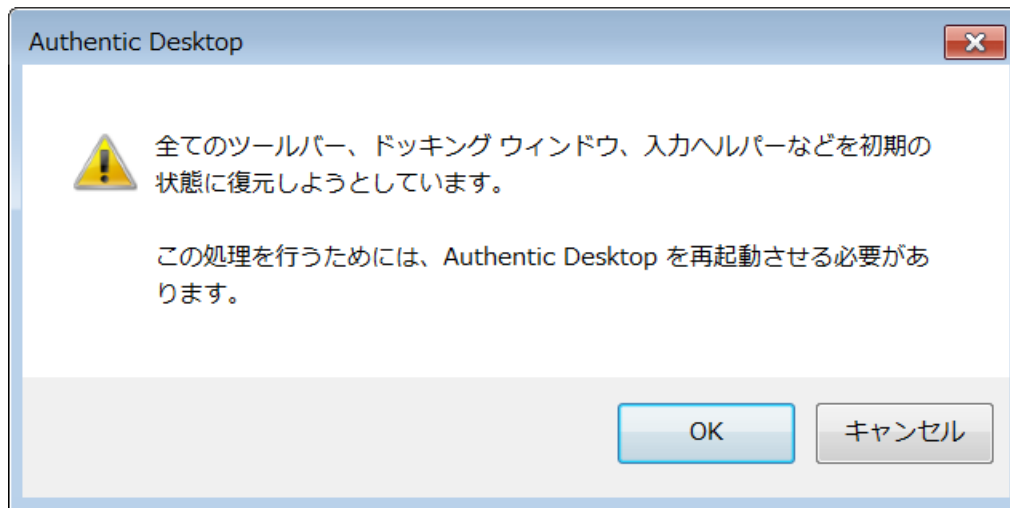
削除されたメニュー、メニューコマンド、ツールバーコマンドを復元

コンテキストメニューのカスタマイズにある削除コマンドにより、メニューやメニューコマンド、ツールバーアイコンが削除された場合、以下の方法により復元することができます:

- **メニュー:** メニューオプションから「[ツール](#) | [カスタマイズ](#) | [メニュー](#)」を選択し、アプリケーションフレームメニューペインにあるセットボタンをクリックしてください。他にも「[ツール](#) | [カスタマイズ](#) | [ツールバー](#)」を選択し、ダイアログ左側でメニューバーを選択した状態でセットボタンをクリックすることもできます。
- **メニューコマンド:** メニューオプションから「[ツール](#) | [カスタマイズ](#) | [コマンド](#)」を選択し、コマンドリストボックスから目的のコマンドをメニューバーへドラッグしてください。
- **ツールバーアイコン:** メニューオプションから「[ツール](#) | [カスタマイズ](#) | [コマンド](#)」を選択し、コマンドリストボックスから目的のコマンドをツールバーへドラッグしてください。

10.9.8 ツールバーとウィンドウの復元

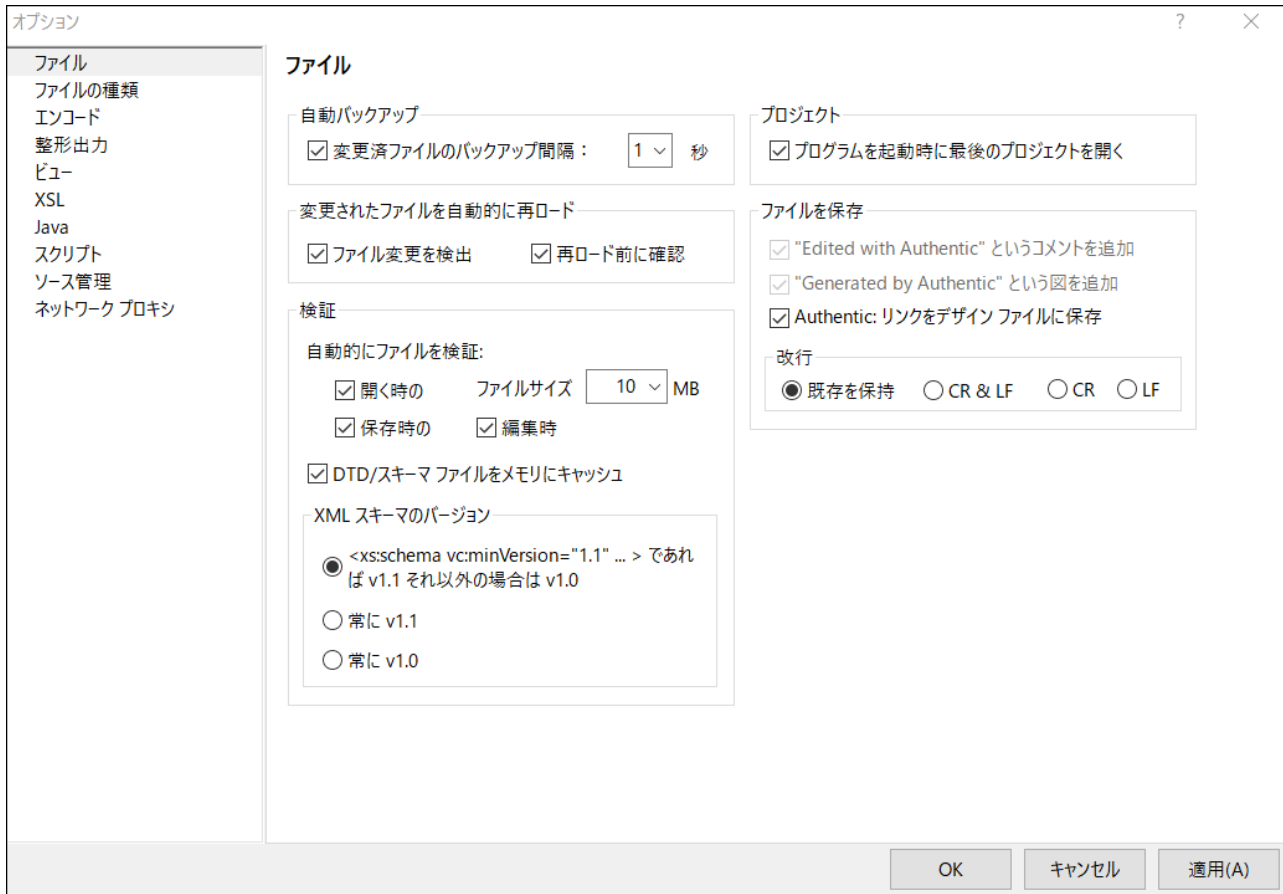
ツールバーとウィンドウの復元コマンドにより、Authentic Desktop をデフォルト設定の状態に戻すことができます。終了する前にAuthentic Desktop の終了を確認するダイアログが表示されます(以下のスクリーンショットを参照)。



このコマンドは、ツールバーやウィンドウのサイズ変更、移動、および非表示にするように指定した後、ツールバーやウィンドウをオリジナルの状態に戻したい場合に使用することができます。

10.9.9 オプション

[ツール | オプション] コマンドによりグローバルなアプリケーション設定を定義することができます。これらの設定は、セクションにより整理されています(下のスクリーンショット)。例えば、[ファイルセクション](#)(下のスクリーンショットを参照)には、Authentic Desktop によるファイルの開き方、及び、保存の方法を指定することができるオプションが含まれています。特定のセクションのオプションを指定するには、左側のペイン内のそのセクションを選択し、必要とするプロパティの値を指定します。「OK」ボタンにより、レジストリへの変更が保存され、ダイアログが閉じられます。「適用」ボタンにより、現在開かれているドキュメントへの変更を表示することができます。



オプションダイアログの各セクションの詳細については、このセクションのサブセクションを参照ください。

10.9.9.1 ファイル

ファイルセクションでは、Authentic Desktop にてファイルを開く、または保存する際の振る舞いを定義することができます。[エンコードセクション](#)にも関連する設定が存在します。

自動的なバックアップ

このオプションが有効化されていると現在編集中のファイルは自動的にバックアップされます。5 秒から 60 秒までの間のバックアップの間隔をコンボボックスから選択、または 300 秒までのカスタム値を入力することができます。詳細に関しては、[の自動的なバックアップ](#)のセクションを参照してください。

変更されたファイルの自動的な再ロード

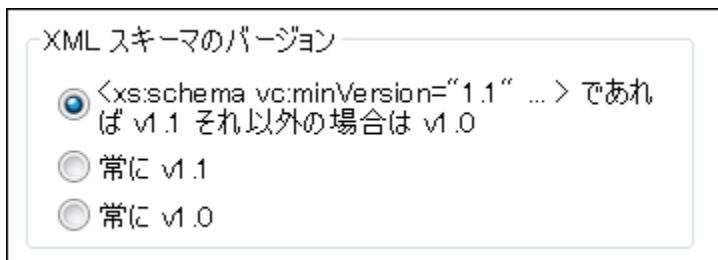
マルチユーザー環境内で作業する場合、または、サーバー上で動的に生成されるファイルと作業する場合、インターフェイス内で現在開かれているファイルへの変更を確認することができます。Authentic Desktop が開かれているドキュメント内の変更を検知する都度、変更されたファイルを再ロードするがプロンプトされます。

検証

DTD やスキーマによりXMLドキュメントの構造を定義している場合、ドキュメントを開くまたは保存する際に検証を行うかどうか指定することができます。自動的に行われる検証は、ドキュメントの上限を定めることで、効果的な編集作業を行うことができます。ドキュメントが妥当でない場合、エラーメッセージが表示されます。ドキュメントが妥当な場合は、メッセージが出力されることなく、処理が行われます。Authentic Desktop ではこれらのファイルをメモリ上にキャッシュすることで、(例えば、参照されているスキーマがURLによりアクセスされた場合など) 不必要な再ロードを抑えることができます。スキーマ位置の宣言にURLを使用している場合、「DTD/スキーマファイルをメモリにキャッシュ」オプションを無効にすることで、スキーマに対して行われ変更を、キャッシュを介さず直接保存することができます。

XML スキーマバージョン

スキーマビュー内で有効化されたXSD モードは以下により異なります (i) 属性の存在/不在—存在する場合、XSD ドキュメントの `/xs:schema/@vc:minVersion` 属性の値 (ii) オプションダイアログのファイルセグメン(「ツール|オプション」内で選択されるXML スキーマバージョンのオプション。以下のスクリーンショット)。



以下のシチュエーションが可能になります。下のテーブル内 XML スキーマバージョンは、上に表示されるXML スキーマバージョン ペイン内の選択を参照して見ます。テーブル内の `vc:minVersion` 値は、XML スキーマドキュメント内の `xs:schema/@vc:minVersion` 属性の値を参照して見ます。

XML スキーマバージョン	<code>vc:minVersion</code> 属性	XSD モード
常に v1.0	不在、または、存在していても値を持っていない	1.0
常に v1.1	不在、または、存在していても値を持っていない	1.1
@ <code>vc:minVersion</code> の値	属性は、1.1 の値を持っている	1.1
@ <code>vc:minVersion</code> の値	属性が不在、または、存在していても値が1.1以外	1.0

プロジェクト

Authentic Desktop を起動すると、最後に使用されたプロジェクトが自動的に開くように指定することができます。

ファイルを保存

XML ドキュメントを保存する際、Authentic Desktop により、`<!-- Edited with Authentic Desktop http://www.altova.com -->` コメントがドキュメントの上部に挿入されます。このオプションを無効にするにはユーザーのアクセス登録が必要で、拡張グリッドビューまたはスキーマデザインビューにて編集を行い、ファイルを保存した際に有効になります。

XML ファイルに StyleVision Power Stylesheet が関連付けられている場合、「Authentic : リンクをデザインファイルに保存」オプションを有効にすることで、XML とともに StyleVision Power Stylesheet を保存することができます。

改行

既存を保持が選択されていると、ファイルを開いたときにドキュメント内にあった改行コード文字がそのまま保持されます。または、3つのコード内のコードの変更を選択することができます: **CR&LF** (PC)、**CR** (MacOS)、または **LF** (Unix)。

保存して完了

設定が完了したら、「OK」をクリックして完了します。

10.9.9.2 ファイルタイプ

ファイルタイプセクションでは、各種ファイルに対する Authentic Desktop の振る舞いを指定することができます。ファイルタイプリストボックスからファイルの種類を選択し、その種類のファイルに対する動作をカスタマイズすることができます。

Windows エクスプローラー設定

Windows エクスプローラーで使用されるファイルタイプの説明と、MIMIE 準拠のコンテンツタイプを定義するほか、この種類のドキュメントを編集するデフォルトデータとして Authentic Desktop を指定することができます。

準拠

Authentic Desktop では、様々な種類のファイルに対して使用することのできる数多くの機能が搭載されています。特定のファイルタイプに対して使用されるオプションの種類は、この準拠オプションによりセットされます。ファイルの型の大半はファイル型に適切なデフォルトの準拠により定義されています。新規のファイル型を追加、または、準拠の他の型にファイルを設定することを希望する場合を除いてこれらの設定を変更しないことが奨励されます。

デフォルトビュー

このグループでは、各ファイルの種類に対して使用されるデフォルトのビューを指定することができます。

テキストビュー

このチェックボックスにより、特定の種類のファイルに対して、構文の色分けを行うよう指定することができます。

自動検証を無効化

このオプションにより、各ファイルに対して自動検証を有効にすることができます。自動検証は通常、ファイルが開かれたとき、またはビューが切り替えられたときに行われます。

空の要素を省略して<E/>で保存する

XML ドキュメントや、XML ドキュメントにより出力されたドキュメントを使用するアプリケーションの中には、XML 1.0 仕様により定義されている(<Element/> という形式の) 空要素が原因でエラーを起こすことがあります。このオプションを無効にすることで、Authentic Desktop は、<Element></Element> という形式で要素の保存を行います。

新たなファイル拡張子を追加

新しいファイルタイプを、ファイルタイプリストに追加します。ファイルタイプを追加した後には、このタブにあるオプションを使用することで、このファイルタイプの定義を行う必要があります。

選択されたファイル拡張子を削除

現在選択されているファイルタイプと、全ての関連する設定を削除します。

保存して完了する

設定が完了したら、「OK」をクリックして完了します。

10.9.9.3 エンコード

エンコードセクションでは、ファイルのエンコーディングを指定することができます。

新規XMLファイルのデフォルトエンコーディング

Unicode UTF-8

リトルエンディアンバイトオーダー
 ビッグエンディアンバイトオーダー

デフォルトエンコーディング

Unicode UTF-8

XMLでないファイルを開く際のエンコーディング

Shift-JIS (Japanese)

BOM (バイト・オーダー・マーク)

UTF-8 でない場合、常に BOM を作成
 保存時に検出した BOM を保持

新規 XML ファイルのデフォルトエンコーディング

ドロップダウンリストから新規 XML ファイルで使用されるデフォルトエンコーディングを選択することができます。ここで指定したエンコーディングが、新たなドキュメントを作成する際、XML 宣言にて使用されます。デフォルトエンコーディングは2バイトまたは4バイトのエンコーディング (UTF-16、UCS-2 または UCS-4) をデフォルトエンコーディングとして選択した場合、リトルエンディアン、またはビッグエンディアンを選択することができます。

既存のXML ファイルにおけるエンコーディングはそのまま保持されますが、「[ファイル | エンコード](#)」コマンドにより変更することができます。

XML ファイルを未知のエンコードを使用して開く

XML ファイルのエンコーディングを決定することができない、またはXMLドキュメントにてエンコーディングの指定がなされていない場合、このコンボボックスで選択したエンコーディングがファイルを開く際に使用されます。

XML でないファイルを開く際のエンコーディング

既存の、そして新規で作成されたXML 以外のファイルは、このコンボボックスで選択したエンコーディングにより開かれます。ドキュメントのエンコーディングは、「[ファイル | エンコード](#)」コマンドにより変更することができます。

BOM (バイト・オーダー・マーク)

2バイトまたは4バイトのエンコーディングによりドキュメントを保存する場合、ドキュメントの保存を、(i) リトルエンディアンオーダー、またはリトルエンディアンBOM により行うか(UTF-8 でない場合、常に BOM を作成)、(ii) 検出されたバイトオーダーまたは検出されたBOM により行うか(保存時に検出した BOM を保持)指定することができます。

保存して完了する

「OK」をクリックして完了してください。

10.9.9.4 ビュー

ビューセッションでは、Authentic Desktop におけるXMLドキュメントの表示方法をカスタマイズすることができます。

プログラムロゴ

アプリケーションの起動時に表示されるスプラッシュ画面を無効にすることで、起動時間を短縮することができます。更に、ライセンスを購入されている場合、プログラムロゴ、著作権情報、登録に関する詳細の印刷をXMLSpy 1にて行わないように指定することができます。

ウィンドウタイトル

各ドキュメントウィンドウのためのウィンドウタイトルは、ファイル名、または、フルパス名になります。

保存して完了

設定が完了したら、「OK」をクリックして完了します。

10.9.9.5 XSL

XSL セグメントでは、アプリケーション内から行われる [XSLT 変換](#) と [XSL-FO 変換](#) におけるオプションを定義することができます

XSLT 変換

Authentic Desktop には Altova XSLT 1.0 エンジンならびに Altova XSLT 2.0 エンジンが搭載されており、XSLT 変換に使用することができます。xsl:stylesheet または xsl:transform 要素の version 属性の値により、適切な XSLT エンジン(1.0 または 2.0)が変換には使用されます。

XSLT を使った XML ドキュメントの変換には、以下のどれかを使用することができます:

- 内蔵の Altova XSLT エンジン (Altova (XSLT 1.0、XSLT 2.0、および XSLT 3.0))。
- MSXML 3.0、4.0、または 6.0 パーサー: お使いのコンピューターで使用されている MSXML のバージョンが明らかな場合、選択を行うことができ、そうでない場合、アプリケーションによるバージョンの選択を行わせるべきでしょう(バージョンを自動的に選択オプションがデフォルトでは選択されています)。このオプションが選択されている場合、利用できる最新バージョンが選択されます。
- 選択した外部 XSLT プロセッサ: 外部 XSLT プロセッサを指定するコマンドライン文字列を入力する必要があります。コマンドライン文字列を構築する際に、以下の変数を使用することができます:

%1 = 処理する XML ドキュメント
 %2 = 生成する出力ファイル
 %3 = 使用する XSLT スタイルシート (XML ドキュメント内に、スタイルシートへの参照が含まれない場合)

例えば以下のコマンドにより、Saxon (XSLT 1.0) を使った単純な変換を行うことができます:

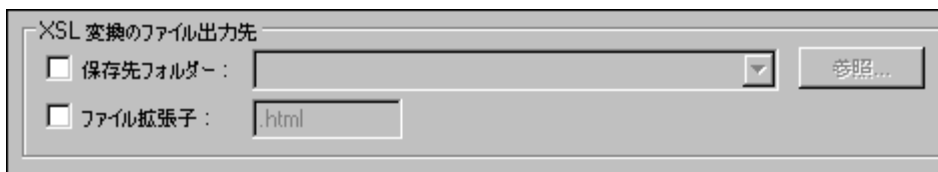
```
myxsltengine.exe -o output.xml input.xml stylesheet.xslt parameter-name=parameter-value
```

Authentic Desktop からこのコマンドを実行するには、外部 XSL 変換プログラムラジオボタンを選択し、以下の文字列をテキストボックスに入力してください。

```
c:\myxsltengine\myxsltengine.exe -o %2 %1 %3 parameter-name=parameter-value
```

適切なチェックボックスにチェックを入れることで、外部プログラムから得られた出力やエラーメッセージを Authentic Desktop のメッセージウィンドウに表示することができます。

アウトプットウィンドウを再利用オプションを使用することで、後続変換の結果を、同じ出力ウィンドウにて表示することができます。XML ファイルがプロジェクトに格納されており、アウトプットウィンドウを再利用オプションが無効になっている場合、この設定は [プロジェクトプロパティ](#) の出力ファイルパスにおける保存先フォルダも無効になっている場合のみ適用されます。



XSL-FO 変換

FO ドキュメントは FO プロセッサにより処理され、FO プロセッサの実行可能ファイルへのパスは XSL-FO 変換エンジンのテキストボックスにて指定する必要があります。変換処理は「[XSL/XQuery | XSL-FO 変換](#)」メニューコマンドから行うことができます。ソースファイル (IDE 内部からコマンドが指定された場合は、アクティブなドキュメント) が XSL-FO ドキュメントの場合、変換に FO プロセッサが呼び出されます。ソースドキュメントが XML ドキュメントの場合、まずは XSLT 変換を使って XML ドキュメントを XSL-FO ドキュメントへ変換する必要があります。この XSLT 変換では、アプリケーションのデフォルトエンジンとして指定した XSLT エンジンのほか、デフォルト FO プロセッサとして指定した FO プロセッサ内部に組み込まれた XSLT エンジンを使用することもできます ([上を参照](#))。これらオプションを選択するには、適切なラジオボタンをクリックしてください。

設定が完了したら、「OK」をクリックして、オプションダイアログを閉じてください。

メモ: [Apache XML プロジェクト](#) の FOP プロセッサをインストールするオプションを解除しない限り、次のフォルダ内にインストールされます C:\ProgramData\Altova\SharedBetweenVersions。インストールされると、XSL-FO エンジン入力ボックス内にパスが自動的に入力されます。FO プロセッサへのパスを希望するように設定することができます。しかしながら、FO プロセッサを使用する他の Altova 製品により同じパスが使用される場合、FO プロセッサを選択して設定をすることもできます (StyleVision と Authentic Desktop)。

保存して完了する

設定が完了したら、「OK」をクリックして、オプションダイアログを閉じてください。

10.9.9.6 スクリプト

スクリプトタブでは、アプリケーションの起動時に [スクリプト環境](#) を有効にすることができます。有効にするには、スクリプトをアクティブにするチェックボックスにチェックを入れてください (以下のスクリーンショットを参照)。

スクリプト

アクティブにする

スクリプトをアクティブにする

グローバル スクリプト プロジェクト ファイル:

C:\Examples\SampleScripts.asprj 参照...

自動スクリプト 処理

Authentic Desktop のスタート時にオートマクロを実行

イベントを処理

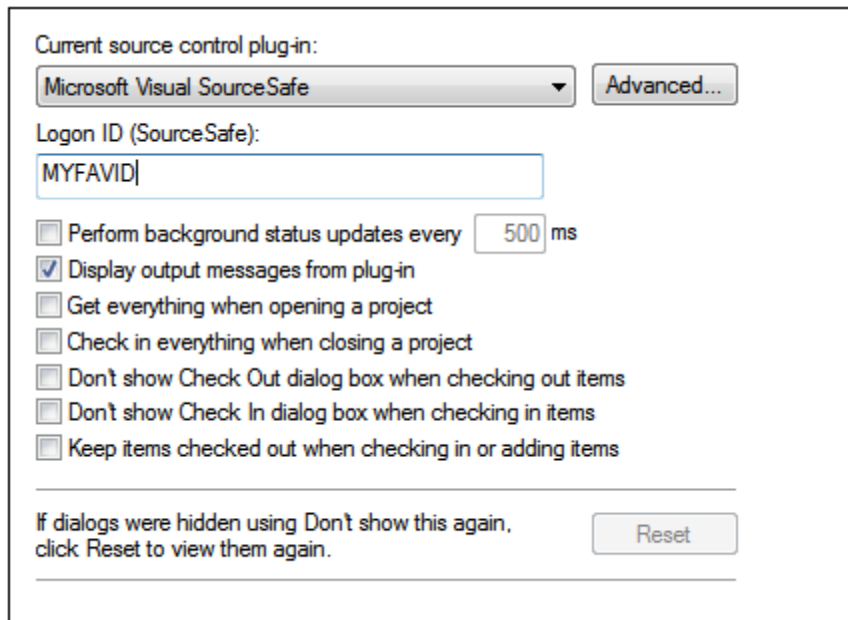
Authentic Desktop に対してグローバルスクリプトプロジェクトを設定するには、スクリプトをアクティブにするチェックボックスにチェックを入れた後、必要な Altova スクリプトプロジェクト (.asprj) ファイルを指定します。その他にも、(i) Authentic Desktop の起動時にマクロを自動的に実行するか、(ii) プロジェクト内のアプリケーションイベントハンドラーを自動的に実行するかを有効に(または無効に)することができます。

保存して完了する

「OK」をクリックして完了してください。グローバルリソーススクリプトプロジェクト内のマクロがマクロコマンドのサブメニューに表示されます。

10.9.9.7 ソース管理

ソース管理セクションではソース管理プロバイダーの指定を行い、デフォルトのログインID やユーザー設定を各種ソース管理プロバイダーに対して指定することができます。



ソース管理プラグイン

現在インストールされている中から、使用するソース管理プラグインをエボボックスのドロップダウンリストを使って選択することができます。目的のソース管理を選択したら、次のテキストボックスにてログインIDを入力します。「詳細」ボタンをクリックすることで、選択されたソース管理プラグインに特化したダイアログが表示され、ソース管理プラグインに対して設定を行うことができます。これら設定の内容はソース管理プラグインにより変化します。

ユーザー設定

以下のようなユーザー設定を利用することができます：

- ユーザーが指定した時間間隔ごとに、バックグラウンドでステータスの更新を行うか、この機能を無効にすることもできます。巨大なソース管理データベースを使用する場合、多くのCPUパワーならびにネットワークリソースが消費されることとなります。バックグラウンドのステータス更新機能を無効にするか、インターバルの時間を長く取ることで、システムのスピードを回復することができます。
- プロジェクトを開くならびに閉じる時に、それぞれファイルのチェックアウトやチェックインを自動的に実行することができます。
- チェックアウトならびにチェックインを行う際に、ダイアログの表示を無効にすることができます。
- ダイアログボックスにて次回から表示しないオプションを選択した場合、リセットボタンが有効となります。リセットボタンをクリックすることで、ダイアログが再度表示されるようになります。

保存して完了

設定が完了したら、「OK」をクリックして完了します。

10.9.9.8 ネットワーク プロキシ

ネットワークプロキシセクションでは、カスタムのプロキシの設定を構成することができます（下のスクリーンショット）。デフォルトは、システムのプロキシの設定が使用されますが、設定はユーザーの介入無しで作動します。代替のネットワークプロキシを設定する場合、設定を定義するオプションを使用してください。

ネットワーク プロキシ

システムのプロキシ設定を使用(U)

自動プロキシ構成(A)

自動検知の設定(D)

スクリプト URL(L) 再ロード(R)

手動のプロキシ構成(M)

HTTP プロキシ(H) ポート

このプロキシサーバーをすべてのプロトコルのために使用する(P)

SSL プロキシ(S) ポート

プロキシ無し(N)

プロキシのサーバーをローカルのアドレスのために使用しないでください(X)

現在のプロキシの設定(C)

URL のテスト(T) ↻

IE 自動プロキシ構成が見つかりました。
 メソッド WPAD (テスト URL http://www.example.com を使用)
 エラー 12180:
 IE プロキシ構成が見つかりませんでした。
 WinHTTP プロキシ構成が見つかりませんでした。

システムプロキシの設定の使用

システムプロキシ設定を介して構成可能なインターネットエクスプローラー(IE)設定を使用します。`netsh.exe winhttp` を介して構成される設定が必要とされます。

自動プロキシの構成

以下のオプションを使用することができます:

- **自動検知の設定:** DHCP または DNS を使用して WPAD スクリプト (`http://wpad.LOCALDOMAIN/wpad.dat`) を検索し、プロキシセットアップのためにこのスクリプトを使用します。
- **スクリプト URL:** プロキシセットアップのために使用されるプロキシ自動構成 (.pac) スクリプトに対する HTTP URL を指定します。
- **再ロード:** 現在の自動プロキシ構成をリセットして再ロードします。このアクションには Windows 8 または以降が必要とされ、30 秒程度の時間が必要です。

手動のプロキシの構成

ホスト名とポートを対応する製品のプロキシのために手動で指定します。サポートされるスキームはホスト名に含まれている場合があります (例: `http://hostname`)。プロキシがスキームをサポートする場合、対応するプロトコルと同じである必要はありません。

以下のオプションを使用することができます:

- *HTTP プロキシ*: HTTP プロトコルのための指定済みのホスト名とポートを使用します。全てのプロトコルのためにこのプロキシサーバーを使用するか選択されている場合、HTTP プロキシがすべてのプロトコルのために使用されます。
- *SSL プロキシ*: SSL プロトコルのために指定されているホスト名とポートが使用されます。
- *プロキシ無し*: セミコロン(;) により区別されているプロキシを使用しないホスト名、ドメイン名、またはホストのためのIP アドレスのリスト。IP アドレスは切り捨てられず、IPv6 アドレスは角かっこで囲まれる必要があります(例: [2606:2800:220:1:248:1893:25c8:1946])。ドメイン名はドットと共に開始される必要があります(例: .example.com)。
- *プロキシのサーバーをローカルアドレスのために使用しない*: チェックされている場合、*プロキシ無し*リストのために <local> を追加します。このオプションが選択されている場合、次の場合、プロキシは使用されません (i) 127.0.0.1、(ii) [::1]、(iii) (.) ドット文字を含んでいないホスト名すべて。

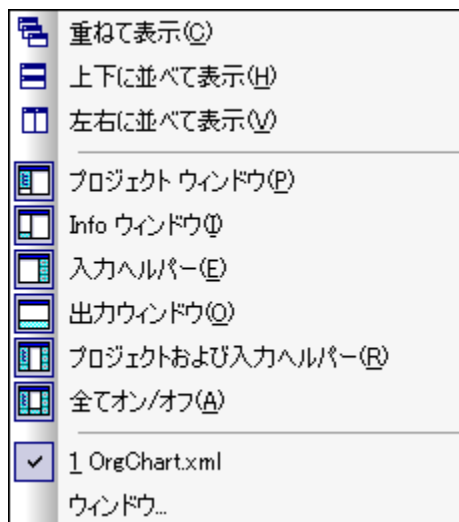
現在のプロキシの設定

プロキシの検知の詳細なログを提供します。URL のテスト フィールドの右の「更新」ボタンを使用して更新することができます(例、URL のテスト を変更する場合、または、プロキシの設定が変更された場合)。

- *URL のテスト*: URL のテストを使用して、どのプロキシが特定のURL ために使用されているかを確認することができます。URL を使用して、I/O はされません。プロキシの自動構成が選択されている場合、このフィールドは空にしておく必要があります(システムプロキシの設定の使用、または、自動プロキシの構成 を使用して)。

10.10 ウィンドウメニュー

ウィンドウメニューには、殆どのWindows アプリケーションに備わっているコマンドが含まれており、Authentic Desktop セッションで開かれた個々のドキュメントウィンドウを管理することができます。



開かれているドキュメントウィンドウを重ねて表示したり、上下や左右に並べて表示、最小化しながらドキュメントウィンドウへアクセスすることができます。各種入力ヘルパーを有効/無効にすることができるため、ウィンドウダイアログから、目的のドキュメントウィンドウへ直接アクセスすることもできます。

10.10.1 重ねて表示

このコマンドにより開かれている全てのドキュメントウィンドウが重ね合わさって表示されるよう配置されます。

10.10.2 上下に並べて表示

このコマンドにより、開かれている全てのドキュメントが上下に並べて配置され、複数のドキュメントを同時に閲覧することができます。

10.10.3 左右に並べて表示

このコマンドにより、開かれている全てのドキュメントが左右に並べて配置され、複数のドキュメントを同時に閲覧することができます。

10.10.4 プロジェクトウィンドウ

このコマンドによりプロジェクトウィンドウの表示/非表示を切り替えることができます。

プロジェクトウィンドウをドッキングすることのできるウィンドウで、タイトルバーをドラッグすることで、オリジナルの位置から切り離してフロートウィンドウにすることができます。タイトルバーを右クリックしてウィンドウをドッキングしたり隠したりすることができます。

10.10.5 情報ウィンドウ

このコマンドにより [情報ウィンドウ](#) の表示/非表示を切り替えることができます。

情報ウィンドウをドッキングすることのできるウィンドウで、タイトルバーをドラッグすることで、オリジナルの位置から切り離してフロートウィンドウにすることができます。タイトルバーを右クリックしてウィンドウをドッキングしたり隠したりすることができます。

10.10.6 入力ヘルパー

このコマンドにより全ての入力ヘルパーの表示/非表示を切り替えることができます。

3つある入力ヘルパーウィンドウをドッキングすることのできるウィンドウで、タイトルバーをドラッグすることで、オリジナルの位置から切り離してフロートウィンドウにすることができます。タイトルバーを右クリックしてウィンドウをドッキングしたり隠したりすることができます。

10.10.7 出カウィンドウ

出カウィンドウは、(検証結果のようなメッセージを表示する)メッセージウィンドウや、(XPath 条件式評価の結果を表示する)XPath ウィンドウをタブ化したウィンドウです。初期状態ではメインウィンドウの下部に表示されます。「出カウィンドウ」コマンドにより、出カウィンドウの表示/非表示を切り替えることができます。

出カウィンドウのウィンドウをドッキングすることのできるウィンドウで、タイトルバーをドラッグすることで、オリジナルの位置から切り離してフロートウィンドウにすることができます。タイトルバーを右クリックしてウィンドウをドッキングしたり隠したりすることができます。

出カウィンドウに関する詳細は、テキストビューセグメントの [出カウィンドウ](#) を参照ください。

10.10.8 プロジェクトおよび入力ヘルパー

このコマンドによりプロジェクトウィンドウならびに入力ヘルパーの表示/非表示を選択することができます。

10.10.9 全てオン/オフ



このコマンドにより、以下にあるドッキング可能な全てのウィンドウの表示/非表示を切り替えることができます：

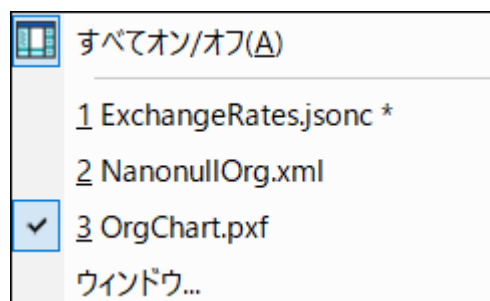
- [プロジェクトウィンドウ](#)
- [情報ウィンドウ](#)

- 3つの入力ヘルパーウィンドウ
- [出力ウィンドウ](#)

このコマンドによりドキュメント以外の全ウィンドウを素早く隠して、ドキュメントの表示を作業エリア全体に広げることができます。

10.10.10 現在開かれているウィンドウのリスト

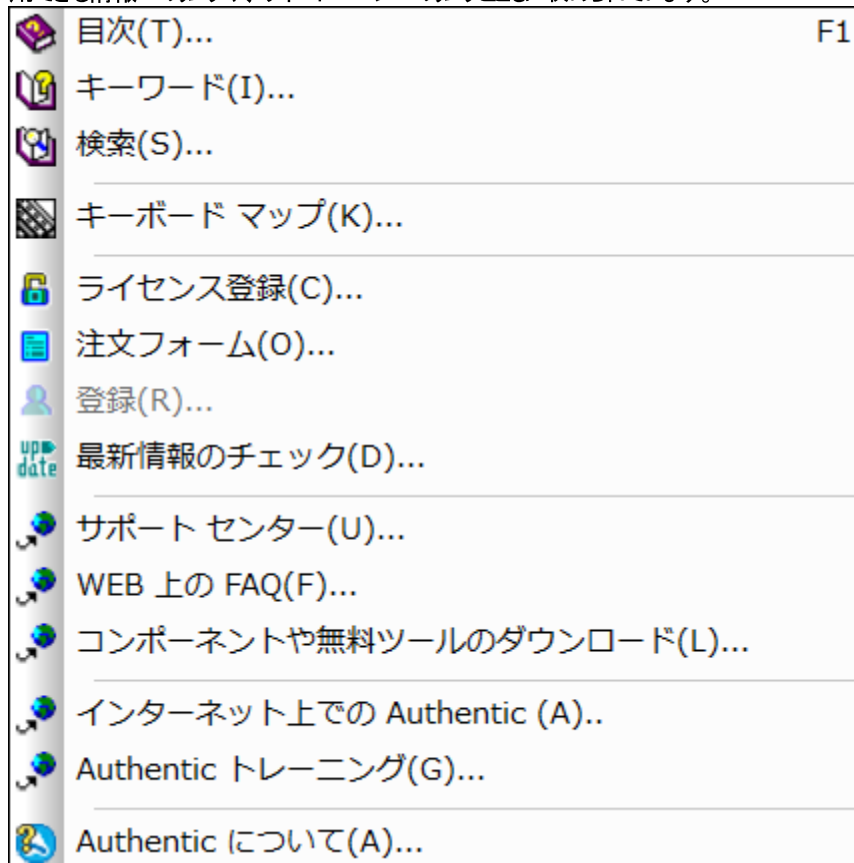
このリストは現在開かれているすべてのウィンドウを表示し、ウィンドウ間で素早く切り替えることができます。



また **CTRL+F6** キーボードショートカットを使用して、開かれているウィンドウ間を循環することができます。

10.11 ヘルプメニュー

ヘルプメニューにはヘルプならびに Authentic Desktop に関する更に詳しい情報を得るために必要なコマンドが、ウェブサーバー上で利用できる情報へのリンクや、サポートページへのリンクとともに収められています。



ヘルプメニューからは [登録ダイアログ](#) を起動することもでき、購入した製品のライセンスキーを入力することができます。

10.11.1 目次、インデックス、検索

▼ 目次

☐ 説明

ヘルプウィンドウの左側のペインに目次を表示した、Authentic Desktop の画面上のヘルプマニュアルを開きます。目次はヘルプドキュメント全体の概要を表示しています。目次のエントリをクリックしてトピックに移動することができます。

▼ インデックス

☐ 説明

ヘルプウィンドウの左側のペインにキーワード インデックスを表示した Authentic Desktop の画面上のヘルプマニュアルを開きます。目次はヘルプドキュメント全体の概要を表示しています。インデックスはキーワードをリストし、キーワードをダブルクリックすることでトピックに移動することができます。キーワードが一つ以上のトピックにリンクされている場合は、トピックのリストが表示されます。

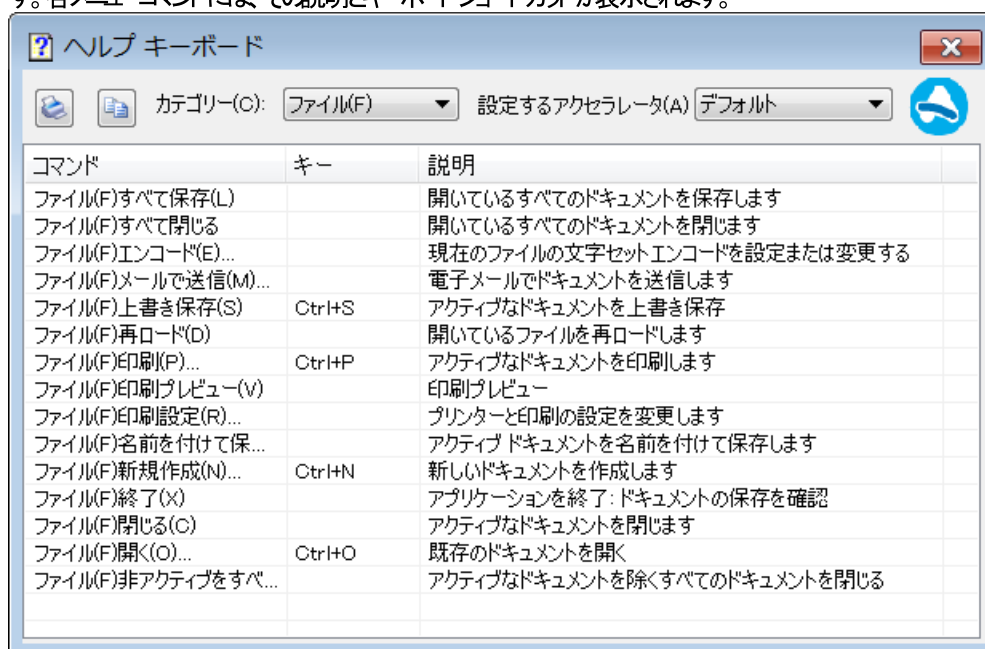
▼ 検索

☐ 説明

ヘルプウィンドウの左側のペインに検索ダイアログを表示したAuthentic Desktop の画面上のヘルプメニューを開きます。単語を検索するには、入力フィールドに検索対象を入力して、(i)「Return」を押す、または(ii)「トピックのリスト」をクリックします。を押します。ヘルプシステムは、ヘルプドキュメント全体で全文検索を行いヒットしたリストを返します。アイテムを表示するためにはアイテムをダブルクリックします。

10.11.2 キーボードマップ

「ヘルプ | キーボードマップ」コマンドにより、Authentic Desktop 内にある全てのコマンドが、メニューごとに分かれて表示されます。各メニューコマンドには、その説明とキーボードショートカットが表示されます。



特定のメニューに収められているコマンドを確認するには、カテゴリコンボボックスからメニュー名を選択します。印刷アイコンをクリックすることで、コマンドを印刷することができます。

10.11.3 ライセンス登録、注文フォーム、登録、最新情報のチェック

▼ ソフトウェアのライセンス認証

☐ 説明

Altova 製品ソフトウェアをダウンロードすると、無料評価キーまたは購入されたライセンスキーを使用して、製品にライセンスを供与、または、ライセンスの認証を行うことができます。

- **無料 評価ライセンス**初めて製品のがダウンロードとインストールを行うと、ソフトウェアライセンス認証ダイアログが表示さ

れます。ダイアログでは無料 評価 ライセンスをリクエストすることができます。ユーザーの名前、所属会社名、そして電子メールアドレスを表示されるダイアログに入力し、リクエストをクリックします。ライセンスファイルが入力された電子メールアドレスに送信されます。この手順は数分を要します。ライセンスファイルを適切な場所に保存します。リクエストをクリックすると、リクエストダイアログの下に入力フィールドが表示されます。このフィールドはライセンスファイルのパスを取ります。ライセンスファイルを参照 またはライセンスファイルへのパスを入力し「OK」をクリックします。(「ソフトウェアのライセンス認証ダイアログ」内で「新規のライセンスをアップロードする」をクリックしてライセンスファイルへのパスを入力するダイアログにアクセスすることができます。)ソフトウェアは30日の間アンロックされます。

- **永続的なライセンスキー:** ソフトウェアライセンス認証ダイアログには永続的なライセンスキーを購入するためのボタンが含まれています。このボタンをクリックすると、製品の永続的なライセンスキーを購入することのできる Altova オンラインショップに移動することができます。受信する電子メールはライセンスデータを含むライセンスファイルの書式で送信されています。3つの種類の永続的なライセンスが存在します: インストール済み、同時使用ユーザー、名前を持つユーザー。
インストール済みのライセンス は単一のコンピュータ上のソフトウェアのロックを解除します。インストール済みのライセンスをN台のコンピュータのために購入するとN台までのコンピュータ上でソフトウェアを使用することができます。
同時使用ユーザーライセンス はN人の同時使用ユーザーにN人のユーザーに同時にソフトウェアを使用することを許可します。(10N台のコンピュータにソフトウェアをインストールすることができます。) **名前を持つユーザーライセンス** は特定のユーザーがN台の異なるコンピュータ上でソフトウェアを使用することを許可します。ソフトウェアのライセンスを認証するときは「新規のライセンスをアップロードする」をクリックして表示されるダイアログ内でライセンスファイルを参照、またはライセンスファイルのパスを入力し「OK」をクリックします。

メモ 複数のユーザーライセンスのために各ユーザーは各自の名前を入力するようプロンプトされます。

ライセンスの電子メールと Altova 製品へのライセンス供与(有効化)の異なる方法

Altova から受信するライセンス電子メールにはライセンスファイルが添付ファイルとして含まれています。ライセンスファイルは .altova_licenses ファイル拡張子を有しています。

Altova 製品のライセンスを認証するには、以下のうち1つ行ってください!

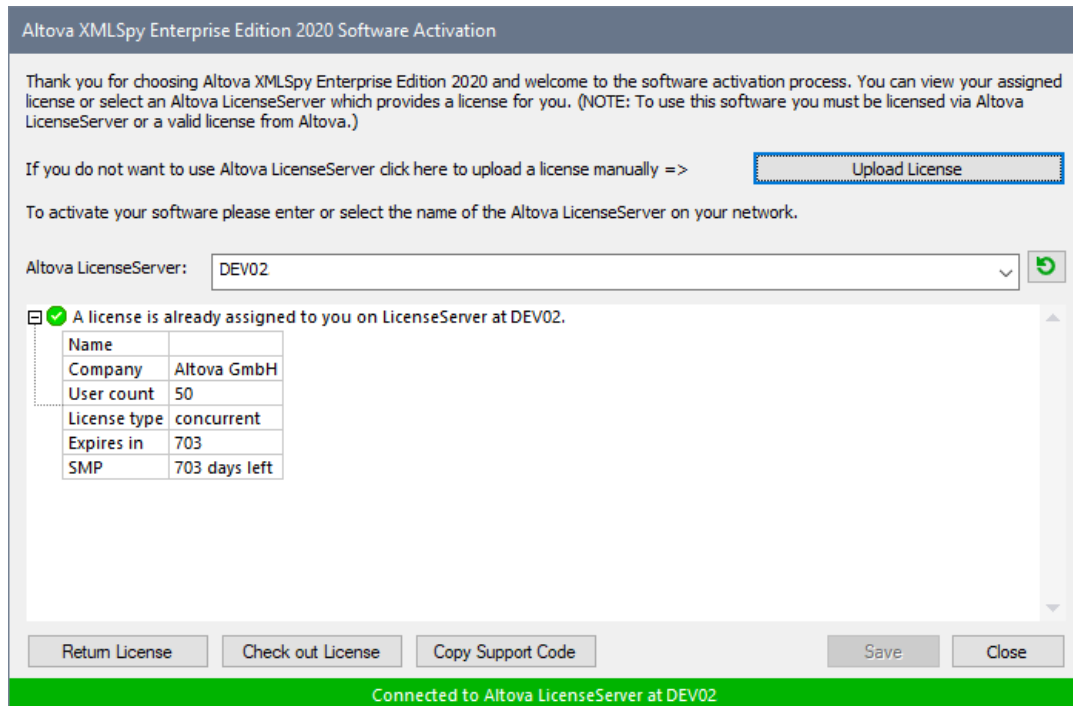
- 適切な場所にライセンスファイル(.altova_licenses)を保存し、ライセンスファイルをダブルクリックし、表示されるダイアログに必要な情報を入力し、「キーの適用」をクリックして完了します。
- ライセンスファイル(.altova_licenses)を適切な場所に保存します。Altova 製品内では「ヘルプ | ライセンス登録メニューコマンド」を選択し、新規のライセンスをアップロードします。ライセンスファイルへのパスを入力し「OK」をクリックします。
- 適切な場所にライセンスファイル(.altova_licenses)を保存し、Altova LicenseServer のライセンスプールにアップロードします。以下を行うことができます: (i) 製品のソフトウェアライセンス認証ダイアログから Altova 製品からライセンスを取得します。(以下を参照)または(ii) Altova LicenseServer から製品へのライセンスを割り当てます。LicenseServer の使用の詳細に関しては、下記の記事を参照してください!

ソフトウェアライセンス認証ダイアログ(下のスクリーンショット)は、「ヘルプ | ソフトウェア アクティベーション」をクリックすることにより常にアクセスすることができます。

以下の方法によりソフトウェアをアクティブ化することができます:

- **ソフトウェアライセンス認証ダイアログでライセンスを登録する方法:** ダイアログ内で、「新規のライセンスをアップロード」をクリックして、ライセンスファイルを参照し選択します。「OK」をクリックしてライセンスファイルへのパスを確認します(複数のユーザーライセンスの場合は個人の名前です)。「保存」をクリックして完了します。
- **ネットワーク上の Altova LicenseServer を使用してライセンス供与する方法:** ネットワーク上の Altova LicenseServer を使用してライセンスを取得するにはソフトウェアのライセンスの認証ダイアログの下にある **Altova LicenseServer** を使用するをクリックします。使用する LicenseServer がインストールされているマシンを選択します。License Servers の自動検知は LAN 上で配信が送信されることを意味します。これらの配信がサブネッ

トに制限されているため、License Server は自動検知のためのクライアントマシンと同じサブネット上に存在する必要ががあります。自動検知が作動しない場合、サーバーの名前を入力します。Altova LicenseServer はライセンスプール内で Altova 製品のためのライセンスを有している必要ががあります。LicenseServer プール内に存在する場合、ソフトウェアライセンス認証ダイアログ内に表示されます (*Altova XMLSpy 内のダイアログで表示されている例を参照してください*)。「保存」をクリックしてライセンスを取得します。



マシン固有のライセンスが LicenseServer からインストールされると、7日間、LicenseServer に戻すことができず、7日過ぎると、「ライセンスを戻す」をクリックしてマシンのライセンスを LicenseServer に戻すことができ、このライセンスは他のクライアントにより LicenseServer から取得することができます。LicenseServer 管理者は、LicenseServer の Web UI を使用して、取得されたライセンスの割り当てを解除することができます。ライセンスの返却は、マシン固有のライセンスのみ適用され、現在使用中のライセンスには適用されないのでご注意ください。

ライセンスのチェックアウト

ライセンスが製品マシン上に保管されるように、ライセンスをライセンスプールから 30 日間チェックアウトすることができます。これにより、オフラインで作業することが可能になります。この機能はとても役に立ちます。Altova LicenseServer にアクセスできない環境 (例えば、旅行中に Altova 製品がインストールされたラップトップコンピューターで作業する場合など) が挙げられます。ライセンスはチェックアウトされていますが、LicenseServer は、ライセンスが使用中と表示し、ライセンスは他のマシンで使用することができません。ライセンスはチェックアウトの期間が終了すると自動的にチェックインされた状態に戻ります。または、チェックアウトされたライセンスはソフトウェアのライセンスの認証ダイアログのボタンを使用して「チェックイン」することができます。

ライセンスをチェックアウトする場合は以下をおこないます: (i) ソフトウェアのライセンスの認証ダイアログで「ライセンスのチェックアウト」をクリックします (上のスクリーンショット参照)。 (ii) ライセンスのチェックアウトダイアログ内から、チェックアウトの期間を選択し、「チェックアウト」をクリックします。ライセンスがチェックアウトされます。ライセンスのチェックアウト後 2 つの状態が発生します: (i) ソフトウェアのライセンス認証ダイアログは時刻およびチェックアウトの期間を含むチェックアウトに関する情報を表示します。 (ii) ダイアログ内の「ライセンスのチェックアウト」ボタンは「チェックイン」ボタンに変更されます。「チェックイン」ボタンをクリックして、ライセンスをチェックインすることができます。チェックアウト期間の期限が切れると、ライセンスは自動的にチェックイン状態に戻されるため、選択したチェックアウトの期間がオフラインで作業する期間をカバーするように確認してください。

メモ ライセンスのチェックアウトを可能にするには、LicenseServer 上でチェックアウト機能が有効化されている必要があります。チェックアウトを試みる際この機能が有効化されていない場合、エラーメッセージが表示されます。この場合、LicenseServer 管理者に連絡してください。

サポートコードのコピー

「サポートコードのコピー」をクリックして、ライセンスの詳細をクリップボードにコピーしてください。これは[オンラインサポートフォーム](#)を使用してサポートをリクエストする際に必要なデータです。

Altova LicenseServer を使用することにより、IT 管理者は、リアルタイムでネットワーク上の全てのライセンスの概要、および、クライアントの割り当てと、クライアントのライセンスの使用状況を確認することができます。LicenseServer を使用する利点は、ですから、多数の Altova ライセンスを管理することのできる管理機能です。Altova LicenseServer は、[Altova Web サイト](#)で無料で提供されています。Altova LicenseServer および Altova LicenseServer を使用したライセンスの供与に関する詳細は、[Altova LicenseServer ドキュメントを参照してください](#)。

▼ 注文フォーム

☐ 説明

ソフトウェア製品のライセンス許与バージョンを注文する準備が整っている場合、(前のセクション参照) ソフトウェアライセンス認証ダイアログ内の「永久ライセンスの購入」ボタン、または「注文フォーム」コマンドを使用して Altova オンラインショップで移動して注文することができます。

▼ 登録

☐ 説明

Altova 製品登録ページをブラウザのタブに表示します。Altova ソフトウェアを登録することにより、最新の製品の情報が得られます。

▼ 更新のチェック

☐ 説明

Altova サーバに接続して、より新しいバージョンの製品が利用可能かどうかチェックし、その結果を表示します。

10.11.4 他のコマンド

▼ サポートセンター

☐ 説明

インターネット上にある Altova サポートセンターへのリンクとなっています。サポートセンターには FAQ やディスカッションフォーラムが含まれており、問題の解決方法を探り、Altova の技術サポートスタッフへアクセスすることができます(現在英語のみの提供となります)。

▼ WEB 上の FAQ

☐ 説明

インターネット上にある Altova の FAQ へのリンクとなっています。FAQ データベースは Altova のサポートスタッフにより常時更新されています。

▼ コンポーネントのダウンロード

☐ 説明

インターネット上にある Altova のコンポーネントダウンロードセンターへのリンクとなっています。このリンク先から様々なコンポーネントソフトウェアをダウンロードして、Altova 製品とともに使用することができます。ソフトウェアコンポーネントは XSLT や XSL-FO プロセッサからアプリケーションサービスプラットフォームまで、幅広く提供されています。コンポーネントダウンロードセンターにてご利用いただけるソフトウェアは、通常無料でご利用いただけます。

▼ インターネット上の Authentic Desktop

☐ 説明

インターネット上にある [Altova Web サイト](#) へのリンクとなっています。[Altova Web サイト](#) では、Authentic Desktop や関連するテクノロジーについて確認することができます。

▼ Authentic Desktop について

☐ 説明

スプラッシュ画面と製品のバージョン番号が表示されます。Authentic Desktop の 64 ビットバージョンを使用している場合、これはアプリケーション名の後のサフィックス (x64) により示されています。32 ビットバージョンにはサフィックスは存在しません。

10.12 コマンドライン

コマンドラインから Authentic Desktop に搭載されているいくつかのコマンドを実行することができます。以下のコマンドを利用することができます:

ファイルを開く

コマンド: `authentic.exe file.xml`

動作: Authentic Desktop にてファイル `file.xml` を開く。

複数のファイルを開く

コマンド: `authentic.exe file1.XML file2.xml`

動作: Authentic Desktop にてファイル `file1.xml` ならびに `file2.xml` を開く。

SPS ファイルを XML ファイルに割り当てて、Authentic ビューによる編集を行う

コマンド: `authentic.exe myxml.xml /sps mysps.sps`

動作: `mysps.sps` を SPS ファイルとして、ファイル `myxml.xml` を Authentic ビューで開く。/`sps` フラグにより、あとで続く SPS ファイルが /`sps` フラグの前にある XML ファイルに (Authentic ビュー編集を行うため) 割り当てられます。

SPS ファイルから新たな XML テンプレートを開く

コマンド: `authentic.exe mysps.sps`

動作: 新規 XML ファイルを Authentic ビューで開きます。表示は SPS をベースとしたもので、SPS スキーマをベースとした新規 XML ファイルが作成されます。XML ファイルの保存を行う際に、ファイルの名前を指定する必要があります。

11 Visual Studio 内の Authentic Desktop

Authentic Desktop は Microsoft Visual Studio IDE のバージョン 2010/2012/2013/2015/2017/2019 に統合することができます。この機能により、高度な XML 編集機能を、Visual Studio の高度な開発環境内で使用できるようになります。

このセクションは以下について説明されています:

- [大まかなインストールの流れ](#) と Authentic Desktop プラグインを Visual Studio へ統合
- Visual Studio バージョンとスタンドアロンバージョンの[違い](#)

11.1 Visual Studio のための Authentic Desktop プラグインのインストール

Visual Studio で使える Authentic Desktop プラグインをインストールするには、以下の手順を踏みます：

1. Microsoft Visual Studio 2010/2012/2013/2015/2017/2019 のインストール。
2. Authentic Desktop (Enterprise または Professional Edition) のインストール
3. Authentic Desktop の Microsoft Visual Studio Integration Package をダウンロードして実行。パッケージは www.altova.com にある Authentic Desktop (Enterprise または Professional Edition) ダウンロードページにてご利用いただけます。(注意: Authentic Desktop バージョンに対応する Integration Package を使用する必要があります (現在のバージョンは 2021 です。))

Integration Package をインストールすると Visual Studio 環境から Authentic Desktop を使用できるようになります。

11.2 スタンドアロンバージョンとの違い

このセクションでは Authentic Desktop のスタンドアロンバージョンとの違いについて説明されています。

入力ヘルパー (Visual Studio のツールウィンドウ)

Authentic Desktop の入力ヘルパーは Visual Studio のツールウィンドウとして使用することができます。以下の点に注意してください。(入力ヘルパーと Authentic Desktop の GUI に関しては [ヒント](#) を参照ください。)

- 入力ヘルパーウィンドウは開発環境内のどのような場所でもドラッグすることができます。
- 入力ヘルパータブを右クリックすることで、インターフェースを更にカスタマイズすることができます。入力ヘルパーのオプションは: フローティング、ドッキング可能、タブ付きドキュメント、自動的に隠す、です。

Visual Studio コマンドとしての Authentic Desktop コマンド

Authentic Desktop コマンドの一部は Visual Studio GUI 内の Visual Studio コマンドとして存在しています。これは以下の通りです:

- *元に戻す*、*やり直す*: これらの Visual Studio コマンドは、Visual Studio 開発環境全てに対して適用されます。
- *プロジェクト*: Authentic Desktop プロジェクトは Visual Studio プロジェクトとして扱われます。
- *ツールバーのカスタマイズ*、*コマンドのカスタマイズ* (以下のスクリーンショットにある) ユーザー設定ダイアログ ([ツール] ユーザー設定) 内のツールバーならびにコマンドタブには Visual Studio と Authentic Desktop のコマンドが含まれます。
- *表示*: [表示] メニュー [Authentic ツールウィンドウ] サブメニューには入力ヘルパーウィンドウや他のサイドバーの切り替え、編集ビュー間の切り替え、そして特定の入力ガイドの有効/無効を切り替えるオプションが含まれています。
- *Authentic ヘルプ*: この Authentic Desktop メニューは Visual Studio [ヘルプ] メニューのサブメニューに表示されます。

追加のメモ

追加のメモヒントは以下にリストされるとおぼです:

- Authentic プラグインを使用して XML ファイルを編集するには [ファイル] 開く コマンドを選択してください。そして、「ファイルを開く」ダイアログ内で、「開く」オプションを使用して Authentic プラグインを選択します。

12 Eclipse 内の XMLSpy

Eclipse はオープンソースのフレームワークで、様々なアプリケーション開発環境をプラグインという形で統合します。

Eclipse 用の Authentic Desktop プラグインを使えば、Eclipse 2020-12, 2020-09, 2020-06, 2020-03 プラットフォームから Authentic Desktop の機能にアクセスできるようになります。プラグインは Windows プラットフォームにてご利用になれます。このセクションでは Eclipse 用の Authentic Desktop プラグインの[インストールの方法](#)と、[Authentic パースペクティブ](#)の設定方法について説明します。これらの操作を行うと、Authentic Desktop GUI と Authentic Desktop メニューコマンドを Eclipse GUI から使用できるようになります。

12.1 Eclipse のための Authentic Desktop プラグインのインストール

必要条件

- Eclipse 2020-12, 2020-09, 2020-06, 2020-03 (<http://www.eclipse.org>)。
- the 64ビットプラットフォームのためのJava Runtime Environment (JRE) またはJava Development Kit (JDK)
- Authentic Desktop Enterprise またはProfessional Edition 64ビット
- Authentic Desktop Integration Package 64ビット を以下でダウンロードすることができます <https://www.altova.com/ja/components/download>。下記の通りEclipse 統合をインストールの一部として行うことができます。

メモ 上記の必要条件は64ビットプラットフォームを必要としています。古いEclipse 32ビットプラットフォームとの統合は作動する場合がありますが、サポートされていません。

Eclipse のための Authentic Desktop プラグインをインストールする方法

Eclipse のための Authentic Desktop プラグインの統合を以下の方法の一つで行うことができます:

1. 自動的に Authentic Desktop Integration Package のインストール中に統合することができます(これは奨励されるオプションです)。
2. Authentic Desktop Integration Package を最初にインストールし、Eclipse からプラグインを手動で統合する。

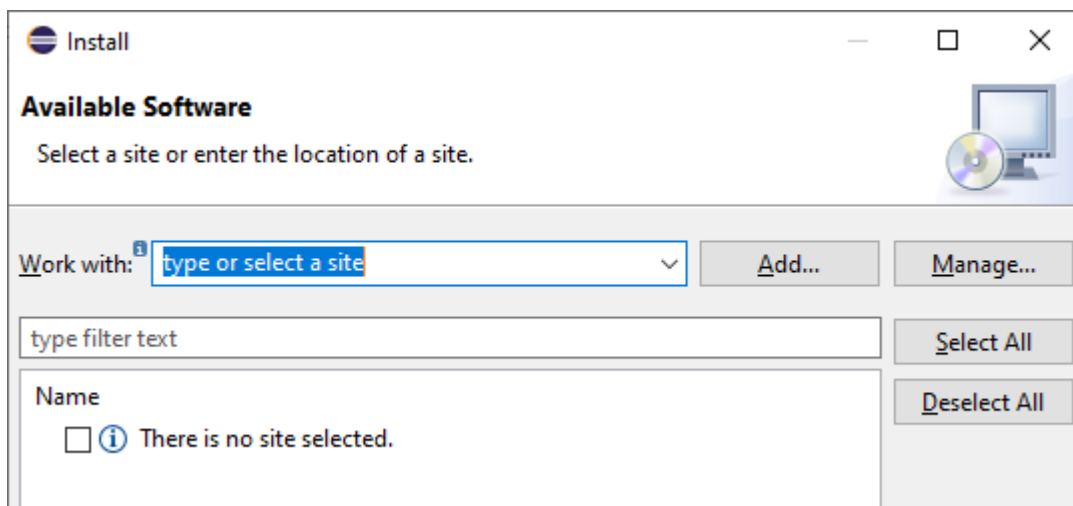
Eclipse のための Authentic Desktop プラグインを統合する方法

1. インストールウィザードを開始するために Authentic Desktop Integration Package を実行します。
2. プロンプトされると Eclipse プラグインのインストールオプションを選択し [次へ] をクリックします。
3. プロンプトされると Authentic Desktop プラグインの Eclipse への統合方法を選択します。以下の一つを行ってください
 - プラグインインストールを自動的に完了する場合 このウィザードに *Altova Authentic Desktop プラグインを Eclipse に統合する* を選択し、Eclipse 実行可能ファイル(eclipse.exe) がロードされている場所を参照します。
 - Eclipse 内で後でプラグインのインストールを完了する場合、clear the このウィザード... チェックボックスをクリアします。
4. [次] をクリックすることでインストールを完了します。自動統合を選択する場合 Authentic Desktop パースペクティブとメニューが Eclipse を次回起動する際に Eclipse 内で使用可能になります。

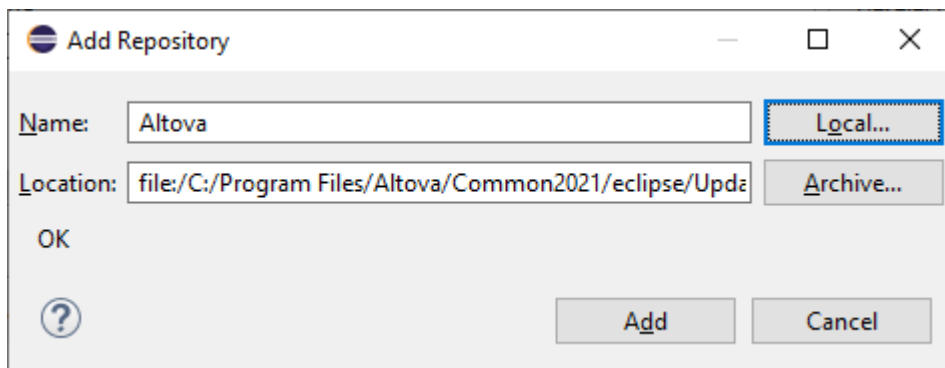
メモ Authentic Desktop Integration Package をインストールまたはアンインストールする場合 Eclipse を閉じる必要があります。

Eclipse のための Authentic Desktop プラグインを手動で統合する方法

1. Eclipse [にてメニューオプションからヘルプ | 新規ソフトウェアのインストール] を選択します。
2. [インストール] ダイアログボックス内で [追加] をクリックします。



3. [レポジトリの追加] ダイアログボックス内で[ローカル]をクリックします。フォルダーC:\Program Files\Altova\Common2021\eclipse\UpdateSite を参照して選択します。("Altova" などの) サイトの名前を与えます。



4. ステップ2 から3 を繰り返します、今度は、フォルダーC:\Program Files\Altova\Authentic2021\eclipse\UpdateSite を選択し、Authentic Desktop などの名前を与えます。
5. インストールダイアログボックスで ローカルのサイトのみを選択します。次に[Altova category] フォルダを選択し [次へ]をクリックします。
6. インストールされるアイテムをレビューし [次へ] をクリックして継続します。
7. 使用許諾契約書に合意するコは対応するチェックボックスを選択します。
8. [完了] をクリックすることでインストールを完了します。

メモ (不足するアイコンなど)プラグインの使用に関する問題が発生した場合 `-clean` フラグを使用してコマンドラインからEclipse を起動します。

12.2 Eclipse 内の Authentic Desktop エントリポイント

Eclipse 内にある以下のエントリポイントを使って Authentic Desktop の機能にアクセスすることができます:

- [Authentic Desktop パースペクティブ](#) から Authentic Desktop の GUI 機能に Eclipse GUI からアクセスすることができます。
- [Authentic Desktop メニューとツールバー](#)

Authentic Desktop パースペクティブ

Eclipse における パースペクティブとは、特定の機能に結びつけられた GUI ビューのことです。Eclipse 用の Authentic Desktop プラグインが Eclipse に統合されると、Authentic Desktop パースペクティブも自動的に作成されます。この パースペクティブは、編集ビュー、メニュー、入力ヘルパー、そしてサイドバーなど、Authentic Desktop の GUI 要素を含む パースペクティブとなります。

Authentic Desktop と関連付けられている種類のファイル(例えば .xml) が開かれた時、このファイルを Authentic Desktop パースペクティブで編集することができます。同様に、他の種類のファイルも Eclipse の別の パースペクティブで開くことができます。更に、アクティブなファイルに対しては、パースペクティブを変更することができ、異なる環境で同じファイルの編集を行うことができます。従って、パースペクティブには2つの利点があります:

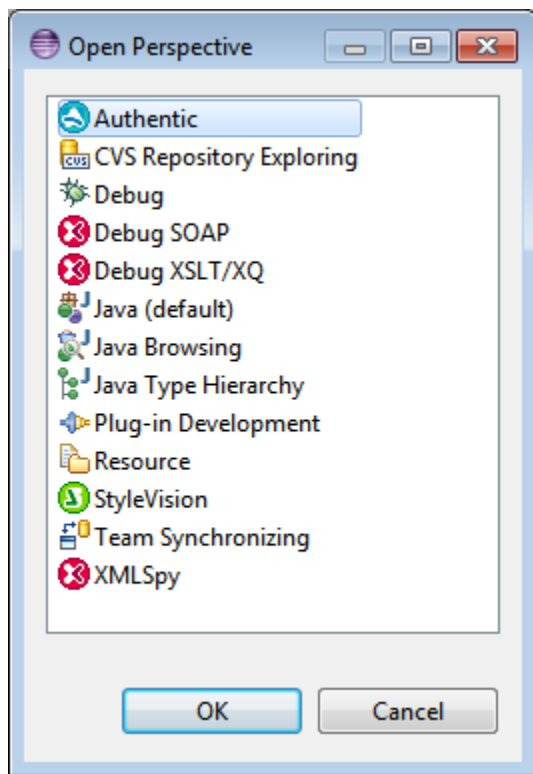
1. アクティブなファイルに対して、作業環境を素早く変更することができる。
2. 新たな開発環境を開くことなくファイルの切り替えを行うことができる(パースペクティブに結びつけられた環境も有効になります)。

Authentic Desktop パースペクティブを使う作業には以下のようなものがあります:

- Authentic Desktop パースペクティブへの切り替え
- Authentic Desktop パースペクティブの環境設定
- Authentic Desktop パースペクティブのカスタマイズ

Authentic Desktop パースペクティブへの切り替え

Eclipse にて、メニューコマンドから「**ウィンドウ | パースペクティブ | パースペクティブを開く | その他**」を選択します。表示されるダイアログボックス内で、下のスクリーンショット) select **Authentic Desktop**を選択して[開く]をクリックします。



空のドキュメント、またはアクティブなドキュメントに対して Authentic Desktop パースペクティブが開かれます。この方法でユーザーはパースペクティブの切り替えを行うことができます。目的のパースペクティブをパースペクティブを開くサブメニュー（のその他アイテムの上）に表示させることで、パースペクティブに素早くアクセスすることができます。

この設定はカスタマイズダイアログで行うことができます。ファイルが開かれた時、またはアクティブなドキュメントでパースペクティブを自動的に切り替えることもできます。そのファイルが最初に開かれたときは、ファイルの種類に関連付けられているアプリケーションのパースペクティブが自動的に開かれ、このファイルの種類に対して、このパースペクティブをデフォルトのパースペクティブとして割り当てることができる場合があります。この種類のファイルを開くたびに尋ねられなくなる場合は、今後このメッセージを表示しないにチェックを入れ「OK」をクリックします。

Authentic Desktop パースペクティブの環境設定

パースペクティブの優先順位には以下が含まれます: Authentic Desktop パースペクティブの環境設定、パースペクティブの設定には (i) 関連付けられた種類のファイルが開かれたときに、自動的にパースペクティブを切り替える（上の参照）(ii) 各 Authentic Desktop ツールバーを含めるオプションが含まれます。左のペインにあるパースペクティブのリストから Authentic Desktop を選択し、必要な設定を選択します。「OK」ボタンをクリックして完了します。

Authentic Desktop パースペクティブのカスタマイズ

カスタマイズオプションを使うことで、ショートカットやコマンドをパースペクティブに加えることができます。（以下のスクリーンショットにある）パースペクティブのカスタマイズダイアログにアクセスするには、（Authentic Desktop パースペクティブの場合は）パースペクティブをアクティブにして、メニューコマンドから「[ウィンドウ] | パースペクティブ | パースペクティブのカスタマイズ」を選択します。ツールバー可視性とメニュー可視性タブ内では、度のツールバーとメニューが表示されるかを指定することができます。アクションセットの使用可能タブでは、親メニューとツールバーにアクションセットを追加することができます。アクショングループ可用性タブでは、アクションセットを親メニューやツールバーへ追加することができます。アクショングループを有効にするには、対応するチェックボックスにチェックを入れてください。カスタマイズパースペクティブダイアログのショートカットタブでは、サブメニューのショートカットをセトすることができます。サブメニューエディタボックスから目的のサブメニューを選択し、ショートカットのカテゴリを選択し、パースペクティブに追加したショートカットにチェックを入れます。[適用 & 閉じる] をクリックしてカスタマイズを完了すると変更が反映されます。

Authentic Desktop メニューとツールバー

The **Authentic Desktop** メニューは Authentic Desktop (により認識されるドキュメントの型が現在 Eclipse 内で開かれていない場合でも関連性のあるコマンドが含まれています。Authentic Desktop のスタンダードバージョンではコマンドの一部が「ファイル」メニュー内に存在しています。

Authentic Desktop ツールバーには次のアイコンが含まれています (下のスクリーンショット参照)。



これは左側から (i) Authentic Desktop ヘルプを開く (ii) (**Authentic Desktop** メニューからアクセスすることの代替として Authentic Desktop コマンドにアクセス

13 プログラマーのレファレンス

Authentic Desktop は自動化サーバーです。自動化クライアントは直接にオブジェクト および自動化サーバーにより使用することができる機能にアクセスすることができます。この結果、自動化クライアントは、自動化サーバーにより使用することができるオブジェクトと機能に直接アクセスすることができます。ですから、Authentic Desktop の自動化クライアントは Authentic Desktop の XML 検証機能を使用することができます。開発者は Authentic Desktop の既製の機能を持つアプリケーションを使用してアプリケーションを拡張することができます。

Authentic Desktop のプログラムすることのできるオブジェクトは、COM API である Authentic Desktop のアプリケーション API を使用して、自動化クライアントで使用することができます。アプリケーション API のオブジェクト モデル および使用可能なすべてのオブジェクトの詳細はこのドキュメントで説明されています([アプリケーション API](#) のセクションを参照してください)。

API は、以下の環境の内部からアクセスすることができます:

- [スクリプトエディター](#)
- [IDE プラグイン](#)
- [外部プログラム](#)
- [ActiveX 統合](#)

これらの環境について下で簡単に説明がされています。

スクリプトエディター: Authentic Desktop 機能のカスタム化と変更

Authentic Desktop のインストールを機能を変更および追加することでカスタム化することができます。ユーザー入力のためのフォームを作成し、新しいメニューコマンド およびツールバーショートカットを含むユーザーインターフェイスを変更することもできます。これらすべての機能は、アプリケーション API のオブジェクトとの作動するスクリプトを書くことにより達成することができます。これらのタスクを効率的に実行するために、Authentic Desktop は、ビルドインのスクリプトエディターを搭載しています。機能の完全な説明と使用方法はこのドキュメントの [スクリプトエディター](#) セクションで説明されています。サポートされるプログラミング言語は JScript および VBScript です。

IDE プラグイン: Authentic Desktop のためのプラグインの作成

Authentic Desktop により自身のプラグインを作成し、Authentic Desktop に統合することができます。これを行うには、Authentic Desktop のプラグインのための特別なインターフェイスを使用します。プラグイン作成に関する詳細は、[Authentic Desktop IDE プラグイン](#) のセクションで説明されています。

アプリケーション オブジェクトは IDE プラグインにより実装され、アプリケーションにより呼び出される必要があるメソッドにマッピングされます。IDE プラグインの実装のために使用される 典型的な言語は、C# および C++ です。詳細に関しては、[Authentic Desktop IDE プラグイン](#) のセクションを参照してください。

外部プログラム

さらに、外部スクリプトともに Authentic Desktop を操作することができます。例えば、Authentic Desktop を指定した時刻に起動し、Authentic Desktop 内のファイルを開き、ファイルを検証し、印刷するスクリプトを作成することができます。外部スクリプトは、アプリケーション API を使用して、これらのタスクを実行します。アプリケーション API の詳細に関しては、[アプリケーション API](#) のセクションを参照してください。

Authentic Desktop から Application API を使用するには、Authentic Desktop のインスタンスが最初に開始される必要があります。実行方法は使用されるプログラミング言語により異なります。個別の言語に関する情報は、[プログラミング言語](#) のセクションを参照してください。

基本的には、Authentic Desktop は COM 登録を使用して起動されます。そして、Authentic Desktop インスタンスと関連する Application オブジェクトを返します。COM 設定により、既に実行中の Authentic Desktop に関連したオブジェクトが返されます。COM オブジェクトの作成と呼び出しをサポートするプログラミング言語を使用することができます。共通する項目は、下にリストされています。

- [JScript](#) および [VBScript](#) スクリプト ファイルは、シンプルな構文を持ち、COM オブジェクトにアクセスするためにデザインされています。DOS コマンド ラインから直接、または Windows Explorer をダブルクリックして実行することができます。これらのスクリプトはシンプルな自動化 タスクのために使用されます。
- [C#](#) は、広範囲の既存の機能を搭載した完全なプログラミング言語です。COM オブジェクトへのアクセスは、自動的にラップされた C# を使用します。
- C++ は、COM への直接的なコントロールを与えますが、他の言語より多量のコードが必要です。
- [Java](#): Altova 製品には、アプリケーション API をラップして、Java の完全な外観と使用感を与える、ネイティブの Java クラスが搭載されています。
- 役に立つ代替の他のプログラミング言語は、次のとおりです: アプリケーション、Perl、および Python のための Visual Basic

ActiveX 統合

Authentic Desktop ActiveX コントロールを使用してアプリケーション API にアクセスする特別のケースです。この機能は [Authentic Desktop 統合 パッケージ](#) がインストールされている場合のみ使用することができます。すべての ActiveX コントロールには、元になる機能のための対応する COM オブジェクトを返すプロパティがあります。マネージャーコントロールは Application オブジェクト、ドキュメントコントロール、Document オブジェクト、およびプレースホルダオブジェクトを与え、プロジェクトツリーを含む箇所では、Project オブジェクトを返します。これらのオブジェクトによりサポートされるメソッドは、[アプリケーション API のインターフェイスのセクション](#) で説明されています。ActiveX コントロール統合のコンテキストでは、このメソッドを使用する必要がなくこの点を考慮する必要があります。詳細に関しては [ActiveX 統合](#) を参照してください。

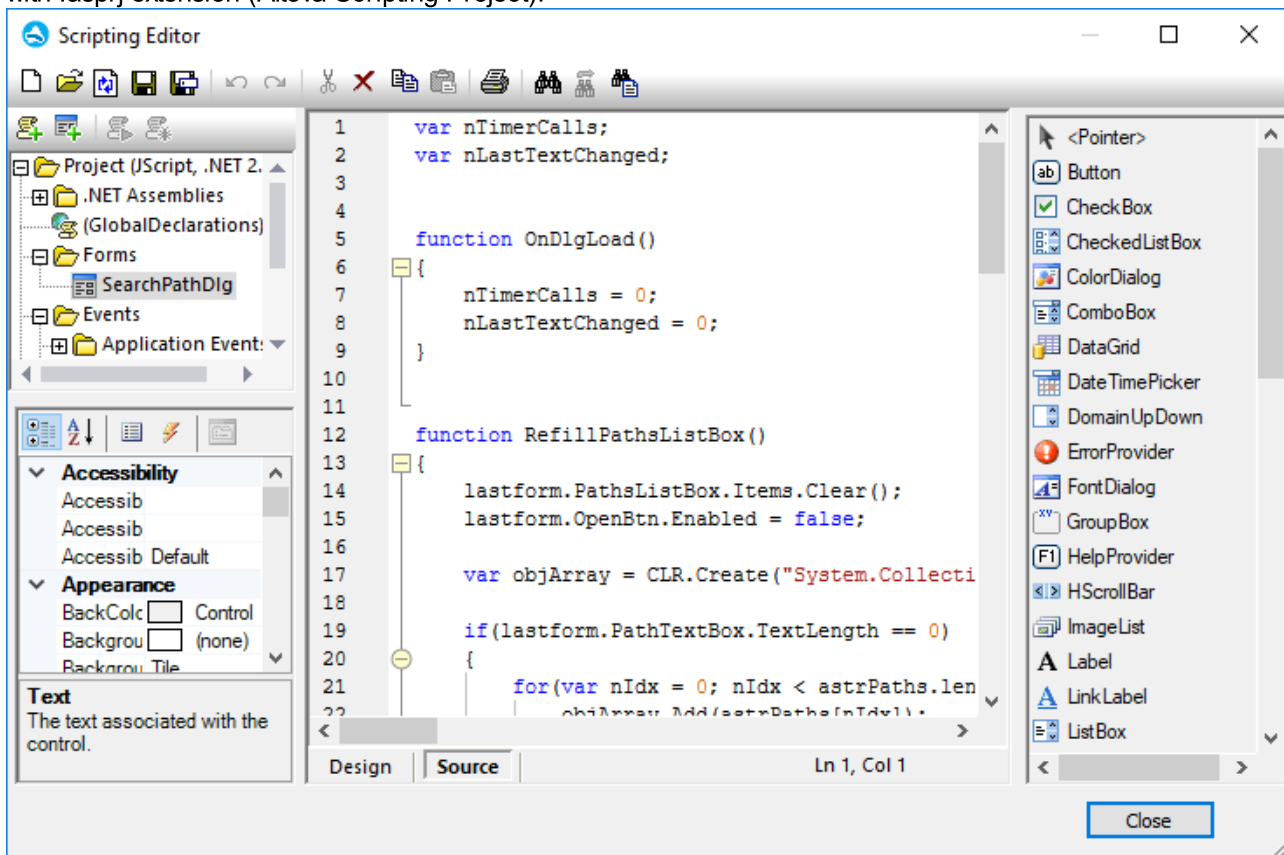
プログラマーのレファレンスについて

Authentic Desktop のプログラマーのレファレンスは、次のセクションから構成されています:

- [スクリプトエディタ](#): Authentic Desktop で使用することができるスクリプト環境のためのユーザー参照。
- [IDE プラグイン](#): Authentic Desktop のためのプラグイン作成の説明
- [アプリケーション API](#): アプリケーション API のためのレファレンス
- [ActiveX 統合](#): ActiveX コントロールを使用した、< % AUTH% > GUI および Authentic Desktop GUI 機能の統合の方法の案内および参照。

13.1 Scripting Editor

Scripting Editor is a development environment built into Authentic Desktop from where you can customize the functionality of Authentic Desktop with the help of JScript or VBScript scripts. For example, you can add a new menu item to perform a custom project task, or you can have Authentic Desktop trigger some behavior each time when a document is opened or closed. To make this possible, you create scripting projects—files with .asprj extension (Altova Scripting Project).



Scripting Editor

Scripting projects typically include one or several macros—these are programs that perform miscellaneous custom tasks when invoked. You can run macros either explicitly from a menu item (or a toolbar button, if configured), or you can set up a macro to run automatically whenever Authentic Desktop starts. The scripting environment also integrates with the Authentic Desktop COM API. For example, your VBScript or JScript scripts can handle application or document events such as starting or shutting down Authentic Desktop, opening or closing a project, and so on. Scripting projects can include Windows Forms that you can design visually, in a way similar to Visual Studio. In addition, several built-in commands are available that help you instantiate and use .NET classes from VBScript or JScript code.

Once your scripting project is complete, you can enable it either globally in Authentic Desktop, or only for specific projects.

Scripting Editor requires .NET Framework 2.0 or later to be installed before Authentic Desktop is installed.

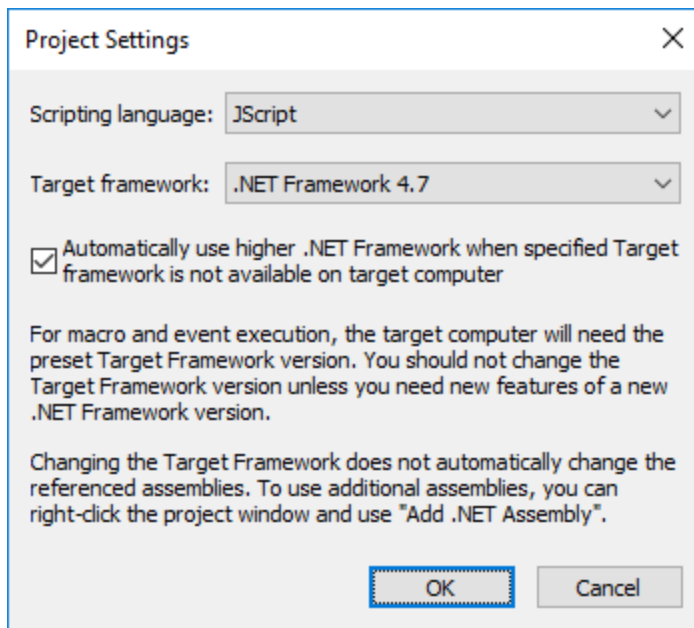
13.1.1 Creating a Scripting Project

All scripts and scripting information created in the Scripting Editor are stored in Altova Scripting Projects (.asprj files). A scripting project may contain macros, application event handlers, and forms (which can have their own event handlers). In addition, you can add global variables and functions to a "Global Declarations" script—this makes such variables and functions accessible across the entire project.

To start a new project, run the menu command **Tools | Scripting Editor**.

The languages supported for use in a scripting project are JScript and VBScript (not to be confused with Visual Basic, which is not supported). These scripting engines are available by default on Windows and have no special requirements to run. You can select a scripting language as follows:

1. Right-click the **Project** item in the upper-left pane, and select **Project settings** from the context menu.
2. Select a language (JScript or VBScript), and click **OK**.



From the Project settings dialog box above, you can also change the target .NET Framework version. This is typically necessary if your scripting project requires features available in a newer .NET Framework version. Note that any clients using your scripting project will need to have the same .NET Framework version installed (or a later compatible version).

By default, a scripting project references several .NET assemblies, like `System`, `System.Data`, `System.Windows.Forms`, and others. If necessary, you can import additional .NET assemblies, including assemblies from .NET Global Assembly Cache (GAC) or custom .dll files. You can import assemblies as follows:

1. Statically, by adding them manually to the project. Right-click **Project** in the top-left pane, and select **Add .NET Assembly** from the context menu.
2. Dynamically, at runtime, by calling the `CLR.LoadAssembly` command from the code.

You can create multiple scripting projects if necessary. You can save a scripting project to the disk, and then load it back into the Scripting Editor later. To do this, use the standard Windows buttons available in the toolbar: **New**, **Open**, **Save**, **Save As**. Once the scripting project has been tested and is ready for deployment,

you can load it into Authentic Desktop and run any of its macros or event handlers. For more information, see [Enabling Scripts and Macros](#).

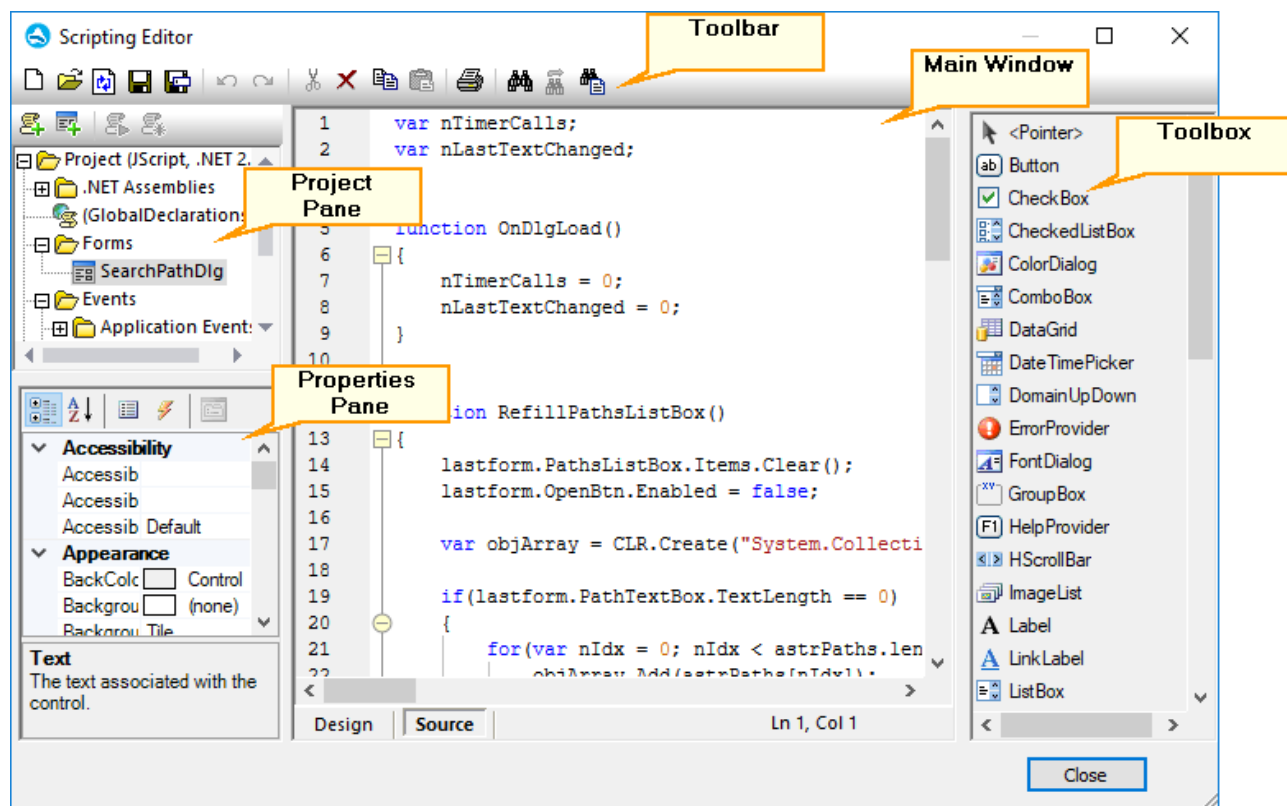
You can also find an example scripting project at the following path: **C:\Users\\Documents\Altova\Authentic2021\AuthenticExamples\SampleScripts.asprj**.

The next sections focus on the parts that your scripting project may need: global declarations, macros, forms, and events.

13.1.1.1 Overview of the Environment

The Scripting Editor consists of the following parts:

- Toolbar
- Project pane
- Properties pane
- Main window
- Toolbox



Toolbar

The toolbar includes standard Windows file management commands (**New**, **Open**, **Save**, **Save As**) commands and editor commands (**Copy**, **Cut**, **Delete**, **Paste**). When editing source code, the **Find** and **Replace** commands are additionally available, as well as the **Print** command.

Project pane

The project pane helps you view and manage the structure of the project. A scripting project consists of several components that can work together and may be created in any order:

- A *"Global Declarations"* script. As the name suggests, this script stores information available globally across the project. You can declare in this script any variables or functions that you need to be available in all forms, event handler scripts, and macros.
- *Forms*. Forms are typically necessary to collect user input, or provide some informative dialog boxes. A form is invoked by a call to it either within a function (in the Global Declarations script) or directly in a macro.
- *Events*. The "Events" folder displays Authentic Desktop application events provided by the COM API. To write a script that will be executed when an event occurs, double-click any event, and then type the handling code in the editor. The application events should not be confused with form events; the latter are handled at form level, as further detailed below.
- *Macros*. A macro is a script that can be invoked either on demand from a context menu or be executed automatically when Authentic Desktop starts. Macros do not have parameters or return values. A macro can access all variables and functions declared in the Global Declarations script and it can also display forms.

Right-click any of the components to see the available context menu commands and their shortcuts. Double-click any file (such as a form or a script) to open it in the main window.

The toolbar buttons provide the following quick commands:



New macro Adds a new macro to the project, in the **Macros** directory.



New form Adds a new form to the project, in the **Forms** directory.



Run macro Runs the selected macro.





Debug macro Runs the selected macro in debug mode.

Properties pane

The Properties pane is very similar to the one in Visual Studio. It displays the following:

- Form properties, when a form is selected
- Object properties, when an object in a form is selected
- Form events, when a form is selected
- Object events, when an object in a form is selected

To switch between the properties and events of the selected component, click the **Properties**  or **Events**  buttons, respectively.

The **Categorized**  and **Alphabetical**  icons display the properties or events either organized by category or organized in ascending alphabetical order.

When a property or event is selected, a short description of it is displayed at the bottom of the Properties pane.

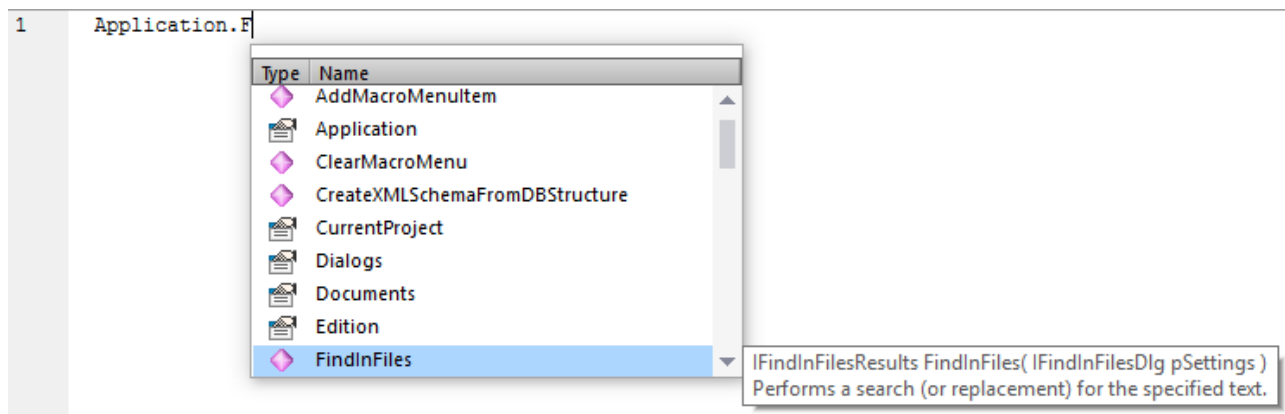
Main window

The main window is the working area where you can enter source code or modify the design of the form. When editing forms, you can work in two tabs: the **Design** tab and the **Source** tab. The **Design** tab shows the layout of the form, while the **Source** tab contains the source code such as handler methods for the form events.

The source code editor provides code editing aids such as syntax coloring, source code folding, highlighting of starting and ending braces, zooming, autocompletion suggestions, bookmarks.

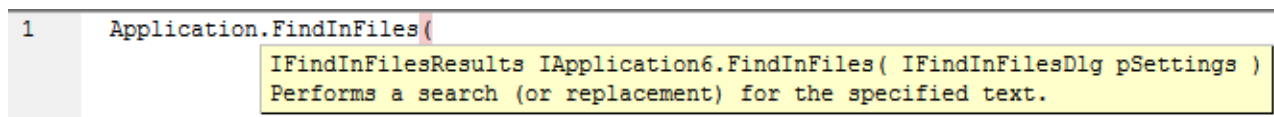
Autocompletion suggestions

JScript and VBScript are untyped languages, so autocompletion is limited to COM API names and Authentic Desktop built-in [commands](#). The full method or property signature is shown next to the autocompletion entry helper.



If names start with `objDocument`, `objProject`, `objXMLData`, or `objAuthenticRange`, members of the corresponding interface will be shown.

Placing the mouse over a known method or property displays its signature (and documentation if available), for example:



The auto-completion entry helper is normally shown automatically during editing, but it can also be obtained on demand by pressing **Ctrl+Space**.

Bookmarks

- To set or remove a bookmark, click inside a line, and then press **Ctrl+F2**
- To navigate to the next bookmark, press **F2**
- To navigate to the previous bookmark, press **Shift+F2**
- To delete all bookmarks, press **Ctrl+Shift+F2**

Zooming in/out

- To zoom in or out, hold the **Ctrl** key pressed and then press the "+" or "-" keys or rotate the mouse wheel.

Text view settings

To trigger text settings, right-click inside the editor, and select **Text View Settings** from the context menu.

Font settings

To change the font, right-click inside the editor, and select **Text View Font** from the context menu.

Toolbox

The Toolbox contains all the objects that are available for designing forms, such as buttons, text boxes, combo boxes, and so on.

To add a Toolbox item to a form:

1. Create or open a form and make sure that the **Design** tab is selected.
2. Click the Toolbox object (for example, **Button**), and then click at the location in the form where you wish to insert it. Alternatively, drag the object directly onto the form.

Some objects such as `Timer` are not added to the Form but are created in a tray at the bottom of the main window. You can select the object in the tray and set properties and event handlers for the object from the Properties pane. For an example of handling tray components from the code, see [Handling form events](#).

You can also add registered ActiveX controls to the form. To do this, right-click the Toolbox area and select **Add ActiveX Control** from the context menu.

13.1.1.2 Global Declarations

The "Global Declarations" script is present by default in any scripting project; you do not need to create it explicitly. Any variables or functions that you add to this script are considered global across the entire project. Consequently, you can refer to such variables and functions from any of the project's macros and events. The following is an example of a global declarations script that imports the `System.Windows.Forms` namespace into the project. To achieve that, the code below invokes the `CLR.Import` command built into Scripting Editor.

```
// import System.Windows.Forms namespace for all macros, forms and events:  
CLR.Import( "System.Windows.Forms" );
```

Note: Every time a macro is executed or an event handler is called, the global declarations are re-initialized.

13.1.1.3 Macros

Macros are scripts that contain JScript (or VBScript, depending on your project's language) statements, such as variable declarations and functions.

If your projects should use macros, you can add them as follows: right-click inside the Project pane, select **Add Macro** from the context menu, and then enter the macro's code in the main form. The code of a macro could be as simple as an alert, for example:

```
alert("Hello, I'm a macro!");
```



More advanced macros can contain variables and local functions. Macros can also contain code that invokes forms from the project. The listing below illustrates an example of a macro that shows a form. It is assumed that this form has already been created in the "Forms" folder and has the name "SampleForm", see also [Forms](#).

```
// display a form
ShowForm( "SampleForm" );
```

In the code listing above, `ShowForm` is a command built into Scripting Editor. For reference to other similar commands that you can use to work with forms and .NET objects, the [Built-in Commands](#).

You can add multiple macros to the same project, and you can designate any macro as "auto-macro". When a macro is designated as "auto-macro", it runs automatically when Authentic Desktop starts. To designate a macro as auto-macro, right-click it, and select **Set as Auto-Macro** from the context menu.

Only one macro can be run at a time. After a macro (or event) is executed, the script is closed and global variables lose their values.

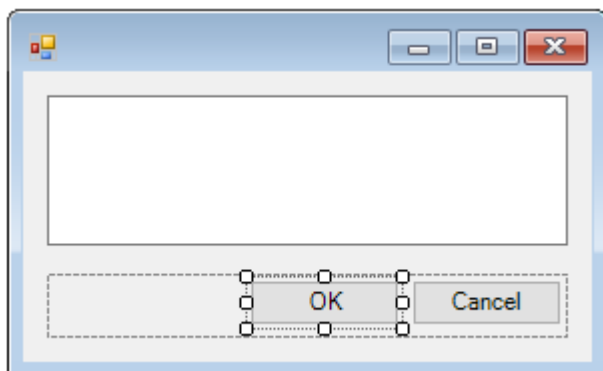
To run a macro directly in Script Editor, click **Run Macro** . To debug a macro using the Visual Studio debugger, click **Debug Macro** . For information about enabling and running macros in Authentic Desktop, see [Enabling Scripts and Macros](#).

13.1.1.4 Forms

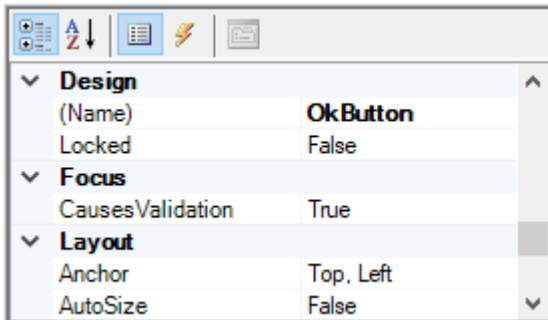
Forms are particularly useful if you need to collect input data from users or display data to users. A form can contain miscellaneous controls to facilitate this, such as buttons, check boxes, combo boxes, and so on.

To add a form, right-click inside the Project pane, and then select **Add Form** from the context menu. To add a control to a form, drag it from the Toolbox available to the right side of Scripting Editor and drop it onto the form.

You can change the position and size of the controls directly on the form, by using the handles that appear when you click any control, for example:




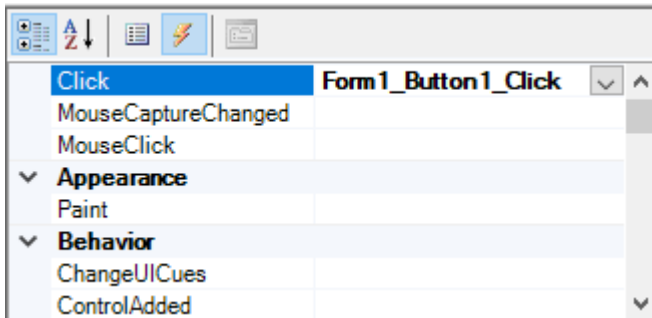
All form controls have properties that you can easily adjust in the Properties pane. To do this, first select the control on the form, and then edit the required properties in the Properties pane.



Handling form events

Each form control also exposes various events to which your scripting project can bind. For example, you might want to invoke some Authentic Desktop COM API method whenever a button is clicked. To create a function that binds to a form event, do the following:

1. In the Properties pane, click **Events** .
2. In the **Action** column, double-click the event where you need the method (for example, in the image below, the handled event is "Click").



You can also add handler methods by double-clicking a control on the form. For example, double-clicking a button in the form design generates a handler method for the "Click" event of that button.

Once the body of the handler method is generated, you can type code that handles this event, for example:

```
//Occurs when the component is clicked.
function MyForm_ButtonClick( objSender, e_EventArgs )
{
    alert("A button was clicked");
}
```

To display a work-in-progress form detached from the Scripting Editor, right-click the form, and select **Test Form** from the context menu. Note that the **Test Form** command just displays the form; the form's events (such as button clicks) are still disabled. To have the form react to events, call it from a macro, for example:

```
// Instantiate and display a form
ShowForm( "SampleForm" );
```

Accessing form controls

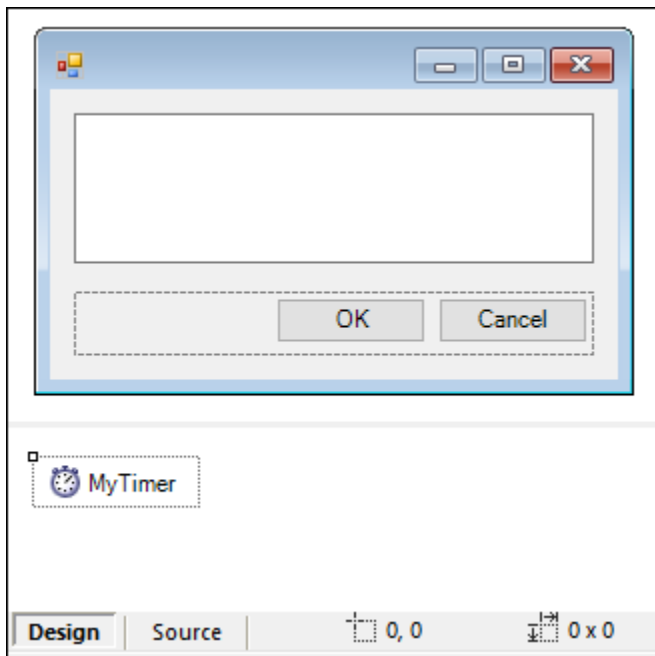
You can access any components on a form from your code by using field access syntax. For example, suppose there is a form designed as follows:

```
// MyForm
//   ButtonPanel
//     OkButton
//     CancelButton
//   TextEditor
//   AxMediaPlayer1
// TrayComponents
//   MyTimer
```

The code below shows how to instantiate the form, access some of its controls using field access syntax, and then display the form:

```
// Instantiate the form
var objForm = CreateForm("MyForm");
// Disable the OK button
objForm.ButtonPanel.OkButton.Enabled = false;
// Change the text of TextEditor
objForm.TextEditor.Text = "Hello";
// Show the form
objForm.ShowDialog();
```

When you add certain controls such as timers to the form, they are not displayed on the form; instead, they are shown as tray components at the base of the form design, for example:



To access controls from the tray, use the `GetTrayComponent` method on the form object, and supply the name of the control as argument. In this example, to get a reference to `MyTimer` and enable it, use the following code:

```
var objTimer = objForm.GetTrayComponent("MyTimer");
objTimer.Enabled = true;
```

For ActiveX Controls, you can access the underlying COM object via the `OCX` property:

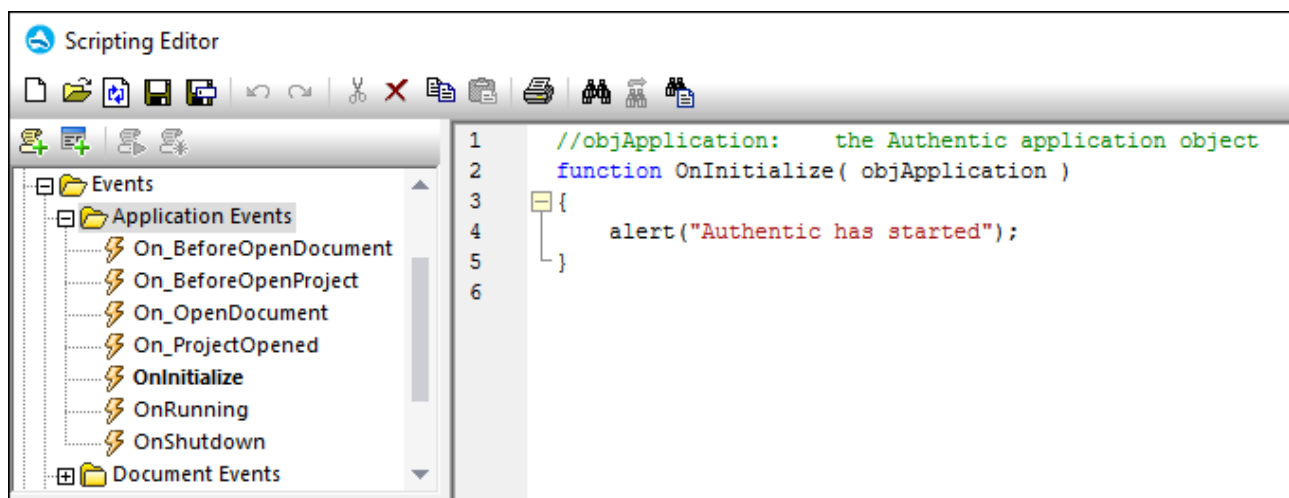
```
var ocx = lastform.AxMediaPlayer1.OCX; // get underlying COM object
ocx.enableContextMenu = true;
ocx.URL = "mms://apasf.apa.at/fm4_live_worldwide";
```

13.1.1.5 Events

Your scripting project may optionally include scripts that handle Authentic Desktop events such as opening, closing, or saving a document, starting or closing Authentic Desktop, adding an element to a diagram, and others. These events are provided by the Authentic Desktop COM API, and you can find them in the "Events" folder of your scripting project. Note that these events are Authentic Desktop-specific, as opposed to form events. Events are organized into folders as follows:

- Application Events
- Document Events
- AuthenticView Events
- GridView Events
- TextView Events

To create an event handler script, right-click an event, and select **Open** from the context menu (or double-click the event). The event handler script is displayed in the main window, where you can start editing it. For example, the event handler illustrated below displays an alert each time Authentic Desktop starts:



Note the following:

- The `alert` command is applicable to JScript. The VBScript equivalent is `MsgBox`. See also [alert](#).

- The name of the event handler function must not be changed; otherwise, the event handler script will not be called.
- In order for events to be processed, the **Process Events** check box must be selected when you enable the scripting project in Authentic Desktop. For more information, see [Enabling Scripts and Macros](#).

You can optionally define local variables and helper functions within event handler scripts, for example:

```
var local;

function OnInitialize( objApplication )
{
    local = "OnInitialize";
    Helper();
}

function Helper()
{
    alert("I'm a helper function for " + local);
}
```

13.1.1.6 JScript Programming Tips

Below are a few JScript programming tips that you may find useful while developing a scripting project in Authentic Desktop Scripting Editor.

Out parameters

Out parameters from methods of the .NET Framework require special variables in JScript. For example:

```
var dictionary =
CLR.Create("System.Collections.Generic.Dictionary<System.String, System.String>");
dictionary.Add("1", "A");
dictionary.Add("2", "B");

// use JScript method to access out-parameters
var strOut = new Array(1);
if ( dictionary.TryGetValue("1", strOut) ) // TryGetValue will set the out parameter
    alert( strOut[0] ); // use out parameter
```

Integer arguments

.NET Methods that require integer arguments should not be called directly with JScript number objects which are floating point values. For example, instead of:

```
var objCustomColor = CLR.Static("System.Drawing.Color").FromArgb(128,128,128);
```

use:

```
var objCustomColor =
CLR.Static("System.Drawing.Color").FromArgb(Math.floor(128),Math.floor(128),Math.floor(
128));
```

Iterating .NET collections

To iterate .NET collections, the JScript Enumerator as well as the .NET iterator technologies can be used, for example:

```
// iterate using the JScript iterator
var itr = new Enumerator( coll );
for ( ; !itr.atEnd(); itr.moveNext() )
    alert( itr.item() );

// iterate using the .NET iterator
var itrNET = coll.GetEnumerator();
while( itrNET.MoveNext() )
    alert( itrNET.Current );
```

.NET templates

.NET templates can be instantiated as shown below:

```
var coll = CLR.Create( "System.Collections.Generic.List<System.String>" );
```

or

```
CLR.Import( "System" );
CLR.Import( "System.Collections.Generic" );
var dictionary = CLR.Create( "Dictionary<String,Dictionary<String,String>>" );
```

.NET enumeration values

.NET enumeration values are accessed as shown below:

```
var enumValStretch = CLR.Static( "System.Windows.Forms.ImageLayout" ).Stretch;
```

Enumeration literals

The enumeration literals from the Authentic Desktop API can be accessed as shown below (there is no need to know their numerical value).

```
objExportXMIFileDlg.XMIType = eXMI21ForUML23;
```

13.1.1.7 Example Scripting Project

A demo project that illustrates scripting with Authentic Desktop is available at the following path: **C:\Users<user>\Documents\Altova\Authentic2021\AuthenticExamples\SampleScripts.asprj**. This scripting project consists of a few macros and a Windows form.

To load the scripting project into Scripting Editor:

1. On the **Tools** menu, click **Scripting Editor**.
2. Click **Open** and browse for the **SampleScripts.asprj** file from the path above.

The project contains several macros in the "Macros" directory.

Macro	Description
AddMacroMenu	<p>This macro adds a new menu item to Authentic Desktop, by invoking the <code>Application.AddMacroMenuItem</code> method of the COM API. The first argument of the <code>AddMacroMenuItem</code> method is the name of the macro to be added (in this example, "CloseAllButActiveDoc") and the second argument is the display text for the menu item.</p> <p>Whenever this macro is run, a new menu command called "CloseAllButActiveDoc") is added under the Tools menu. To clear macro menu items created previously, either restart Authentic Desktop or create a macro that calls the <code>Application.ClearMacroMenu</code> API method.</p>
CloseAllButActiveDocument	<p>When executed, the macro iterates through the currently open documents in Authentic Desktop and closes all of them, except for the active document.</p>
SearchPath	<p>This macro displays a form that lets users perform search for files within the current project. The form is available in the "Forms" directory, where you can view its design and the associated event handlers.</p> <p>The <code>GetAllPathsFromProject()</code> method returns all the file paths that belong to the currently opened project, as an array. The definition of this method is in the GlobalDeclarations script of the project. The <code>InsertStringInArrayUnique</code> method ensures that only unique paths are added to the array. Next, the form is initialized with CreateForm. Finally, the array is converted to a .NET type with the help of the CLR.Create method and the form is populated with the resulting <code>ArrayList</code> collection.</p> <p>The Open button of the form has a handler that calls the <code>Application.Documents.OpenFile</code> API method to open the currently selected file.</p>

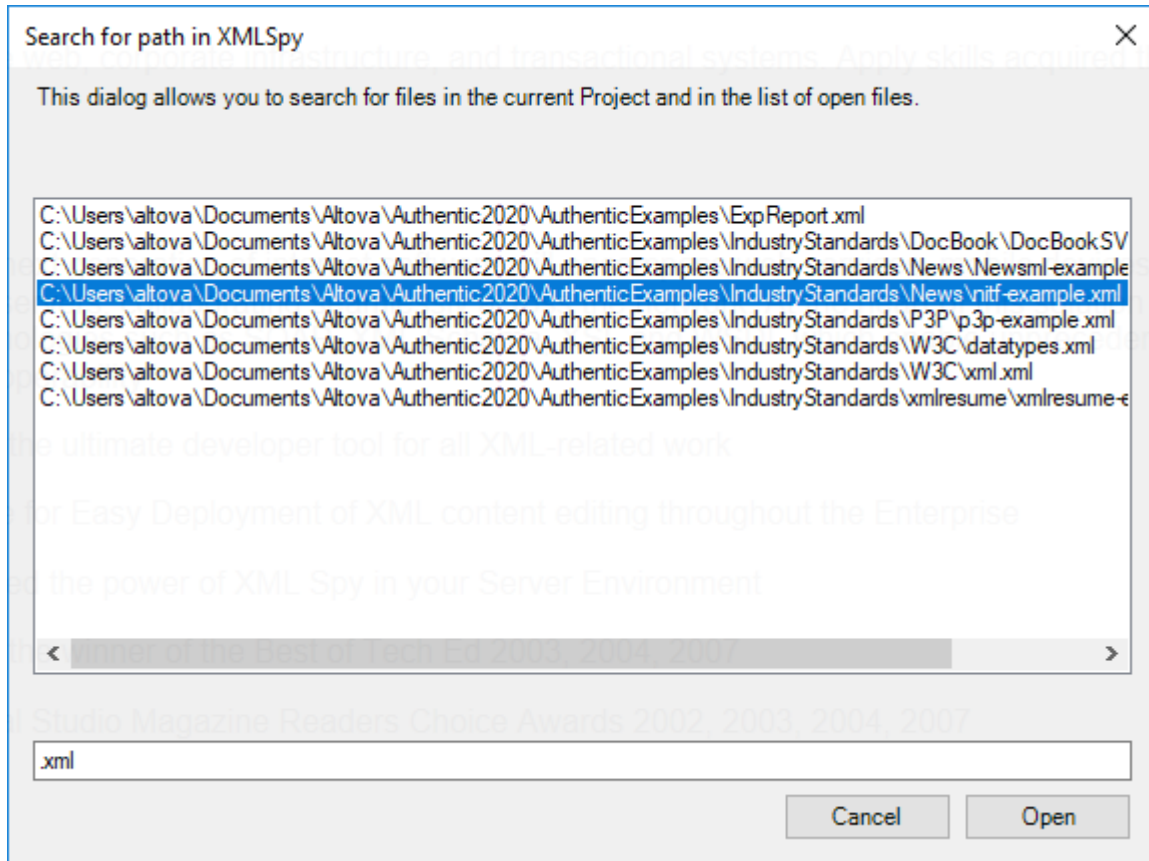
To enable the scripting project as global Authentic Desktop scripting project:

1. On the **Tools** menu, click **Options**.
2. Click the **Scripting** tab.
3. Under "Global scripting project file", click **Browse** and select the **SampleScripts.asprj** file from the path above.
4. This scripting project does not have auto-macros and application event handlers; therefore, you don't need to select either the **Run auto-macros...** or **Process events** check boxes.
5. Click **Apply**.

At this stage, several new menu items (one for each macro) become available under the **Tools | Macros** menu.

To run the "SearchPath" macro:

1. Open an Authentic Desktop project that contains several files (in this example, **C:\Users<user>\Documents\Altova\Authentic2021\AuthenticExamples\Examples.spp**).
2. On the **Tools** menu, click **Macros**, and then click **Search Path**.
3. Type the search term (in this example, ".xml").



As shown above, all file names that contain the search term are now listed. You can click any element in the list, and then click **Open** to display it in the main editor.

13.1.2 Built-in Commands

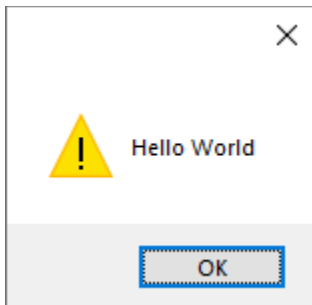
This section provides reference to all the commands you can use in the Authentic Desktop Scripting Editor.

- [alert](#)
- [confirm](#)
- [CLR.Create](#)
- [CLR.Import](#)
- [CLR.LoadAssembly](#)
- [CLR.ShowImports](#)

- [CLR.ShowLoadedAssemblies](#)
- [CLR.Static](#)
- [CreateForm](#)
- [doevents](#)
- [lastform](#)
- [prompt](#)
- [ShowForm](#)
- [watchdog](#)

13.1.2.1 alert

Displays a message box that shows a given message and the "OK" button. To proceed, the user will have to click "OK".



Signature

For JScript, the signature is:

```
alert(strMessage : String) -> void
```

For VBScript, the signature is:

```
MsgBox(strMessage : String) -> void
```

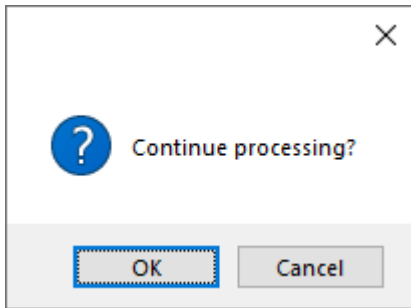
Example

The following JScript code displays a message box with the text "Hello World".

```
alert("Hello World");
```

13.1.2.2 confirm

Opens a dialog box that shows a given message, a confirmation button, and a cancel button. The user will have to click either "OK" or "Cancel" to proceed. Returns a Boolean that represents the user's answer. If the user clicked "OK", the function returns **true**; if the user clicked "Cancel", the function returns **false**.



Signature

```
confirm(strMessage : String) -> result : Boolean
```

Example (JScript)

```
if ( confirm( "Continue processing?" ) == false )  
    alert("You have cancelled this action");
```

Example (VBScript)

```
If ( confirm( "Continue processing?" ) = false ) Then  
    MsgBox("You have cancelled this action")  
End If
```

13.1.2.3 CLR.Create

Creates a new .NET object instance of the type name supplied as argument. If more than one argument is passed, the successive arguments are interpreted as the arguments for the constructor of the .NET object. The return value is a reference to the created .NET object

Signature

```
CLR.Create(strTypeNameCLR : String, constructor arguments ... ) -> object
```

Example

The following JScript code illustrates how to create instances of various .NET classes.

```
// Create an ArrayList  
var objArray = CLR.Create("System.Collections.ArrayList");  
// Create a ListViewItem  
var newItem = CLR.Create( "System.Windows.Forms.ListViewItem", "NewItemText" );  
// Create a List<string>  
var coll = CLR.Create( "System.Collections.Generic.List<System.String>" );  
// Import required namespaces and create a Dictionary object
```

```
CLR.Import( "System" );
CLR.Import( "System.Collections.Generic" );
var dictionary = CLR.Create( "Dictionary< String, Dictionary< String, String > >" );
```

13.1.2.4 CLR.Import

Imports a namespace. This is the scripting equivalent of C# `using` and VB.Net `imports` keyword. Calling `CLR.Import` makes it possible to leave out the namespace part in subsequent calls like `CLR.Create()` and `CLR.Static()`.

Note: Importing a namespace does not add or load the corresponding assembly to the scripting project. You can add assemblies to the scripting project dynamically (at runtime) in the source code by calling [CLR.LoadAssembly](#).

Signature

```
CLR.Import(strNamespaceCLR : String) -> void
```

Example

Instead of having to use fully qualified namespaces like:

```
if ( ShowForm( "FormName" ) == CLR.Static( "System.Windows.Forms.DialogResult" ).OK )
{
    var sName = lastform.textboxFirstName.Text + " " + lastform.textboxLastName.Text;
    CLR.Static( "System.Windows.Forms.MessageBox" ).Show( "Hello " + sName );
}
```

One can import namespaces first and subsequently use the short form:

```
CLR.Import( "System.Windows.Forms" );

if ( ShowForm( "FormName" ) == CLR.Static( "DialogResult" ).OK )
{
    var sName = lastform.textboxFirstName.Text + " " + lastform.textboxLastName.Text;
    CLR.Static( "MessageBox" ).Show( "Hello " + sName );
}
```

13.1.2.5 CLR.LoadAssembly

Loads the .NET assembly with the given long assembly name or file path. Returns Boolean **true** if the assembly could be loaded; **false** otherwise.

Signature

```
CLR.LoadAssembly(strAssemblyNameCLR : String, showLoadErrors : Boolean) -> result : Boolean
```

Example

The following JScript code attempts to set the clipboard text by loading the required assembly dynamically.

```
// set clipboard text (if possible)
// System.Windows.Clipboard is part of the PresentationCore assembly, so load this
// assembly first:
if ( CLR.LoadAssembly( "PresentationCore, Version=3.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35", true ) )
{
    var clipboard = CLR.Static( "System.Windows.Clipboard" );
    if ( clipboard != null )
        clipboard.SetText( "HelloClipboard" );
}
```

13.1.2.6 CLR.ShowImports

Opens a message box that shows the currently imported namespaces. The user will have to click "OK" to proceed.

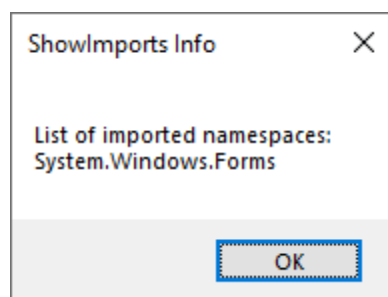
Signature

```
CLR.ShowImports() -> void
```

Example

The following JScript code first imports a namespace, and then displays the list of imported namespaces:

```
CLR.Import( "System.Windows.Forms" );
CLR.ShowImports();
```



13.1.2.7 CLR.ShowLoadedAssemblies

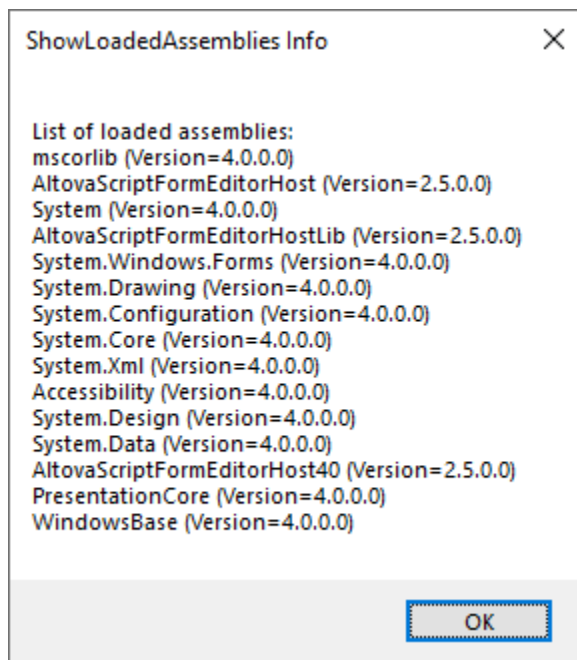
Opens a message box that shows the currently loaded assemblies. The user will have to click "OK" to proceed.

Signature

```
CLR.ShowLoadedAssemblies() -> void
```

Example

```
CLR.ShowLoadedAssemblies();
```



13.1.2.8 CLR.Static

Returns a reference to a static .NET object. You can use this function to get access to .NET types that have no instances and contain only static members.

Signature

```
CLR.Static(strTypeNameCLR : String) -> object
```

Example (JScript)

```
// Get the value of a .NET Enum into a variable
```

```
var enumValStretch = CLR.Static( "System.Windows.Forms.ImageLayout" ).Stretch

// Set the value of the Windows clipboard
var clipboard = CLR.Static( "System.Windows.Clipboard" );
clipboard.SetText( "HelloClipboard" );

// Check the buttons pressed by the user on a dialog box
if ( ShowForm( "FormName" ) == CLR.Static( "System.Windows.Forms.DialogResult" ).OK )
    alert( "ok" );
else
    alert( "cancel" );
```

13.1.2.9 CreateForm

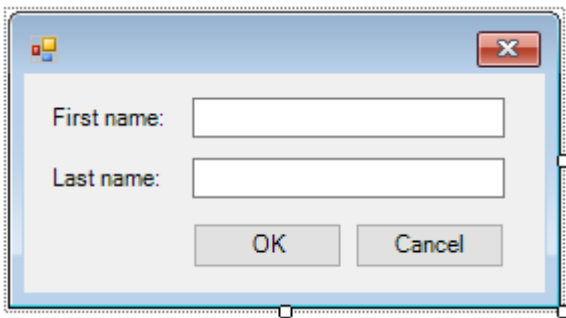
Instantiates the `Form` object identified by the name supplied as argument. The form must exist in the "Forms" folder of the scripting project. Returns the form object (`System.Windows.Forms.Form`) corresponding to the given name, or `null` if no form with such name exists.

Signature

```
CreateForm (strFormName : String) -> System.Windows.Forms.Form | null
```

Example

Let's assume that a form called "FormName" exists in the scripting project.



The following JScript code instantiates the form with some default values and displays it to the user.

```
var myForm = CreateForm( "FormName" );
if ( myForm != null )
{
    myForm.textboxFirstName.Text = "Daniela";
    myForm.textboxLastName.Text = "Heidegger";
    var dialogResult = myForm.ShowDialog();
}
```

The `dialogResult` can subsequently be evaluated as follows:

```
if ( dialogResult == CLR.Static( "System.Windows.Forms.DialogResult" ).OK )
```

```
    alert( "ok" );  
else  
    alert( "cancel" );
```

Note: The code above will work only if the **DialogResult** property of the "OK" and "Cancel" buttons is set correctly from the Properties pane (for example, it must be **OK** for the "OK" button).

13.1.2.10 doevents

Processes all Windows messages currently in the message queue.

Signature

```
doevents() -> void
```

Example (JScript)

```
for ( i=0; i < nLongLastingProcess; ++i )  
{  
    // do long lasting process  
  
    doevents(); // process Windows messages; give UI a chance to update  
}
```

13.1.2.11 lastform

This is a global field that returns a reference to the last form object that was created via `CreateForm()` or `ShowForm()`.

Signature

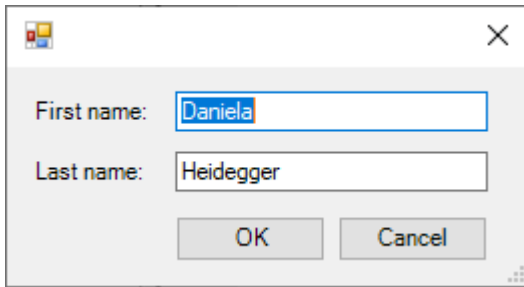
```
lastform -> formObj : System.Windows.Forms.Form
```

Example

The following JScript code shows the form "FormName" as a dialog box.

```
CreateForm( "FormName" );  
if ( lastform != null )  
{  
    lastform.textboxFirstName.Text = "Daniela";  
    lastform.textboxLastName.Text = "Heidegger";  
    var dialogResult = lastform.ShowDialog();  
}
```

The values of both `textbox` controls are initialized with the help of `lastform`.



13.1.2.12 prompt

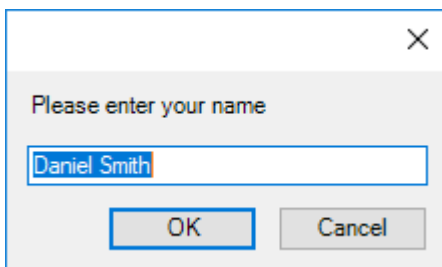
Opens a dialog box that shows a message and a textbox control with a default answer. This can be used to let the user input a simple string value. The return value is a string that contains the textbox value or null if the user selected "Cancel".

Signature

```
prompt(strMessage : String, strDefault : String) -> val : String
```

Example

```
var name = prompt( "Please enter your name", "Daniel Smith" );  
if ( name != null )  
    alert( "Hello " + name + "!" );
```



13.1.2.13 ShowForm

Instantiates a new form object from the given form name and immediately shows it as dialog box. The return value is an integer that represents the generated DialogResult (System.Windows.Forms.DialogResult). For the list of possible values, refer to the documentation of the DialogResult Enum (<https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.dialogresult?view=netframework-4.8>).

Signature

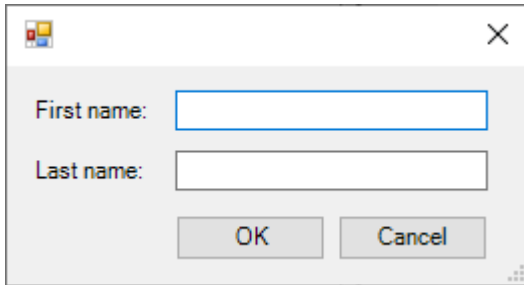
```
ShowForm(strFormName : String) -> result : Integer
```

Example

The following JScript code

```
var dialogResult = ShowForm( "FormName" );
```

Shows the form "FormName" as a dialog box:



The `DialogResult` can subsequently be evaluated, for example:

```
if ( dialogResult == CLR.Static( "System.Windows.Forms.DialogResult" ).OK )  
    alert( "ok" );  
else  
    alert( "cancel" );
```

Note: The code above will work only if the **DialogResult** property of the "OK" and "Cancel" buttons is set correctly from the Properties pane (for example, it must be **OK** for the "OK" button).

13.1.2.14 watchdog

Long running CPU-intensive scripts may ask the user if the script should be terminated. The `watchdog()` method is used to disable or enable this behavior. By default, the watchdog is enabled.

Calling `watchdog(true)` can also be used to reset the watchdog. This can be useful before executing long running CPU-intensive tasks to ensure they have the maximum allowed script processing quota.

Signature

```
watchdog(bEnable : boolean) -> void
```

Example

```
watchdog( false ); // disable watchdog - we know the next statement is CPU intensive  
but it will terminate for sure  
doCPUIntensiveScript();  
watchdog( true ); // re-enable watchdog
```

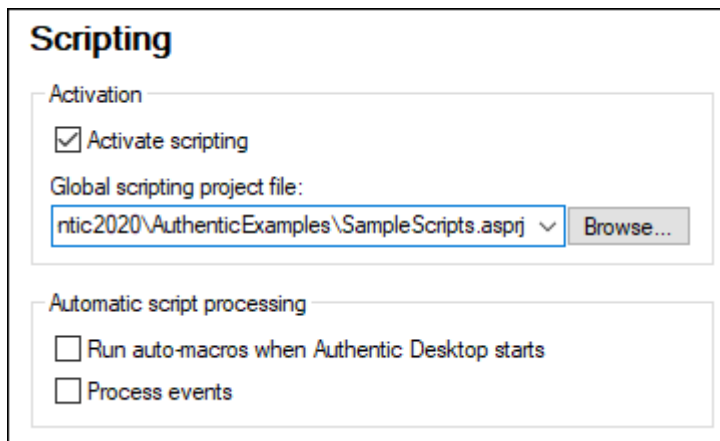
13.1.3 Enabling Scripts and Macros

Once a scripting project is complete and tested, you can use it in the following ways:

1. As the global scripting project for Authentic Desktop. This means that all the scripts and macros from the scripting project are available to Authentic Desktop.
2. At project level. This means that a reference to the .asprj file is saved together with the Authentic Desktop project. When the Authentic Desktop project is opened, its associated scripts and macros can be called.

To set a scripting project as global:

1. On the **Tools** menu, click **Options**.
2. Click the **Scripting** tab.
3. Select the **Activate scripting** check box and browse for the .asprj file to be used as global scripting project.



You can optionally enable the following additional script processing options:

Run auto-macros when Authentic Desktop starts	If you select this check box, any macros that were set as "Auto-macro" in the project will be triggered automatically when Authentic Desktop starts.
Process events	Select this check box if your scripts bind to any application events. Clear the check box to prevent the scripts from reacting to events.

To enable a scripting project at project level:

1. Open the project.
2. On the **Project** menu, click **Script Settings**.
3. Select the **Activate project scripts** check box and browse for the .asprj file.

The **Run-auto macros...** check box has the same meaning as already described above.

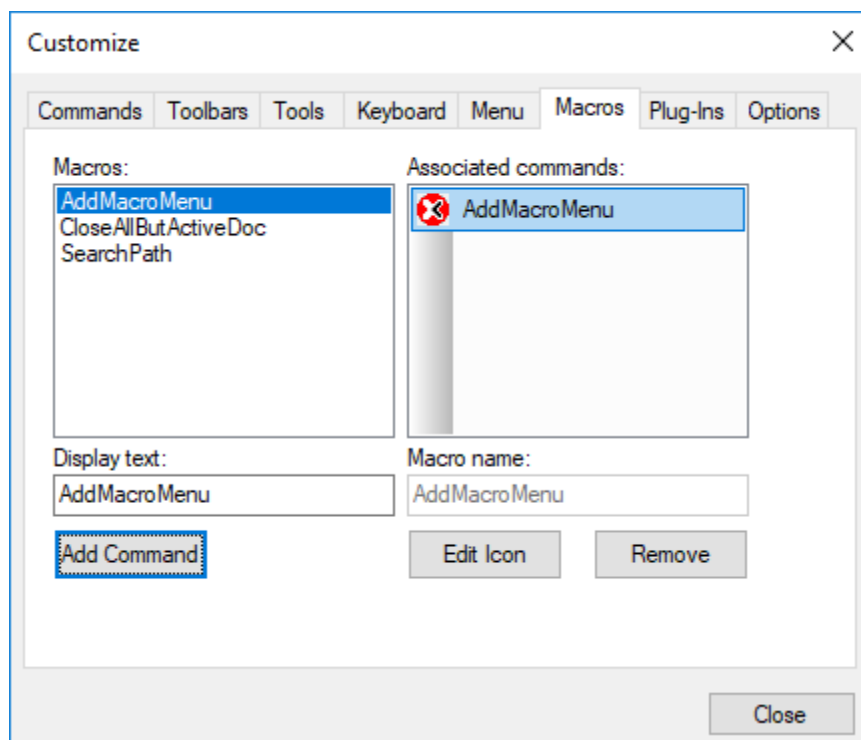
13.1.3.1 Running Macros

When a scripting project is active in Authentic Desktop, any macros available in that project are displayed in the **Tools | Macros** menu. Therefore, you can run a macro at any time, by triggering the respective menu command, for example **Tools | Macros | <SomeMacro>**.

Macros that were configured as auto-macros will run automatically whenever Authentic Desktop starts, provided that this behavior is enabled from options, as described in [Enabling Scripts and Macros](#).

For convenience, you can create toolbar buttons for macros, as follows:

1. On the **Tools** menu, click **Customize**.
2. Click the **Macros** tab. Any macros that are available at application level (in the *global* scripting project) are listed.
3. Click **Add Command**.



4. Optionally, click **Edit icon** and draw a new icon for the new macro. You can also assign a shortcut to the macro, from the **Keyboard** tab.
5. Drag the macro from the **Associated commands** pane onto the toolbar where you would like it to appear.

To remove a macro from a toolbar:

1. On the **Tools** menu, click **Customize**.
2. Click the **Macros** tab.
3. Drag the macro from the toolbar where it appears back into the **Associated commands** pane.

13.2 IDE Plugins

Authentic Desktop allows you to create your own IDE plug-ins and integrate them into Authentic Desktop.

Use plug-ins to:

- Configure your version of Authentic Desktop, add commands through menus, icons, buttons etc.
- React to events from Authentic Desktop.
- Run your specific code within Authentic Desktop with access to the complete Authentic Desktop API

Authentic Desktop expects your plug-in to implement the [IXMLSpyPlugin](#) interface. VB.NET, C# and C++ are the currently supported languages, and examples using these languages are included with your installation package and are located in the `Authentic2021\AuthenticExamples\IDEPlugin` folder of your Authentic Desktop installation.

Windows 7, 8, 10	C:/Users/<username>/Documents
------------------	-------------------------------

See [ATL sample files](#) for an example using C++.

13.2.1 Registration of IDE Plugins

Authentic Desktop maintains a specific key in the Registry where it stores all registered IDE plug-ins:

```
HKEY_CURRENT_USER\Software\Altova\XML Spy\Plugin
```

All values of this key are treated as references to registered plug-ins and must conform to the following format:

Value name:	ProgID of the plug-in
Value type:	must be REG_SZ
Value data:	CLSID of the component

Each time the application starts the values of the `Plugin` key is scanned, and the registered plug-ins are loaded.

Register plug-in manually

To register a plug-in manually, use the Customize dialog box of Authentic Desktop's **Tools** menu. Use the **Add Plug-In** button to specify the DLL that implements your plug-in. Authentic Desktop registers the DLL as a COM server and adds the corresponding entry in its `Plugin` key.

If you experience problems with manual registration, check whether the CLSID of your plug-in is correctly registered in the `Plugin` key. If the registration is incorrect, then the name of your plug-in DLL was probably not sufficiently unique. Use a different name or perform direct registration.

Register plug-in directly

A plug-in can be directly registered as an IDE plug-in by first registering the DLL and then adding the appropriate value to the `PLUGINS` key of Authentic Desktop. (This can be done, for example, during plug-in setup.) The new plug-in will be activated the next time Authentic Desktop is launched.

Creating plug-ins

Source code for sample plug-ins has been provided in the application's [\(My\) Documents folder](#): `Examples\IDEPlugin` folder. To build a plug-in from such source code, do the following:

1. Open the solution you want to build as a plug-in in Visual Studio.
2. Build the plug-in with the command in the Build menu.
3. The plug-in's DLL file will be created in the `Bin` or `Debug` folder. This DLL file is the file that must be added as a plug-in (see above).

Note: VB.NET, C# and C++ are the currently supported languages.

13.2.2 ActiveX Controls

ActiveX controls are supported. Any IDE PlugIn which is also an ActiveX control will be displayed in a Dialog Control Bar. A sample PlugIn that is also an ActiveX control is included in the `IDEPlugin` folder in the `Examples` folder of your application folder.

13.2.3 Configuration XML

The IDE plug-in allows you to change the user interface (UI) of Authentic Desktop. This is done by describing each separate modification using an XML data stream. The XML configuration is passed to Authentic Desktop using the [GetUIModifications](#) method of the `IXMLSpyPlugIn` interface.

The XML file containing the UI modifications for the IDE PlugIn, must have the following structure:

```
<ConfigurationData>
  <ImageFile>Path to image file</ImageFile>
  <Modifications>
    <Modification>
      ...
    </Modification>
    ...
  </Modifications>
</ConfigurationData>
```

You can define icons or toolbar buttons for new menu items that are added to the UI of Authentic Desktop by the plug-in. The path to the file containing the images is set using the `ImageFile` element. Each image must be 16x16 pixels using maximum 256 colors. The image references must be arranged from left to right in a single `<ImageFile>` element. The rightmost image index value is zero.

The `Modifications` element can have any number of `Modification` child elements. Each `Modification` element defines a specific change to the standard UI of Authentic Desktop. The modifications you can carry out are described in the next section below.

Structure of Modification elements

A `Modification` element has two child elements:

```
<Modification>
  <Action>Type of action</Action>
  <UIElement Type="Type of UI element" />
</Modification>
```

Valid values for the `Action` element are:

Add: to add the following UI element to Authentic Desktop

Hide: to hide the following UI element in Authentic Desktop

Remove: to remove the UI element from the "Commands" list box, in the customize dialog

Multiple modifications can be combined in an `Action` element, like this: "Add Hide"

The `UIElement` element defines any new or existing UI element and may be one of the the following types: toolbars, buttons, menus, or menu items. The `type` attribute specifies which of these types the UI element belongs to. The structure of `UIElement` is described in the sections below.

Common UIElement children

The `ID` and `Name` elements are defined for all types of UI element. In the case of some types, however, one of these elements is ignored. For example, `Name` is ignored for a separator.

```
<ID></ID>
<Name></Name>
```

If `UIElement` describes an existing element of the UI, the value of the `ID` element is predefined by Authentic Desktop. Normally these ID values are not known to the public. If the XML fragment describes a new part of the UI, then the ID is arbitrary and the value should be less than 1000. The `name` element sets the textual value. Existing UI elements can be identified just by name; for example, menus and menu items that have sub menus. For new UI elements, the `Name` element sets the caption (for example, the title of a toolbar) or the text of a menu item.

Toolbars and Menus

To define a toolbar it is necessary to specify the ID and/or the name of the toolbar. An existing toolbar can be specified using only the name or ID (if the latter is known). To create a **new** toolbar, both values must be set. The `type` attribute must have a value of `ToolBar`.

```
<UIElement Type="ToolBar">
  <ID>1</ID>
  <Name>TestPlugIn</Name>
</UIElement>
```

To specify an Authentic Desktop menu you need two parameters:

- The ID of the menu bar which contains the menu. If no XML documents are open in the main window, the menu bar ID is 128. If one or more XML documents are open, the menu bar ID is 129.
- The menu name. Menus do not have an associated ID value. The following example defines the "Edit" menu of the menu bar which is active, when at least one XML document is open:

```
<UIElement Type="Menu">
  <ID>129</ID>
  <Name>Edit</Name>
</UIElement>
```

An additional element is used if you want to create a new menu. The `Place` element defines the position of the new menu in the menu bar:

```
<UIElement Type="Menu">
  <ID>129</ID>
  <Name>PlugIn Menu</Name>
  <Place>12</Place>
</UIElement>
```

A value of `-1` for the `Place` element sets the new button or menu item at the end of the menu or toolbar.

Commands

If you add a new command (through a toolbar button or a menu item), the `UIElement` fragment can contain any of these sub elements:

```
<MacroName></MacroName>
<Info></Info>
<ImageID></ImageID>
```

If `MacroName` is specified, Authentic Desktop searches for a macro with the same name in the scripting environment and executes it each time this command is processed. The `Info` element contains a description string that is displayed in the status bar when the mouse pointer is over the associated command (button or menu item). `ImageID` defines the index of the icon in the image file. Note that all icons are stored in one image file.

To define a toolbar button, create an `UIElement` with this structure:

```
<UIElement Type="ToolBarItem">
  <!--don't reuse local IDs even the commands do the same-->
  <ID>5</ID>
  <Name>Open file from repository...</Name>
  <!--Set Place To -1 If this is the first button To be inserted-->
  <Place>-1</Place>
  <ImageID>0</ImageID>
  <ToolBarID>1</ToolBarID>
  <!--instead of the toolbar ID the toolbar name could be used-->
  <ToolBarName>TestPlugIn</ToolBarName>
</UIElement>
```

Additional elements to declare a toolbar button are `Place`, `ToolBarID` and `ToolBarName`. The `ToolBarID` and `ToolBarName` elements are used to identify the toolbar which contains the new or existing button. The textual value of `ToolBarName` is case-sensitive. The (`UIElement`) `type` attribute must be `ToolBarItem`.

To define a menu item, the elements `MenuID`, `Place` and `Parent` are available in addition to the standard elements used to declare a command. The content of the `MenuID` element can be either `128` or `129`. See the section "Toolbars and Menus" above for more information.

The `Parent` element is used to identify the menu where the new menu entry should be inserted. As sub menu items have no unique Windows ID, we need some other way to identify the parent of the menu item. We do this by setting the content of the `Parent` element to be the path to the menu item. The steps in the path are indicated by a colon. The pattern would be `ParentMenu:SubMenu`. If the menu has no parent (because it is not

a submenu), add a colon to the beginning of the name (see example below). The `type` attribute must be set to `MenuItem`.

The example below defines a menu item, where the containing menu is not a sub menu:

```
<UIElement Type="MenuItem">
  <!--the following element is a Local command ID-->
  <ID>3</ID>
  <Name>Open file from repository...</Name>
  <Place>-1</Place>
  <MenuID>129</MenuID>
  <Parent>:PlugIn Menu</Parent>
  <ImageID>0</ImageID>
</UIElement>
```

You can add toolbar separators and menus if the value of the `ID` element is set to 0.

13.2.4 ATL Sample Files

This section shows how to create a simple Authentic Desktop IDE plug-in DLL using ATL. You must know how to work with MS VisualStudio, ATL, and the wizards that generate new ATL objects. To access the API, the implementation imports the Type Library of Authentic Desktop. The code reads various properties and calls methods using the smart pointers provided by the `#import` statement of the code. In addition, the sample code uses the MFC class `CString` and ATL conversion macros such as `W2T`.

The broad steps to create an ATL DLL are as follows:

1. Open VisualStudio and select **File | New**.
2. Select the *Projects* tab.
3. Select *ATL COM AppWizard*, and type in a project name.
4. Select *Support for MFC* if you want to use MFC classes or if you want to create a project for the sample code.

Having created the project files you can add an ATL object to implement the `IXMLSpyPlugIn` interface:

1. Select **Insert | New ATL Object**.
2. Select *Simple Object* from the wizard. and click **Next**.
3. Type in a name for the object.
4. On the *Attributes* tab, select *Custom* for the type of interface and disable *Aggregation*.

These steps produce the skeleton code for the implementation of the IDE plug-in interface. See the following pages for information about how to modify the code and specify some basic functionality.

13.2.4.1 Interface description (IDL)

The IDL of the newly created ATL object contains a declaration for one COM interface.

- This interface declaration must be replaced by the declaration of `IXMLSpyPlugIn` as shown below.
- The IDL must also contain the definition of the `SPYUpdateAction` enumeration.

- Replace the generated default interface name (created by the wizard) with `IXMLSpyPlugIn` in the `coclass` declaration.

The IDL should then look something like the example code below. After creating the ATL object, you need to implement the IDE plug-in interface of Authentic Desktop.

```
import "oaidl.idl";
import "ocidl.idl";

// ----- please insert the following block into your IDL file -----
typedef enum {
    spyEnable = 1,
    spyDisable = 2,
    spyCheck = 4,
    spyUncheck = 8
} SPYUpdateAction;

// ----- end insert block ----

// ----- E.g. Interface entry automatically generated by the ATL wizard -----
// [
//     object,
//     uuid(AB7CD86A-8145-429A-A1F3-270692E08AFC),
//
//     helpstring("IXMLSpyPlugIn Interface")
//     pointer_default(unique)
// ]
// interface IXMLSpyPlugIn : IUnknown
// {
// };
// ----- end automatically generated Interface Entry

// ----- replace the Interface Entry (shown above) generated for you by the ATL wizard,
// with the following block -----

[
    odl,
    uuid(88F2A622-4B7E-42CD-8D04-3C0E5389DD85),
    helpstring("IXMLSpyPlugIn Interface")
]
interface IXMLSpyPlugIn : IUnknown
{
    HRESULT _stdcall OnCommand([in] long nID, [in] IDispatch* pXMLSpy);
    HRESULT _stdcall OnUpdateCommand([in] long nID, [in] IDispatch* pXMLSpy,
[out, retval] SPYUpdateAction* pAction);
    HRESULT _stdcall OnEvent([in] long nEventID, [in] SAFEARRAY(VARIANT)*
arrayParameters, [in] IDispatch* pXMLSpy, [out, retval] VARIANT* pReturnValue);
    HRESULT _stdcall GetUIModifications([out, retval] BSTR* pModificationsXML);
    HRESULT _stdcall GetDescription([out, retval] BSTR* pDescription);
};

// ----- end replace block -----

// ----- The code below is automatically generated by the ATL wizard and will look
// slightly different in your case -----

[
```

```

        uuid(24FE0D1B-3FC0-494E-B36E-1D4CE412B014),
        version(1.0),
        helpstring("XMLSpyIDEPlugInDLL 1.0 Type Library")
    ]
    library XMLSPYIDEPLUGINDLLLib
    {
        importlib("stdole32.tlb");
        importlib("stdole2.tlb");

        [
            uuid(3800E791-7F6B-4ACD-9E32-2AC184444501),
            helpstring("XMLSpyIDEPlugIn Class")
        ]
        coclass XMLSpyIDEPlugIn
        {
            [default] interface IXMLSpyPlugIn; // ----- define IXMLSpyPlugIn as the
default interface -----
        };
    };
};

```

13.2.4.2 Class definition

In the class definition of the ATL object, the following changes must be made:

- The class has to derive from `IXMLSpyPlugIn`
- The "Interface Map" needs an entry for `IXMLSpyPlugIn`
- The methods of the IDE plug-in interface must be declared

These changes can be made as shown below:

```

#ifndef __XMLSPYIDEPLUGIN_H_
#define __XMLSPYIDEPLUGIN_H_

#include "resource.h" // main symbols

////////////////////////////////////
// CXMLSpyIDEPlugIn
class ATL_NO_VTABLE CXMLSpyIDEPlugIn :
public CComObjectRootEx<CComSingleThreadModel>,
public CComCoClass<CXMLSpyIDEPlugIn, &CLSID_XMLSpyIDEPlugIn>,
public IXMLSpyPlugIn
{
public:
    CXMLSpyIDEPlugIn ()
    {
    }

    DECLARE_REGISTRY_RESOURCEID(IDR_XMLSPYIDEPLUGIN)
    DECLARE_NOT_AGGREGATABLE(CXMLSpyIDEPlugIn)

    DECLARE_PROTECT_FINAL_CONSTRUCT()

    BEGIN_COM_MAP(CXMLSpyIDEPlugIn)
        COM_INTERFACE_ENTRY(IXMLSpyPlugIn)
    END_COM_MAP()

```

```

// IXMLSpyIDEPlugIn
public:
    virtual HRESULT _stdcall OnCommand(long nID, IDispatch* pXMLSpy);
    virtual HRESULT _stdcall OnUpdateCommand(long nID, IDispatch* pXMLSpy,
SPYUpdateAction* pAction);
    virtual HRESULT _stdcall OnEvent(long nEventID, SAFEARRAY **arrayParameters,
IDispatch* pXMLSpy, VARIANT* pReturnValue);
    virtual HRESULT _stdcall GetUIModifications(BSTR* pModificationsXML);
    virtual HRESULT _stdcall GetDescription(BSTR* pDescription);
};

#endif // __XMLSPYIDEPLUGIN_H_

```

13.2.4.3 Implementation

The code below shows a simple implementation of an Authentic Desktop IDE plug-in. It adds a menu item and a separator (available with Authentic Desktop) to the Tools menu. Inside the OnUpdateCommand() method, the new command is only enabled when the active document is displayed using the Grid View. The command searches for the XML element which has the current focus, and opens any URL starting with "http://", from the textual value of the element.

```

////////////////////////////////////
// CXMLSpyIDEPlugIn

#import "XMLSpy.tlb"
using namespace XMLSpyLib;

HRESULT CXMLSpyIDEPlugIn::OnCommand(long nID, IDispatch* pXMLSpy)
{
    USES_CONVERSION;

    if(nID == 1)    {
        IApplicationPtr    ipSpyApp;

        if(pXMLSpy)    {
            if(SUCCEEDED(pXMLSpy->QueryInterface(__uuidof(IApplication), (void **)
&ipSpyApp)))    {
                IDocumentPtr    ipDocPtr = ipSpyApp->ActiveDocument;

                // we assume that grid view is active
                if(ipDocPtr)    {
                    IGridViewPtr    ipGridPtr = ipDocPtr->GridView;

                    if(ipGridPtr)    {
                        IXMLDataPtr    ipXMLData = ipGridPtr->CurrentFocus;

                        CString    strValue = W2T(ipXMLData->TextValue);

                        if(!strValue.IsEmpty() && (strValue.Left(7) == _T("http://")))
                            ::ShellExecute(NULL, _T("open"), W2T(ipXMLData->
>TextValue), NULL, NULL, SW_SHOWNORMAL);
                    }
                }
            }
        }
    }
}

```

```

    return S_OK;
}

HRESULT CXMLSpyIDEPlugIn::OnUpdateCommand(long nID, IDispatch* pXMLSpy, SPYUpdateAction*
pAction)
{
    *pAction = spyDisable;

    if(nID == 1) {
        IApplicationPtr ipSpyApp;

        if(pXMLSpy) {
            if(SUCCEEDED(pXMLSpy->QueryInterface(__uuidof(IApplication), (void **)
&ipSpyApp))) {
                IDocumentPtr ipDocPtr = ipSpyApp->ActiveDocument;

                // only enable if grid view is active
                if((ipDocPtr != NULL) && (ipDocPtr->CurrentViewMode == spyViewGrid))
                    *pAction = spyEnable;
            }
        }
    }

    return S_OK;
}

HRESULT CXMLSpyIDEPlugIn::OnEvent(long nEventID, SAFEARRAY **arrayParameters, IDispatch*
pXMLSpy, VARIANT* pReturnValue)
{
    return S_OK;
}

HRESULT CXMLSpyIDEPlugIn::GetUIModifications(BSTR* pModificationsXML)
{
    CComBSTR bstrMods = _T(" \
    <ConfigurationData> \
    <Modifications> ");
    // add "Open URL..." to Tools menu
    bstrMods.Append (_T(" \
        <Modification> \
        <Action>Add</Action> \
        <UIElement type=\"MenuItem\"> \
        <ID>1</ID> \
        <Name>Open URL...</Name> \
        <Place>0</Place> \
        <MenuID>129</MenuID> \
        <Parent>:Tools</Parent> \
        </UIElement> \
    </Modification>"));
    // add Seperator to Tools menu
    bstrMods.Append (_T(" \
        <Modification> \
        <Action>Add</Action> \
        <UIElement type=\"MenuItem\"> \
        <ID>0</ID> \
        <Place>1</Place> \
        <MenuID>129</MenuID> \
        <Parent>:Tools</Parent> \
        </UIElement> \
    </Modification>"));
}

```

```

    // finish modification description
    bstrMods.Append (_T(" \
        </Modifications> \
        </ConfigurationData>"));

    return bstrMods.CopyTo(pModificationsXML);
}

HRESULT CXMLSpyIDEPlugIn::GetDescription(BSTR* pDescription)
{
    CComBSTR bstrDescr = _T("ATL C++ XMLSpy IDE PlugIn;This PlugIn demonstrates the
implementation of a simple ATL DLL as a IDE PlugIn for XMLSpy.");
    return bstrDescr.CopyTo(pDescription);
}

```

13.2.5 IXMLSpyPlugIn

Methods

[OnCommand](#)
[OnUpdateCommand](#)
[OnEvent](#)
[GetUIModifications](#)
[GetDescription](#)

Description

If a DLL is added to Authentic Desktop as an IDE plug-in, it is necessary that it registers a COM component that answers to an `IXMLSpyPlugIn` interface with the reserved `uuid(88F2A622-4B7E-42CD-8D04-3C0E5389DD85)`. This is required for it to be recognized as a plug-in.

13.2.5.1 OnCommand

Declaration

```
OnCommand(nID as long, pXMLSpy as IDispatch)
```

Description

The `OnCommand()` method of the interface implementation is called each time a command added by the the IDE plug-in (menu item or toolbar button) is processed. `nID` stores the command ID defined by the `ID` element of the respective `UIElement`. `pXMLSpy` holds a reference to the dispatch interface of the Application object of Authentic Desktop.

Example

```

Public Sub IXMLSpyPlugIn_OnCommand(ByVal nID As Long, ByVal pXMLSpy As Object)
    If (Not (pXMLSpy Is Nothing)) Then
        Dim objDlg
        Dim objDoc As XMLSpyLib.Document
        Dim objSpy As XMLSpyLib.Application
        Set objSpy = pXMLSpy

        If nID = 3 Or nID = 5 Then
            Set objDlg = CreateObject("MSComDlg.CommonDialog")
            objDlg.Filter = "XML Files (*.xml)|*.xml|All Files (*.*)|*.*||"
            objDlg.FilterIndex = 1

```

```

objDlg.ShowOpen

If Len(objDlg.FileName) > 0 Then
    Set objDoc = objSpy.Documents.OpenFile(objDlg.FileName, False)
    Set objDoc = Nothing
End If
End If

If nID = 4 Or nID = 6 Then
    Set objDlg = CreateObject("MSComDlg.CommonDialog")
    objDlg.Filter = "All Files (*.*)|*.*||"
    objDlg.Flags = cdIOFNPathMustExist
    objDlg.ShowSave

    If Len(objDlg.FileName) > 0 Then
        Set objDoc = objSpy.ActiveDocument

        If Not (objDoc Is Nothing) Then
            objDoc.SetPathName objDlg.FileName
            objDoc.Save
            Set objDoc = Nothing
        End If
    End If
End If

Set objSpy = Nothing
End If
End Sub

```

13.2.5.2 OnUpdateCommand

Declaration

OnUpdateCommand(nID as long, pXMLSpy as IDispatch) as SPYUpdateAction

Description

The `OnUpdateCommand()` method is called each time the visible state of a button or menu item needs to be set. `nID` stores the command ID defined by the `ID` element of the respective `UIElement`. `pXMLSpy` holds a reference to the dispatch interface of the `Application` object.

Possible return values to set the update state are:

```

spyEnable      = 1
spyDisable     = 2
spyCheck       = 4
spyUncheck     = 8

```

Example

```

Public Function IXMLSpyPlugIn_OnUpdateCommand(ByVal nID As Long, ByVal pXMLSpy As Object)
As SPYUpdateAction
    IXMLSpyPlugIn_OnUpdateCommand = spyDisable

    If (Not (pXMLSpy Is Nothing)) Then
        Dim objSpy As XMLSpyLib.Application
        Set objSpy = pXMLSpy

        If nID = 3 Or nID = 5 Then
            IXMLSpyPlugIn_OnUpdateCommand = spyEnable
        End If
    End If

```

```

    If nID = 4 Or nID = 6 Then
        If objSpy.Documents.Count > 0 Then
            IXMLSpyPlugIn_OnUpdateCommand = spyEnable
        Else
            IXMLSpyPlugIn_OnUpdateCommand = spyDisable
        End If
    End If
End If
End If
End Function

```

13.2.5.3 OnEvent

Declaration

OnEvent(nEventID as long, arrayParameters as SAFEARRAY(VARIANT), pXMLSpy as IDispatch) as VARIANT

Description

OnEvent() is called each time an event is raised from Authentic Desktop.

Possible values for nEventID are:

On_BeforeStartEditing	= 1
On_EditingFinished	= 2
On_FocusChanged	= 3
On_Beforedrag	= 4
On_BeforeDrop	= 5
On_OpenProject	= 6
On_OpenDocument	= 7
On_CloseDocument	= 8
On_SaveDocument	= 9
On_DocEditDragOver	= 10
On_DocEditDrop	= 11
On_DocEditKeyDown	= 12
On_DocEditKeyUp	= 13
On_DocEditKeyPressed	= 14
On_DocEditMouseMove	= 15
On_DocEditButtonUp	= 16
On_DocEditButtonDown	= 17
On_DocEditContextMenu	= 18
On_DocEditPaste	= 19
On_DocEditCut	= 20
On_DocEditCopy	= 21
On_DocEditClear	= 22
On_DocEditSelectionChanged	= 23

<code>On_DocEditDragOver</code>	= 10
<code>On_BeforeOpenProject</code>	= 25
<code>On_BeforeOpenDocument</code>	= 26
<code>On_BeforeSaveDocument</code>	= 27
<code>On_BeforeCloseDocument</code>	= 28
<code>On_ViewActivation</code>	= 29
<code>On_DocEditKeyboardEvent</code>	= 30
<code>On_DocEditMouseEvent</code>	= 31
<code>On_BeforeValidate</code>	= 32
<code>On_BeforeShowSuggestions</code>	= 33
<code>On_ProjectOpened</code>	= 34
<code>On_Char</code>	= 35
<code>On_Initialize</code>	= 36
<code>On_Running</code>	= 37
<code>On_Shutdown</code>	= 38
<code>On_AuthenticBeforeSave</code>	= 39
<code>On_AuthenticContextMenuActivated</code>	= 40
<code>On_AuthenticLoad</code>	= 41
<code>On_AuthenticToolBarButtonClicked</code>	= 42
<code>On_AuthenticToolBarButtonExecuted</code>	= 43
<code>On_AuthenticUserAddedXMLNode</code>	= 44

The names of the events are the same as they appear in the Scripting Environment of Authentic Desktop. For IDE plug-ins the names used are immaterial. The events are identified using the ID value.

`arrayParameters` is an array which is filled with the parameters of the currently raised event. Order, type, and meaning of the single parameters are available through the scripting environment of Authentic Desktop. The events module of a scripting project contains predefined functions for all events prior to version 4.4. The parameters passed to the predefined functions are identical to the array elements of the `arrayParameters` parameter.

Events raised from the Authentic View of Authentic Desktop do not pass any parameters directly. An "event" object is used instead. The event object can be accessed through the `Document` object of the active document.

`pXMLSpy` holds a reference to the dispatch interface of the `Application` object of Authentic Desktop.

If the return value of `OnEvent()` is set, then neither the IDE plug-in nor an event handler inside of the scripting environment will get this event afterwards. Please note that all IDE plug-ins get/process the event before the Scripting Environment does.

13.2.5.4 GetUIModifications

Declaration

`GetUIModifications()` as String

Description

The `GetUIModifications()` method is called during initialization of the plug-in, to get the configuration XML data that defines the changes to the UI of Authentic Desktop. The method is called when the plug-in is loaded for the first time, and at every start of Authentic Desktop. See also [Configuration XML](#) for a detailed description how to change the UI.

Example

```
Public Function IXMLSpyPlugIn_GetUIModifications() As String
    ' GetUIModifications() gets the XML file with the specified modifications of
    ' the UI from the config.xml file in the plug-in folder
    Dim strPath As String
    strPath = App.Path

    If Len(strPath) > 0 Then
        Dim fso As New FileSystemObject
        Dim file As file
        Set file = fso.GetFile(strPath & "\config.xml")

        If (Not (file Is Nothing)) Then
            Dim stream As TextStream
            Set stream = file.OpenAsTextStream(ForReading)

            ' this replaces the token '**path**' from the XML file with
            ' the actual installation path of the plug-in to get the image file
            Dim strMods As String
            strMods = stream.ReadAll
            strMods = Replace(strMods, "**path**", strPath)

            IXMLSpyPlugIn_GetUIModifications = strMods
        Else
            IXMLSpyPlugIn_GetUIModifications = ""
        End If
    End If
End Function
```

13.2.5.5 GetDescription

Declaration

```
GetDescription() as String
```

Description

`GetDescription()` is used to define the description string for the plug-in entries visible in the Customize dialog box.

Example

```
Public Function IXMLSpyPlugIn_GetDescription() As String
    IXMLSpyPlugIn_GetDescription = "Sample Plug-in for XMLSpy;This Plug-in demonstrates
the implementation of a simple VisualBasic DLL as a Plug-in for XMLSpy."
End Function
```

13.3 Application API

The COM-based API of Authentic Desktop (also called the Application API from now on) enables other applications to use the functionality of Authentic Desktop. As a result, it is possible to automate a wide range of tasks, from validating an XML file to modifying complex XML content (with the [XMLData](#) interface).

Authentic Desktop and its Application API follow the common specifications for automation servers set out by Microsoft. It is possible to access the methods and properties of the Application API from common development environments, such as those using C#, C++, VisualBasic, and Delphi, and with scripting languages like JScript and VBScript.

Execution environments for the Application API

The Application API can be accessed from the following execution environments:

- External programs (described [below](#) and in the [Overview](#) part of this section)
- From within the built-in Scripting Editor of Authentic Desktop. For a description of the scripting environment, see the section, [Scripting Editor](#).
- Authentic Desktop allows you to create and integrate your own plug-ins into the application using a special interface for plug-ins. A description of how to create plug-ins is given in the section [IDE Plug-ins](#).
- Via an ActiveX Control, which is available if the [integration package](#) is installed. For more information, see the section [ActiveX Integration](#).

External programs

In the [Overview](#) part of this section, we describe how the functionality of Authentic Desktop can be accessed and automated from external programs.

Using the Application API from outside Authentic Desktop requires an instance of Authentic Desktop to be started first. How this is done depends on the programming language used. See the section, [Programming Languages](#), for information about individual languages.

Essentially, Authentic Desktop will be started via its COM registration. Then the `Application` object associated with the Authentic Desktop instance is returned. Depending on the COM settings, an object associated with an already running Authentic Desktop can be returned. Any programming language that supports creation and invocation of COM objects can be used. The most common of these are listed below.

- JScript and [VBScript](#) script files have a simple syntax and are designed to access COM objects. They can be run directly from a DOS command line or with a double click on Windows Explorer. They are best used for simple automation tasks.
- [C#](#) is a full-fledged programming language that has a wide range of existing functionality. Access to COM objects can be automatically wrapped using `C#`.
- C++ provides direct control over COM access but requires relatively larger amounts of code than the other languages.
- [Java](#): Altova products come with native Java classes that wrap the Application API and provide a full Java look-and-feel.
- Other programming languages that make useful alternatives are: Visual Basic for Applications, Perl, and Python.

Programming points

The following limitations must be considered in your client code:

- Be aware that if your client code crashes, instances of Authentic Desktop may still remain in the system.
- Don't hold references to objects in memory longer than you need them, especially those from the `XMLData` interface. If the user interacts between two calls of your client, then there is no guarantee that these references are still valid.
- Don't forget to disable dialogs if the user interface is not visible.
- See [Error handling in JScript](#) (and in [C#](#) and [Java](#)) for details of how to avoid annoying error messages.
- Free references explicitly if you are using C# or C++.

This documentation

This documentation section about the Application API is broadly divided into two parts.

- The first part consists of an [Overview](#), which describes the object model for the API and explains how the API is accessed via various [programming languages](#).
- The second part is a reference section ([Interfaces](#) and [Enumerations](#)) that contains descriptions of the interface objects of the Application API.

13.3.1 Overview

This overview of the Application API is organized as follows:

- [The Object Model](#) describes the relationships between the objects of the Application API.
- [Programming Languages](#) explains how the most commonly used programming languages (JScript, VBScript, C#, and Java) can be used to access the functionality of the Application API. Code listings from the example files supplied with your application package are used to describe basic mechanisms.

13.3.1.1 Object Model

The starting point for every application which uses the Application API is the [Application](#) object. This object contains general methods like import/export support and references to the open documents and any open project.

The `Application` object is created differently in various programming languages. In scripting languages such as JScript or VBScript, this involves calling a function which initializes the application's COM object. For examples, see the [Programming Languages](#) section.

XMLSpy.Application or AuthenticDesktop.Application

Authentic Desktop installs a TypeLibrary containing the XMLSpyLib. If this TypeLibrary has been added to the development environment (VB development environment, for example) then an object of the `Application` type can be created with:

```
Set objSpy = New XMLSpyLib.Application
```

If only Authentic Desktop is installed (and not XMLSpy), then

```
Set objSpy = GetObject("", "XMLSpy.Application")
```

does not work, because there won't be any object registered in the Registry with a ProgID of `XMLSpy.Application`. In this case, the registered object is `AuthenticDesktop.Application`.

The code listings in this documentation assume that both Authentic Desktop and XMLSpy have been installed. If, however, only Authentic Desktop has been installed, then please modify code fragments to take account of this difference.

The application object consists of the following parts:

1. Document collection and reference to the active document.
2. Reference to current project and methods for creating and opening projects.
3. Methods to support the export to and import from databases, text files, and Word documents.
4. URL management.
5. Methods for macro menu items.

Once you have created an Application object you can start using the functionality of Authentic Desktop. In most cases, you either open a project and access the documents from there or you directly open a document via the [Documents](#) interface.

13.3.1.2 Programming Languages

Programming languages differ in the way they support COM access. A few examples for the most frequently used languages (*links below*) will help you get started. The code listings in this section show how basic functionality can be accessed. The files in the `API` subfolder of the `Examples` folder can be used to test this functionality:

Windows 7、Windows 8、 Windows 10	C:\Users\<<username>\Documents\ Altova\Authentic\2021\%APPNAME%\Examples
------------------------------------	---

JScript

The JScript listings demonstrate the following basic functionality:

- [Start application or attach to a running instance](#)
- [Simple document access](#)
- [Iteration](#)
- [Error handling](#)
- [Events](#)

VBScript

VBScript is different than JScript only syntactically; otherwise it works in the same way. The listings below describe is an example of how VBScript can be used. For more information, refer to the [JScript examples](#).

- [Events](#): Shows how events are handled using VBScript.

C#

C# can be used to access the Application API functionality. The code listings show how to access the API for certain basic functionality.

- [Start Authentic Desktop](#): Starts Authentic Desktop, which is registered as an automation server, or activates the application if it is already running.
- [Open OrgChart.pxf](#): Locates one of the example documents installed with Authentic Desktop and opens it. If this document is already open it becomes the active document.
- [OnDocumentOpened Event On/Off](#): Shows how to listen to Authentic Desktop events. When turned on, a message box will pop up after a document has been opened.
- [Open ExpReport.xml](#): Opens another example document.
- [Toggle View Mode](#): Changes the view of all open documents between Browser View and Authentic View. The code shows how to iterate through open documents.
- [Validate](#): Validates the active document and shows the result in a message box. The code shows how to handle errors and COM output parameters.
- [Shutdown Authentic Desktop](#): Stops Authentic Desktop.

Java

The Authentic Desktop API can be accessed from Java code. [The Java sub-section of this section](#) explains how some basic Authentic Desktop functionality can be accessed from Java code. It is organized into the following sub-sections:

- [Mapping Rules for the Java Wrapper](#)
- [Example Java Project](#)
- [Application Startup and Shutdown](#)
- [Simple Document Access](#)
- [Iterations](#)
- [Use of Out-Parameters](#)
- [Event Handlers](#)

13.3.1.2.1 JScript

This section contains listings of JScript code that demonstrate the following basic functionality:

- [Start application or attach to a running instance](#)
- [Simple document access](#)
- [Iteration](#)
- [Error handling](#)
- [Events](#)

Example files

The code listings in this section are available in example files that you can test as is or modify to suit your needs. The JScript example files are located in the `JScript` subfolder of the API Examples folder:

Windows 7、Windows 8、	C:\Users\ <username>\Documents\<</username>
----------------------	--

Windows 10

Altova\Authentic\2021\%APPNAME%\Examples

The example files can be run in one of two ways:

- *From the command line:* Open a command prompt window, change the directory to the path above, and type the name of one of the example scripts (for example, `Start.js`).
- *From Windows Explorer:* In Windows Explorer, browse for the JScript file and double-click it.

The script is executed by Windows Script Host that is packaged with Windows operating system. For more information about Windows Script Host, refer to MSDN documentation (<https://msdn.microsoft.com>).

13.3.1.2.1.1 Start Application

The JScript below starts the application and shuts it down. If the COM object of the 32-bit Authentic Desktop cannot be found, the code attempts to get the COM object of the 64-bit application; otherwise, an error is thrown. If an instance of the application is already running, the running instance will be returned.

メモ 32-bit Authentic Desktop では登録された名前、またはCOM オブジェクトのプログラム識別子 (ProgId) は以下の通りです: `MapForce.ApplicationAuthenticDesktop.Application`。64-bit Authentic Desktop では名前は以下の通りです: `MapForce_x64.ApplicationAuthenticDesktop_x64.Application`。プログラムの呼び出しは自身のレジストリタイプ、またはグループ(32-bit または 64-bit)内の CLASSES レジストリエントリにアクセスすることにご注意ください。ですから標準のコマンドプロンプトと 64-bit Windows の Windows Explorer を使用してスクリプトを実行すると、64-bit Authentic Desktop を指す 64-bit レジストリエントリにアクセスされます。この理由のため、Authentic Desktop 32-bit と 64-bit の両方がインストールされている場合、32-bit Authentic Desktop を呼び出すために特別な処理が必要になります。例えば、スクリプトホストがプログラムを呼び出す場合以下を行います:

1. ディレクトリを `C:\Windows\SysWOW64` に変更します。
2. コマンドライン上で `wscript.exe` を入力し実行するスクリプトパスを以下のように入力します:

```
wscript.exe "C:\Users\...\Documents\Altova\Authentic2021\AuthenticExamples\API\JScript\start.js"
```

```
// Initialize application's COM object. This will start a new instance of the
// application and
// return its main COM object. Depending on COM settings, the main COM object of an
// already
// running application might be returned.

try {   objAuthentic = WScript.GetObject("", "AuthenticDesktop.Application");   }
catch(err) {}

if( typeof( objAuthentic ) == "undefined" )
{
  try   {   objAuthentic = WScript.GetObject("", "AuthenticDesktop_x64.Application")
  }
  catch(err)
  {
    WScript.Echo( "Can't access or create AuthenticDesktop.Application" );
    WScript.Quit();
  }
}
```

```
}  
  
// if newly started, the application will start without its UI visible. Set it to  
visible.  
objAuthentic.Visible = true;  
  
WScript.Echo(objAuthentic.Edition + " has successfully started. ");  
  
objAuthentic.Visible = false; // will shutdown application if it has no more COM  
connections  
//objAuthentic.Visible = true; // will keep application running with UI visible
```

The JScript code listed above is available in the sample file `Start.js` (see [Example Files](#)).

13.3.1.2.1.2 Simple Document Access

After you have started the application as shown in [Start Application](#), you will most likely want to programmatically open a document in order to work with it. The JScript code listing below illustrates how to open two documents from the Authentic Desktop Examples folder and set one of them as the active document.

```
// Locate examples via USERPROFILE shell variable. The path needs to be adapted to  
major release versions.  
objWshShell = WScript.CreateObject("WScript.Shell");  
majorVersionYear = objAuthentic.MajorVersion + 1998  
strExampleFolder = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\\My  
Documents\\Altova\\Authentic" + majorVersionYear + "\\AuthenticExamples\\";  
  
// Tell Authentic to open two documents. No dialogs  
objDoc1 = objAuthentic.Documents.OpenFile(strExampleFolder + "OrgChart.pxf", false);  
objAuthentic.Documents.OpenFile(strExampleFolder + "ExpReport.xml", false);  
  
// The document currently active can be easily located.  
objDoc2 = objAuthentic.ActiveDocument;  
  
// Let us make sure that the document is shown in Authentic view.  
objDoc2.SwitchViewMode(5); // SPYViewModes.spyViewAuthentic = 5  
  
// Now switch back to the document opened first  
objDoc1.SetActiveDocument();
```

The JScript code listed above is available in the sample file `DocumentAccess.js` (see [Example Files](#)).

13.3.1.2.1.3 Iteration

The JScript listing below shows how to iterate through the open documents. It is assumed that you have already started the application and opened some documents as shown in the previous sections.

```
// go through all open documents using a JScript Enumerator
```



```
bRequiresSaving = false;
for (var iterDocs = new Enumerator(objAuthentic.Documents); !iterDocs.atEnd());
iterDocs.moveNext()
{
    if (iterDocs.item().IsModified)
        bRequiresSaving = true;

    var strErrorText = new Array(1);
    var nErrorNumber = new Array(1);
    var errorData = new Array(1);

    if (!iterDocs.item().IsValid(strErrorText, nErrorNumber, errorData))
    {
        var text = strErrorText;
        // access that XMLData object only if filled in
        if (errorData[0] != null)
            text += "(" + errorData[0].Name + "/" + errorData[0].TextValue + ")";

        WScript.Echo("Document \"" + iterDocs.item().Name + "\" validation error[" +
nErrorNumber + "]: " + text);
    }
    else
    {
        // The COM call succeeded and the document is valid.
        WScript.Echo("Document \"" + iterDocs.item().Name + "\" is valid.");
    }
}

// go through all open documents using index-based access to the document collection
for (i = objAuthentic.Documents.Count; i > 0; i--)
    objAuthentic.Documents.Item(i).Close(false);
```

The JScript code listed above is available in the sample file `DocumentAccess.js` (see [Example Files](#)).

13.3.1.2.1.4 Error Handling

The Application API returns errors in two different ways:

- The `HRESULT` returned by every API method
- The `IErrorInfo` interface of the Application API

Every API method returns an `HRESULT`. This return value gives the caller information about errors during execution of the method. If the call was successful, the return value is `S_OK`. The `HRESULT` option is commonly used in C/C++ programs.

However, programming languages such as VisualBasic and scripting languages (and other high-level development environments) don't give the programmer access to the `HRESULT` return of a COM call. Such languages use the `IErrorInfo` interface, which is also supported by the Application API. If an error occurs, the Application API creates a new object that implements the `IErrorInfo` interface. The information provided by the `IErrorInfo` interface is imported by the development environment into its own error-handling mechanism.

For example, the JScript code listing below causes an error to be thrown by incorrectly declaring an array. Additional information about the error object is provided by its properties `number` and `description`.

```
try {
    var arr = new Array(-1);
}
catch (err) {
    WScript.Echo("Error : (" + (err.number & 0xffff) + ") " + err.description);
}
```

13.3.1.2.1.5 Events

COM specifies that a client must register itself at a server for callbacks using the connection point mechanism. The automation interface for XMLSpy defines the necessary event interfaces. The way to connect to those events depends on the programming language you use in your client. The following code listing shows how this is done using JScript.

The method `WScript.ConnectObject` is used to receive events.

```
// The event-handler function
function DocEvent_OnBeforeCloseDocument(objDocument)
{
    WScript.Echo("Received event - before closing document");
}

// Create or connect to XMLSpy (or Authentic Desktop)
try
{
    // Create the environment and XMLSpy (or Authentic Desktop)
    objWshShell = WScript.CreateObject("WScript.Shell");
    objFSO = WScript.CreateObject("Scripting.FileSystemObject");
    objSpy = WScript.GetObject("", "XMLSpy.Application");

    // If only Authentic Desktop is installed (and XMLSpy is not installed) use:
    // objSpy = WScript.GetObject("", "AuthenticDesktop.Application")
}
catch(err)
{ WScript.Echo ("Can't create WScript.Shell object or XMLSpy"); }

// Create document object and connect to its events
objSpy.Visible = true;
majorVersionYear = objSpy.MajorVersion + 1998
docPath = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\\Documents\\Altova\\
XMLSpy" + majorVersionYear + "\\Examples\\ExpReport.xml";
objDoc = objSpy.Documents.OpenFile (docPath, false);
WScript.ConnectObject(objDoc, "DocEvent_");

// Keep running while waiting for the event
// In the meanwhile close this document in XMLSpy (or Authentic Desktop) manually
WScript.Echo ("Sleeping for 10 seconds ...");
WScript.Sleep (10000);

objDoc = null;
WScript.Echo ("Stopped listening for event");
objSpy.Quit();
```

13.3.1.2.2 VBScript

VBScript is syntactically different than JScript but works in the same way. This section contains a listing showing [how events are used with VBScript](#) and an [example](#).

For information about other functionality, refer to the JScript examples listed below:

- [Start application or attach to a running instance](#)
- [Simple document access](#)
- [Iteration](#)
- [Error handling](#)

13.3.1.2.2.1 Events

COM specifies that a client must register itself at a server for callbacks using the connection point mechanism. The automation interface for XMLSpy defines the necessary event interfaces. The way to connect to those events depends on the programming language you use in your client. The following code listing shows how this is done using VBScript.

The method `WScript.ConnectObject` is used to receive events.

To run this code, paste it into a file with `.vbs` extension, and either double-click in Windows Explorer, or run it from a command prompt.

```
' the event handler function
Function DocEvent_OnBeforeCloseDocument(objDocument)
    Call WScript.Echo("received event - before closing document")
End Function

' create or connect to XmlSpy
Set objWshShell = WScript.CreateObject("WScript.Shell")
Set objFSO = WScript.CreateObject("Scripting.FileSystemObject")
Set objSpy = WScript.GetObject("", "XMLSpy.Application")
' If only Authentic is installed (and XMLSpy is not installed) use:
' Set objSpy = WScript.GetObject("", "AuthenticDesktop.Application")
' If only XMLSpy 64-bit is installed, use:
' Set objSpy = WScript.GetObject("", "XMLSpy_x64.Application")

' create document object and connect to its events
objSpy.Visible = True

' Find out user's personal folder and locate one of the installed examples.
personalFolder = objWshShell.ExpandEnvironmentStrings("%UserProfile%")
majorVersionYear = objSpy.MajorVersion + 1998
xmlspyExamplesFolder = personalFolder & "\\Documents\\Altova\\XMLSpy" & majorVersionYear
& "\\Examples\\"
docPath = xmlspyExamplesFolder & "ExpReport.xml"

' open a document
Set objDoc = objSpy.Documents.OpenFile(docPath, False)
Call WScript.ConnectObject(objDoc, "DocEvent_")
```

```
' keep running while waiting on the event
' in the meantime close the document in XMLSPY manually
Call WScript.Echo ("sleeping for 10 seconds ...")
Call WScript.Sleep (10000)

Set objDoc = Nothing
Call WScript.Echo ("stopped listening for event")
Call objSpy.Quit
```

メモ 32-bit Authentic Desktop では、登録された名前、または COM オブジェクトのプログラム識別子 (ProgId) は以下の通りです: `MapForce.ApplicationAuthenticDesktop.Application`。64-bit Authentic Desktop では、名前は以下の通りです: `MapForce_x64.ApplicationAuthenticDesktop_x64.Application`。プログラムの呼び出しは自身のレジストリタイプ、またはグループ (32-bit または 64-bit) 内の CLASSES レジストリエントリにアクセスすることにご注意ください。ですから、標準のコマンドプロンプトと 64-bit Windows の Windows Explorer を使用してスクリプトを実行すると、64-bit Authentic Desktop を指す 64-bit レジストリエントリにアクセスされます。この理由のため、Authentic Desktop 32-bit と 64-bit の両方がインストールされている場合、32-bit Authentic Desktop を呼び出すために特別な処理が必要になります。例えば、スクリプトホストがプログラムを呼び出す場合以下を行います:

1. ディレクトリを `C:\Windows\SysWOW64` に変更します。
2. コマンドライン上で `wscript.exe` を入力し実行するスクリプトパスを以下のように入力します:

```
wscript.exe "C:\Users\...\
\Documents\Altova\Authentic2021\AuthenticExamples\API\JScript\start.js"
```

13.3.1.2.2.2 Example: Using Events

Authentic View supports event connection on a per-object basis. Implementation of this feature is based on COM connection points and is available in environments that support this mechanism.

The following example is a VBScript code example that shows how to use events from within a VBScript project.

```
' -----
' VBScript example that demonstrates how to use events.
' -----

' Event handler for OnSelectionChanged event of AuthenticView
Function AuthenticViewEvent_OnSelectionChanged(objAuthenticRange)
    If objAuthenticRange.FirstTextPosition <> objAuthenticRange.LastTextPosition Then
        Call WScript.Echo("Selection: " & objAuthenticRange.Text & vbNewLine &
vbNewLine & "Close this dialog.")
    Else
        Call WScript.Echo("Cursor position: " & objAuthenticRange.FirstTextPosition &
vbNewLine & vbNewLine & "Close this dialog.")
    End If
End Function

' Start/access XMLSpy and connect to its automation interface.
Set WshShell = WScript.CreateObject("WScript.Shell")
Set objSpy = GetObject("", "XMLSpy.Application")
' Make the UI of XMLSpy visible.
```

```
objSpy.Visible = True

' Find out user's personal folder and locate one of the installed XMLSpy examples.
personalFolder = WshShell.ExpandEnvironmentStrings("%UserProfile%")
majorVersionYear = objSpy.MajorVersion + 1998
xmlspyExamplesFolder = personalFolder & "\Documents\Altova\XMLSpy" & majorVersionYear
& "\Examples\"
docPath = xmlspyExamplesFolder & "ExpReport.xml"

' Create object to access windows file system and test if the our document exists.
Set fso = CreateObject("Scripting.FileSystemObject")
If fso.FileExists(docPath) Then
    ' open the document
    Call objSpy.Documents.OpenFile(docPath, False)
    set objDoc = objSpy.ActiveDocument

    ' switch active document to authentic view
    objDoc.SwitchViewMode 4 ' spyViewAuthentic

    ' Register for connection point events on the authentic view of the active
document.
    ' Any function with a valid event name prefixed with "AuthenticViewEvent_" will
    ' be called when the corresponding event gets triggered on the specified object.
    set objView = objDoc.AuthenticView
    Call WScript.ConnectObject(objView, "AuthenticViewEvent_")
    Call WScript.Echo("Events are connected." & vbNewLine & vbNewLine & "Now set or
move the cursor in XMLSpy." & vbNewLine & vbNewLine & "Close this dialog to shut down
XMLSpy.")

    ' To disconnect from the events delete the reference to the object.
    set objView = Nothing
Else
    Call WScript.Echo("The file " & docPath & " does not exist.")
End If

' shut down XMLSpy when this script ends
objSpy.Visible = False
```

13.3.1.2.3 C#

The C# programming language can be used to access the Application API functionality. You could use Visual Studio 2010/2012/2013/2015/2017/2019 to create the C# code, saving it in a Visual Studio project. Create the project as follows:

1. In Microsoft Visual Studio, add a new project using **File | New | Project**.
2. Add a reference to the Authentic Desktop Type Library by clicking **Project | Add Reference**. The Add Reference dialog appears. Browse for the Authentic Desktop Type Library component, which is located in the Authentic Desktop application folder, and add it.
3. Enter the code you want.
4. Compile the code and run it.

Example C# project

Your Authentic Desktop package contains an example C# project, which is located in the `API\C#` subfolder of the `Examples` folder :

Windows 7、Windows 8、 Windows 10	C:\Users\<>username>\Documents\ Altova\Authentic\2021\%APPNAME%\Examples
------------------------------------	---

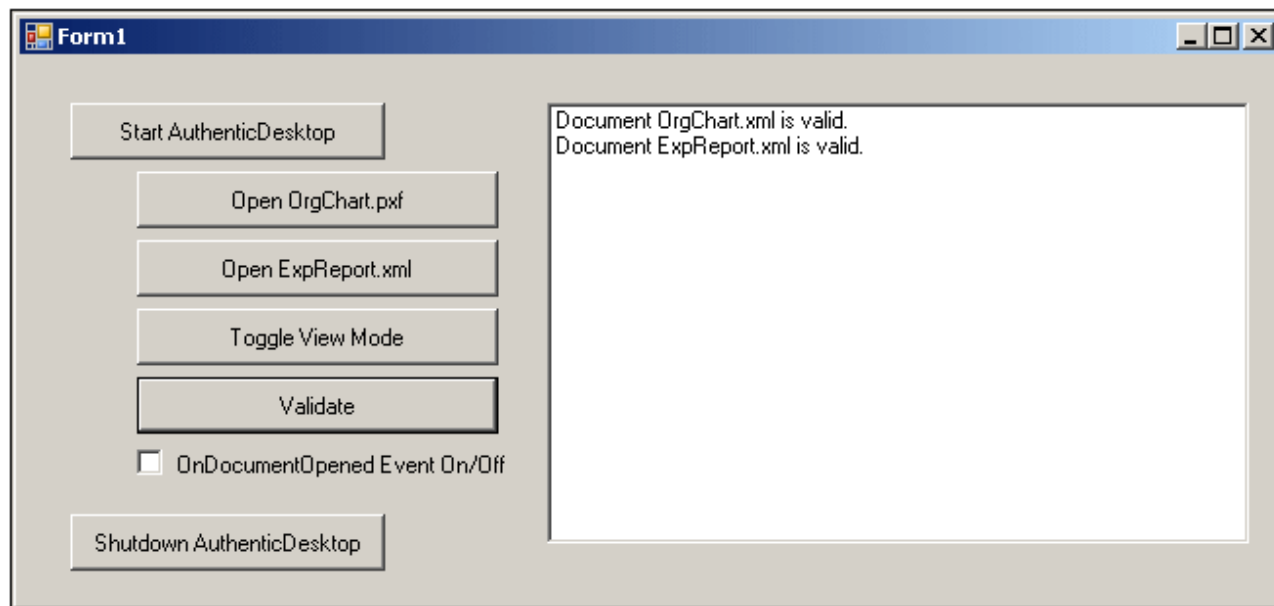
You can compile and run the project from within Visual Studio 2010/2012/2013/2015/2017/2019. The code listing below shows how basic application functionality can be used. This code is similar to the example C# project in the API Examples folder of your application package, but might differ slightly.

Platform configuration

If you have a 64-bit operating system and are using a 32-bit installation of Authentic Desktop, you must add the x86 platform in the solution's Configuration Manager and build the sample using this configuration. A new x86 platform (for the active solution in Visual Studio) can be created in the New Solution Platform dialog (**Build | Configuration Manager | Active solution platform | <New...>**).

What the code listing below does

The example code listing below creates a simple user interface (*screenshot below*) with buttons that invoke basic Authentic Desktop operations:



- [Start Authentic Desktop](#): Starts Authentic Desktop, which is registered as an automation server, or activates the application if it is already running.
- [Open OrgChart.pxf](#): Locates one of the example documents installed with Authentic Desktop and opens it. If this document is already open it becomes the active document.
- [Open ExpReport.xml](#): Opens another example document.
- [Toggle View Mode](#): Changes the view of all open documents between Text View and Authentic View. The code shows how to iterate through open documents.
- [Validate](#): Validates the active document and shows the result in a message box. The code shows how to handle errors and COM output parameters.
- [Shut down Authentic Desktop](#): Stops Authentic Desktop.

You can modify the code (of the code listing below or of the example C# project in the API Examples folder) in any way you like and run it.

Compiling and running the example

In the API Examples folder, double-click the file `AutomateAuthenticDesktop_VS2008.sln` or the file `AutomateAuthenticDesktop_VS2010.sln` (to open in Visual Studio 2010/2012/2013/2015/2017/2019). Alternatively the file can be opened from within Visual Studio (with **File | Open | Project/Solution**). To compile and run the example, select **Debug | Start Debugging** or **Debug | Start Without Debugging**.

Code listing of the example

Given below is the C# code listing of the basic functionality of the form (`Form1.cs`) created in the `AutomateAuthenticDesktop` example. Note that the code listed below might differ slightly from the code in the API Examples form. The listing below is commented for ease of understanding. Parts of the code are also presented separately in the sub-sections of this section, according to the Application API functionality they access.

The code essentially consists of a series of handlers for the buttons in the user interface shown in the screenshot above.

```
namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        // An instance of AuthenticDesktop is accessed via its automation interface.
        XMLSpyLib.Application AuthenticDesktop;

        // Location of examples installed with AuthenticDesktop
        String strExamplesFolder;

        private void Form1_Load(object sender, EventArgs e)
        {
            // Locate examples installed with AuthenticDesktop.
            // REMARK: You might need to adapt this if you have a different major version
of the product.
            strExamplesFolder = Environment.GetEnvironmentVariable("USERPROFILE") + "\\My
Documents\\Altova\\Authentic2012\\AuthenticExamples\\";
        }

        // Handler for the "Start AuthenticDesktop" button
        private void StartAuthenticDesktop_Click(object sender, EventArgs e)
        {
            if (AuthenticDesktop == null)
            {
                Cursor.Current = Cursors.WaitCursor;

                // If there is no AuthenticDesktop instance, create one and make it
visible.
                AuthenticDesktop = new XMLSpyLib.Application();
                AuthenticDesktop.Visible = true;

                Cursor.Current = Cursors.Default;
            }
        }
    }
}
```

```
    }
    else
    {
        // If an AuthenticDesktop instance is already running, make sure it's
visible.
        if (!AuthenticDesktop.Visible)
            AuthenticDesktop.Visible = true;
    }
}

// Handler for the "Open OrgChart.pxf" button
private void openOrgChart_Click(object sender, EventArgs e)
{
    // Make sure there's a running Authentic Desktop instance, and that it's
visible
    StartAuthenticDesktop_Click(null, null);

    // Open a sample file installed with the product.
    AuthenticDesktop.Documents.OpenFile(strExamplesFolder + "OrgChart.pxf",
false);
}

// Handler for the "Open ExpReport.xml" button
private void openExpReport_Click(object sender, EventArgs e)
{
    // Make sure there's a running Authentic Desktop instance, and that it's
visible
    StartAuthenticDesktop_Click(null, null);

    // Open a sample file installed with the product.
    AuthenticDesktop.Documents.OpenFile(strExamplesFolder + "ExpReport.xml",
false);
}

// Handler for the "Toggle View Mode" button
private void toggleView_Click(object sender, EventArgs e)
{
    // Make sure there's a running Authentic Desktop instance, and that it's
visible
    StartAuthenticDesktop_Click(null, null);

    // Iterate through all open documents and toggle the current view between
Text View and Authentic View.
    foreach (XMLSpyLib.Document doc in AuthenticDesktop.Documents)
        if (doc.CurrentViewMode == XMLSpyLib.SPYViewModes.spyViewAuthentic)
            doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewBrowser);
        else
            doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewAuthentic);
}

// Handler for the "Shut down AuthenticDesktop" button
// Shut down application instance by explicitly releasing the COM object.
private void shutdownAuthenticDesktop_Click(object sender, EventArgs e)
{
    if (AuthenticDesktop != null)
    {
        // Allow shut down of AuthenticDesktop by releasing the UI
        AuthenticDesktop.Visible = false;

        // Explicitly release the COM object
        try
        {
```



```
        while
(System.Runtime.InteropServices.Marshal.ReleaseComObject(AuthenticDesktop) > 0) ;
    }
    finally
    {
        // Avoid subsequent access to this object.
        AuthenticDesktop = null;
    }
}

// Handler for the "Validate" button
private void validate_Click(object sender, EventArgs e)
{
    // COM errors get returned to C# as exceptions. Use a try/catch block to
handle them.
    try
    {
        // Method 'IsValid' is one of the few functions that use output
parameters.
        // Use 'object' type for these parameters.
        object strErrorText = "";
        object nErrorNumber = 0;
        object errorData = null;

        if (!AuthenticDesktop.ActiveDocument.IsValid(ref strErrorText, ref
nErrorNumber, ref errorData))
        {
            // The COM call succeeds but the document is not valid.
            // A detailed description of the problem is returned in strErrorText,
nErrorNumber and errorData.
            listBoxMessages.Items.Add("Document " +
AuthenticDesktop.ActiveDocument.Name + " is not valid.");
            listBoxMessages.Items.Add("\tErrorText : " + strErrorText);
            listBoxMessages.Items.Add("\tErrorNumber: " + nErrorNumber);
            listBoxMessages.Items.Add("\tElement : " + (errorData != null ?
((XMLSpyLib.XMLData)errorData).TextValue : "null"));
        }
        else
        {
            // The COM call succeeds and the document is valid.
            listBoxMessages.Items.Add("Document " +
AuthenticDesktop.ActiveDocument.Name + " is valid.");
        }
    }
    catch (Exception ex)
    {
        // The COM call was not successful.
        // Probably no application instance has been started or no document is
open.
        listBoxMessages.Items.Add("Error validating active document: " +
ex.Message);
    }
}

delegate void addListBoxItem_delegate(string sText);
// Called from the UI thread
private void addListBoxItem(string sText)
{
    listBoxMessages.Items.Add(sText);
}
// Wrapper method to call UI control methods from a worker thread
```

```

        void syncWithUiThread(Control ctrl, addListBoxItem_delegate methodToInvoke,
String sText)
        {
            // Control.Invoke: Executes on the UI thread, but calling thread waits for
completion before continuing.
            // Control.BeginInvoke: Executes on the UI thread, and calling thread doesn't
wait for completion.
            if (ctrl.InvokeRequired)
                ctrl.BeginInvoke(methodToInvoke, new Object[] { sText });
        }

// Event handler for OnDocumentOpened event
private void handleOnDocumentOpened(XMLSpyLib.Document i_ipDocument)
{
    String sText = "";

    if (i_ipDocument.Name.Length > 0)
        sText = "Document " + i_ipDocument.Name + " was opened!";
    else
        sText = "An empty document was created.";

    // Synchronize the calling thread with the UI thread because
// COM events are triggered from a working thread
addListBoxItem_delegate methodToInvoke = new
addListBoxItem_delegate(addListBoxItem);
// Call syncWithUiThread with the following arguments:
// 1 - listBoxMessages - list box control to display messages from COM events
// 2 - methodToInvoke - a C# delegate which points to the method which will
be called from the UI thread
// 3 - sText - the text to be displayed in the list box
syncWithUiThread(listBoxMessages, methodToInvoke, sText);
}

private void checkBoxEventOnOff_CheckedChanged(object sender, EventArgs e)
{
    if (AuthenticDesktop != null)
    {
        if (checkBoxEventOnOff.Checked)
            AuthenticDesktop.OnDocumentOpened += new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);
        else
            AuthenticDesktop.OnDocumentOpened -= new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);
    }
}
}
}

```

13.3.1.2.3.1 Add Reference to Authentic Desktop API

Add the application's type library as a reference in a .NET project as follows: With the .NET project open, click **Project | Add Reference**. Then browse for the type library, which is called `Authentic.tlb`, and is located in the Authentic Desktop application folder.

Then declare a variable to access the Authentic Desktop API:

```
// An instance of Authentic Desktop is accessed via its automation interface.
```

```
XMLSpyLib.Application Authentic Desktop;
```

13.3.1.2.3.2 Application Startup and Shutdown

In the code snippets below, the methods `StartAuthenticDesktop_Click` and `ShutdownAuthenticDesktop_Click` are those assigned to buttons in the [AutomateAuthenticDesktop example](#) that, respectively, start up and shut down the application. This example is located in the C# subfolder of the API Examples folder (see the file *Form1.cs*):

Windows 7、Windows 8、 Windows 10	C:\Users\<<username>\Documents\ Altova\Authentic\2021\%APPNAME%\Examples
------------------------------------	---

You can compile and run the project from within Visual Studio 2010/2012/2013/2015/2017/2019.

Starting Authentic Desktop

The following code snippet from the [AutomateAuthenticDesktop example](#) shows how to start up the application.

```
// Handler for the "Start AuthenticDesktop" button
private void StartAuthenticDesktop_Click(object sender, EventArgs e)
{
    if (AuthenticDesktop == null)
    {
        Cursor.Current = Cursors.WaitCursor;

        // If there is no AuthenticDesktop instance, create one and make it
visible.
        AuthenticDesktop = new XMLSpyLib.Application();
        AuthenticDesktop.Visible = true;

        Cursor.Current = Cursors.Default;
    }
    else
    {
        // If an instance of Authentic Desktop is already running, make sure it's
visible
        if (!AuthenticDesktop.Visible)
            AuthenticDesktop.Visible = true;
    }
}
```

Shutting down Authentic Desktop

The following code snippet from the [AutomateAuthenticDesktop example](#) shows how to shut down the application.

```
// Handler for the "Shut down AuthenticDesktop" button
// Shut down application instance by explicitly releasing the COM object.
private void shutdownAuthenticDesktop_Click(object sender, EventArgs e)
{
    if (AuthenticDesktop != null)
    {
        // Allow shut down of AuthenticDesktop by releasing the UI
        AuthenticDesktop.Visible = false;
    }
}
```

```

        // Explicitly release the COM object
        try
        {
            while
(System.Runtime.InteropServices.Marshal.ReleaseComObject(AuthenticDesktop) > 0) ;
        }
        finally
        {
            // Avoid subsequent access to this object.
            AuthenticDesktop = null;
        }
    }
}

```

13.3.1.2.3.3 Opening Documents

The code snippets below (from the [AutomateAuthenticDesktop example](#)) show how two files are opened via two separate methods assigned to two buttons in the user interface. Both methods use the same Application API access mechanism: [Documents.OpenFile\(string, boolean\)](#).

The [AutomateAuthenticDesktop example](#) (see the file *Form1.cs*) is located in the C# subfolder of the API Examples folder:

Windows 7、Windows 8、 Windows 10	C:\Users\<<username>\Documents\ Altova\Authentic\2021\%APPNAME%\Examples
------------------------------------	---

You can compile and run the project from within Visual Studio 2010/2012/2013/2015/2017/2019.

Code snippet

```

// Handler for the "Open OrgChart.pxf" button
private void openOrgChart_Click(object sender, EventArgs e)
{
    // Make sure there's a running Authentic Desktop instance, and that it's
visible
    StartAuthenticDesktop_Click(null, null);

    // Open a sample file installed with the product.
false);
    AuthenticDesktop.Documents.OpenFile(strExamplesFolder + "OrgChart.pxf",
}

// Handler for the "Open ExpReport.xml" button
private void openExpReport_Click(object sender, EventArgs e)
{
    // Make sure there's a running Authentic Desktop instance, and that it's
visible
    StartAuthenticDesktop_Click(null, null);

    // Open a sample file installed with the product.
false);
    AuthenticDesktop.Documents.OpenFile(strExamplesFolder + "ExpReport.xml",
}

```

The file opened last will be the active file.

13.3.1.2.3.4 Iterating through Open Documents

The code snippet below (from the [AutomateAuthenticDesktop example](#); see the file *Form1.cs*) shows how to iterate through open documents. A condition is then tested within the iteration loop, and the document view is switched between Browser View and Authentic View.

```
// Handler for the "Toggle View Mode" button
private void toggleView_Click(object sender, EventArgs e)
{
    // Make sure there's a running Authentic Desktop instance, and that it's
visible
    StartAuthenticDesktop_Click(null, null);

    // Iterate through all open documents and toggle the current view between
Browser View and Authentic View.
    foreach (XMLSpyLib.Document doc in AuthenticDesktop.Documents)
        if (doc.CurrentViewMode == XMLSpyLib.SPYViewModes.spyViewAuthentic)
            doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewBrowser);
        else
            doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewAuthentic);
}
```

The [AutomateAuthenticDesktop example](#) example is located in the C# subfolder of the API Examples folder:

Windows 7、Windows 8、 Windows 10	C:\Users\<<username>\Documents\ Altova\Authentic\2021\%APPNAME%\Examples
------------------------------------	---

You can compile and run the project from within Visual Studio 2010/2012/2013/2015/2017/2019.

13.3.1.2.3.5 Errors and COM Output Parameters

The code snippet below (from the [AutomateAuthenticDesktop example](#)) shows how to handle errors and COM output parameters. The method [AuthenticDesktop.ActiveDocument.IsValid\(ref strErrorText, ref nErrorNumber, ref errorData\)](#) uses output parameters that are used, in the code snippet below, to generate an error-message text.

The [AutomateAuthenticDesktop example](#) (see the file *Form1.cs*) is located in the C# subfolder of the API Examples folder:

Windows 7、Windows 8、 Windows 10	C:\Users\<<username>\Documents\ Altova\Authentic\2021\%APPNAME%\Examples
------------------------------------	---

You can compile and run the project from within Visual Studio 2010/2012/2013/2015/2017/2019.

Code snippet

```

// Handler for the "Validate" button
private void validate_Click(object sender, EventArgs e)
{
    // COM errors get returned to C# as exceptions. Use a try/catch block to
handle them.
    try
    {
        // Method 'IsValid' is one of the few functions that use output
parameters.
        // Use 'object' type for these parameters.
        object strErrorText = "";
        object nErrorNumber = 0;
        object errorData = null;

        if (!AuthenticDesktop.ActiveDocument.IsValid(ref strErrorText, ref
nErrorNumber, ref errorData))
        {
            // The COM call succeeds but the document is not valid.
            // A detailed description of the problem is returned in strErrorText,
nErrorNumber and errorData.
            listBoxMessages.Items.Add("Document " +
AuthenticDesktop.ActiveDocument.Name + " is not valid.");
            listBoxMessages.Items.Add("\tErrorText : " + strErrorText);
            listBoxMessages.Items.Add("\tErrorNumber: " + nErrorNumber);
            listBoxMessages.Items.Add("\tElement : " + (errorData != null ?
((XMLSpyLib.XMLData)errorData).TextValue : "null"));
        }
        else
        {
            // The COM call succeeds and the document is valid.
            listBoxMessages.Items.Add("Document " +
AuthenticDesktop.ActiveDocument.Name + " is valid.");
        }
    }
    catch (Exception ex)
    {
        // The COM call was not successful.
        // Probably no application instance has been started or no document is
open.
        listBoxMessages.Items.Add("Error validating active document: " +
ex.Message);
    }
}

```

13.3.1.2.3.6 Events

The code snippet below (from the [AutomateAuthenticDesktop example](#)) lists the code for two event handlers. The [AutomateAuthenticDesktop example](#) (see the file *Form1.cs*) is located in the C# subfolder of the API Examples folder:

Windows 7、Windows 8、 Windows 10	C:\Users\<<username>\Documents\ Altova\Authentic\2021\%APPNAME%\Examples
------------------------------------	---

You can compile and run the project from within Visual Studio 2010/2012/2013/2015/2017/2019.

Code snippet

```
delegate void addListBoxItem_delegate(string sText);
// Called from the UI thread
private void addListBoxItem(string sText)
{
    listBoxMessages.Items.Add(sText);
}
// Wrapper method to call UI control methods from a worker thread
void syncWithUiThread(Control ctrl, addListBoxItem_delegate methodToInvoke,
String sText)
{
    // Control.Invoke: Executes on the UI thread, but calling thread waits for
completion before continuing.
    // Control.BeginInvoke: Executes on the UI thread, and calling thread doesn't
wait for completion.
    if (ctrl.InvokeRequired)
        ctrl.BeginInvoke(methodToInvoke, new Object[] { sText });
}

// Event handler for OnDocumentOpened event
private void handleOnDocumentOpened(XMLSpyLib.Document i_ipDocument)
{
    String sText = "";

    if (i_ipDocument.Name.Length > 0)
        sText = "Document " + i_ipDocument.Name + " was opened!";
    else
        sText = "An empty document was created.";

    // Synchronize the calling thread with the UI thread because
// COM events are triggered from a working thread
addListBoxItem_delegate methodToInvoke = new
addListBoxItem_delegate(addListBoxItem);
// Call syncWithUiThread with the following arguments:
// 1 - listBoxMessages - list box control to display messages from COM events
// 2 - methodToInvoke - a C# delegate which points to the method which will
be called from the UI thread
// 3 - sText - the text to be displayed in the list box
syncWithUiThread(listBoxMessages, methodToInvoke, sText);
}

private void checkBoxEventOnOff_CheckedChanged(object sender, EventArgs e)
{
    if (AuthenticDesktop != null)
    {
        if (checkBoxEventOnOff.Checked)
            AuthenticDesktop.OnDocumentOpened += new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);
        else
            AuthenticDesktop.OnDocumentOpened -= new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);
    }
}
```

13.3.1.2.4 Java

The Application API can be accessed from Java code. To allow accessing the Authentic Desktop automation server directly from Java code, the libraries listed below must reside in the `classpath`. They are installed in the folder: `JavaAPI` in the Authentic Desktop application folder.

- `AltovaAutomation.dll`: a JNI wrapper for Altova automation servers (`AltovaAutomation_x64.dll` in the case of 64-bit versions)
- `AltovaAutomation.jar`: Java classes to access Altova automation servers
- `AuthenticAPI.jar`: Java classes that wrap the Authentic Desktop automation interface
- `AuthenticAPI_JavaDoc.zip`: a Javadoc file containing help documentation for the Java API

Note: In order to use the Java API, the DLL and Jar files must be on the Java Classpath.

Example Java project

An example Java project is supplied with your product installation. You can test the Java project and modify and use it as you like. For more details of the example Java project, see the section, [Example Java Project](#).

Rules for mapping the Application API names to Java

The rules for mapping between the Application API and the Java wrapper are as follows:

- **Classes and class names**
For every interface of the Authentic Desktop automation interface a Java class exists with the name of the interface.
- **Method names**
Method names on the Java interface are the same as used on the COM interfaces but start with a small letter to conform to Java naming conventions. To access COM properties, Java methods that prefix the property name with `get` and `set` can be used. If a property does not support write-access, no setter method is available. Example: For the `Name` property of the `Document` interface, the Java methods `getName` and `setName` are available.
- **Enumerations**
For every enumeration defined in the automation interface, a Java enumeration is defined with the same name and values.
- **Events and event handlers**
For every interface in the automation interface that supports events, a Java interface with the same name plus 'Event' is available. To simplify the overloading of single events, a Java class with default implementations for all events is provided. The name of this Java class is the name of the event interface plus 'DefaultHandler'. For example:
Application: Java class to access the application
ApplicationEvents: Events interface for the Application
ApplicationEventsDefaultHandler: Default handler for ApplicationEvents

Exceptions to mapping rules

There are some exceptions to the rules listed above. These are listed below:

Interface	Java name
-----------	-----------

Document, method SetEncoding	setFileEncoding
AuthenticView, method Goto	gotoElement
AuthenticRange, method Goto	gotoElement
AuthenticRange, method Clone	cloneRange

This section

This section explains how some basic Authentic Desktop functionality can be accessed from Java code. It is organized into the following sub-sections:

- [Example Java Project](#)
- [Application Startup and Shutdown](#)
- [Simple Document Access](#)
- [Iterations](#)
- [Use of Out-Parameters](#)
- [Event Handlers](#)

13.3.1.2.4.1 Example Java Project

The Authentic Desktop installation package contains an example Java project, located in the the `API\Java` subfolder of the `Examples` folder :

Windows 7、Windows 8、 Windows 10	C:\Users\ <username>\documents\ </username>\documents\ Altova\Authentic\2021\%APPNAME%\Examples
------------------------------------	--

This folder contains Java examples for the Authentic Desktop API. You can test it directly from the command line using the batch file `BuildAndRun.bat`, or you can compile and run the example project from within Eclipse. See below for instructions on how to use these procedures.

File list

The Java examples folder contains all the files required to run the example project. These files are listed below. If you are using a 64-bit version of the application, some filenames contain `_x64` in the name. These filenames are indicated with `(_x64)`.

<code>AltovaAutomation(_x64).dll</code>	Java-COM bridge: DLL part
<code>AltovaAutomation.jar</code>	Java-COM bridge: Java library part
<code>AuthenticAPI.jar</code>	Java classes of the Authentic Desktop API
<code>RunAuthenticDesktop.java</code>	Java example source code
<code>BuildAndRun.bat</code>	Batch file to compile and run example code from the command line prompt. Expects folder where Java Virtual Machine resides as parameter.
<code>.classpath</code>	Eclipse project helper file
<code>.project</code>	Eclipse project file
<code>Authentic_JavaDoc.zip</code>	Javadoc file containing help documentation for the Java API

What the example does

The example starts up Authentic Desktop and performs a few operations, including opening and closing documents. When done, Authentic Desktop stays open. You must close it manually.

- [Start Authentic Desktop](#): Starts Authentic Desktop, which is registered as an automation server, or activates Authentic Desktop if it is already running.
- [Open example files](#): Locates example documents installed with Authentic Desktop and opens them.
- [Iteration and Changing the View Mode](#): Changes the view of all open documents to Browser View. The code also shows how to iterate through open documents.
- [Iteration, validation, output parameters](#): Validates the active document and shows the result in a message box. The code shows how to use output parameters.
- [Event Handling](#): Shows how to handle Authentic Desktop events.
- [Shut down Authentic Desktop](#): Shuts down Authentic Desktop.

You can modify the example in any way you like and run it.

Running the example from the command line

To run the example from the command line, open a command prompt window, go to the Java folder of the API Examples folder (*see above for location*), and then type:

```
buildAndRun.bat "<Path-to-the-Java-bin-folder>"
```

The Java binary folder must be that of a JDK 1.5 or later installation on your computer. Press the **Return** key. The Java source in `RunAuthenticDesktop.java` will be compiled and then executed.

Loading the example in Eclipse

Open Eclipse and use the **Import | Existing Projects into Workspace** command to add the Eclipse project file (`.project`) located in the Java folder of the API Examples folder (*see above for location*). The project `RunAuthenticDesktop` will then appear in your Package Explorer or Navigator. Select the project and then the command **Run as | Java Application** to execute the example.

Note: You can select a class name or method of the Java API and press F1 to get help for that class or method.

Java source code listing

The Java source code in the example file `RunAuthenticDesktop.java` is listed below with comments.

```
01 // Access general JAVA-COM bridge classes
02 import com.altova.automation.libs.*;
03
04 // Access AuthenticDesktop Java-COM bridge
05 import com.altova.automation.AuthenticDesktop.*;
06 import com.altova.automation.AuthenticDesktop.Enums.SPYViewModes;
07
08 /**
09  * A simple example that starts AuthenticDesktop COM server and performs a view
10  * operations on it.
11  * Feel free to extend.
12  */
12 public class RunAuthenticDesktop
```

```
13 {
14     public static void main(String[] args)
15     {
16         // An instance of the application.
17         Application authenticDesktop = null;
18
19         // Instead of COM error-handling, use Java exception mechanism.
20         try
21         {
22             // Start AuthenticDesktop as COM server.
23             authenticDesktop = new Application();
24             // COM servers start up invisible so we make it visible
25             authenticDesktop.setVisible(true);
26
27             // Locate samples installed with the product.
28             String strExamplesFolder = System.getenv("USERPROFILE") + "\\My Documents\\
29             \\Altova\\Authentic2012\\AuthenticExamples\\";
30
31             // Open two files from the product samples.
32             authenticDesktop.getDocuments().openFile(strExamplesFolder + "OrgChart.pxf",
33             false);
34             authenticDesktop.getDocuments().openFile(strExamplesFolder + "ExpReport.xml",
35             false);
36
37             // Iterate through all open documents and set the View Mode to 'Text'.
38             for (Document doc:authenticDesktop.getDocuments())
39                 if ( doc.getCurrentViewMode() != SPYViewModes.spyViewText)
40                     doc.switchViewMode(SPYViewModes.spyViewText);
41
42             // An alternative iteration mode is index-based. COM indices are typically zero-
43             based.
44             Documents documents = authenticDesktop.getDocuments();
45             for (int i = 1; i <= documents.getCount(); i++)
46             {
47                 Document doc = documents.getItem(i);
48
49                 // Validation is one of the few methods that have output parameters.
50                 // The class JVariant is the correct type for parameters in these cases.
51                 // To get values back mark them with the by-reference flag.
52                 JVariant validationErrorText = new JVariant.JStringVariant("");
53                 validationErrorText.setByRefFlag();
54                 JVariant validationErrorCount = new JVariant.JIntVariant(0);
55                 validationErrorCount.setByRefFlag();
56                 JVariant validationErrorXMLData = new JVariant.JIDispatchVariant(0);
57                 validationErrorXMLData.setByRefFlag();
58                 if (!doc.isValid(validationErrorText, validationErrorCount,
59                 validationErrorXMLData))
60                     System.out.println("Document " + doc.getName() + " is not wellformed - " +
61                 validationErrorText.getStringValue());
62                 else
63                     System.out.println("Document " + doc.getName() + " is wellformed.");
64             }
65
66             // The following lines attach to the document events using a default
67             implementation
68             // for the events and override one of its methods.
69             // If you want to override all document events it is better to derive your
70             listener class
71             // from DocumentEvents and implement all methods of this interface.
72             Document doc = authenticDesktop.getActiveDocument();
73             doc.addListener(new DocumentEventsDefaultHandler()
74             {
```

```
64         @Override
65         public boolean onBeforeCloseDocument(Document i_ipDoc) throws
AutomationException
66         {
67             System.out.println("Document " + i_ipDoc.getName() + " requested closing.");
68
69             // Allow closing of document
70             return true;
71         }
72     });
73     doc.close(true);
74     doc = null;
75
76     System.out.println("Watch AuthenticDesktop!");
77 }
78 catch (AutomationException e)
79 {
80     // e.printStackTrace();
81 }
82 finally
83 {
84     // Make sure that AuthenticDesktop can shut down properly.
85     if (authenticDesktop != null)
86         authenticDesktop.dispose();
87
88     // Since the COM server was made visible and still is visible, it will keep
running
89     // and needs to be closed manually.
90     System.out.println("Now close AuthenticDesktop!");
91 }
92 }
93 }
```

13.3.1.2.4.2 Application Startup and Shutdown

The code listings below show how the application can be started up and shut down.

Application startup

Before starting up the application, the appropriate classes must be imported (see *below*).

```
01 // Access general JAVA-COM bridge classes
02 import com.altova.automation.libs.*;
03
04 // Access AuthenticDesktop Java-COM bridge
05 import com.altova.automation.AuthenticDesktop.*;
06 import com.altova.automation.AuthenticDesktop.Enums.SPYViewModes;
07
08 /**
09  * A simple example that starts AuthenticDesktop COM server and performs a view
operations on it.
10  * Feel free to extend.
11  */
12 public class RunAuthenticDesktop
13 {
14     public static void main(String[] args)
15     {
16         // An instance of the application.
```

```
17 Application authenticDesktop = null;
18
19 // Instead of COM error-handling, use Java exception mechanism.
20 try
21 {
22     // Start AuthenticDesktop as COM server.
23     authenticDesktop = new Application();
24     // COM servers start up invisible so we make it visible
25     authenticDesktop.setVisible(true);
26
27 ...
28 }
29 }
30 }
```

Application shutdown

The application can be shut down as shown below.

```
1 {
2     // Make sure that AuthenticDesktop can shut down properly.
3     if (authenticDesktop != null)
4         authenticDesktop.dispose();
5
6     // Since the COM server was made visible and still is visible, it will keep
running
7     // and needs to be closed manually.
8     System.out.println("Now close AuthenticDesktop!");
9 }
```

13.3.1.2.4.3 Simple Document Access

The code listing below shows how to open a document.

```
1 // Locate samples installed with the product.
2 String strExamplesFolder = System.getenv("USERPROFILE") + "\\My Documents\\Altova\\
\\Authentic2012\\AuthenticExamples\\";
3
4 // Open two files from the product samples.
5 authenticDesktop.getDocuments().openFile(strExamplesFolder + "OrgChart.pxf", false);
6 authenticDesktop.getDocuments().openFile(strExamplesFolder + "ExpReport.xml", false);
```

13.3.1.2.4.4 Iterations

The listing below shows how to iterate through open documents.

```
01 // Iterate through all open documents and set the View mode to 'Browser'.
02 for (Document doc:authenticDesktop.getDocuments())
03     if ( doc.getCurrentViewMode() != SPYViewModes.spyViewBrowser)
04         doc.switchViewMode(SPYViewModes.spyViewBrowser);
05
06 // An alternative iteration mode is index-based. COM indices are typically zero-based.
```

```
07 Documents documents = authenticDesktop.getDocuments();
08 for (int i = 1; i <= documents.getCount(); i++)
09 {
10     Document doc = documents.getItem(i);
11     ...
12 }
```

13.3.1.2.4.5 Use of Out-Parameters

The code listing below iterates through open documents and validates each of them. For each validation, a message is generated using the output parameters of the Validation method.

```
01 // Iterate through all open documents and set the View mode to 'Text'.
02 for (Document doc:authenticDesktop.getDocuments())
03     if ( doc.getCurrentViewMode() != SPYViewModes.spyViewText)
04         doc.switchViewMode(SPYViewModes.spyViewText);
05
06 // An alternative iteration mode is index-based. COM indices are typically zero-based.
07 Documents documents = authenticDesktop.getDocuments();
08 for (int i = 1; i <= documents.getCount(); i++)
09 {
10     Document doc = documents.getItem(i);
11
12     // Validation is one of the few methods that have output parameters.
13     // The class JVariant is the correct type for parameters in these cases.
14     // To get values back, mark them with the by-reference flag.
15     JVariant validationErrorText = new JVariant.JStringVariant("");
validationErrorText.setByRefFlag();
16     JVariant validationErrorCount = new JVariant.JIntVariant(0);
validationErrorCount.setByRefFlag();
17     JVariant validationErrorXMLData = new JVariant.JIDispatchVariant(0);
validationErrorXMLData.setByRefFlag();
18     if (!doc.isValid(validationErrorText, validationErrorCount, validationErrorXMLData))
19         System.out.println("Document " + doc.getName() + " is not wellformed - " +
validationErrorText.getStringValue());
20     else
21         System.out.println("Document " + doc.getName() + " is wellformed.");
22 }
```

13.3.1.2.4.6 Event Handlers

The listing below shows how to listen for and use events.

```
01 // The following lines attach to the document events using a default implementation
02 // for the events and override one of its methods.
03 // If you want to override all document events, it is better to derive your listener
class
04 // from DocumentEvents and implement all methods of this interface.
05 Document doc = authenticDesktop.getActiveDocument();
06 doc.addListener(new DocumentEventsDefaultHandler())
07 {
08     @Override
09     public boolean onBeforeCloseDocument(Document i_ipDoc) throws AutomationException
```

```
10 {
11     System.out.println("Document " + i_ipDoc.getName() + " requested closing.");
12
13     // allow closing of document
14     return true;
15 }
16 });
17 doc.close(true);
18 doc = null;
```

13.3.2 Interfaces

Object Hierarchy

[Application](#)

[SpyProject](#)

[SpyProjectItems](#)

[SpyProjectItem](#)

[Documents](#)

[Document](#)

[GridView](#)

[AuthenticView](#)

[AuthenticRange](#)

[AuthenticDataTransfer](#) (previously DocEditDataTransfer)

[AuthenticDataTransfer](#) (previously DocEditDataTransfer)

[TextView](#)

[XMLData](#)

[Dialogs](#)

[CodeGeneratorDlg](#)

[FileSelectionDlg](#)

[SchemaDocumentationDlg](#)

[GenerateSampleXMLDlg](#)

[DTDSchemaGeneratorDlg](#)

[FindInFilesDlg](#)

[DatabaseConnection](#)

[ExportSettings](#)

[TextImportExportSettings](#)

[ElementList](#)

[ElementListItem](#)

[Enumerations](#)

Description

This chapter contains the reference of the Authentic Desktop 1.5 Type Library.

Most of the given examples are written in VisualBasic. These code snippets assume that there is a variable defined and set, called **objSpy of type Application**. There are also some code samples written in JavaScript.

13.3.2.1 Application

Methods

[GetDatabaseImportElementList](#)

[GetDatabaseSettings](#)
[GetDatabaseTables](#)
[ImportFromDatabase](#)
[CreateXMLSchemaFromDBStructure](#)

[GetTextImportElementList](#)
[GetTextImportExportSettings](#)
[ImportFromText](#)

[ImportFromWord](#)

[ImportFromSchema](#)

[GetExportSettings](#)

[NewProject](#)
[OpenProject](#)

[AddMacroMenuItem](#)
[ClearMacroMenu](#)

[ShowForm](#)

[ShowApplication](#)

[URLDelete](#)
[URLMakeDirectory](#)

[AddXSLT_XQParameter](#)
[GetXSLT_XQParameterCount](#)
[GetXSLT_XQParameterName](#)
[GetXSLT_XQParameterXPath](#)
[RemoveXSLT_XQParameter](#)

[FindInFiles](#)

[Quit](#)

Properties

[Application](#)
[Parent](#)

[ActiveDocument](#)
[Documents](#)

[CurrentProject](#)

[Dialogs](#)

[WarningNumber](#)
[WarningText](#)

[Status](#)
[MajorVersion](#)
[MinorVersion](#)

[Edition](#)
[IsAPISupported](#)
[ServicePackVersion](#)

Description

Application is the root for all other objects. It is the only object you can create by CreateObject (VisualBasic) or other similar COM related functions.

Example

```
Dim objSpy As Application
Set objSpy = CreateObject("XMLSpy.Application")
```

13.3.2.1.1 Events

13.3.2.1.1.1 OnBeforeOpenDocument

Event: OnBeforeOpenDocument(*objDialog* as [FileSelectionDlg](#))

Description

This event gets fired whenever a document gets opened via the OpenFile or OpenURL menu command. It is sent after a document file has been selected but before the document gets opened. The file selection dialog object is initialized with the name of the selected document file. You can modify this selection. To continue the opening of the document leave the [FileSelectionDlg.DialogAction](#) property of *io_objDialog* at its default value [spyDialogOK](#). To abort the opening of the document set this property to [spyDialogCancel](#).

Examples

Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:

```
Function On_BeforeOpenDocument(objDialog)
End Function
```

XMLSpy scripting environment - JScript:

```
function On_BeforeOpenDocument(objDialog)
{
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (26, ...)// nEventId = 26
```

13.3.2.1.1.2 OnBeforeOpenProject

Event: OnBeforeOpenProject(*objDialog* as [FileSelectionDlg](#))

Description

This event gets fired after a project file has been selected but before the project gets opened. The file selection dialog object is initialized with the name of the selected project file. You can modify this selection. To continue

the opening of the project leave the [FileSelectionDlg.DialogAction](#) property of *io_objDialog* at its default value [spyDialogOK](#). To abort the opening of the project set this property to [spyDialogCancel](#).

Examples

Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:

```
Function On_BeforeOpenProject(objDialog)
End Function
```

XMLSpy scripting environment - JScript:

```
function On_BeforeOpenProject(objDialog)
{
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (25, ...) // nEventId = 25
```

13.3.2.1.1.3 OnDocumentOpened

Event: OnDocumentOpened(*objDocument* as [Document](#))

Description

This event gets fired whenever a document opens in Authentic Desktop. This can happen due to opening a file with the OpenFile or OpenURL dialog, creating a new file or dropping a file onto Authentic Desktop. The new document gets passed as parameter. The operation cannot be canceled.

Examples

Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:

```
Function On_OpenDocument(objDocument)
End Function
```

XMLSpy scripting environment - JScript:

```
function On_OpenDocument(objDocument)
{
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (7, ...) // nEventId = 7
```

13.3.2.1.1.4 OnProjectOpened

Event: OnProjectOpened(*objProject* as [SpyProject](#))

Description

This event gets fired whenever a project gets opened in Authentic Desktop. The new project gets passed as parameter.

Examples

Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:

```
Function On_OpenProject(objProject)
End Function
```

XMLSpy scripting environment - JScript:

```
function On_OpenProject(objProject)
{
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (6, ...) // nEventId = 6
```

13.3.2.1.2 ActiveDocument

Property: ActiveDocument as [Document](#)

Description

Reference to the active document. If no document is open, ActiveDocument is null (nothing).

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.3 AddMacroMenuItem

Method: AddMacroMenuItem(*strMacro* as String,*strDisplayText* as String)

Description

Adds a menu item to the **Tools** menu. This new menu item invokes the macro defined by *strMacro*. See also [Example Scripting Project](#).

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1108 Number of macro items is limited to 16 items.

13.3.2.1.4 AddXSLT_XQParameter

Method: AddXSLT_XQParameter(name as String, XPath as String)

Description

Adds an XSLT or XQuery parameter. The parameter's name and value are the two arguments of the method.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.
- 1124 The XPath expression is not set.
- 1125 Not a QName.
- 1126 The specified XPath is not valid. Reason for invalidity appended.
- 1127 A parameter with the submitted name already exists.

13.3.2.1.5 Application

Property: Application as [Application](#) (read-only)

Description

Accesses the Authentic Desktop application object.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.6 ClearMacroMenu

Method: ClearMacroMenu()

Return Value

None

Description

Removes from the **Tools** menu those menu items that were added by calling [AddMacroMenuItem](#). See also [Example Scripting Project](#).

Errors

- 1111 The application object is no longer valid.

13.3.2.1.7 CreateXMLSchemaFromDBStructure

Method: CreateXMLSchemaFromDBStructure(pImportSettings as [DatabaseConnection](#), pTables as [ElementList](#))

Description

CreateXMLSchemaFromDBStructure creates from a database specified in plImportSettings for the defined tables in pTables new XML Schema document(s) describing the database tables structure.

The parameter pTables specifies which table structures the XML Schema document should contain. This parameter can be NULL, specifying that all table structures will be exported.

See also [GetDataBaseTables](#).

Errors

- 1112 Invalid database specified.
- 1120 Database import failed.

13.3.2.1.8 CurrentProject

Property: CurrentProject as [SpyProject](#)

Description

Reference to the active document. If no project is open, CurrentProject is null (nothing).

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.9 Dialogs

Property: Dialogs as [Dialogs](#) (read-only)

Description

Access the built-in dialogs of Authentic Desktop.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.10 Documents

Property: Documents as [Documents](#)

Description

Collection of all open documents.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.11 Edition

Property: Edition as String

Description

Returns the edition of the application, for example `Altova Authentic Desktop Enterprise Edition` for the Enterprise edition.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.12 FindInFiles

Method: FindInFiles(*pSettings* as [FindInFilesDlg](#)) as [FindInFilesResults](#)

Description

Returns a [FindInFilesResults](#) object containing information about the files that matched the specified settings.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.13 GetDatabaseImportElementList

Method: GetDatabaseImportElementList(*pImportSettings* as [DatabaseConnection](#)) as [ElementList](#)

Description

The function returns a collection of [ElementListItem](#)s where the properties [ElementListItem.Name](#) contain the names of the fields that can be selected for import and the properties [ElementListItem.ElementKind](#) are initialized either to `spyXMLDataAttr` or `spyXMLDataElement`, depending on the value passed in [DatabaseConnection.AsAttributes](#). This list serves as a filter to what finally gets imported by a future call to [ImportFromDatabase](#). Use [ElementList.RemoveElement](#) to exclude fields from import.

Properties mandatory to be filled out for the database connection are one of [DatabaseConnection.File](#), [DatabaseConnection.ADOConnection](#) and [DatabaseConnection.ODBCConnection](#), as well as [DatabaseConnection.SQLSelect](#). Use the property [DatabaseConnection.AsAttributes](#) to initialize [ElementListItem.ElementKind](#) of the resulting element list to either `spyXMLDataAttr` or `spyXMLDataElement`, respectively.

Example

See example at [ImportFromDatabase](#).

Errors

- 1111 The application object is no longer valid.

- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1107 Import from database failed.
- 1112 Invalid database specified.
- 1114 Select statement is missing.
- 1119 database element list import failed.

13.3.2.1.14 GetDatabaseSettings

Method: GetDatabaseSettings() as [DatabaseConnection](#)

Description

GetDatabaseSettings creates a new object of database settings. The object is used to specify database connection parameters for the methods [GetDatabaseTables](#), [GetDatabaseImportElementList](#), [ImportFromDatabase](#), [ImportFromSchema](#) and [ExportToDatabase](#).

Example

See example of [ImportFromDatabase](#).

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.15 GetDatabaseTables

Method: GetDatabaseTables(*pImportSettings* as [DatabaseConnection](#)) as [ElementList](#)

Description

GetDatabaseTables reads the table names from the database specified in *pImportSettings*. Properties mandatory to be filled out for the database connection are one of [DatabaseConnection.File](#), [DatabaseConnection.ADOConnection](#) and [DatabaseConnection.ODBCConnection](#). All other properties are ignored.

The function returns a collection of [ElementListItem](#)s where the properties [ElementListItem.Name](#) contain the names of tables stored in the specified database. The remaining properties of [ElementListItem](#) are unused.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1112 Invalid database specified.
- 1113 Error while reading database table information.
- 1118 Database table query failed.

Example

```
Dim objImpSettings As DatabaseConnection
Set objImpSettings = objSpy.GetDatabaseSettings
objImpSettings.ADOConnection = TxtADO.Text
```

```
'store table names in list box
ListTables.Clear

Dim objList As ElementList
Dim objItem As ElementListItem
On Error GoTo ErrorHandler
Set objList = objSpy.GetDatabaseTables(objImpSettings)

    For Each objItem In objList
        ListTables.AddItem objItem.Name
    Next
```

13.3.2.1.16 GetExportSettings

Method: GetExportSettings() as [ExportSettings](#) (read-only)

Description

GetExportSettings creates a new object of common export settings. This object is used to pass the parameters to the export functions and defines the behaviour of the export calls. See also the export functions from [Document](#).

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.17 GetTextImportElementList

Method: GetTextImportElementList(pImportSettings as [TextImportExportSettings](#)) as [ElementList](#)

Description

GetTextImportElementList retrieves importing information about the text-file as specified in pImportSettings. The function returns a collection of ElementListItems where the properties [ElementListItem.Name](#) contain the names of the fields found in the file. The values of remaining properties are undefined.

If the text-file does not contain a column header, set pImportSettings.[HeaderRow](#) to false. The resulting element list will contain general column names like 'Field1' and so on.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1107 Import from database failed.
- 1115 Error during text element list import. Cannot create parser for import file.
- 1116 Error during text element list import.

Example

```
'-----
' VBA client code fragment - import selected fields from text file
'-----
    Dim objImpSettings As TextImportExportSettings
```



```
Set objImpSettings = objSpy.GetTextImportExportSettings

objImpSettings.ImportFile = "C:\ImportMe.txt"
objImpSettings.HeaderRow = False

Dim objList As ElementList
Set objList = objSpy.GetTextImportElementList(objImpSettings)

'exclude first column
objList.RemoveItem 1

Dim objImpDoc As Document
On Error Resume Next
Set objImpDoc = objSpy.ImportFromText(objImpSettings, objList)
CheckForError
```

13.3.2.1.18 [GetTextImportExportSettings](#)

Method: [GetTextImportExportSettings\(\)](#) as [TextImportExportSettings](#) (read-only)

Description

[GetTextImportExportSettings](#) creates a new object of common import and export settings for text files. See also the example for [Application.GetTextImportElementList](#).

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.19 [GetXSLT_XQParameterCount](#)

Method: [GetXSLT_XQParameterCount\(\)](#) as Long

Description

Returns the number of XSLT and XQuery parameters.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.20 [GetXSLT_XQParameterName](#)

Method: [GetXSLT_XQParameterName\(index as Long\)](#) as String

Description

Returns the name of the XSLT or XQuery parameter identified by the supplied index.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.21 GetXSLT_XQParameterXPath

Method: GetXSLT_XQParameterXPath(index as Long) as String

Description

Returns the XPath expression of the XSLT or XQuery parameter identified by the supplied index.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.22 ImportFromDatabase

Method: ImportFromDatabase(pImportSettings as [DatabaseConnection](#), pElementList as [ElementList](#)) as [Document](#)

Return Value

Creates a new document containing the data imported from the database.

Description

ImportFromDatabase imports data from a database as specified in pImportSettings and creates a new document containing the data imported from the database. Properties mandatory to be filled out are one of [DatabaseConnection.File](#), [DatabaseConnection.ADOConnection](#) or [DatabaseConnection.ODBCConnection](#) and [DatabaseConnection.SQLSelect](#). Additionally, you can use [DatabaseConnection.AsAttributes](#), [DatabaseConnection.ExcludeKeys](#), [DatabaseConnection.IncludeEmptyElements](#) and [NumberDateTimeFormat](#) to further parameterize import.

The parameter pElementList specifies which fields of the selected data gets written into the newly created document, and which are created as elements and which as attributes. This parameter can be NULL, specifying that all selected fields will be imported as XML elements.

See [GetDatabaseSettings](#) and [GetDatabaseImportElementList](#) for necessary steps preceding any import of data from a database.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1107 Import from database failed.
- 1112 Invalid database specified.
- 1114 Select statement is missing.

- 1117 Transformation to XML failed.
- 1120 Database import failed.

Example

```
Dim objImpSettings As DatabaseConnection
Set objImpSettings = objSpy.GetDatabaseSettings

objImpSettings.ADOConnection = strADOConnection
objImpSettings.SQLSelect = "SELECT * FROM MyTable"

Dim objDoc As Document
On Error Resume Next
Set objDoc = objSpy.ImportFromDatabase(objImpSettings,
objSpy.GetDatabaseImportElementList(objImpSettings))
' CheckForError here
```

13.3.2.1.23 ImportFromSchema

Method: ImportFromSchema(*pImportSettings* as [DatabaseConnection](#), *strTable* as String, *pSchemaDoc* as [Document](#)) as [Document](#)

Return Value

Creates a new document filled with data from the specified database as specified by the schema definition in *pSchemaDoc*.

Description

ImportFromSchema imports data from a database specified in *pImportSettings*. Properties mandatory to be filled out are one of [DatabaseConnection.File](#), [DatabaseConnection.ADOConnection](#) or [DatabaseConnection.ODBCConnection](#). Additionally, you can use [DatabaseConnection.AsAttributes](#), [DatabaseConnection.ExcludeKeys](#) and [NumberDateTimeFormat](#) to further parameterize import. All other properties get ignored.

ImportFromSchema does not use an explicit SQL statement to select the data. Instead, it expects a structure definition of the document to create in form of an XML schema document in *pSchemaDoc*. From this definition the database select statement is automatically deduced. Specify in *strTable* the table name of the import root that will become the root node in the new document.

See [GetDatabaseSettings](#) and [GetDatabaseTables](#) for necessary steps preceding an import from a database based on a schema definition. To create the schema definition file use command 'create database schema' from the 'convert' menu of Authentic Desktop.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1107 Import from database failed.
- 1112 Invalid database specified.
- 1120 Database import failed.
- 1121 Could not create validator for the specified schema.
- 1122 Failed parsing schema for database import.

13.3.2.1.24 ImportFromText

Method: ImportFromText(*pImportSettings* as [TextImportExportSettings](#), *pElementList* as [ElementList](#)) as [Document](#)

Description

ImportFromText imports the text file as specified in *pImportSettings*. The parameter *pElementList* can be used as import filter. Either pass the list returned by a previous call to [GetTextImportElementList](#) or null to import all columns. To avoid import of unnecessary columns use [ElementList.RemoveElement](#) to remove the corresponding field names from *pElementList* before calling `ImportFromText`.

The method returns the newly created document containing the imported data. This document is the same as the active document of Authentic Desktop.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1107 Import from text file failed.
- 1117 Transformation to XML failed.

Example

```
'-----  
' VBA client code fragment - import from text file  
'-----  
    Dim objImpSettings As TextImportExportSettings  
    Set objImpSettings = objSpy.GetTextImportExportSettings  
  
    objImpSettings.ImportFile = strFileName  
    objImpSettings.HeaderRow = False  
  
    Dim objImpDoc As Document  
    On Error Resume Next  
    Set objImpDoc = objSpy.ImportFromText(objImpSettings,  
objSpy.GetTextImportElementList(objImpSettings))  
  
    CheckForError
```

13.3.2.1.25 ImportFromWord

Method: ImportFromWord(*strFile* as String) as [Document](#)

Description

ImportFromWord imports the MS-Word Document *strFile* into a new XML document.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
Import from document failed.

13.3.2.1.26 IsAPISupported

Property: IsAPISupported as Boolean

Description

Returns whether the API is supported in this version or not.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.27 MajorVersion

Property: MajorVersion as Integer

Description

Returns the application version's major number, for example 15 for 2013 versions, and 16 for 2014 versions..

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.28 MinorVersion

Property: MinorVersion as Integer

Description

Returns the application version's minor number.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.29 NewProject

Method: NewProject(*strPath* as String, *bDiscardCurrent* as Boolean)

Description

NewProject creates a new project.

If there is already a project open that has been modified and *bDiscardCurrent* is false, then NewProject() fails.

Errors

- 1111 The application object is no longer valid.
- 1102 A project is already open but *bDiscardCurrent* is true.
- 1103 Creation of new project failed.

13.3.2.1.30 OpenProject

Method: OpenProject(*strPath* as String,*bDiscardCurrent* as Boolean,*bDialog* as Boolean)

Parameters

strPath

Path and file name of the project to open. Can be empty if *bDialog* is true.

bDiscardCurrent

Discard currently open project and possibly lose changes.

bDialog

Show dialogs for user input.

Return Value

None

Description

OpenProject opens an existing project. If there is already a project open that has been modified and *bDiscardCurrent* is false, then OpenProject() fails.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1101 Cannot open specified project.
- 1102 A project is already open but *bDiscardCurrent* is true.

13.3.2.1.31 Parent

Property: Parent as [Application](#) (read-only)

Description

Accesses the Authentic Desktop application object.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.32 Quit

Method: Quit()

Return Value

None

Description

This method terminates Authentic Desktop. All modified documents will be closed without saving the changes. This is also true for an open project.

If Authentic Desktop was automatically started as an automation server by a client program, the application will not shut down automatically when your client program shuts down if a project or any document is still open. Use the Quit method to ensure automatic shut-down.

Errors

1111 The application object is no longer valid.

13.3.2.1.33 ReloadSettings

Method: ReloadSettings

Return Value**Description**

The application settings are reloaded from the registry.

Available with TypeLibrary version 1.5

Errors

1111 The application object is no longer valid.

13.3.2.1.34 RemoveXSLT_XQParameter

Method: RemoveXSLT_XQParameter(index as Long)

Description

Removes the XSLT or XQuery parameter identified by the supplied index.

Errors

1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

13.3.2.1.35 RunMacro

Method: RunMacro(*strMacro* as String)

Return Value**Description**

Calls the specified macro either from the project scripts (if present) or from the global scripts.

Available with TypeLibrary version 1.5

Errors

1111 The application object is no longer valid.

13.3.2.1.36 ScriptingEnvironment

Property: ScriptingEnvironment as IUnknown (read-only)

Description

Reference to any active scripting environment. This property makes it possible to access the TypeLibrary of the XMLSpyFormEditor.exe application which is used as the current scripting environment.

Available with TypeLibrary version 1.5

Errors

1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

13.3.2.1.37 ServicePackVersion

Property: ServicePackVersion as Long

Description

Returns the Service Pack version number of the application. Eg: 1 for 2010 R2 SP1

Errors

1111 The application object is no longer valid.
1100 Invalid address for the return parameter was specified.

13.3.2.1.38 ShowApplication

Method: ShowApplication(*bShow* as Boolean)

Return Value

None

Description

The method shows (*bShow* = True) or hides (*bShow* = False) Authentic Desktop.

Errors

1110 The application object is no longer valid.

13.3.2.1.39 ShowFindInFiles

Method: ShowFindInFiles(pSettings as [FindInFilesDlg](#)) as Boolean

Return Value

Returns false if the user pressed the Cancel button, true otherwise.

Description

Displays the FindInFiles dialog preset with the given settings. The user modifications of the settings are stored in the passed dialog object.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.

13.3.2.1.40 ShowForm

Method: ShowForm(strFormName as String) as Long

Return Value

Returns zero if the user pressed a Cancel button or the form calls TheView.Cancel().

Description

Displays the form strFormName.

Forms, event handlers and macros can be created with the Scripting Environment. Select "Switch to scripting environment" from the **Tools** menu to invoke the Scripting Environment.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.

13.3.2.1.41 Status

Property: Status as [ENUMApplicationStatus](#)

Description

Returns the current status of the running application.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.42 URLDelete

Method: URLDelete(*strURL* as String,*strUser* as String,*strPassword* as String)

Return Value

None

Description

The method deletes the file at the URL *strURL*.

Errors

- 1111 The application object is no longer valid.
- 1109 Error deleting file at specified URL.

13.3.2.1.43 URLMakeDirectory

Method: URLMakeDirectory(*strURL* as String,*strUser* as String,*strPassword* as String)

Return Value

None

Description

The method creates a new directory at the URL *strURL*.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter specified.

13.3.2.1.44 Visible

Property: Visible as VARIANT_BOOL

Description

Sets or gets the visibility attribute of Authentic Desktop. This standard automation property makes usage of [ShowApplication](#) obsolete.

Errors

- 1110 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.45 WarningNumber

Property: WarningNumber as integer

Description

Some methods fill the property `WarningNumber` with additional information if an error occurs.

Currently just [Documents.OpenFile](#) fills this property.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.1.46 `WarningText`

Property: `WarningText` as `String`

Description

Some methods fill the property `WarningText` with additional information if an error occurs.

Currently just [Documents.OpenFile](#) fills this property.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

13.3.2.2 `AuthenticContextMenu`

The context menu interface provides the means for the user to customize the context menus shown in `Authentic`. The interface has the methods listed in this section.

13.3.2.2.1 `CountItems`

Method: `CountItems()` `nItems` as `long`

Return Value

Returns the number of menu items.

Errors

- 2501 Invalid object.

13.3.2.2.2 `DeleteItem`

Method: `DeleteItem(IndexPosition as long)`

Return Value

Deletes the menu item that has the index position submitted in the first parameter.

Errors

- 2501 Invalid object

2502 Invalid index

13.3.2.2.3 GetItemText

Method: `GetItemText(IndexPosition as long) MenuItemName as string`

Return Value

Gets the name of the menu item located at the index position submitted in the first parameter.

Errors

2501 Invalid object
2502 Invalid index

13.3.2.2.4 InsertItem

Method: `InsertItem(IndexPosition as long, MenuItemName as string, MacroName as string)`

Return Value

Inserts a user-defined menu item at the position in the menu specified in the first parameter and having the name submitted in the second parameter. The menu item will start a macro, so a valid macro name must be submitted.

Errors

2501 Invalid object
2502 Invalid index
2503 No such macro
2504 Internal error

13.3.2.2.5 SetItemText

Method: `SetItemText(IndexPosition as long, MenuItemName as string)`

Return Value

Sets the name of the menu item located at the index position submitted in the first parameter.

Errors

2501 Invalid object
2502 Invalid index

13.3.2.3 AuthenticDataTransfer

Renamed from DocEditDataTransfer to AuthenticDataTransfer

The DocEditView object is renamed to OldAuthenticView.
DocEditSelection is renamed to AuthenticSelection.
DocEditEvent is renamed to AuthenticEvent.
DocEditDataTransfer is renamed to AuthenticDataTransfer.

Their usage—except for AuthenticDataTransfer—is no longer recommended. We will continue to support existing functionality for a yet undefined period of time but no new features will be added to these interfaces.

For examples on migrating from DocEdit to Authentic see the description of the different methods and properties of the different DocEdit objects.

Methods

[getData](#)

Properties

[dropEffect](#)
[ownDrag](#)
[type](#)

Description

The events OnDragOver and OnBeforeDrop provide information about the object being dragged with an instance of type AuthenticDataTransfer. It contains a description of the dragged object and its content. The latter is available either as string or a pointer to a COM object supporting the IUnkown interface.

13.3.2.3.1 dropEffect

Property: dropEffect as long

Description

The property stores the drop effect from the default event handler. You can set the drop effect if you change this value and return TRUE for the event handler.

Errors

2101 Invalid address for the return parameter was specified.

13.3.2.3.2 getData

Method: getData() as Variant

Description

Retrieve the data associated with the dragged object. Depending on [AuthenticDataTransfer.type](#), that data is either a string or a COM interface pointer of type IUnknown.

Errors

2101 Invalid address for the return parameter was specified.

13.3.2.3.3 ownDrag

Property: ownDrag as Boolean (read-only)

Description

The property is TRUE if the current dragging source comes from inside Authentic View.

Errors

2101 Invalid address for the return parameter was specified.

13.3.2.3.4 type

Property: type as String (read-only)

Description

Holds the type of data you get with the [DocEditDataTransfer.getData](#) method.

Currently supported data types are:

OWN	data from Authentic View itself
TEXT	plain text
UNICODETEXT	plain text as UNICODE

Errors

2101 Invalid address for the return parameter was specified.

13.3.2.4 AuthenticEventContext

The `EventContext` interface gives access to many properties of the context in which a macro is executed.

13.3.2.4.1 EvaluateXPath

Method: EvaluateXPath (strExpression as string) as strValue as string

Return Value

The method evaluates the XPath expression in the context of the node within which the event was triggered and returns a string.

Description

`EvaluateXPath()` executes an XPath expression with the given event context. The result is returned as a string, in the case of a sequence it is a space-separated string.

Errors

- 2201 Invalid object.
- 2202 No context.
- 2209 Invalid parameter.
- 2210 Internal error.
- 2211 XPath error.

13.3.2.4.2 `GetEventContextType`

Method: `GetEventContextType()` Type as `AuthenticEventContextType` enumeration

Return Value

Returns the context node type.

Description

`GetEventContextType` allows the user to determine whether the macro is in an XML node or in an XPath atomic item context. The enumeration `AuthenticEventContextType` is defined as follows:

`authenticEventContextXML`,
`authenticEventContextAtomicItem`,
`authenticEventContextOther`

If the context is a normal XML node, the `GetXMLNode()` function gives access to it (returns `NULL` if not).

Errors

- 2201 Invalid object.
- 2202 No context.
- 2209 Invalid parameter.

13.3.2.4.3 `GetNormalizedTextValue`

Method: `GetNormalizedTextValue()` `strValue` as string

Return Value

Returns the value of the current node as string

Errors

- 2201 Invalid object.
- 2202 No context.
- 2203 Invalid context
- 2209 Invalid parameter.

13.3.2.4.4 GetVariableValue

Method: `GetVariableValue(strName as string) strValue as string`

Return Value

Gets the value of the variable submitted as the parameter.

Description

`GetVariableValue` gets the variable's value in the scope of the context.

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

Errors

- 2201 Invalid object.
- 2202 No context.
- 2204 No such variable in scope
- 2205 Variable cannot be evaluated
- 2206 Variable returns sequence
- 2209 Invalid parameter

13.3.2.4.5 GetXMLNode

Method: `GetXMLNode() Node as XMLData object`

Return Value

Returns the context XML node or `NULL`

Errors

- 2201 Invalid object.
- 2202 No context.
- 2203 Invalid context
- 2209 Invalid parameter.

13.3.2.4.6 IsAvailable

Method: `IsAvailable() as Boolean`

Return Value

Returns true if `EventContext` is set, false otherwise.

Errors

- 2201 Invalid object.

13.3.2.4.7 SetVariableValue

Method: `SetVariableValue(strName as string, strValue as string)`

Return Value

Sets the value (second parameter) of the variable submitted in the first parameter.

Description

`SetVariableValue` sets the variable's value in the scope of the context.

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

Errors

- 2201 Invalid object.
- 2202 No context.
- 2204 No such variable in scope
- 2205 Variable cannot be evaluated
- 2206 Variable returns sequence
- 2207 Variable read-only
- 2208 No modification allowed

13.3.2.5 AuthenticRange

The first table lists the properties and methods of `AuthenticRange` that can be used to navigate through the document and select specific portions.

Properties	Methods	
Application	Clone	MoveBegin
FirstTextPosition	CollapsToBegin	MoveEnd
FirstXMLData	CollapsToEnd	NextCursorPosition
FirstXMLDataOffset	ExpandTo	PreviousCursorPosition
LastTextPosition	Goto	Select
LastXMLData	GotoNext	SelectNext
LastXMLDataOffset	GotoPrevious	SelectPrevious
Parent	IsEmpty	SetFromRange
	IsEqual	

The following table lists the content modification methods, most of which can be found on the right/button mouse menu.

Properties	Edit operations	Dynamic table operations
Text	Copy	AppendRow
	Cut	DeleteRow
	Delete	DuplicateRow
	IsCopyEnabled	InsertRow
	IsCutEnabled	IsFirstRow
	IsDeleteEnabled	IsInDynamicTable
	IsPasteEnabled	IsLastRow
	Paste	MoveRowDown
		MoveRowUp

The following methods provide the functionality of the Authentic entry helper windows for range objects.

Operations of the entry helper windows		
Elements	Attributes	Entities
CanPerformActionWith	GetElementAttributeValue	GetEntityNames
CanPerformAction	GetElementAttributeNames	InsertEntity
PerformAction	GetElementHierarchy	
	HasElementAttribute	
	IsTextStateApplied	
	SetElementAttributeValue	

Description

AuthenticRange objects are the 'cursor' selections of the automation interface. You can use them to point to any cursor position in the Authentic view, or select a portion of the document. The operations available for AuthenticRange objects then work on this selection in the same way, as the corresponding operations of the user interface do with the current user interface selection. The main difference is that you can use an arbitrary number of AuthenticRange objects at the same time, whereas there is exactly one cursor selection in the user interface.

To get to an initial range object use [AuthenticView.Selection](#), to obtain a range corresponding with the current cursor selection in the user interface. Alternatively, some trivial ranges are accessible via the read/only properties [AuthenticView.DocumentBegin](#), [AuthenticView.DocumentEnd](#), and [AuthenticView.WholeDocument](#). The most flexible method is [AuthenticView.Goto](#), which allows navigation to a specific portion of the document within one call. For more complex selections, combine the above with the various navigation methods on range objects listed in the first table on this page.

Another method to select a portion of the document is to use the position properties of the range object. Two positioning systems are available and can be combined arbitrarily:

- **Absolute** text cursor positions, starting with position 0 at the document beginning, can be set and retrieved for the beginning and end of a range. For more information see [FirstTextPosition](#) and [LastTextPosition](#). This method requires complex internal calculations and should be used with care.
- The **XMLData** element and a text position inside this element, can be set and retrieved for the beginning and end of a range. For more information see [FirstXMLData](#), [FirstXMLDataOffset](#), [LastXMLData](#), and [LastXMLDataOffset](#). This method is very efficient but requires knowledge of the underlying document structure. It can be used to locate XMLData objects and perform operations on them otherwise not accessible through the user interface.

Modifications to the document content can be achieved by various methods:

- The [Text](#) property allows you to retrieve the document text selected by the range object. If set, the selected document text gets replaced with the new text.
- The standard document edit functions [Cut](#), [Copy](#), [Paste](#) and [Delete](#).
- Table operations for tables that can grow dynamically.
- Methods that map the functionality of the Authentic entry helper windows.
- Access to the [XMLData](#) objects of the underlying document to modify them directly.

13.3.2.5.1 AppendRow

Method: AppendRow() as Boolean

Description

If the beginning of the range is inside a dynamic table, this method inserts a new row at the end of the selected table. The selection of the range is modified to point to the beginning of the new row. The function returns *true* if the append operation was successful, otherwise *false*.

Errors

- 2001 The authentic range object or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

Examples

```
'-----
' Scripting environment - VBScript
' Append row at end of current dynamically growable table
'-----
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' check if we can insert something
If objRange.IsInDynamicTable Then
    objRange.AppendRow
    ' objRange points to beginning of new row
    objRange.Select
End If
```

13.3.2.5.2 Application

Property: Application as [Application](#) (read-only)

Description

Accesses the Authentic Desktop application object.

Errors

- 2001 The authentic range object or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.3 CanPerformAction

Method: CanPerformAction (*eAction* as [SPYAuthenticActions](#), *strElementName* as String) as Boolean

Description

CanPerformAction and its related methods enable access to the entry-helper functions of Authentic. This function allows easy and consistent modification of the document content, without having to know exactly where the modification will take place. The beginning of the range object is used to locate the next valid location where the specified action can be performed. If the location can be found, the method returns *True*, otherwise it returns *False*.

HINT: To find out all valid element names for a given action, use [CanPerformActionWith](#).

Errors

- 2001 The authentic range object or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.
- 2007 Invalid action was specified.

Examples

See [PerformAction](#).

13.3.2.5.4 CanPerformActionWith

Method: CanPerformActionWith (*eAction* as [SPYAuthenticActions](#), *out_arrElementNames* as Variant)

Description

PerformActionWith and its related methods, enable access to the entry-helper functions of Authentic. This function allows easy and consistent modification of the document content without having to know exactly where the modification will take place.

This method returns an array of those element names that the specified action can be performed with.

HINT: To apply the action use [CanPerformActionWith](#).

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

2007 Invalid action was specified.

Examples

See [PerformAction](#).

13.3.2.5.5 Clone

Method: Clone() as [AuthenticRange](#)

Description

Returns a copy of the range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.6 CollapsToBegin

Method: CollapsToBegin() as [AuthenticRange](#)

Description

Sets the end of the range object to its begin. The method returns the modified range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.7 CollapsToEnd

Method: CollapsToEnd() as [AuthenticRange](#)

Description

Sets the beginning of the range object to its end. The method returns the modified range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.8 Copy

Method: Copy() as Boolean

Description

Returns *False* if the range contains no portions of the document that may be copied.

Returns *True* if text, and in case of fully selected XML elements the elements as well, has been copied to the copy/paste buffer.

Errors

- 2001 The authentic range object or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.9 Cut

Method: Cut() as Boolean

Description

Returns *False* if the range contains portions of the document that may not be deleted.

Returns *True* after text, and in case of fully selected XML elements the elements as well, has been deleted from the document and saved in the copy/paste buffer.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.10 Delete

Method: Delete() as Boolean

Description

Returns *False* if the range contains portions of the document that may not be deleted.

Returns *True* after text, and in case of fully selected XML elements the elements as well, has been deleted from the document.

Errors

- 2001 The authentic range object or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.11 DeleteRow

Method: DeleteRow() as Boolean

Description

If the beginning of the range is inside a dynamic table, this method deletes the selected row. The selection of the range gets modified to point to the next element after the deleted row. The function returns *true*, if the delete operation was successful, otherwise *false*.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

Examples

```
'-----  
' Scripting environment - VBScript  
' Delete selected row from dynamically growing table  
'-----  
Dim objRange  
' we assume that the active document is open in authentic view mode  
Set objRange = Application.ActiveDocument.AuthenticView.Selection  
  
' check if we are in a table  
If objRange.IsInDynamicTable Then  
    objRange.DeleteRow  
End If
```

13.3.2.5.12 DuplicateRow

Method: DuplicateRow() as Boolean

Description

If the beginning of the range is inside a dynamic table, this method inserts a duplicate of the current row after the selected one. The selection of the range gets modified to point to the beginning of the new row. The function returns *true* if the duplicate operation was successful, otherwise *false*.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

Examples

```
'-----  
' Scripting environment - VBScript  
' duplicate row in current dynamically growable table  
'-----  
Dim objRange  
' we assume that the active document is open in authentic view mode  
Set objRange = Application.ActiveDocument.AuthenticView.Selection  
  
' check if we can insert something  
If objRange.IsInDynamicTable Then  
    objRange.DuplicateRow  
    ' objRange points to beginning of new row  
    objRange.Select  
End If
```

13.3.2.5.13 EvaluateXPath

Method: EvaluateXPath (strExpression as string) strValue as string

Return Value

The method returns a string

Description

`EvaluateXPath()` executes an XPath expression with the context node being the beginning of the range selection. The result is returned as a string, in the case of a sequence it is a space-separated string. If XML context node is irrelevant, the user may provide any node, like `AuthenticView.XMLDataRoot`.

Errors

- 2001 Invalid object
- 2005 Invalid parameter
- 2008 Internal error
- 2202 Missing context node
- 2211 XPath error

13.3.2.5.14 ExpandTo

Method: `ExpandTo` (*eKind* as [SPYAuthenticElementKind](#)), as [AuthenticRange](#)

Description

Selects the whole element of type *eKind*, that starts at, or contains, the first cursor position of the range. The method returns the modified range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2003 Range expansion would be beyond end of document.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.15 FirstTextPosition

Property: `FirstTextPosition` as Long

Description

Set or get the left-most text position index of the range object. This index is always less or equal to [LastTextPosition](#). Indexing starts with 0 at document beginning, and increments with every different position that the text cursor can occupy. Incrementing the text position by 1, has the same effect as the cursor-right key. Decrementing the text position by 1 has the same effect as the cursor-left key.

If you set `FirstTextPosition` to a value greater than the current [LastTextPosition](#), [LastTextPosition](#) gets set to the new `FirstTextPosition`.

HINT: Use text cursor positions with care, since this is a costly operation compared to XMLData based cursor positioning.

Errors

- 2001 The authentic range object, or its related view object is not valid.
- 2005 Invalid address for the return parameter was specified.
- 2006 A text position outside the document was specified.

Examples

```
'-----  
' Scripting environment - VBScript
```



```
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

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

' let's create a range that selects the whole document
' in an inefficient way
Dim objRange
' we need to get a (any) range object first
Set objRange = objAuthenticView.DocumentBegin
objRange.FirstTextPosition = nDocStartPosition
objRange.LastTextPosition = nDocEndPosition

' let's check if we got it right
If objRange.IsEqual(objAuthenticView.WholeDocument) Then
    MsgBox "Test using direct text cursor positioning was ok"
Else
    MsgBox "Oops!"
End If
```

13.3.2.5.16 FirstXMLData

Property: FirstXMLData as [XMLData](#)

Description

Set or get the first XMLData element in the underlying document that is partially, or completely selected by the range. The exact beginning of the selection is defined by the [FirstXMLDataOffset](#) attribute.

Whenever you set FirstXMLData to a new data object, [FirstXMLDataOffset](#) gets set to the first cursor position inside this element. Only XMLData objects that have a cursor position may be used. If you set FirstXMLData / [FirstXMLDataOffset](#) selects a position greater than the current [LastXMLData](#) / [LastXMLDataOffset](#), the latter gets moved to the new start position.

HINT: You can use the [FirstXMLData](#) and [LastXMLData](#) properties to directly access and manipulate the underlying XML document in those cases where the methods available with the [AuthenticRange](#) object are not sufficient.

Errors

- 2001 The authentic range object, or its related view object is not valid.
- 2005 Invalid address for the return parameter was specified.
- 2008 Internal error
- 2009 The XMLData object cannot be accessed.

Examples

```
' -----
' Scripting environment - VBScript
' show name of currently selected XMLData element
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
```

```
Set objAuthenticView = Application.ActiveDocument.AuthenticView
```

```
Dim objXMLData
Set objXMLData = objAuthenticView.Selection.FirstXMLData
' authentic view adds a 'text' child element to elements
' of the document which have content. So we have to go one
' element up.
Set objXMLData = objXMLData.Parent
MsgBox "Current selection selects element " & objXMLData.Name
```

13.3.2.5.17 FirstXMLDataOffset

Property: FirstXMLDataOffset as Long

Description

Set or get the cursor position offset inside [FirstXMLData](#) element for the beginning of the range. Offset positions are based on the characters returned by the [Text](#) property, and start with 0. When setting a new offset, use -1 to set the offset to the last possible position in the element. The following cases require specific attention:

- The textual form of entries in Combo Boxes, Check Boxes and similar controls can be different from what you see on screen. Although the data offset is based on this text, there only two valid offset positions, one at the beginning and one at the end of the entry. An attempt to set the offset to somewhere in the middle of the entry, will result in the offset being set to the end.
- The textual form of XML Entities might differ in length from their representation on the screen. The offset is based on this textual form.

If [FirstXMLData](#) / [FirstXMLDataOffset](#) selects a position after the current [LastXMLData](#) / [LastXMLDataOffset](#), the latter gets moved to the new start position.

Errors

- 2001 The authentic range object, or its related view object is not valid.
- 2005 Invalid offset was specified.
Invalid address for the return parameter was specified.

Examples

```
' -----
' Scripting environment - VBScript
' Select the complete text of an XMLData element
' using XMLData based selection and ExpandTo
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

' first we use the XMLData based range properties
' to select all text of the first XMLData element
' in the current selection
Dim objRange
Set objRange = objAuthenticView.Selection
objRange.FirstXMLDataOffset = 0 ' start at beginning of element text
objRange.LastXMLData = objRange.FirstXMLData ' select only one element
```

```
objRange.LastXMLDataOffset = -1 ' select till its end

' the same can be achieved with the ExpandTo method
Dim objRange2
Set objRange2 = objAuthenticView.Selection.ExpandTo(spyAuthenticTag)

' were we successful?
If objRange.IsEqual(objRange2) Then
    objRange.Select()
Else
    MsgBox "Oops"
End If
```

13.3.2.5.18 GetElementAttributeNames

Method: GetElementAttributeNames (*strElementName* as String, *out_arrAttributeNames* as Variant)

Description

Retrieve the names of all attributes for the enclosing element with the specified name. Use the element/attribute pairs, to set or get the attribute value with the methods [GetElementAttributeValue](#) and [SetElementAttributeValue](#).

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid element name was specified.
Invalid address for the return parameter was specified.

Examples

See [SetElementAttributeValue](#).

13.3.2.5.19 GetElementAttributeValue

Method: GetElementAttributeValue (*strElementName* as String, *strAttributeName* as String) as String

Description

Retrieve the value of the attribute specified in *strAttributeName*, for the element identified with *strElementName*. If the attribute is supported but has no value assigned, the empty string is returned. To find out the names of attributes supported by an element, use [GetElementAttributeNames](#), or [HasElementAttribute](#).

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid element name was specified.
Invalid attribute name was specified.
Invalid address for the return parameter was specified.

Examples

See [SetElementAttributeValue](#).

13.3.2.5.20 GetElementHierarchy

Method: GetElementHierarchy (*out_arrElementNames* as Variant)

Description

Retrieve the names of all XML elements that are parents of the current selection. Inner elements get listed before enclosing elements. An empty list is returned whenever the current selection is not inside a single XMLData element.

The names of the element hierarchy, together with the range object uniquely identify XMLData elements in the document. The attributes of these elements can be directly accessed by [GetElementAttributeNames](#), and related methods.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.21 GetEntityNames

Method: GetEntityNames (*out_arrEntityNames* as Variant)

Description

Retrieve the names of all defined entities. The list of retrieved entities is independent of the current selection, or location. Use one of these names with the [InsertEntity](#) function.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

Examples

See: [GetElementHierarchy](#) and [InsertEntity](#).

13.3.2.5.22 GetVariableValue

Method: GetVariableValue (*strName* as string) *strVal* as string

Return Value

Gets the value of the variable named as the method's parameter.

Errors

- 2001 Invalid object.
- 2202 No context.
- 2204 No such variable in scope
- 2205 Variable cannot be evaluated
- 2206 Variable returns sequence

2209 Invalid parameter

13.3.2.5.23 Goto

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

Description

Sets the range to point to the beginning of the *nCount* element of type *eKind*. The start position is defined by the parameter *eFrom*.

Use positive values for *nCount* to navigate to the document end. Use negative values to navigate to the beginning of the document. The method returns the modified range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2003 Target lies after end of document.
- 2004 Target lies before begin of document.
- 2005 Invalid element kind specified.
Invalid start position specified.
Invalid address for the return parameter was specified.

13.3.2.5.24 GotoNext

Method: GotoNext (*eKind* as [SPYAuthenticElementKind](#)) as [AuthenticRange](#)

Description

Sets the range to the beginning of the next element of type *eKind*. The method returns the modified range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2003 Target lies after end of document.
- 2005 Invalid element kind specified.
Invalid address for the return parameter was specified.

Examples

```
'-----  
' Scripting environment - VBScript  
' Scan through the whole document word-by-word  
'-----  
Dim objAuthenticView  
' we assume that the active document is open in authentic view mode  
Set objAuthenticView = Application.ActiveDocument.AuthenticView  
  
Dim objRange  
Set objRange = objAuthenticView.DocumentBegin  
Dim bEndOfDocument
```

```
bEndOfDocument = False

On Error Resume Next
While Not bEndOfDocument
    objRange.GotoNext(spyAuthenticWord).Select
    If ((Err.number - vbObjecterror) = 2003) Then
        bEndOfDocument = True
        Err.Clear
    ElseIf (Err.number <> 0) Then
        Err.Raise ' forward error
    End If
Wend
```

13.3.2.5.25 GotoNextCursorPosition

Method: GotoNextCursorPosition() as [AuthenticRange](#)

Description

Sets the range to the next cursor position after its current end position. Returns the modified object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2003 Target lies after end of document.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.26 GotoPrevious

Method: GotoPrevious (eKind as [SPYAuthenticElementKind](#)) as [AuthenticRange](#)

Description

Sets the range to the beginning of the element of type eKind which is before the beginning of the current range. The method returns the modified range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2004 Target lies before beginning of document.
- 2005 Invalid element kind specified.
Invalid address for the return parameter was specified.

Examples

```
' -----
' Scripting environment - VBScript
' Scan through the whole document tag-by-tag
' -----

Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentEnd
```

```
Dim bBeginOfDocument
bBeginOfDocument = False

On Error Resume Next
While Not bBeginOfDocument
    objRange.GotoPrevious(spyAuthenticTag).Select
    If ((Err.number - vbObjecterror) = 2004) Then
        bBeginOfDocument = True
        Err.Clear
    ElseIf (Err.number <> 0) Then
        Err.Raise ' forward error
    End If
Wend
```

13.3.2.5.27 GotoPreviousCursorPosition

Method: GotoPreviousCursorPosition() as [AuthenticRange](#)

Description

Set the range to the cursor position immediately before the current position. Returns the modified object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2004 Target lies before begin of document.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.28 HasElementAttribute

Method: HasElementAttribute (*strElementName* as String, *strAttributeName* as String) as Boolean

Description

Tests if the enclosing element with name *strElementName*, supports the attribute specified in *strAttributeName*.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid element name was specified.
Invalid address for the return parameter was specified.

13.3.2.5.29 InsertEntity

Method: InsertEntity (*strEntityName* as String)

Description

Replace the ranges selection with the specified entity. The specified entity must be one of the entity names returned by [GetEntityNames](#).

Errors

- 2001 The authentic range object, or its related view object is no longer valid.

2005 Unknown entry name was specified.

Examples

```
'-----  
' Scripting environment - VBScript  
' Insert the first entity in the list of available entities  
'-----  
Dim objRange  
' we assume that the active document is open in authentic view mode  
Set objRange = Application.ActiveDocument.AuthenticView.Selection  
  
' first we get the names of all available entities as they  
' are shown in the entry helper of XMLSpy  
Dim arrEntities  
objRange.GetEntityNames arrEntities  
  
' we insert the first one of the list  
If UBound(arrEntities) >= 0 Then  
    objRange.InsertEntity arrEntities(0)  
Else  
    MsgBox "Sorry, no entities are available for this document"  
End If
```

13.3.2.5.30 InsertRow

Method: InsertRow() as Boolean

Description

If the beginning of the range is inside a dynamic table, this method inserts a new row before the current one. The selection of the range gets modified to point to the beginning of the newly inserted row. The function returns *true* if the insert operation was successful, otherwise *false*.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

Examples

```
'-----  
' Scripting environment - VBScript  
' Insert row at beginning of current dynamically growing table  
'-----  
Dim objRange  
' we assume that the active document is open in authentic view mode  
Set objRange = Application.ActiveDocument.AuthenticView.Selection  
  
' check if we can insert something  
If objRange.IsInDynamicTable Then  
    objRange.InsertRow  
    ' objRange points to beginning of new row  
    objRange.Select  
End If
```


13.3.2.5.31 IsCopyEnabled

Property: IsCopyEnabled as Boolean (read-only)

Description

Checks if the copy operation is supported for this range.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.32 IsCutEnabled

Property: IsCutEnabled as Boolean (read-only)

Description

Checks if the cut operation is supported for this range.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.33 IsDeleteEnabled

Property: IsDeleteEnabled as Boolean (read-only)

Description

Checks if the delete operation is supported for this range.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.34 IsEmpty

Method: IsEmpty() as Boolean

Description

Tests if the first and last position of the range are equal.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.35 IsEqual

Method: IsEqual (*objCmpRange* as [AuthenticRange](#)) as Boolean

Description

Tests if the start and end of both ranges are the same.

Errors

- 2001 One of the two range objects being compared, is invalid.
- 2005 Invalid address for a return parameter was specified.

13.3.2.5.36 IsFirstRow

Property: IsFirstRow as Boolean (read-only)

Description

Test if the range is in the first row of a table. Which table is taken into consideration depends on the extent of the range. If the selection exceeds a single row of a table, the check is if this table is the first element in an embedding table. See the entry helpers of the user manual for more information.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.37 IsInDynamicTable

Method: IsInDynamicTable() as Boolean

Description

Test if the whole range is inside a table that supports the different row operations like 'insert', 'append', duplicate, etc.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.38 IsLastRow

Property: IsLastRow as Boolean (read-only)

Description

Test if the range is in the last row of a table. Which table is taken into consideration depends on the extent of the range. If the selection exceeds a single row of a table, the check is if this table is the last element in an embedding table. See the entry helpers of the user manual for more information.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.39 IsPasteEnabled

Property: IsPasteEnabled as Boolean (read-only)

Description

Checks if the paste operation is supported for this range.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.40 IsSelected

Property: IsSelected as Boolean

Description

Returns true() if selection is present. The selection range still can be empty: that happens when e.g. only the cursor is set.

13.3.2.5.41 IsTextStateApplied

Method: IsTextStateApplied (*i_strElementName* as String) as Boolean

Description

Checks if all the selected text is embedded into an XML Element with name *i_strElementName*. Common examples for the parameter *i_strElementName* are "strong", "bold" or "italic".

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.42 LastTextPosition

Property: LastTextPosition as Long

Description

Set or get the rightmost text position index of the range object. This index is always greater or equal to [FirstTextPosition](#). Indexing starts with 0 at the document beginning, and increments with every different position that the text cursor can occupy. Incrementing the test position by 1, has the same effect as the cursor-right key. Decreasing the test position by 1 has the same effect as the cursor-left key.

If you set `LastTextPosition` to a value less than the current [FirstTextPosition](#), `FirstTextPosition` gets set to the new `LastTextPosition`.

HINT: Use text cursor positions with care, since this is a costly operation compared to XMLData based cursor positioning.

Errors

- 2001 The authentic range object, or its related view object is not valid.
- 2005 Invalid address for the return parameter was specified.
- 2006 A text position outside the document was specified.

Examples

```
'-----  
' Scripting environment - VBScript  
'-----  
Dim objAuthenticView  
' we assume that the active document is open in authentic view mode  
Set objAuthenticView = Application.ActiveDocument.AuthenticView  
  
nDocStartPosition = objAuthenticView.DocumentBegin.FirstTextPosition  
nDocEndPosition = objAuthenticView.DocumentEnd.FirstTextPosition  
  
' let's create a range that selects the whole document  
' in an inefficient way  
Dim objRange  
' we need to get a (any) range object first  
Set objRange = objAuthenticView.DocumentBegin  
objRange.FirstTextPosition = nDocStartPosition  
objRange.LastTextPosition = nDocEndPosition  
  
' let's check if we got it right  
If objRange.IsEqual(objAuthenticView.WholeDocument) Then  
    MsgBox "Test using direct text cursor positioning was ok"  
Else  
    MsgBox "Oops!"  
End If
```

13.3.2.5.43 LastXMLData

Property: LastXMLData as [XMLData](#)

Description

Set or get the last XMLData element in the underlying document that is partially or completely selected by the range. The exact end of the selection is defined by the [LastXMLDataOffset](#) attribute.

Whenever you set `LastXMLData` to a new data object, [LastXMLDataOffset](#) gets set to the last cursor position inside this element. Only XMLData objects that have a cursor position may be used. If you set `LastXMLData` / [LastXMLDataOffset](#), select a position less than the current [FirstXMLData](#) / [FirstXMLDataOffset](#), the latter gets moved to the new end position.

HINT: You can use the [FirstXMLData](#) and [LastXMLData](#) properties to directly access and manipulate the underlying XML document in those cases, where the methods available with the [AuthenticRange](#) object are not sufficient.

Errors

- 2001 The authentic range object, or its related view object is not valid.
- 2005 Invalid address for the return parameter was specified.
- 2008 Internal error
- 2009 The XMLData object cannot be accessed.

13.3.2.5.44 LastXMLDataOffset

Property: LastXMLDataOffset as Long

Description

Set or get the cursor position inside [LastXMLData](#) element for the end of the range.

Offset positions are based on the characters returned by the [Text](#) property and start with 0. When setting a new offset, use -1 to set the offset to the last possible position in the element. The following cases require specific attention:

- The textual form of entries in Combo Boxes, Check Boxes and similar controls can be different from what you see on the screen. Although, the data offset is based on this text, there only two valid offset positions, one at the beginning and one at the end of the entry. An attempt to set the offset to somewhere in the middle of the entry, will result in the offset being set to the end.
- The textual form of XML Entities might differ in length from their representation on the screen. The offset is based on this textual form.

If [LastXMLData](#) / [LastXMLDataOffset](#) selects a position before [FirstXMLData](#) / [FirstXMLDataOffset](#), the latter gets moved to the new end position.

Errors

- 2001 The authentic range object, or its related view object is not valid.
- 2005 Invalid offset was specified.
Invalid address for the return parameter was specified.

Examples

```
'-----
' Scripting environment - VBScript
' Select the complete text of an XMLData element
' using XMLData based selection and ExpandTo
'-----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

' first we use the XMLData based range properties
' to select all text of the first XMLData element
' in the current selection
Dim objRange
Set objRange = objAuthenticView.Selection
objRange.FirstXMLDataOffset = 0 ' start at beginning of element text
```

```
objRange.LastXMLData = objRange.FirstXMLData ' select only one element
objRange.LastXMLDataOffset = -1 ' select till its end
```

' the same can be achieved with the ExpandTo method

```
Dim objRange2
```

```
Set objRange2 = objAuthenticView.Selection.ExpandTo(spyAuthenticTag)
```

' were we successful?

```
If objRange.IsEqual(objRange2) Then
```

```
    objRange.Select()
```

```
Else
```

```
    MsgBox "Ooops"
```

```
End If
```

13.3.2.5.45 MoveBegin

Method: MoveBegin (eKind as [SPYAuthenticElementKind](#), nCount as Long) as [AuthenticRange](#)

Description

Move the beginning of the range to the beginning of the nCount element of type eKind. Counting starts at the current beginning of the range object.

Use positive numbers for nCount to move towards the document end, use negative numbers to move towards document beginning. The end of the range stays unmoved, unless the new beginning would be larger than it. In this case, the end is moved to the new beginning. The method returns the modified range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2003 Target lies after end of document.
- 2004 Target lies before beginning of document.
- 2005 Invalid element kind specified.
Invalid address for the return parameter was specified.

13.3.2.5.46 MoveEnd

Method: MoveEnd (eKind as [SPYAuthenticElementKind](#), nCount as Long) as [AuthenticRange](#)

Description

Move the end of the range to the begin of the nCount element of type eKind. Counting starts at the current end of the range object.

Use positive numbers for nCount to move towards the document end, use negative numbers to move towards document beginning. The beginning of the range stays unmoved, unless the new end would be less than it. In this case, the beginning gets moved to the new end. The method returns the modified range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2003 Target lies after end of document.
- 2004 Target lies before begin of document.
- 2005 Invalid element kind specified.
Invalid address for the return parameter was specified.

13.3.2.5.47 MoveRowDown

Method: MoveRowDown() as Boolean

Description

If the beginning of the range is inside a dynamic table and selects a row which is not the last row in this table, this method swaps this row with the row immediately below. The selection of the range moves with the row, but does not otherwise change. The function returns *true* if the move operation was successful, otherwise *false*.

Errors

- 2001 The authentic range object or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.48 MoveRowUp

Method: MoveRowUp() as Boolean

Description

If the beginning of the range is inside a dynamic table and selects a row which is not the first row in this table, this method swaps this row with the row above. The selection of the range moves with the row, but does not change otherwise. The function returns *true* if the move operation was successful, otherwise *false*.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.49 Parent

Property: Parent as [AuthenticView](#) (read-only)

Description

Access the view that owns this range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.50 Paste

Method: Paste() as Boolean

Description

Returns *False* if the copy/paste buffer is empty, or its content cannot replace the current selection.

Otherwise, deletes the current selection, inserts the content of the copy/paste buffer, and returns *True*.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.5.51 PerformAction

Method: PerformAction (*eAction* as [SPYAuthenticActions](#), *strElementName* as String) as Boolean

Description

PerformAction and its related methods, give access to the entry-helper functions of Authentic. This function allows easy and consistent modification of the document content without a need to know exactly where the modification will take place. The beginning of the range object is used to locate the next valid location where the specified action can be performed. If no such location can be found, the method returns *False*. Otherwise, the document gets modified and the range points to the beginning of the modification.

HINT: To find out element names that can be passed as the second parameter use [CanPerformActionWith](#).

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.
- 2007 Invalid action was specified.

Examples

```
'-----
' Scripting environment - VBScript
' Insert the innermost element
'-----
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' we determine the elements that can be inserted at the current position
Dim arrElements()
objRange.CanPerformActionWith spyAuthenticInsertBefore, arrElements

' we insert the first (innermost) element
If UBound(arrElements) >= 0 Then
    objRange.PerformAction spyAuthenticInsertBefore, arrElements(0)
    ' objRange now points to the beginning of the inserted element
    ' we set a default value and position at its end
    objRange.Text = "Hello"
    objRange.ExpandTo(spyAuthenticTag).CollapsToEnd().Select
Else
    MsgBox "Can't insert any elements at current position"
End If
```


13.3.2.5.52 Select

Method: Select()

Description

Makes this range the current user interface selection. You can achieve the same result using:
'objRange.Parent.Selection = objRange'

Errors

2001 The authentic range object or its related view object is no longer valid.

Examples

```
'-----  
' Scripting environment - VBScript  
'-----  
Dim objAuthenticView  
' we assume that the active document is open in authentic view mode  
Set objAuthenticView = Application.ActiveDocument.AuthenticView  
  
' set current selection to end of document  
objAuthenticView.DocumentEnd.Select()
```

13.3.2.5.53 SelectNext

Method: SelectNext (eKind as [SPYAuthenticElementKind](#)) as [AuthenticRange](#)

Description

Selects the element of type eKind after the current end of the range. The method returns the modified range object.

Errors

2001 The authentic range object, or its related view object is no longer valid.
2003 Target lies after end of document.
2005 Invalid element kind specified.
Invalid address for the return parameter was specified.

Examples

```
'-----  
' Scripting environment - VBScript  
' Scan through the whole document word-by-word  
'-----  
Dim objAuthenticView  
' we assume that the active document is open in authentic view mode  
Set objAuthenticView = Application.ActiveDocument.AuthenticView  
  
Dim objRange  
Set objRange = objAuthenticView.DocumentBegin  
Dim bEndOfDocument  
bEndOfDocument = False
```

```
On Error Resume Next
While Not bEndOfDocument
    objRange.SelectNext(spyAuthenticWord).Select
    If ((Err.number - vbObjecterror) = 2003) Then
        bEndOfDocument = True
        Err.Clear
    ElseIf (Err.number <> 0) Then
        Err.Raise ' forward error
    End If
Wend
```

13.3.2.5.54 SelectPrevious

Method: GotoPrevious (eKind as [SPYAuthenticElementKind](#)) as [AuthenticRange](#)

Description

Selects the element of type eKind before the current beginning of the range. The method returns the modified range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2004 Target lies before begin of document.
- 2005 Invalid element kind specified.
Invalid address for the return parameter was specified.

Examples

```
' -----
' Scripting environment - VBScript
' Scan through the whole document tag-by-tag
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentEnd
Dim bBeginOfDocument
bBeginOfDocument = False

On Error Resume Next
While Not bBeginOfDocument
    objRange.SelectPrevious(spyAuthenticTag).Select
    If ((Err.number - vbObjecterror) = 2004) Then
        bBeginOfDocument = True
        Err.Clear
    ElseIf (Err.number <> 0) Then
        Err.Raise ' forward error
    End If
Wend
```

13.3.2.5.55 SetElementAttributeValue

Method: SetElementAttributeValue (*strElementName* as String, *strAttributeName* as String, *strAttributeValue* as String)

Description

Set the value of the attribute specified in *strAttributeName* for the element identified with *strElementName*. If the attribute is supported but has no value assigned, the empty string is returned. To find out the names of attributes supported by an element, use [GetElementAttributeNames](#), or [HasElementAttribute](#).

Errors

- 2001 The authentic range object or its related view object is no longer valid.
- 2005 Invalid element name was specified.
Invalid attribute name was specified.
Invalid attribute value was specified.

Examples

```
' -----  
' Scripting environment - VBScript  
' Get and set element attributes  
' -----  
  
Dim objRange  
' we assume that the active document is open in authentic view mode  
Set objRange = Application.ActiveDocument.AuthenticView.Selection  
  
' first we find out all the elements below the beginning of the range  
Dim arrElements  
objRange.GetElementHierarchy arrElements  
  
If IsArray(arrElements) Then  
    If UBound(arrElements) >= 0 Then  
        ' we use the top level element and find out its valid attributes  
        Dim arrAttrs()  
        objRange.GetElementAttributeNames arrElements(0), arrAttrs  
  
        If UBound(arrAttrs) >= 0 Then  
            ' we retrieve the current value of the first valid attribute  
            Dim strAttrVal  
            strAttrVal = objRange.GetElementAttributeValue (arrElements(0), arrAttrs(0))  
            msgbox "current value of " & arrElements(0) & "/" & arrAttrs(0) & " is: " & strAttrVal  
  
            ' we change this value and read it again  
            strAttrVal = "Hello"  
            objRange.SetElementAttributeValue arrElements(0), arrAttrs(0), strAttrVal  
            strAttrVal = objRange.GetElementAttributeValue (arrElements(0), arrAttrs(0))  
            msgbox "new value of " & arrElements(0) & "/" & arrAttrs(0) & " is: " & strAttrVal  
  
        End If  
    End If  
End If
```

13.3.2.5.56 SetFromRange

Method: SetFromRange (*objSrcRange* as [AuthenticRange](#))

Description

Sets the range object to the same beginning and end positions as *objSrcRange*.

Errors

- 2001 One of the two range objects, is invalid.
- 2005 Null object was specified as source object.

13.3.2.5.57 SetVariableValue

Method: SetVariableValue (*strName* as string, *strValue* as string)

Return Value

Sets the value (second parameter) of the variable named in the first parameter.

Errors

- 2201 Invalid object.
- 2202 No context.
- 2204 No such variable in scope
- 2205 Variable cannot be evaluated
- 2206 Variable returns sequence
- 2207 Variable read-only
- 2208 No modification allowed

13.3.2.5.58 Text

Property: Text as String

Description

Set or get the textual content selected by the range object.

The number of characters retrieved are not necessarily identical, as there are text cursor positions between the beginning and end of the selected range. Most document elements support an end cursor position different to the beginning cursor position of the following element. Drop-down lists maintain only one cursor position, but can select strings of any length. In the case of radio buttons and check boxes, the text property value holds the string of the corresponding XML element.

If the range selects more than one element, the text is the concatenation of the single texts. XML entities are expanded so that '&' is expected as '&'.

Setting the text to the empty string, does not delete any XML elements. Use [Cut](#), [Delete](#) or [PerformAction](#) instead.

Errors

- 2001 The authentic range object or its related view object is no longer valid.
- 2005 Invalid address for a return parameter was specified.

13.3.2.6 AuthenticView**Properties**

[Application](#)
[AsXMLString](#)
[DocumentBegin](#)
[DocumentEnd](#)
[Event](#)
[MarkupVisibility](#)
[Parent](#)
[Selection](#)
[XMLDataRoot](#)
[WholeDocument](#)

Methods

[Goto](#)
[IsRedoEnabled](#)
[IsUndoEnabled](#)
[Print](#)
[Redo](#)
[Undo](#)
[UpdateXMLInstanceEntities](#)

Events

[OnBeforeCopy](#)
[OnBeforeCut](#)
[OnBeforeDelete](#)
[OnBeforeDrop](#)
[OnBeforePaste](#)
[OnDragOver](#)
[OnKeyBoardEvent](#)
[OnMouseEvent](#)
[OnSelectionChanged](#)

Description

AuthenticView and its child objects [AuthenticRange](#) and AuthenticDataTransfer provide you with an interface for **Authentic View**, which allow easy and consistent modification of document contents. These interfaces replace the following interfaces which are marked now as **obsolete**:

OldAuthenticView (old name was DocEditView)

AuthenticSelection (old name was DocEditSelection, superseded by [AuthenticRange](#))

AuthenticEvent (old name was DocEditEvent)

AuthenticView gives you easy access to specific features such as printing, the multi-level undo buffer, and the current cursor selection, or position.

AuthenticView uses objects of type [AuthenticRange](#) to make navigation inside the document straight-forward, and to allow for the flexible selection of logical text elements. Use the properties [DocumentBegin](#), [DocumentEnd](#), or [WholeDocument](#) for simple selections, while using the [Goto](#) method for more complex selections. To navigate relative to a given document range, see the methods and properties of the [AuthenticRange](#) object.

13.3.2.6.1 Events*13.3.2.6.1.1 OnBeforeCopy*

Event: OnBeforeCopy() as Boolean

Scripting environment - VBScript:

```

Function On_AuthenticBeforeCopy()
    ' On_AuthenticBeforeCopy = False ' to disable operation
End Function

```

Scripting environment - JScript:

```
function On_AuthenticBeforeCopy()
{
    // return false; /* to disable operation */
}
```

IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (21, ...)// nEventId = 21
```

Description

This event gets triggered before a copy operation gets performed on the document. Return *True* (or nothing) to allow copy operation. Return *False* to disable copying.

13.3.2.6.1.2 OnBeforeCut

Event: OnBeforeCut() as Boolean

Scripting environment - VBScript:

```
Function On_AuthenticBeforeCut()
    ' On_AuthenticBeforeCut = False ' to disable operation
End Function
```

Scripting environment - JScript:

```
function On_AuthenticBeforeCut()
{
    // return false; /* to disable operation */
}
```

IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (20, ...)// nEventId = 20
```

Description

This event gets triggered before a cut operation gets performed on the document. Return *True* (or nothing) to allow cut operation. Return *False* to disable operation.

13.3.2.6.1.3 OnBeforeDelete

Event: OnBeforeDelete() as Boolean

Scripting environment - VBScript:

```
Function On_AuthenticBeforeDelete()
    ' On_AuthenticBeforeDelete = False ' to disable operation
End Function
```

Scripting environment - JScript:

```
function On_AuthenticBeforeDelete()
{
    // return false; /* to disable operation */
}
```

IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (22, ...)// nEventId = 22
```

Description

This event gets triggered before a delete operation gets performed on the document. Return *True* (or nothing) to allow delete operation. Return *False* to disable operation.

13.3.2.6.1.4 OnBeforeDrop

Event: OnBeforeDrop (*i_nXPos* as Long, *i_nYPos* as Long, *i_ipRange* as [AuthenticRange](#), *i_ipData* as cancelBoolean)

Scripting environment - VBScript:

```
Function On_AuthenticBeforeDrop(nXPos, nYPos, objRange, objData)
    ' On_AuthenticBeforeDrop = False ' to disable operation
End Function
```

Scripting environment - JScript:

```
function On_AuthenticBeforeDrop(nXPos, nYPos, objRange, objData)
{
    // return false; /* to disable operation */
}
```

IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (11, ...)// nEventId = 11
```

Description

This event gets triggered whenever a previously dragged object gets dropped inside the application window. All event related information gets passed as parameters.

The first two parameters specify the mouse position at the time when the event occurred. The parameter *objRange* passes a range object that selects the XML element below the mouse position. The value of this parameter might be *NULL*. Be sure to check before you access the range object. The parameter *objData* allows to access information about the object being dragged.

Return *False* to cancel the drop operation. Return *True* (or nothing) to continue normal operation.

13.3.2.6.1.5 OnBeforePaste

Event: OnBeforePaste (*objData* as Variant, *strType* as String) as Boolean

Scripting environment - VBScript:

```
Function On_AuthenticBeforePaste(objData, strType)
    ' On_AuthenticBeforePaste = False ' to disable operation
End Function
```

Scripting environment - JScript:

```
function On_AuthenticBeforePaste(objData, strType)
{
    // return false; /* to disable operation */
}
```

IDE Plugin:

IXMLSpyPlugIn.OnEvent (19, ...)// nEventId = 19

Description

This event gets triggered before a paste operation gets performed on the document. The parameter *strType* is one of "TEXT", "UNICODETEXT" or "IUNKNOWN". In the first two cases *objData* contains a string representation of the object that will be pasted. In the later case, *objData* contains a pointer to an IUnknown COM interface.

Return *True* (or nothing) to allow paste operation. Return *False* to disable operation.

13.3.2.6.1.6 OnBeforeSave

Event: OnBeforeSave (SaveAs flag) as Boolean

Description: *OnBeforeSave* gives the opportunity to e.g. warn the user about overwriting the existing XML document, or to make the document read-only when specific circumstances are not met. The event will be fired before the file dialog is shown.

13.3.2.6.1.7 OnDragOver

Event: OnDragOver (*nXPos* as Long, *nYPos* as Long, *eMouseEvent* as [SPYMouseEvent](#), *objRange* as [AuthenticRange](#), *objData* as AuthenticDataTransfer) as Boolean

Scripting environment - VBScript:

```
Function On_AuthenticDragOver(nXPos, nYPos, eMouseEvent, objRange, objData)
    ' On_AuthenticDragOver = False ' to disable operation
End Function
```

Scripting environment - JScript:

```
function On_AuthenticDragOver(nXPos, nYPos, eMouseEvent, objRange, objData)
{
    // return false; /* to disable operation */
}
```

IDE Plugin:

IXMLSpyPlugIn.OnEvent (10, ...)// nEventId = 10

Description

This event gets triggered whenever an object from within or outside of Authentic View gets dragged with the mouse over the application window. All event related information gets passed as parameters.

The first three parameters specify the mouse position, the mouse button status and the status of the virtual keys at the time when the event occurred. The parameter *objRange* passes a range object that selects the XML element below the mouse position. The value of this parameter might be *NULL*. Be sure to check before you access the range object. The parameter *objData* allows to access information about the object being dragged.

Return *False* to cancel the drag operation. Return *True* (or nothing) to continue normal operation.

13.3.2.6.1.8 OnKeyboardEvent

Event: OnKeyboardEvent (eKeyEvent as [SPYKeyEvent](#), nKeyCode as Long, nVirtualKeyStatus as Long) as Boolean

Scripting environment - VBScript:

```
Function On_AuthenticKeyboardEvent(eKeyEvent, nKeyCode, nVirtualKeyStatus)
    ' On_AuthenticKeyboardEvent = True ' to cancel bubbling of event
End Function
```

Scripting environment - JScript:

```
function On_AuthenticKeyboardEvent(eKeyEvent, nKeyCode, nVirtualKeyStatus)
{
    // return true; /* to cancel bubbling of event */
}
```

IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (30, ...)// nEventId = 30
```

Description

This event gets triggered for *WM_KEYDOWN*, *WM_KEYUP* and *WM_CHAR* Windows messages.

The actual message type is available in the *eKeyEvent* parameter. The status of virtual keys is combined in the parameter *nVirtualKeyStatus*. Use the bit-masks defined in the enumeration datatype [SPYVirtualKeyMask](#), to test for the different keys or their combinations.

13.3.2.6.1.9 OnLoad

Event: OnLoad ()

Description: OnLoad can be used e.g. to restrict some AuthenticView functionality, as shown in the example below:

```
function On_AuthenticLoad( )
{
    // We are disabling all entry helpers in order to prevent user from manipulating XML tree
    AuthenticView.DisableElementEntryHelper();
    AuthenticView.DisableAttributeEntryHelper();

    // We are also disabling the markup buttons for the same purpose
    AuthenticView.SetToolBarButtonState( 'AuthenticMarkupSmall', authenticToolBarButtonDisabled );
    AuthenticView.SetToolBarButtonState( 'AuthenticMarkupLarge', authenticToolBarButtonDisabled );
    AuthenticView.SetToolBarButtonState( 'AuthenticMarkupMixed', authenticToolBarButtonDisabled );
}
```

In the example the status of the Markup Small, Markup Large, Markup Mixed toolbar buttons are manipulated with the help of button identifiers. See [complete list](#).

13.3.2.6.1.10 *OnMouseEvent*

Event: OnMouseEvent (*nXPos* as Long, *nYPos* as Long, *eMouseEvent* as [SPYMouseEvent](#), *objRange* as [AuthenticRange](#)) as Boolean

Scripting environment - VBScript:

```
Function On_AuthenticMouseEvent(nXPos, nYPos, eMouseEvent, objRange)
    ' On_AuthenticMouseEvent = True ' to cancel bubbling of event
End Function
```

Scripting environment - JScript:

```
function On_AuthenticMouseEvent(nXPos, nYPos, eMouseEvent, objRange)
{
    // return true; /* to cancel bubbling of event */
}
```

IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (31, ...)// nEventId = 31
```

Description

This event gets triggered for every mouse movement and mouse button Windows message.

The actual message type and the mouse buttons status, is available in the *eMouseEvent* parameter. Use the bit-masks defined in the enumeration datatype [SPYMouseEvent](#) to test for the different messages, button status, and their combinations.

The parameter *objRange* identifies the part of the document found at the current mouse cursor position. The range object always selects a complete tag of the document. (This might change in future versions, when a more precise positioning mechanism becomes available). If no selectable part of the document is found at the current position, the range object is *null*.

13.3.2.6.1.11 *OnSelectionChanged*

Event: OnSelectionChanged (*objNewSelection* as [AuthenticRange](#))

Scripting environment - VBScript:

```
Function On_AuthenticSelectionChanged (objNewSelection)
End Function
```

Scripting environment - JScript:

```
function On_AuthenticSelectionChanged (objNewSelection)
{
}
```

IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (23, ...)// nEventId = 23
```

Description

This event gets triggered whenever the selection in the user interface changes.

13.3.2.6.1.12 *OnToolbarButtonClicked*

Event: OnToolbarButtonClicked (Button identifier)

Description: OnToolbarButtonClicked is fired when a toolbar button was clicked by user. The parameter button identifier helps to determine which button was clicked. The list of predefined button identifiers is below:

- AuthenticPrint
- AuthenticPrintPreview
- AuthenticUndo
- AuthenticRedo
- AuthenticCut
- AuthenticCopy
- AuthenticPaste
- AuthenticClear
- AuthenticMarkupHide
- AuthenticMarkupLarge
- AuthenticMarkupMixed
- AuthenticMarkupSmall
- AuthenticValidate
- AuthenticChangeWorkingDBXMLCell
- AuthenticSave
- AuthenticSaveAs
- AuthenticReload
- AuthenticTableInsertRow
- AuthenticTableAppendRow
- AuthenticTableDeleteRow
- AuthenticTableInsertCol
- AuthenticTableAppendCol
- AuthenticTableDeleteCol
- AuthenticTableJoinCellRight
- AuthenticTableJoinCellLeft
- AuthenticTableJoinCellAbove
- AuthenticTableJoinCellBelow
- AuthenticTableSplitCellHorizontally
- AuthenticTableSplitCellVertically
- AuthenticTableAlignCellContentTop
- AuthenticTableCenterCellVertically
- AuthenticTableAlignCellContentBottom
- AuthenticTableAlignCellContentLeft
- AuthenticTableCenterCellContent
- AuthenticTableAlignCellContentRight
- AuthenticTableJustifyCellContent
- AuthenticTableInsertTable
- AuthenticTableDeleteTable
- AuthenticTableProperties
- AuthenticAppendRow
- AuthenticInsertRow
- AuthenticDuplicateRow

- AuthenticMoveRowUp
- AuthenticMoveRowDown
- AuthenticDeleteRow
- AuthenticDefineEntities
- AuthenticXMLSignature

For custom buttons the user might add his own identifiers. Please, note that the user must take care, as the identifiers are not checked for uniqueness. The same identifiers can be used to identify buttons in the `Set/GetToolBarState()` COM API calls. By adding code for different buttons, the user is in the position to completely redefine the AuthenticView toolbar behavior, adding own methods for table manipulation, etc.

13.3.2.6.1.13 *OnToolBarButtonExecuted*

Event: OnToolBarButtonExecuted (Button identifier)

Description: OnToolBarButtonClicked is fired when a toolbar button was clicked by user. The parameter button identifier helps to determine which button was clicked. See the list of [predefined button identifiers](#).

OnToolBarButtonExecuted is fired after the toolbar action was executed. It is useful e.g. to add update code, as shown in the example below:

```
//event fired when a toolbar button action was executed
function On_AuthenticToolBarButtonExecuted( varBtnIdentifier )
{
    // After whatever command user has executed - make sure to update toolbar button states
    UpdateOwnToolBarButtonStates();
}
```

In this case `UpdateOwnToolBarButtonStates` is a user function defined in the Global Declarations.

13.3.2.6.1.14 *OnUserAddedXMLNode*

Event: OnUserAddedXMLNode (XML node)

Description: OnUserAddedXMLNode will be fired when the user adds an XML node as a primary action. This happens in the situations, where the user clicks on

- auto-add hyperlinks (see example OnUserAddedXMLNode.sps)
- the Insert..., Insert After..., Insert Before... context menu items
- Append row, Insert row toolbar buttons
- Insert After..., Insert Before... actions in element entry helper (outside StyleVision)

The event doesn't get fired on Duplicate row, or when the node was added externally (e.g. via COM API), or on Apply (e.g. Text State Icons), or when in XML table operations or in DB operations.

The event parameter is the XML node object, which was added giving the user an opportunity to manipulate the XML node added. An elaborate example for an event handler can be found in the `OnUserAddedXMLNode.sps` file.

13.3.2.6.2 Application

Property: Application as [Application](#) (read-only)

Description

Accesses the Authentic Desktop application object.

Errors

- 2000 The authentic view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.6.3 AsXMLString

Property: AsXMLString as String

Description

Returns or sets the document content as an XML string. Setting the content to a new value does not change the schema file or sps file in use. If the new XMLString does not match the actual schema file error 2011 gets returned.

Errors

- 2000 The authentic view object is no longer valid.
- 2011 AsXMLString was set to a value which is no valid XML for the current schema file.

13.3.2.6.4 ContextMenu

Property: ContextMenu() as ContextMenu

Description

The property `ContextMenu` gives access to customize the context menu. The best place to do it is in the event handler `OnContextMenuActivated`.

Errors

- 2000 Invalid object.
- 2005 Invalid parameter.

13.3.2.6.5 CreateXMLNode

Method: CreateXMLNode (*nKind* as [SPYXMLDataKind](#)) as [XMLData](#)

Return Value

The method returns the new [XMLData](#) object.

Description

To create a new XMLData object use the `CreateXMLNode()` method.

Errors

- 2000 Invalid object.
- 2012 Cannot create XML node.

13.3.2.6.6 DisableAttributeEntryHelper

Method: DisableAttributeEntryHelper ()

Description

DisableAttributeEntryHelper () disables the attribute entry helper in XMLSpy, Authentic Desktop and Authentic Browser plug-in.

Errors

- 2000 Invalid object.

13.3.2.6.7 DisableElementEntryHelper

Method: DisableElementEntryHelper ()

Description

DisableElementEntryHelper () disables the element entry helper in XMLSpy, Authentic Desktop and Authentic Browser plug-in.

Errors

- 2000 Invalid object.

13.3.2.6.8 DisableEntityEntryHelper

Method: DisableEntityEntryHelper ()

Description

DisableEntityEntryHelper () disables the entity entry helper in XMLSpy, Authentic Desktop and Authentic Browser plug-in.

Errors

- 2000 Invalid object.

13.3.2.6.9 DocumentBegin

Property: DocumentBegin as [AuthenticRange](#) (read-only)

Description

Retrieve a range object that points to the beginning of the document.

Errors

- 2000 The authentic view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.6.10 DocumentEnd

Property: DocumentEnd as [AuthenticRange](#) (read-only)

Description

Retrieve a range object that points to the end of the document.

Errors

- 2000 The authentic view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.6.11 DoNotPerformStandardAction

Method: DoNotPerformStandardAction ()

Description

`DoNotPerformStandardAction()` serves as cancel bubble for macros, and stops further execution after macro has finished.

Errors

- 2000 Invalid object.

13.3.2.6.12 EvaluateXPath

Method: EvaluateXPath (XMLData as [XMLData](#), strExpression as string) strValue as string

Return Value

The method returns a string

Description

`EvaluateXPath()` executes an XPath expression with the given XML context node. The result is returned as a string, in the case of a sequence it is a space-separated string.

Errors

- 2000 Invalid object.
- 2005 Invalid parameter.
- 2008 Internal error.
- 2013 XPath error.

13.3.2.6.13 Event

Property: Event as AuthenticEvent (read-only)

Description

This property gives access to parameters of the last event in the same way as `OldAuthenticView.event` does. Since all events for the scripting environment and external clients are now available with parameters this Event property should only be used from within IDE-Plugins.

Errors

- 2000 The authentic view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.6.14 EventContext

Property: EventContext() as EventContext

Description

`EventContext` property gives access to the running macros context. See the [EventContext](#) interface description for more details.

Errors

- 2000 Invalid object.

13.3.2.6.15 GetToolBarButtonState

Method: GetToolBarButtonState (`ButtonIdentifier` as string) as `AuthenticToolBarButtonState`

Return Value

The method returns `AuthenticToolBarButtonState`

Description

`Get/SetToolBarButtonState` queries the status of a toolbar button, and lets the user disable or enable the button, identified via its button identifier ([see list above](#)). One usage is to disable toolbar buttons permanently. Another usage is to put `SetToolBarButtonState` in the `OnSelectionChanged` event handler, as toolbar buttons are updated regularly when the selection changes in the document.

Toolbar button states are given by the [listed enumerations](#).

The default state means that the enable/disable of the button is governed by `AuthenticView`. When the user sets the button state to enable or disable, the button remains in that state as long as the user does not change it.

Errors

- 2000 Invalid object.
- 2005 Invalid parameter.

2008 Internal error.
2014 Invalid button identifier.

13.3.2.6.16 Goto

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

Description

Retrieve a range object that points to the beginning of the *nCount* element of type *eKind*. The start position is defined by the parameter *eFrom*. Use positive values for *nCount* to navigate to the document end. Use negative values to navigate towards the beginning of the document.

Errors

2000 The authentic view object is no longer valid.
2003 Target lies after end of document.
2004 Target lies before beginning of document.
2005 Invalid element kind specified.
The document position to start from is not one of *spyAuthenticDocumentBegin* or *spyAuthenticDocumentEnd*.
Invalid address for the return parameter was specified.

Examples

```
'-----
' Scripting environment - VBScript
'-----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

On Error Resume Next
Dim objRange
' goto beginning of first table in document
Set objRange = objAuthenticView.Goto (spyAuthenticTable, 1, spyAuthenticDocumentBegin)
If (Err.number = 0) Then
    objRange.Select()
Else
    MsgBox "No table found in document"
End If
```

13.3.2.6.17 IsRedoEnabled

Property: IsRedoEnabled as Boolean (read-only)

Description

True if redo steps are available and [Redo](#) is possible.

Errors

2000 The authentic view object is no longer valid.
2005 Invalid address for the return parameter was specified.

13.3.2.6.18 IsUndoEnabled

Property: IsUndoEnabled as Boolean (read-only)

Description

True if undo steps are available and [Undo](#) is possible.

Errors

- 2000 The authentic view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.6.19 MarkupVisibility

Property: MarkupVisibility as [SPYAuthenticMarkupVisibility](#)

Description

Set or get current visibility of markup.

Errors

- 2000 The authentic view object is no longer valid.
- 2005 Invalid enumeration value was specified.
Invalid address for the return parameter was specified.

13.3.2.6.20 Parent

Property: Parent as [Document](#) (read-only)

Description

Access the document shown in this view.

Errors

- 2000 The authentic view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.6.21 Print

Method: Print (*bWithPreview* as Boolean, *bPromptUser* as Boolean)

Description

Print the document shown in this view. If *bWithPreview* is set to *True*, the print preview dialog pops up. If *bPromptUser* is set to *True*, the print dialog pops up. If both parameters are set to *False*, the document gets printed without further user interaction.

Errors

2000 The authentic view object is no longer valid.

13.3.2.6.22 Redo

Method: Redo() as Boolean

Description

Redo the modification undone by the last undo command.

Errors

2000 The authentic view object is no longer valid.

2005 Invalid address for the return parameter was specified.

13.3.2.6.23 Selection

Property: Selection as [AuthenticRange](#)

Description

Set or get current text selection in user interface.

Errors

2000 The authentic view object is no longer valid.

2002 No cursor selection is active.

2005 Invalid address for the return parameter was specified.

Examples

```
'-----  
' Scripting environment - VBScript  
'-----  
Dim objAuthenticView  
' we assume that the active document is open in authentic view mode  
Set objAuthenticView = Application.ActiveDocument.AuthenticView  
  
' if we are the end of the document, re-start at the beginning  
If (objAuthenticView.Selection.IsEqual(objAuthenticView.DocumentEnd)) Then  
    objAuthenticView.Selection = objAuthenticView.DocumentBegin  
Else  
    ' objAuthenticView.Selection = objAuthenticView.Selection.GotoNextCursorPosition()  
    ' or shorter:  
    objAuthenticView.Selection.GotoNextCursorPosition().Select  
End If
```

13.3.2.6.24 SetToolBarButtonState

Method: SetToolBarButtonState (ButtonIdentifier as string, AuthenticToolBarButtonState state)

Description

Get/SetToolBarButtonState queries the status of a toolbar button, and lets the user disable or enable the button, identified via its button identifier ([see list above](#)). One usage is to disable toolbar buttons permanently. Another usage is to put SetToolBarButtonState in the OnSelectionChanged event handler, as toolbar buttons are updated regularly when the selection changes in the document.

ToolBar button states are given by the [listed enumerations](#).

The default state means that the enable/disable of the button is governed by AuthenticView. When the user sets the button state to enable or disable, the button remains in that state as long as the user does not change it.

Errors

- 2000 Invalid object.
- 2008 Internal error.
- 2014 Invalid button identifier.

13.3.2.6.25 Undo

Method: Undo() as Boolean

Description

Undo the last modification of the document from within this view.

Errors

- 2000 The authentic view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.6.26 UpdateXMLInstanceEntities

Method: UpdateXMLInstanceEntities()

Description

Updates the internal representation of the declared entities, and refills the entry helper. In addition, the validator is reloaded, allowing the XML file to validate correctly. Please note that this may also cause schema files to be reloaded.

Errors

The method never returns an error.

Example

```
// -----  
// Scripting environment - JavaScript  
// -----
```

```
if(Application.ActiveDocument && (Application.ActiveDocument.CurrentViewMode == 4))
{
    var objDocType;
    objDocType = Application.ActiveDocument.DocEditView.XMLRoot.GetFirstChild(10);

    if(objDocType)
    {
        var objEntity = Application.ActiveDocument.CreateChild(14);
        objEntity.Name = "child";
        objEntity.TextValue = "SYSTEM \\child.xml\\";
        objDocType.AppendChild(objEntity);

        Application.ActiveDocument.AuthenticView.UpdateXMLInstanceEntities();
    }
}
```

13.3.2.6.27 WholeDocument

Property: WholeDocument as [AuthenticRange](#) (read-only)

Description

Retrieve a range object that selects the whole document.

Errors

- 2000 The authentic view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.6.28 XMLDataRoot

Property: XMLDataRoot as [XMLData](#) (read-only)

Description

Returns or sets the top-level XMLData element of the current document. This element typically describes the document structure and would be of kind spyXMLDataXMLDocStruct, spyXMLDataXMLEntityDocStruct or spyXMLDataDTDDocStruct..

Errors

- 2000 The authentic view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

13.3.2.7 CodeGeneratorDlg

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Properties and Methods

Standard automation properties

[Application](#)

[Parent](#)

Programming language selection properties

[ProgrammingLanguage](#)

[TemplateFileName](#)

Settings for C++ code

[CPPSettings_DOMType](#)

[CPPSettings_LibraryType](#)

[CPPSettings_UseMFC](#)

[CPPSettings_GenerateVC6ProjectFile](#)

[CPPSettings_GenerateVSProjectFile](#)

Settings for C# code

[CSharpSettings_ProjectType](#)

Dialog handling for above code generation properties

[PropertySheetDialogAction](#)

Output path selection properties

[OutputPath](#)

[OutputPathDialogAction](#)

Presentation of result

[OutputResultDialogAction](#)

Description

Use this object to configure the generation of program code for schema files. The method [GenerateProgramCode](#) expects a CodeGeneratorDlg as parameter to configure code generation as well as the associated user interactions.

13.3.2.7.1 Application

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Property: Application as [Application](#) (read-only)

Description

Access the Authentic Desktop application object.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

13.3.2.7.2 CPPSettings_DOMType

Property: CPPSettings_DOMType as [SPYDOMType](#)

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Defines one of the settings that configure generation of C++ code.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

13.3.2.7.3 CPPSettings_GenerateVC6ProjectFile

Property: CPPSettings_GenerateVC6ProjectFile as Boolean

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Defines one of the settings that configure generation of C++ code.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

13.3.2.7.4 CPPSettings_GenerateGCCMakefile

Property: CPPSettings_GenerateGCCMakefile as Boolean

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Creates makefiles to compile the generated code under Linux with GCC.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

13.3.2.7.5 CPPSettings_GenerateVSProjectFile

Property: CSharpSettings_GenerateVSProjectFile as [SPYProjectType](#)

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Defines one of the settings that configure generation of C++ code. Only `spyVisualStudio2005Project (=4)` and `spyVisualStudio2008Project (=5)` and `spyVisualStudio2010Project (=6)` are valid project types.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

13.3.2.7.6 CPPSettings_LibraryType

Property: CPPSettings_LibraryType as [SPYLibType](#)

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Defines one of the settings that configure generation of C++ code.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

13.3.2.7.7 CPPSettings_UseMFC

Property: CPPSettings_UseMFC as Boolean

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Defines one of the settings that configure generation of C++ code.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

13.3.2.7.8 CSharpSettings_ProjectType

Property: CSharpSettings_ProjectType as [SPYProjectType](#)

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Defines the only setting to configure generation of C# code.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

13.3.2.7.9 OutputPath

Property: OutputPath as String

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Selects the base directory for all generated code.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

13.3.2.7.10 OutputPathDialogAction

Property: OutputPathDialogAction as [SPYDialogAction](#)

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Defines how the sub-dialog for selecting the code generation output path gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog with the current value of the [OutputPath](#) property as default. Use *spyDialogOK(0)* to hide the dialog from the user.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

13.3.2.7.11 OutputResultDialogAction

Property: OutputResultDialogAction as [SPYDialogAction](#)

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Defines how the sub-dialog that asks to show the result of the code generation process gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog. Use *spyDialogOK(0)* to hide the dialog from the user.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

13.3.2.7.12 Parent

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Property: Parent as [Dialogs](#) (read-only)

Description

Access the parent of the object.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

13.3.2.7.13 [ProgrammingLanguage](#)

Property: ProgrammingLanguage as [ProgrammingLanguage](#)

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Selects the output language for the code to be generated.

CAUTION: Setting this property to one of C++, C# or Java, changes the property [TemplateFileName](#) to the appropriate template file delivered with Authentic Desktop as well. If you want to generate C++, C# or Java code based on your own templates, set first the programming language and then select your template file.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

13.3.2.7.14 [PropertySheetDialogAction](#)

Property: PropertySheetDialogAction as [SPYDialogAction](#)

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Defines how the sub-dialog that configures the code generation process gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog with the current values as defaults. Use *spyDialogOK(0)* to hide the dialog from the user.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

13.3.2.7.15 [TemplateFileName](#)

Property: TemplateFileName as String

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Selects the code generation template file. Authentic Desktop comes with template files for C++, C# or Java in the SPL folder of your installation directory.

Setting this property to one of the code generation template files of your Authentic Desktop installation automatically sets the [ProgrammingLanguage](#) property to its appropriate value.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

13.3.2.8 DatabaseConnection

Properties for import and export

[File](#) or
[ADODConnection](#) or
[ODBCConnection](#)

Properties for import only

[DatabaseKind](#)
[SQLSelect](#)
[AsAttributes](#)
[ExcludeKeys](#)
[IncludeEmptyElements](#)
[NumberDateTimeFormat](#)
[NullReplacement](#)
[CommentIncluded](#)

Properties for export only

[CreateMissingTables](#)
[CreateNew](#)
[TextFieldLen](#)
[DatabaseSchema](#)

Properties for XML Schema from DB Structure generation

[PrimaryKeys](#)
[ForeignKeys](#)
[UniqueKeys](#)
[SchemaExtensionType](#)
[SchemaFormat](#)
[ImportColumnsType](#)

Description

DatabaseConnection specifies the parameters for the database connection.

Please note that the properties of the DatabaseConnection interface are referring to the settings of the import and export dialogs of Authentic Desktop.

13.3.2.8.1 ADOConnection

Property: ADOConnection as String

Description

The property ADOConnection contains a connection string. Either use this property or [ODBCConnection](#) or [File](#) to refer to a database.

Errors

No error codes are returned.

Example

```
Dim objSpyConn As DatabaseConnection
Set objSpyConn = objSpy.GetDatabaseSettings

Dim objADO As DataLinks
Set objADO = CreateObject("DataLinks")

If Not (objADO Is Nothing) Then
    Dim objConn As Connection
    Set objConn = objADO.PromptNew
    objSpyConn.ADOConnection = objConn.ConnectionString
End If
```

13.3.2.8.2 AsAttributes

Property: AsAttributes as Boolean

Description

Set AsAttributes to true if you want to initialize all import fields to be imported as attributes. Default is false and will initialize all fields to be imported as elements. This property is used only in calls to [Application.GetDatabaseImportElementList](#).

Errors

No error codes are returned.

13.3.2.8.3 CommentIncluded

Property: CommentIncluded as Boolean

Description

This property tells whether additional comments are added to the generated XML. Default is true. This property is used only when importing from databases.

Errors

No error codes are returned.

13.3.2.8.4 CreateMissingTables

Property: CreateMissingTables as Boolean

Description

If CreateMissingTables is true, tables which are not already defined in the export database will be created during export. Default is true. This property is used only when exporting to databases.

Errors

No error codes are returned.

13.3.2.8.5 CreateNew

Property: CreateNew as Boolean

Description

Set CreateNew true if you want to create a new database on export. Any existing database will be overwritten. See also [DatabaseConnection.File](#). Default is false. This property is used only when exporting to databases.

Errors

No error codes are returned.

13.3.2.8.6 DatabaseKind

Property: DatabaseKind as [SPYDatabaseKind](#)

Description

Select the kind of database that gets access. The default value is spyDB_Unspecified(7) and is sufficient in most cases. This property is used only when importing from databases.

Errors

No error codes are returned.

13.3.2.8.7 DatabaseSchema

Property: DatabaseSchema as String

Description

This property specifies the Schema used for export in Schema aware databases. Default is "". This property is used only when exporting to databases.

Errors

No error codes are returned.

13.3.2.8.8 ExcludeKeys

Property: ExcludeKeys as Boolean

Description

Set ExcludeKeys to true if you want to exclude all key columns from the import data. Default is false. This property is used only when importing from databases.

Errors

No error codes are returned.

13.3.2.8.9 File

Property: File as String

Description

The property File sets the path for the database during export or import. This property can only be used in conjunction with a Microsoft Access database. Either use this property or [ODBCConnection](#) or [ADODConnection](#) to refer to the database.

Errors

No error codes are returned.

13.3.2.8.10 ForeignKeys

Property: ForeignKeys as Boolean

Description

Specifies whether the Foreign Keys constraint is created or not. Default is true. This property is used only when creating a XML Schema from a DB structure.

Errors

No error codes are returned.

13.3.2.8.11 ImportColumnsType

Property: ImportColumnsType as [SPYImportColumnsType](#)

Description

Defines if column information from the DB is saved as element or attribute in the XML Schema. Default is as element. This property is used only when creating a XML Schema from a DB structure.

Errors

No error codes are returned.

13.3.2.8.12 IncludeEmptyElements

Property: IncludeEmptyElements as Boolean

Description

Set IncludeEmptyElements to false if you want to exclude all empty elements. Default is true. This property is used only when importing from databases.

Errors

No error codes are returned.

13.3.2.8.13 NullReplacement

Property: NullReplacement as String

Description

This property contains the text value that is used during import for empty elements (null values). Default is "". This property is used only when importing from databases.

Errors

No error codes are returned.

13.3.2.8.14 NumberDateTimeFormat

Property: NumberDateTimeFormat as [SPYNumberDateTimeFormat](#)

Description

The property NumberDateTimeFormat sets the format of numbers and date- and time-values. Default is [spySystemLocale](#). This property is used only when importing from databases.

Errors

No error codes are returned.

13.3.2.8.15 ODBCConnection

Property: ODBCConnection as String

Description

The property ODBCConnection contains a ODBC connection string. Either use this property or [ADOConnection](#) or [File](#) to refer to a database.

Errors

No error codes are returned.

13.3.2.8.16 PrimaryKeys

Property: PrimaryKeys as Boolean

Description

Specifies whether the Primary Keys constraint is created or not. Default is true. This property is used only when creating a XML Schema from a DB structure.

Errors

No error codes are returned.

13.3.2.8.17 SchemaExtensionType

Property: SchemaExtensionType as [SPYSchemaExtensionType](#)

Description

Defines the Schema extension type used during the Schema generation. This property is used only when creating a XML Schema from a DB structure.

Errors

No error codes are returned.

13.3.2.8.18 SchemaFormat

Property: SchemaFormat as [SPYSchemaFormat](#)

Description

Defines the Schema format used during the Schema generation. This property is used only when creating a XML Schema from a DB structure.

Errors

No error codes are returned.

13.3.2.8.19 SQLSelect

Property: SQLSelect as String

Description

The SQL query for the import is stored in the property SQLSelect. This property is used only when importing from databases.

Errors

No error codes are returned.

13.3.2.8.20 TextFieldLen

Property: TextFieldLen as long

Description

The property TextFieldLen sets the length for created text fields during the export. Default is 255. This property is used only when exporting to databases.

Errors

No error codes are returned.

13.3.2.8.21 UniqueKeys

Property: UniqueKeys as Boolean

Description

Specifies whether the Unique Keys constraint is created or not. Default is true. This property is used only when creating a XML Schema from a DB structure.

Errors

No error codes are returned.

13.3.2.9 Dialogs

Properties and Methods

Standard automation properties

[Application](#)

[Parent](#)

Various dialog objects

[CodeGeneratorDlg](#)

[FileSelectionDlg](#)

[SchemaDocumentationDlg](#)

[GenerateSampleXMLDlg](#)

[DTDSchemaGeneratorDlg](#)

[FindInFilesDlg](#)

Description

The Dialogs object provides access to different built-in dialogs of Authentic Desktop. These dialog objects allow to initialize the fields of user dialogs before they get presented to the user or allow to simulate complete user input by your program.

13.3.2.9.1 Application

Property: Application as [Application](#) (read-only)

Description

Access the Authentic Desktop application object.

Errors

- 2300 The object is no longer valid.
- 2301 Invalid address for the return parameter was specified.

13.3.2.9.2 CodeGeneratorDlg

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Property: CodeGeneratorDlg as [CodeGeneratorDlg](#) (read-only)

Description

Get a new instance of a code generation dialog object. You will need this object to pass the necessary parameters to the code generation methods. Initial values are taken from last usage of the code generation dialog.

Errors

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

13.3.2.9.3 FileSelectionDlg

Property: FileSelectionDlg as [FileSelectionDlg](#) (read-only)

Description

Get a new instance of a file selection dialog object.

File selection dialog objects are passed to you with the some events that signal opening or saving of documents and projects.

Errors

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

13.3.2.9.4 Parent

Property: Parent as [Application](#) (read-only)

Description

Access the Authentic Desktop application object.

Errors

- 2300 The object is no longer valid.
- 2301 Invalid address for the return parameter was specified.

13.3.2.9.5 SchemaDocumentationDlg

Property: SchemaDocumentationDlg as [SchemaDocumentationDlg](#) (read-only)

Description

Get a new instance of a dialog object that parameterizes generation of schema documentation. See [Document.GenerateSchemaDocumentation](#) for its usage.

Errors

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

13.3.2.9.6 GenerateSampleXMLDlg

Property: GenerateSampleXMLDlg as [GenerateSampleXMLDlg](#) (read-only)

Description

Get a new instance of a dialog object that parameterizes generation of a sample XML based on a W3C schema or DTD. See [GenerateSampleXML](#) for its usage.

Errors

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

13.3.2.9.7 DTDSchemaGeneratorDlg

Property: DTDSchemaGeneratorDlg as [DTDSchemaGeneratorDlg](#) (read-only)

Description

Get a new instance of a dialog object that parameterizes generation of a schema or DTD. See [Document.GenerateDTDOrSchemaEx](#) for its usage.

Errors

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

13.3.2.9.8 FindInFilesDlg

Property: FindInFilesDlg as [FindInFilesDlg](#) (read-only)

Description

Get a new instance of a dialog object that parameterizes the search (or replacement) of strings in files. See [Application.FindInFiles](#) for its usage.

Errors

- 2300 The Dialogs object or one of its parents is no longer valid.

2301 Invalid address for the return parameter was specified.

13.3.2.9.9 WSDLDocumentationDlg

Property: WSDLDocumentationDlg as [WSDLDocumentationDlg](#) (read-only)

Description

Get a new instance of a dialog object that parameterizes generation of WSDL documentation. See [Document.GenerateWSDLDocumentation](#) for its usage.

Errors

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

13.3.2.9.10 WSDL20DocumentationDlg

Property: WSDL20DocumentationDlg as [WSDL20DocumentationDlg](#) (read-only)

Description

Get a new instance of a dialog object that parameterizes generation of WSDL 2.0 documentation. See [Document.GenerateWSDL20Documentation](#) for its usage.

Errors

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

13.3.2.9.11 XBRLDocumentationDlg

Property: XBRLDocumentationDlg as [XBRLDocumentationDlg](#) (read-only)

Description

Get a new instance of a dialog object that parameterizes generation of XBRL documentation. See [Document.GenerateXBRLDocumentation](#) for its usage.

Errors

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

13.3.2.10 Document

Properties and Methods

Standard automation properties

[Application](#)
[Parent](#)

Various document properties and methods

[SetActiveDocument](#)[Encoding](#)[SetEncoding \(obsolete\)](#)[Suggestions](#)

XML validation

[IsValid](#)[SetExternalsIsValid](#)

Document conversion and transformation

[AssignDTD](#)[AssignSchema](#)[AssignXSL](#)[AssignXSLFO](#)[ConvertDTDOOrSchema](#)[ConvertDTDOOrSchemaEx](#)[GenerateDTDOOrSchema](#)[GenerateDTDOOrSchemaEx](#)[FlattenDTDOOrSchema](#)[CreateSchemaDiagram](#)[ExecuteXQuery](#)[TransformXSL](#)[TransformXSLEx](#)[TransformXSLFO](#)[GenerateProgramCode](#) (Enterprise Edition only)[GenerateSchemaDocumentation](#)[GenerateSampleXML](#)[ConvertToWSDL20](#)

Document export

[GetExportElementList](#)[ExportToText](#)[ExportToDatabase](#)[CreateDBStructureFromXMLSchema](#)[GetDBStructureList](#)

File saving and naming

[FullName](#)[Name](#)[Path](#)[GetPathName \(obsolete\)](#)[SetPathName \(obsolete\)](#)[Title](#)[IsModified](#)[Saved](#)[SaveAs](#)[Save](#)[SaveInString](#)[SaveToURL](#)[Close](#)

View access

[CurrentViewMode](#)

[SwitchViewMode](#)
[AuthenticView](#)
[GridView](#)
[DocEditView \(obsolete\)](#)

Access to XMLData
[RootElement](#)
[DataRoot](#)
[CreateChild](#)
[UpdateViews](#)
[StartChanges](#)
[EndChanges](#)
[UpdateXMLData](#)

Description

Document objects represent XML documents opened in Authentic Desktop.

Use one of the following properties to access documents that are already open Authentic Desktop:

[Application.ActiveDocument](#)
[Application.Documents](#)

Use one of the following methods to open a new document in Authentic Desktop:

[Documents.OpenFile](#)
[Documents.OpenURL](#)
[Documents.OpenURLDialog](#)
[Documents.NewFile](#)
[Documents.NewFileFromText](#)
[SpyProjectItem.Open](#)
[Application.ImportFromDatabase](#)
[Application.ImportFromSchema](#)
[Application.ImportFromText](#)
[Application.ImportFromWord](#)
[Document.ConvertDTDOOrSchema](#)
[Document.GenerateDTDOOrSchema](#)

13.3.2.10.1 Events

13.3.2.10.1.1 OnBeforeSaveDocument

Event: OnBeforeSaveDocument(*objDocument* as [Document](#), *objDialog* as [FileSelectionDlg](#))

XMLSpy scripting environment - VBScript:

```
Function On_BeforeSaveDocument(objDocument, objDialog)  
End Function
```

```
' old handler - now obsolete  
' return string to save to new file name  
' return empty string to cancel save operation  
' return nothing to save to original name  
Function On_SaveDocument(objDocument, strFilePath)  
End Function
```

XMLSpy scripting environment - JScript:

```
function On_BeforeSaveDocument(objDocument, objDialog)
{
}

// old handler - now obsolete
// return string to save to new file name
// return empty string to cancel save operation
// return nothing to save to original name
function On_SaveDocument(objDocument, strFilePath)
{
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (27, ...)// nEventId = 27
```

Description

This event gets fired on any attempt to save a document. The file selection dialog object is initialized with the name chosen for the document file. You can modify this selection. To continue saving the document leave the [FileSelectionDlg.DialogAction](#) property of *io_objDialog* at its default value [spyDialogOK](#). To abort saving of the document set this property to [spyDialogCancel](#).

13.3.2.10.1.2 OnBeforeCloseDocument

Event: OnBeforeCloseDocument(*objDocument* as [Document](#))as Boolean

XMLSpy scripting environment - VBScript:

```
Function On_BeforeCloseDocument(objDocument)
    ' On_BeforeCloseDocument = False ' to prohibit closing of document
End Function
```

XMLSpy scripting environment - JScript:

```
function On_BeforeCloseDocument(objDocument)
{
    // return false; /* to prohibit closing of document */
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (28, ...)// nEventId = 28
```

Description

This event gets fired on any attempt to close a document. To prevent the document from being closed return false.

13.3.2.10.1.3 OnBeforeValidate

Event: OnBeforeValidate(*objDocument* as [Document](#), *bOnLoading* as Boolean, *bOnCommand* as Boolean) as Boolean

XMLSpy scripting environment - VBScript:

```
Function On_BeforeValidate(objDocument, bOnLoading, bOnCommand)
    On_BeforeValidate = bCancelDefaultValidation 'set by the script if necessary
End Function
```

XMLSpy scripting environment - JScript:

```
function On_BeforeValidate(objDocument, bOnLoading, bOnCommand)
{
    return bCancelDefaultValidation //set by the script if necessary
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (32, ...)// nEventId = 32
```

Description

This event gets fired before the document is validated. It is possible to suppress the default validation by returning false from the event handler. In this case the script should also set the validation result using the [SetExternallsValid](#) method.

bOnLoading is true if the event is raised on the initial validation on loading the document.

bOnCommand is true whenever the user selected the Validate command from the Toolbar or menu.

Available with TypeLibrary version 1.5

13.3.2.10.1.4 OnCloseDocument

Event: OnCloseDocument(objDocument as [Document](#))

XMLSpy scripting environment - VBScript:

```
Function On_Close Document(objDocument)
End Function
```

XMLSpy scripting environment - JScript:

```
function On_Close Document(objDocument)
{
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (8, ...) // nEventId = 8
```

Description

This event gets fired as a result of closing a document. Do not modify the document from within this event.

13.3.2.10.1.5 OnViewActivation

Event: OnViewActivation(objDocument as [Document](#), eViewMode as [SPYViewModes](#), bActivated as Boolean)

XMLSpy scripting environment - VBScript:

```
Function On_ViewActivation(objDocument, eViewMode, bActivated)
End Function
```


XMLSpy scripting environment - JScript:

```
function On_ViewActivation(objDocument, eViewMode, bActivated)
{
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (29, ...)// nEventId = 29
```

Description

This event gets fired whenever a view of a document becomes visible (i.e. becomes the active view) or invisible (i.e. another view becomes the active view or the document gets closed). However, the first view activation event after a document gets opened cannot be received, since there is no document object to get the event from. Use the [Application.OnDocumentOpened](#) event instead.

13.3.2.10.2 Application

Property: Application as [Application](#) (read-only)

Description

Accesses the Authentic Desktop application object.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

13.3.2.10.3 AssignDTD

Method: AssignDTD(*strDTDFile* as String, *bDialog* as Boolean)

Description

The method places a reference to the DTD file "strDTDFile" into the document. Note that no error occurs if the file does not exist, or is not accessible. If bDialog is true Authentic Desktop presents a dialog to set the file.

Errors

- 1400 The object is no longer valid.
- 1409 You are not allowed to assign a DTD to the document.

13.3.2.10.4 AssignSchema

Method: AssignSchema (*strSchemaFile* as String, *bDialog* as Boolean)

Description

The method places a reference to the schema file "strSchemaFile" into the document. Note that no error occurs if the file does not exist or is not accessible. If bDialog is true Authentic Desktop presents a dialog to set the file.

Errors

- 1400 The object is no longer valid.
- 1409 You are not allowed to assign a schema file to the document.

13.3.2.10.5 AssignXSL

Method: AssignXSL (*strXSLFile* as String, *bDialog* as Boolean)

Description

The method places a reference to the XSL file "strXSLFile" into the document. Note that no error occurs if the file does not exist or is not accessible. If *bDialog* is true Authentic Desktop presents a dialog to set the file.

Errors

- 1400 The object is no longer valid.
- 1409 You are not allowed to assign an XSL file to the document.

13.3.2.10.6 AssignXSLFO

Method: AssignXSLFO (*strXSLFOFile* as String, *bDialog* as Boolean)

Description

The method places a reference to the XSLFO file "strXSLFile" into the document. Note that no error occurs if the file does not exist or is not accessible. If *bDialog* is true Authentic Desktop presents a dialog to set the file.

Errors

- 1400 The object is no longer valid.
- 1409 You are not allowed to assign an XSL file to the document.

13.3.2.10.7 AsXMLString

Property: AsXMLString as String

Description

This property can be used to get or set the document content.

Errors

- 1400 The document object is no longer valid.
- 1404 Cannot create XMLData object.
- 1407 View mode cannot be switched.

13.3.2.10.8 AuthenticView

Method: AuthenticView as [AuthenticView](#) (read-only)

Description

Returns an object that gives access to properties and methods specific to Authentic view. The object returned is only valid if the current document is opened in Authentic view mode. The lifetime of an object ends with the next view switch. Any attempt to access objects or any of its children afterwards will result in an error indicating that the object is invalid.

Errors

- 1400 The object is no longer valid.
- 1417 Document needs to be open in authentic view mode.

Examples

```
'-----
' XMLSpy scripting environment - VBScript
' secure access to authentic view object
'-----
Dim objDocument
Set objDocument = Application.ActiveDocument
If (Not objDocument Is Nothing) Then
    ' we have an active document, now check for view mode
    If (objDocument.CurrentViewMode <> spyViewAuthentic) Then
        If (Not objDocument.SwitchViewMode (spyViewAuthentic)) Then
            MsgBox "Active document does not support authentic view mode"
        Else
            ' now it is safe to access the authentic view object
            Dim objAuthenticView
            Set objAuthenticView = objDocument.AuthenticView
            ' now use the authentic view object

            End If
        End If
    Else
        MsgBox "No document is open"
    End If
```

13.3.2.10.9 Close

Method: Close (*bDiscardChanges* as Boolean)

Description

To close the document call this method. If *bDiscardChanges* is true and the document is modified, the document will be closed but not saved.

Errors

- 1400 The object is no longer valid.
- 1401 Document needs to be saved first.

13.3.2.10.10 ConvertDTDOrSchema

Method: ConvertDTDOrSchema (*nFormat* as [SPYDTDSchemaFormat](#), *nFrequentElements* as [SPYFrequentElements](#))

Parameters

nFormat

Sets the schema output format to DTD or W3C.

nFrequentElements

Create complex elements as elements or complex types.

Description

ConvertDTDOrSchema takes an existing schema format and converts it into a different format.

For a finer tuning of DTD/XSD conversion, use [ConvertDTDOrSchemaEx](#).

Errors

1400 The object is no longer valid.

1412 Error during conversion. In the case of DTD to DTD or XSD to XSD conversion, the following errors are returned: *DTD to DTD conversion is not supported. Please use function FlattenDTDOrSchema instead and Schema to schema conversion is not supported. Please use function FlattenDTDOrSchema instead.*

13.3.2.10.11 ConvertDTDOrSchemaEx

Method: ConvertDTDOrSchemaEx (*nFormat* as [SPYDTDSchemaFormat](#), *nFrequentElements* as [SPYFrequentElements](#), *sOutputPath* as String, *nOutputPathDialogAction* as [SPYDialogAction](#))

Parameters

nFormat

Sets the schema output format to DTD, or W3C.

nFrequentElements

Create complex elements as elements or complex types.

sOutputPath

The file path for the newly generated file.

nOutputPathDialogAction

Defines the dialog interaction for this call.

Description

ConvertDTDOrSchemaEx takes an existing schema format and converts it into a different format.

Errors

1400 The object is no longer valid.

1412 Error during conversion. In the case of DTD to DTD or XSD to XSD conversion, the following errors are returned: *DTD to DTD conversion is not supported. Please use function FlattenDTDOrSchema instead and*

Schema to schema conversion is not supported. Please use function FlattenDTDOOrSchema instead.

13.3.2.10.12 ConvertToWSDL20

Method: ConvertToWSDL20 (*sFilePath* as String, *bShowDialogs* as Boolean)

Parameters

sFilePath

This specifies the file name of the converted WSDL. In case the source WSDL includes files which also must be converted, then only the directory part of the given path is used and the file names are generated automatically.

bShowDialogs

Defines whether file/folder selection dialogs are shown.

Description

Converts the WSDL 1.1 document to a WSDL 2.0 file. It will also convert any referenced WSDL files that are referenced from within this document. Note that this functionality is limited to WSDL View only. See [Document.CurrentViewMode](#). and [SPYViewModes](#).

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid parameters have been passed or an empty file name has been specified as output target.
- 1417 The document is not opened in WSDL view, maybe it is not an '.wsdl' file.
- 1421 Feature is not available in this edition.
- 1433 WSDL 1.1 to WSDL 2.0 conversion failed.

13.3.2.10.13 CreateChild

Method: CreateChild (*nKind* as [SPYXMLDataKind](#)) as [XMLData](#)

Return Value

The method returns the new XMLData object.

Description

To create a new XMLData object use the CreateChild() method.

Errors

- 1400 The object is no longer valid.
- 1404 Cannot create XMLData object.
- 1407 Invalid address for the return parameter was specified.

13.3.2.10.14 CreateDBStructureFromXMLSchema

Method: CreateDBStructureFromXMLSchema (*pDatabase* as [DatabaseConnection](#), *pTables* as [ElementList](#), *bDropTableWithExistingName* as Boolean) as String

Description

CreateDBStructureFromXMLSchema exports the given tables to the specified database. The function returns the SQL statements that were necessary to perform the changes.

See also [GetDBStructureList](#).

Errors

- 1429 Database selection missing.
- 1430 Document export failed.

13.3.2.10.15 CreateSchemaDiagram

Method: CreateSchemaDiagram (*nKind* as [SPYSchemaDefKind](#), *strName* as String, *strFile* as String)

Return Value

None.

Description

The method creates a diagram of the schema type *strName* of kind *nKind* and saves the output file into *strFile*. Note that this functionality is limited to Schema View only. See [Document.CurrentViewMode](#), and [SPYViewModes](#).

Errors

- 1400 The object is no longer valid.
- 1414 Failed to save diagram.
- 1415 Invalid schema definition type specified.

13.3.2.10.16 CurrentViewMode

Method: CurrentViewMode as [SPYViewModes](#)

Description

The property holds the current view mode of the document. See also [Document.SwitchViewMode](#).

Errors

- 1400 The object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

13.3.2.10.17 DataRoot

Property: DataRoot as [XMLData](#) (read-only)

Description

This property provides access to the document's first XMLData object of type *spyXMLDataElement*. This is typically the root element for all document content data. See [XMLSpyDocument.RootElement](#) to get the root element of the whole document including XML prolog data. If the [CurrentViewMode](#) is not *spyViewGrid* or *spyViewAuthentic* an [UpdateXMLData](#) may be necessary to get access to the latest [XMLData](#).

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

13.3.2.10.18 DocEditView

Method: DocEditView as DocEditView

Description

Holds a reference to the current Authentic View object.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid address for the return parameter was specified.
- 1417 Document needs to be open in authentic view mode.

13.3.2.10.19 Encoding

Property: Encoding as String

Description

This property provides access to the document's encoding value. However, this property can only be accessed when the document is opened in *spyViewGrid*, *spyViewText* or *spyViewAuthentic*. See [CurrentViewMode](#) on how to detect a document's actual view mode.

This property makes the method [SetEncoding](#) obsolete.

Possible values are, for example:

- 8859-1,
- 8859-2,
- ASCII, ISO-646,
- 850,
- 1252,
- 1255,
- SHIFT-JIS, MS-KANJI,
- BIG5, FIVE,
- UTF-7,

UTF-8,
UTF-16

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.
- 1416 Operation not supported in current view mode.

13.3.2.10.20 EndChanges

Method: EndChanges()

Description

Use the method EndChanges to display all changes since the call to [Document.StartChanges](#).

Errors

- 1400 The object is no longer valid.

13.3.2.10.21 ExecuteXQuery

Method: ExecuteXQuery (*strXMLFileName* as String)

Description

Execute the XQuery statements contained in the document of the document object. Either an XQuery execution or an XQuery Update is performed depending on the file extension of the document. Use the XML file specified in the argument as the XML target document that the XQuery document processes.

- If the document has an XQuery file extension as defined in the Options dialog of Authentic Desktop, then an XQuery execution is performed. By default: `.xq`, `.xql`, and `.xquery` are set as XQuery file extensions in Authentic Desktop.
- If the document has an XQuery Update file extension as defined in the Options dialog of Authentic Desktop, then an XQuery Update action is performed. By default: `.xqu` is set as an XQuery Update file extension in Authentic Desktop.

If your XQuery script does not use an XML source, set the parameter `strXMLFileName` to an empty string.

Errors

- 1400 The document object is no longer valid.
- 1423 XQuery transformation error.
- 1424 Not all files required for operation could be loaded. Most likely, the file specified in `strXMLFileName` does not exist or is not valid.

13.3.2.10.22 ExportToDatabase

Method: ExportToDatabase (*pFromChild* as [XMLData](#), *pExportSettings* as [ExportSettings](#), *pDatabase* as [DatabaseConnection](#))

Description

ExportToDatabase exports the XML document starting with the element *pFromChild*. The parameter *pExportSettings* defines the behaviour of the export (see [Application.GetExportSettings](#)). The parameter *pDatabase* specifies the destination of the export (see [Application.GetDatabaseSettings](#)). [UpdateXMLData\(\)](#) might be indirectly needed as you have to pass the [XMLData](#) as parameter to this function.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.
- 1416 Error during export.
- 1429 Database selection missing.
- 1430 Document export failed.

Example

```
Dim objDoc As Document
Set objDoc = objSpy.ActiveDocument

'set the behaviour of the export with ExportSettings
Dim objExpSettings As ExportSettings
Set objExpSettings = objSpy.GetExportSettings

'set the destination with DatabaseConnection
Dim objDB As DatabaseConnection
Set objDB = objSpy.GetDatabaseSettings

objDB.CreateMissingTables = True
objDB.CreateNew = True
objDB.File = "C:\Export.mdb"

objDoc.ExportToDatabase objDoc.RootElement, objExpSettings, objDB
If Err.Number <> 0 Then
    a = MsgBox("Error: " & (Err.Number - vbObjectError) & Chr(13) &
        "Description: " & Err.Description)
End If
```

13.3.2.10.23 ExportToText

Method: ExportToText (*pFromChild* as [XMLData](#), *pExportSettings* as [ExportSettings](#), *pTextSettings* as [TextImportExportSettings](#))

Description

ExportToText exports tabular information from the document starting at *pFromChild* into one or many text files. Columns of the resulting tables are generated in alphabetical order of the column header names. Use [GetExportElementList](#) to learn about the data that will be exported. The parameter *pExportSettings* defines the

specifics for the export. Set the property [ExportSettings.ElementList](#) to the - possibly modified - list returned by [GetExportElementList](#) to avoid exporting all contained tables. The parameter pTextSettings defines the options specific to text export and import. You need to set the property [TextImportExportSettings.DestinationFolder](#) before you call ExportToText. [UpdateXMLData\(\)](#) might be indirectly needed as you have to pass the [XMLData](#) as parameter to this function.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.
- 1416 Error during export.
- 1430 Document export failed.

Example

```
'-----
' VBA client code fragment - export document to text files
'-----
    Dim objDoc As Document
    Set objDoc = objSpy.ActiveDocument

    Dim objExpSettings As ExportSettings
    Set objExpSettings = objSpy.GetExportSettings
    objExpSettings.ElementList = objDoc.GetExportElementList(
                                                objDoc.RootElement,
                                                objExpSettings)

    Dim objTextExp As TextImportExportSettings
    Set objTextExp = objSpy.GetTextImportExportSettings
    objTextExp.HeaderRow = True
    objTextExp.DestinationFolder = "C:\Exports"

    On Error Resume Next
    objDoc.ExportToText objDoc.RootElement, objExpSettings, objTextExp

    If Err.Number <> 0 Then
        a = MsgBox("Error: " & (Err.Number - vbObjectError) & Chr(13) & "Description: "
        & Err.Description)
    End If
```

13.3.2.10.24 FlattenDTDOOrSchema

Method: FlattenDTDOOrSchema (sOutputPath as String, nOutputPathDialogAction as [SPYDialogAction](#))

Parameters

sOutputPath

The file path for the newly generated file.

nOutputPathDialogAction

Defines the dialog interaction for this call.

Description

`FlattenDTDOrSchema` takes an existing DTD or schema, generates a flattened file, and saves the generated file at the specified location. In the case of DTDs, flattening removes parameter entities and produces a single DTD from a collection of modules; sections marked `IGNORE` are suppressed and unused parameter entities are deleted. When an XML Schema is flattened, (i) the components of all included schemas are added as global components of the active schema, and (ii) included schemas are deleted.

Errors

- 1400 The object is no longer valid.
- 1412 Error during conversion.

13.3.2.10.25 FullName

Property: FullName as String

Description

This property can be used to get or set the full file name - including the path - to where the document gets saved. The validity of the name is not verified before the next save operation.

This property makes the methods [GetPathName](#) and [SetPathName](#) obsolete.

Errors

- 1400 The document object is no longer valid.
- 1402 Empty string has been specified as full file name.

13.3.2.10.26 GenerateDTDOrSchema

Method: `GenerateDTDOrSchema` (*nFormat* as [SPYDTDSchemaFormat](#), *nValuesList* as integer, *nDetection* as [SPYTypeDetection](#), *nFrequentElements* as [SPYFrequentElements](#))

Parameters

nFormat

Sets the schema output format to DTD, or W3C.

nValuesList

Generate not more than this amount of enumeration-facets per type. Set to -1 for unlimited.

nDetection

Specifies granularity of simple type detection.

nFrequentElements

Shall the types for all elements be defined as global? Use the value `spyGlobalComplexType` to define them on global scope. Otherwise, use the value `spyGlobalElements`.

Description

Use this method to automatically generate a DTD or schema for the current XML document.

For a finer tuning of DTD / schema generation, use [GenerateDTDOrSchemaEx](#).

Note that this functionality is not available in ZIP View only. See [Document.CurrentViewMode](#). and [SPYViewModes](#).

Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.

13.3.2.10.27 GenerateDTDOOrSchemaEx

Method: GenerateDTDOOrSchemaEx (*objDlg* as [DTDSchemaGeneratorDlg](#)) as [Document](#)

Description

Use this method to automatically generate a DTD or schema for the current XML document. A [DTDSchemaGeneratorDlg](#) object is used to pass information to the schema/DTD generator. The generation process can be configured to allow user interaction or run without further user input. Note that this functionality is not available in ZIP View only. See [Document.CurrentViewMode](#). and [SPYViewModes](#).

Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.

13.3.2.10.28 GenerateProgramCode

Method: GenerateProgramCode (*objDlg* as [CodeGeneratorDlg](#))

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Generate Java, C++ or C# class files from the XML Schema definitions in your document. A [CodeGeneratorDlg](#) object is used to pass information to the code generator. The generation process can be configured to allow user interaction or run without further user input.

Errors

- 1400 The document object is no longer valid.
- 1407 An empty file name has been specified.
- 1421 Feature not available in this edition

13.3.2.10.29 GenerateSampleXML

Method: GenerateSampleXML (*objDlg* as [GenerateSampleXMLDlg](#)) as [Document](#)

Description

Generates a sample XML if the document is a schema or DTD. Use [Dialogs.GenerateSampleXMLDlg](#) to get an initialized set of options.

Available with TypeLibrary version 1.5

Errors

- 1400 The document object is no longer valid.

13.3.2.10.30 GenerateSchemaDocumentation

Method: GenerateSchemaDocumentation (*objDlg* as [SchemaDocumentationDlg](#))

Description

Generate documentation for a schema definition file in HTML, MS-Word, or RTF format. The parameter *objDlg* is used to parameterize the generation process. Use [Dialogs.SchemaDocumentationDlg](#) to get an initialized set of options. As a minimum, you will need to set the property [SchemaDocumentationDlg.OutputFile](#) before starting the generation process. Note that this functionality is limited to Schema View only. See [Document.CurrentViewMode](#) and [SPYViewModes](#).

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid parameters have been passed or an empty file name has been specified as output target.
- 1417 The document is not opened in schema view, maybe it is not an '.xsd' file.
- 1421 Feature is not available in this edition.
- 1422 Error during generation

13.3.2.10.31 GenerateWSDL20Documentation

Method: GenerateWSDL20Documentation (*objDlg* as [WSDL20DocumentationDlg](#))

Description

Generate documentation for a WSDL definition file in HTML, MS-Word, or RTF format. The parameter *objDlg* is used to parameterize the generation process. Use [Dialogs.WSDL20DocumentationDlg](#) to get an initialized set of options. As a minimum, you will need to set the property [WSDL20DocumentationDlg.OutputFile](#) before starting the generation process. Note that this functionality is limited to WSDL View only. See [Document.CurrentViewMode](#) and [SPYViewModes](#).

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid parameters have been passed or an empty file name has been specified as output target.
- 1417 The document is not opened in schema view, maybe it is not an '.xsd' file.
- 1421 Feature is not available in this edition.
- 1422 Error during generation

13.3.2.10.32 GenerateWSDLDocumentation

Method: GenerateWSDLDocumentation (*objDlg* as [WSDLDocumentationDlg](#))

Description

Generate documentation for a WSDL definition file in HTML, MS-Word, or RTF format. The parameter *objDlg* is used to parameterize the generation process. Use [Dialogs.WSDLDocumentationDlg](#) to get an initialized set of options. As a minimum, you will need to set the property [WSDLDocumentationDlg.OutputFile](#) before starting the generation process. Note that this functionality is limited to WSDL View only. See [Document.CurrentViewMode](#) and [SPYViewModes](#).

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid parameters have been passed or an empty file name has been specified as output target.
- 1417 The document is not opened in schema view, maybe it is not an '.xsd' file.
- 1421 Feature is not available in this edition.
- 1422 Error during generation

13.3.2.10.33 GenerateXBRLDocumentation

Method: GenerateXBRLDocumentation (*objDlg* as [XBRLDocumentationDlg](#))

Description

Generate documentation for an XBRL file in HTML, MS-Word, or RTF format. The parameter *objDlg* is used to parameterize the generation process. Use [Dialogs.XBRLDocumentationDlg](#) to get an initialized set of options. As a minimum, you will need to set the property [XBRLDocumentationDlg.OutputFile](#) before starting the generation process. Note that this functionality is limited to XBRL View only. See [Document.CurrentViewMode](#) and [SPYViewModes](#).

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid parameters have been passed or an empty file name has been specified as output target.
- 1417 The document is not opened in schema view, maybe it is not an '.xsd' file.
- 1421 Feature is not available in this edition.
- 1422 Error during generation

13.3.2.10.34 GetDBStructureList

Method: GetDBStructureList (*pDatabase* as [DatabaseConnection](#)) as [ElementList](#)

Description

GetDBStructureList creates a collection of elements from the Schema document for which tables in the specified database are created. The function returns a collection of [ElementListItem](#)s where the properties [ElementListItem.Name](#) contain the names of the tables.

See also [CreateDBStructureFromXMLSchema](#).

Errors

- 1400 The object is no longer valid.
- 1427 Failed creating parser for the specified XML.
- 1428 Export of element list failed.
- 1429 Database selection missing.

13.3.2.10.35 GetExportElementList

Method: GetExportElementList (*pFromChild* as [XMLData](#), *pExportSettings* as [ExportSettings](#)) as [ElementList](#)

Description

GetExportElementList creates a collection of elements to export from the document, depending on the settings in *pExportSettings* and starting from the element *pFromChild*. The function returns a collection of [ElementListItem](#)s where the properties [ElementListItem.Name](#) contain the names of the tables that can be exported from the document. The property [ElementListItem.FieldCount](#) contains the number of columns in the table. The property [ElementListItem.RecordCount](#) contains the number of records in the table. The property [ElementListItem.ElementKind](#) is unused. [UpdateXMLData\(\)](#) might be indirectly needed as you have to pass the [XMLData](#) as parameter to this function.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.
- 1427 Failed creating parser for the specified XML.
- 1428 Export of element list failed.

13.3.2.10.36 GetPathName (obsolete)

Superseded by [Document.FullName](#)

```
// ----- javascript sample -----  
// instead of:  
// strPathName = Application.ActiveDocument.GetPathName();  
// use now:  
strPathName = Application.ActiveDocument.FullName;
```

Method: GetPathName() as String

Description

The method GetPathName gets the path of the active document.

See also [Document.SetPathName](#) (obsolete).

13.3.2.10.37 GridView

Property: GridView as [GridView](#)

Description

This property provides access to the grid view functionality of the document.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid address for the return parameter was specified.
- 1417 Document needs to be open in enhanced grid view mode.

13.3.2.10.38 IsModified

Property: IsModified as Boolean

Description

True if the document is modified.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

13.3.2.10.39 IsValid

Method: HRESULT IsValid([in, out] VARIANT *strError, [in, out] VARIANT *nErrorPos, [in, out] VARIANT *pBadData, [out,retval] VARIANT_BOOL *bValid);

Return Value

True if the document is valid, false if not. To call `IsValid()`, the application GUI must be visible. (If you wish to validate without the GUI being visible, please use [Altova RaptorXML Server](#).)

Description

`IsValid` validates the document against its associated schema or DTD. `strError` gives you the same error message as when you validate the file within the GUI.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.
- 1408 Unable to validate file.

Example

The following C++ code snippet provides an example of how to use the `IsValid` method.


```

#import "XMLSpy.tlb"

CComPtr< XMLSpyLib::IDocument12> ipDoc = ipXMLSpy->GetActiveDocument();
if ( ipDoc )
{
    // prepare in/out parameters for IsValid call
    CComVariant variantError;
    CComVariant variantErrorPos;
    CComVariant variantBadData;
    // IsValid always shows a dialog with the validation result. This cannot be turned
off.
    bool bIsValid = ipDoc->IsValid( &variantError, &variantErrorPos, &variantBadData )
== VARIANT_TRUE;

    if ( !bIsValid )
    {
        // retrieve values from out parameters
        CString strError = (V_VT( &variantError ) == VT_BSTR ?
V_BSTR( &variantError ) : _T( " " ));
        long npos = (V_VT( &variantErrorPos ) == VT_I4 ? V_I4( &variantErrorPos ) :
-1);
        CComQIPtr< XMLSpyLib::IXMLData > ipXMLBadData = (V_VT( &variantBadData ) ==
VT_DISPATCH ? V_DISPATCH( &variantBadData ) : nullptr);

        if ( ipXMLBadData )
            strError += CString( _T("\n\n Node: ") ) + (LPCWSTR)ipXMLBadData-
>GetName();

        if ( !strError.IsEmpty() )
            AfxMessageBox( "Validation failed - " + strError );
    }
}

```

13.3.2.10.40 IsValidEx

Method: IsValidEx(*nXSDVersion* as [SPYValidateXSDVersion](#), *nErrorLimit* as *int*, *nErrorFormat* as [SPYValidateErrorFormat](#), out *strError* as *Variant*) as *Boolean*

Return Value

True if the document is valid, false if not.

Description

IsValidEx validates the document against its associated schema or DTD.

In parameters:

nXSDVersion which is an enumeration value of [SPYValidateXSDVersion](#) that selects the XSD version to validate against.

nErrorLimit which is an integer. Values must be 1 to 999.

nErrorFormat which is an enumeration value of [SPYValidateErrorFormat](#) that selects the XSD version to validate against.

Out parameter:

`strError` is the error message, and is the same as that received when validating the file within the GUI.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.
- 1408 Unable to validate file.

Example

The following C++ code snippet provides an example of how to use the `IsValidEx` method.

```
#import "XMLSpy.tlb"

CComPtr< XMLSpyLib::IDocument12> ipDoc = ipXMLSpy->GetActiveDocument();
if ( ipDoc )
{
    CComVariant variantErrorEx;
    bool bIsValidEx = ipDoc->IsValidEx( XMLSpyLib::spyValidateXSDVersion_AutoDetect,
100, XMLSpyLib::SPYValidateErrorFormat_LongXML, &variantErrorEx ) == VARIANT_TRUE;

    // // retrieve values from out parameters
    CString strErrorEx = (V_VT( &variantErrorEx ) == VT_BSTR ?
V_BSTR( &variantErrorEx ) : _T( "" ));
    if ( !strErrorEx.IsEmpty() )
        AfxMessageBox( "Validation failed - " + strErrorEx );
}
}
```

13.3.2.10.41 IsWellFormed

Method: `IsWellFormed` (*pData* as XMLData, *bWithChildren* as Boolean, *strError* as Variant, *nErrorPos* as Variant, *pBadXMLData* as Variant) as Boolean

Return Value

True if the document is well formed.

Description

`IsWellFormed` checks the document for well-formedness starting at the element *pData*.

If the document is not well formed, `strError` contains an error message, `nErrorPos` the position in the file and `pBadXMLData` holds a reference to the element which breaks the well-formedness. These out-parameters are defined as VARIANTS to support scripting languages like VBScript.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.

Example

See IsValid.

13.3.2.10.42 Name

Property: Name as String (read-only)

Description

Use this property to retrieve the name - not including the path - of the document file. To change the file name for a document use the property [FullName](#).

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

13.3.2.10.43 Parent

Property: Parent as [Documents](#) (read-only)

Description

Access the parent of the document object.

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

Property: Parent as [Application](#) (read-only)

13.3.2.10.44 Path

Property: Path as String (read-only)

Description

Use this property to retrieve the path - not including the file name - of the document file. To change the file name and path for a document use the property [FullName](#).

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

13.3.2.10.45 RootElement

Property: RootElement as [XMLData](#) (read-only)

Description

The property `RootElement` provides access to the root element of the XML structure of the document including the XML prolog data. To access the first element of a document's content navigate to the first child of kind `spyXMLDataElement` or use the [Document.DataRoot](#) property. If the [CurrentViewMode](#) is not `spyViewGrid` or `spyViewAuthentic` an [UpdateXMLData](#) may be necessary to get access to the latest [XMLData](#).

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

13.3.2.10.46 Save

Method: `Save()`

Description

The method writes any modifications of the document to the associated file. See also [Document.FullName](#).

Errors

- 1400 The document object is no longer valid.
- 1407 An empty file name has been specified.
- 1403 Error when saving file, probably the file name is invalid.

13.3.2.10.47 SaveAs

Method: `SaveAs (strFileName as String)`

Description

Save the document to the file specified. If saving was successful, the [FullName](#) property gets set to the specified file name.

Errors

- 1400 The document object is no longer valid.
- 1407 An empty file name has been specified.
- 1403 Error when saving file, probably the file name is invalid.

13.3.2.10.48 Saved

Property: `Saved as Boolean (read-only)`

Description

This property can be used to check if the document has been saved after the last modifications. It returns the negation of [IsModified](#).

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

13.3.2.10.49 SaveInString

Method: SaveInString (*pData* as [XMLData](#), *bMarked* as Boolean) as String

Parameters

pData

XMLData element to start. Set *pData* to [Document.RootElement](#) if you want to copy the complete file.

bMarked

If *bMarked* is true, only the elements selected in the grid view are copied.

Return Value

Returns a string with the XML data.

Description

SaveInString starts at the element *pData* and converts the XMLData objects to a string representation.

[UpdateXMLData\(\)](#) might be indirectly needed as you have to pass the [XMLData](#) as parameter to this function.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.

13.3.2.10.50 SaveToURL

Method: SaveToURL (*strURL* as String, *strUser* as String, *strPassword* as String)

Return Value

Description

SaveToURL() writes the document to the URL *strURL*. This method does not set the permanent file path of the document.

Errors

- 1400 The object is no longer valid.
- 1402 Invalid URL specified.
- 1403 Error while saving to URL.

13.3.2.10.51 SetActiveDocument

Method: SetActiveDocument()

Description

The method sets the document as the active and brings it to the front.

Errors

- 1400 The object is no longer valid.

13.3.2.10.52 SetEncoding (obsolete)

Superseded by [Document.Encoding](#)

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.SetEncoding("UTF-16");  
// use now:  
Application.ActiveDocument.Encoding = "UTF-16";
```

Method: SetEncoding (*strEncoding* as String)

Description

SetEncoding sets the encoding of the document like the menu item "File/Encoding..." in Authentic Desktop. Possible values for *strEncoding* are, for example:

- 8859-1,
- 8859-2,
- ASCII, ISO-646,
- 850,
- 1252,
- 1255,
- SHIFT-JIS, MS-KANJI,
- BIG5, FIVE,
- UTF-7,
- UTF-8,
- UTF-16

13.3.2.10.53 SetExternalIsValid

Method: SetExternalIsValid (*bValid* as Boolean)

Parameters

bValid
Sets the result of an external validation process.

Description

The internal information set by this method is only queried on cancelling the default validation in any [OnBeforeValidate](#) handler.

Available with TypeLibrary version 1.5

Errors

1400 The object is no longer valid.

13.3.2.10.54 SetPathName (obsolete)

Superseded by [Document.FullName](#)

```
// ---- javascript sample ----  
// instead of:  
// Application.ActiveDocument.SetPathName("C:\\myXMLFiles\\test.xml");  
// use now:  
Application.ActiveDocument.FullName = "C:\\myXMLFiles\\test.xml";
```

Method: SetPathName (*strPath* as String)

Description

The method SetPathName sets the path of the active document. SetPathName only copies the string and does not check if the path is valid. All succeeding save operations are done into this file.

13.3.2.10.55 StartChanges

Method: StartChanges()

Description

After StartChanges is executed Authentic Desktop will not update its editor windows until [Document.EndChanges](#) is called. This increases performance of complex tasks to the XML structure.

Errors

1400 The object is no longer valid.

13.3.2.10.56 Suggestions

Property: Suggestions as Array

Description

This property contains the last valid user suggestions for this document. The XMLSpy generated suggestions can be modified before they are shown to the user in the [OnBeforeShowSuggestions](#) event.

Errors

1400 The object is no longer valid.

1407 Invalid parameter or invalid address for the return parameter was specified.

13.3.2.10.57 SwitchViewMode

Method: SwitchViewMode (*nMode* as [SPYViewModes](#)) as Boolean

Return value

Returns true if view mode is switched.

Description

The method sets the current view mode of the document in Authentic Desktop. See also [Document.CurrentViewMode](#).

Errors

- 1400 The object is no longer valid.
- 1407 Invalid address for the return parameter was specified.
- 1417 Invalid view mode specified.

13.3.2.10.58 TextView

Property: TextView as [TextView](#)

Description

This property provides access to the text view functionality of the document.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

13.3.2.10.59 Title

Property: Title as String (read-only)

Description

Title contains the file name of the document. To get the path and filename of the file use [FullName](#).

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

13.3.2.10.60 TransformXSL

Method: TransformXSL()

Description

TransformXSL processes the XML document via the associated XSL file. See [Document.AssignXSL](#) on how to place a reference to a XSL file into the document.

Errors

- 1400 The document object is no longer valid.
- 1411 Error during transformation process.

13.3.2.10.61 TransformXSLEx

Method: TransformXSLEx(*nAction* as [SPYDialogAction](#))

Description

TransformXSLEx processes the XML document via the associated XSL file. The parameter specifies whether a dialog asking for the result document name should pop up or not. See [Document.AssignXSL](#) on how to place a reference to a XSL file into the document.

Errors

- 1400 The document object is no longer valid.
- 1411 Error during transformation process.

13.3.2.10.62 TransformXSLFO

Method: TransformXSLFO()

Description

TransformXSLFO processes the XML document via the associated XSLFO file. See [AssignXSLFO](#) on how to place a reference to a XSLFO file into the document. You need to assign a FOP processor to Authentic Desktop before you can use this method.

Errors

- 1400 The document object is no longer valid.
- 1411 Error during transformation process.

13.3.2.10.63 TreatXBRLInconsistenciesAsErrors

Property: TreatXBRLInconsistenciesAsErrors as Boolean

Description

If this is set to `true` the `Document.IsValid()` method will return `false` for XBRL instances containing inconsistencies as defined by the XBRL Specification. The default value of this property is `false`.

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

13.3.2.10.64 UpdateViews

Method: UpdateViews()

Description

To redraw the Enhanced Grid View and the Tree View call `UpdateViews`. This can be important after you changed the `XMLData` structure of a document. This method does not redraw the text view of Authentic Desktop.

Errors

1400 The document object is no longer valid.

13.3.2.10.65 UpdateXMLData

Method: UpdateXMLData() as Boolean

Description

The [XMLData](#) tree is updated from the current view. Please note that this can fail in case of the `TextView` if the current XML text is not well-formed. This is not necessary if [CurrentViewMode](#) is `spyViewGrid` or `spyViewAuthentic` because these views keep the [XMLData](#) updated.

Available with TypeLibrary version 1.5

Errors

1400 The document object is no longer valid.

13.3.2.11 Documents

Properties

[Count](#)

[Item](#)

Methods

[NewAuthenticFile](#)

[NewFile](#)

[NewFileFromText](#)

[OpenAuthenticFile](#)

[OpenFile](#)

[OpenURL](#)

[OpenURLDialog](#)

Description

This object represents the set of documents currently open in Authentic Desktop. Use this object to open further documents or iterate through already opened documents.

Examples

```
'-----  
' XMLSpy scripting environment - VBScript  
' iterate through open documents  
'-----  
Dim objDocuments  
Set objDocuments = Application.Documents
```

```
For Each objDoc In objDocuments
    'do something useful with your document
    objDoc.SetActiveDocument()
Next

// -----
// XMLSpy scripting environment - JScript
// close all open documents
// -----
for (var iter = new Enumerator (Application.Documents);
    ! iter.atEnd();
    iter.moveNext())
{
    // MsgBox ("Closing file " + iter.item().Name);
    iter.item().Close (true);
}
```

13.3.2.11.1 Count

Property: Count as long

Description

Count of open documents.

Errors

1600	Invalid Documents object
1601	Invalid input parameter

13.3.2.11.2 Item

Method: Item (*n* as long) as [Document](#)

Description

Gets the document with the index *n* in this collection. Index is 1-based.

Errors

1600	Invalid Documents object
1601	Invalid input parameter

13.3.2.11.3 NewAuthenticFile

Method: NewAuthenticFile (*strSPSPath* as String, *strXMLPath* as String) as [Document](#)

Parameters

strSPSPath

The path to the SPS document.

strXMLPath

The new XML document name.

Return Value

The method returns the new document.

Description

NewAuthenticFile creates a new XML file and opens it in Authentic View using SPS design strSPSPath.

13.3.2.11.4 NewFile

Method: NewFile (*strFile* as String, *strType* as String) as [Document](#)

Parameters

strFile

Full path of new file.

strType

Type of new file as string (i.e. "xml", "xsd", ...)

Return Value

Returns the new file.

Description

NewFile creates a new file of type strType (i.e. "xml"). The newly created file is also the ActiveDocument.

13.3.2.11.5 NewFileFromText

Method: NewFileFromText (*strText* as String, *strType* as String) as [Document](#)

Parameters

strText

The content of the new document in plain text.

strType

Type of the document to create (i.e. "xml").

Return Value

The method returns the new document.

Description

NewFileFromText creates a new document with strText as its content.

13.3.2.11.6 OpenAuthenticFile

Method: OpenAuthenticFile (*strSPSPath* as String, *strXMLPath* as String) as [Document](#)

Parameters

strSPSPath

The path to the SPS document.

strXMLPath

The path to the XML document (can be empty).

Return Value

The method returns the new document.

Description

OpenAuthenticFile opens an XML file or database in Authentic View using SPS design strSPSPath.

13.3.2.11.7 OpenFile

Method: OpenFile (*strPath* as String, *bDialog* as Boolean) as [Document](#)

Parameters

strPath

Path and file name of file to open.

bDialog

Show dialogs for user input.

Return Value

Returns the opened file on success.

Description

OpenFile opens the file strPath. If bDialog is TRUE, a file-dialog will be displayed.

Example

```
Dim objDoc As Document
Set objDoc = objSpy.Documents.OpenFile(strFile, False)
```

13.3.2.11.8 OpenURL

Method: OpenURL (*strURL* as String, *nURLType* as [SPYURLTypes](#), *nLoading* as [SPYLoading](#), *strUser* as String, *strPassword* as String) as [Document](#)

Parameters

strURL

URL to open as document.

nURLType

Type of document to open. Set to -1 for auto detection.

nLoading

Set nLoading to 0 (zero) if you want to load it from cache or proxy. Otherwise set nLoading to 1.

strUser

Name of the user if required. Can be empty.

strPassword

Password for authentication. Can be empty.

Return Value

The method returns the opened document.

Description

OpenURL opens the URL strURL.

13.3.2.11.9 OpenURLDialog

Method: OpenURLDialog (*strURL* as String, *nURLType* as [SPYURLTypes](#), *nLoading* as [SPYLoading](#), *strUser* as String, *strPassword* as String) as [Document](#)

Parameters

strURL

URL to open as document.

nURLType

Type of document to open. Set to -1 for auto detection.

nLoading

Set nLoading to 0 (zero) if you want to load it from cache or proxy. Otherwise set nLoading to 1.

strUser

Name of the user if required. Can be empty.

strPassword

Password for authentication. Can be empty.

Return Value

The method returns the opened document.

Description

OpenURLDialog displays the "open URL" dialog to the user and presets the input fields with the given parameters.

13.3.2.12 DTDSchemaGeneratorDlg

Properties and Methods

Standard automation properties

[Application](#)

[Parent](#)

[DTDSchemaFormat](#)

[ValueList](#)

[TypeDetection](#)

[FrequentElements](#)

[MergeAllEqualNamed](#)

[ResolveEntities](#)

[AttributeTypeDefinition](#)

[GlobalAttributes](#)

[OnlyStringEnums](#)
[MaxEnumLength](#)
[OutputPath](#)
[OutputPathDialogAction](#)

Description

Use this object to configure the generation of a schema or DTD. The method [GenerateDTDOrSchemaEx](#) expects a DTDSchemaGeneratorDlg as parameter to configure the generation as well as the associated user interactions.

13.3.2.12.1 Application

Property: Application as [Application](#) (read-only)

Description

Access the Authentic Desktop application object.

Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

13.3.2.12.2 AttributeTypeDefinition

Property: AttributeTypeDefinition as [SPYAttributeTypeDefinition](#)

Description

Specifies how attribute definitions get merged.

Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

13.3.2.12.3 DTDSchemaFormat

Property: DTDSchemaFormat as [SPYDTDSchemaFormat](#)

Description

Sets the schema output format to DTD, or W3C.

Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

13.3.2.12.4 FrequentElements

Property: FrequentElements as [SPYFrequentElements](#)

Description

Shall the types for all elements be defined as global? Use the value *spyGlobalComplexType* to define them on global scope. Otherwise, use the value *spyGlobalElements*.

Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

13.3.2.12.5 GlobalAttributes

Property: GlobalAttributes as Boolean

Description

Shall attributes with same name and type be resolved globally?

Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

13.3.2.12.6 MaxEnumLength

Property: MaxEnumLength as Integer

Description

Specifies the maximum number of characters allowed for enumeration names. If one value is longer than this, no enumeration will be generated.

Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

13.3.2.12.7 MergeAllEqualNamed

Property: MergeAllEqualNamed as Boolean

Description

Shall types of all elements with the same name be merged into one type?

Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

13.3.2.12.8 OnlyStringEnums

Property: OnlyStringEnums as Boolean

Description

Specifies if enumerations will be created only for plain strings or all types of values.

Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

13.3.2.12.9 OutputPath

Property: OutputPath as String

Description

Selects the file name for the generated schema/DTD.

Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

13.3.2.12.10 OutputPathDialogAction

Property: OutputPathDialogAction as [SPYDialogAction](#)

Description

Defines how the sub-dialog for selecting the schema/DTD output path gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog with the current value of the [OutputPath](#) property as default. Use *spyDialogOK(0)* to hide the dialog from the user.

Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

13.3.2.12.11 Parent

Property: Parent as [Dialogs](#) (read-only)

Description

Access the parent of the object.

Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

13.3.2.12.12 ResolveEntities

Property: ResolveEntities as Boolean

Description

Shall all entities be resolved before generation starts? If yes, an info-set will be built.

Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

13.3.2.12.13 TypeDetection

Property: TypeDetection as [SPYTypeDetection](#)

Description

Specifies granularity of simple type detection.

Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

13.3.2.12.14 ValueList

Property: ValueList as Integer

Description

Generate not more than this amount of enumeration-facets per type. Set to -1 for unlimited.

Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

13.3.2.13 ElementList

Properties

[Count](#)
[Item](#)

Methods

[RemoveElement](#)

Description

Element lists are used for different purposes during export and import of data. Depending on this purpose, different properties of [ElementListItem](#) are used.

It can hold

- a list of table names returned by a call to [Application.GetDatabaseTables](#),
- a list of field names returned by a call to [Application.GetDatabaseImportElementList](#) or [Application.GetTextImportElementList](#),
- a field name filter list used in [Application.ImportFromDatabase](#) and [Application.ImportFromText](#),
- a list of table names and counts for their rows and columns as returned by calls to [GetExportElementList](#) or
- a field name filter list used in [Document.ExportToDatabase](#) and [Document.ExportToText](#).

13.3.2.13.1 Count

Property: Count as long (read-only)

Description

Count of elements in this collection.

13.3.2.13.2 Item

Method: Item(n as long) as [ElementListItem](#)

Description

Gets the element with the index n from this collection. The first item has index 1.

13.3.2.13.3 RemoveElement

Method: RemoveElement(Index as long)

Description

RemoveElement removes the element Index from the collection. The first Item has index 1.

13.3.2.14 ElementListItem

Properties

[Name](#)

[ElementKind](#)

[FieldCount](#)

[RecordCount](#)

Description

An element in an [ElementList](#). Usage of its properties depends on the purpose of the element list. For details see [ElementList](#).

13.3.2.14.1 ElementKind

Property: ElementKind as [SPYXMLDataKind](#)

Description

Specifies if a field should be imported as XML element (data value of spyXMLDataElement) or attribute (data value of spyXMLDataAttr).

13.3.2.14.2 FieldCount

Property: FieldCount as long (read-only)

Description

Count of fields (i.e. columns) in the table described by this element. This property is only valid after a call to [Document.GetExportElementList](#).

13.3.2.14.3 Name

Property: Name as String (read-only)

Description

Name of the element. This is either the name of a table or a field, depending on the purpose of the element list.

13.3.2.14.4 RecordCount

Property: RecordCount as long (read-only)

Description

Count of records (i.e. rows) in the table described by this element. This property is only valid after a call to [Document.GetExportElementList](#).

13.3.2.15 ExportSettings

Properties

[ElementList](#)

[EntitiesToText](#)

[ExportAllElements](#)
[SubLevelLimit](#)

[FromAttributes](#)
[FromSingleSubElements](#)
[FromTextValues](#)

[CreateKeys](#)
[IndependentPrimaryKey](#)

[Namespace](#)

[ExportCompleteXML](#)
[StartFromElement](#)

Description

ExportSettings contains options used during export of XML data to a database or text file.

13.3.2.15.1 CreateKeys

Property: CreateKeys as Boolean

Description

This property turns creation of keys (i.e. primary key and foreign key) on or off. Default is True.

13.3.2.15.2 ElementList

Property: ElementList as [ElementList](#)

Description

Default is empty list. This list of elements defines which fields will be exported. To get the list of available fields use [Document.GetExportElementList](#). It is possible to prevent exporting columns by removing elements from this list with [ElementList.RemoveElement](#) before passing it to [Document.ExportToDatabase](#) or [Document.ExportToText](#).

13.3.2.15.3 EntitiesToText

Property: EntitiesToText as Boolean

Description

Defines if XML entities should be converted to text or left as they are during export. Default is True.

13.3.2.15.4 ExportAllElements

Property: ExportAllElements as Boolean

Description

If set to true, all elements in the document will be exported. If set to false, then [ExportSettings.SubLevelLimit](#) is used to restrict the number of sub levels to export. Default is true.

13.3.2.15.5 ExportCompleteXML

Property: ExportCompleteXML as Boolean

Description

Defines whether the complete XML is exported or only the element specified by [StartFromElement](#) and its children. Default is True.

13.3.2.15.6 FromAttributes

Property: FromAttributes as Boolean

Description

Set FromAttributes to false if no export data should be created from attributes. Default is True.

13.3.2.15.7 FromSingleSubElements

Property: FromSingleSubElements as Boolean

Description

Set FromSingleSubElements to false if no export data should be created from elements. Default is True.

13.3.2.15.8 FromTextValues

Property: FromTextValues as Boolean

Description

Set FromTextValues to false if no export data should be created from text values. Default is True.

13.3.2.15.9 IndependentPrimaryKey

Property: IndependentPrimaryKey as Boolean

Description

Turns creation of independent primary key counter for every element on or off. If [ExportSettings.CreateKeys](#) is False, this property will be ignored. Default is True.

13.3.2.15.10 Namespace

Property: Namespace as [SPYExportNamespace](#)

Description

The default setting removes all namespace prefixes from the element names. In some database formats the colon is not a legal character. Default is spyNoNamespace.

13.3.2.15.11 StartFromElement

Property: StartFromElement as String

Description

Specifies the start element for the export. This property is only considered when [ExportCompleteXML](#) is false.

13.3.2.15.12 SubLevelLimit

Property: SubLevelLimit as Integer

Description

Defines the number of sub levels to include for the export. Default is 0. This property is ignored if [ExportSettings.ExportAllElements](#) is true.

13.3.2.16 FileSelectionDlg

Properties and Methods

Standard automation properties

[Application](#)

[Parent](#)

Dialog properties

[FullName](#)

Acceptance or cancellation of action that caused event

[DialogAction](#)

Description

The dialog object allows you to receive information about an event and pass back information to the event handler in the same way as with a user dialog. Use the [FileSelectionDlg.FullName](#) to select or modify the file path and set the [FileSelectionDlg.DialogAction](#) property to cancel or agree with the action that caused the event.

13.3.2.16.1 Application

Property: Application as [Application](#) (read-only)

Description

Access the Authentic Desktop application object.

Errors

- 2400 The object is no longer valid.
- 2401 Invalid address for the return parameter was specified.

13.3.2.16.2 DialogAction

Property: DialogAction as [SPYDialogAction](#)

Description

If you want your script to perform the file selection operation without any user interaction necessary, simulate user interaction by either setting the property to *spyDialogOK(0)* or *spyDialogCancel(1)*.

To allow your script to fill in the default values but let the user see and react on the dialog, use the value *spyDialogUserInput(2)*. If you receive a FileSelectionDlg object in an event handler, *spyDialogUserInput(2)* is not supported and will be interpreted as *spyDialogOK(0)*.

Errors

- 2400 The object is no longer valid.
- 2401 Invalid value for dialog action or invalid address for the return parameter was specified.

13.3.2.16.3 FullName

Property: FullName as String

Description

Access the full path of the file the gets selected by the dialog. Most events that pass a FileSelectionDlg object to you allow you modify this value and thus influence the action that caused the event (e.g. load or save to a different location).

Errors

- 2400 The object is no longer valid.
- 2401 Invalid address for the return parameter was specified.

13.3.2.16.4 Parent

Property: Parent as [Dialogs](#) (read-only)

Description

Access the parent of the object.

Errors

- 2400 The object is no longer valid.
- 2401 Invalid address for the return parameter was specified.

13.3.2.17 FindInFilesDlg

Properties and Methods

Standard automation properties

[Application](#)

[Parent](#)

[Find](#)

[RegularExpression](#)

[Replace](#)

[DoReplace](#)

[ReplaceOnDisk](#)

[MatchWholeWord](#)

[MatchCase](#)

[SearchLocation](#)

[StartFolder](#)

[IncludeSubfolders](#)

[SearchInProjectFilesDoExternal](#)

[FileExtension](#)

[AdvancedXMLSearch](#)

[XMLElementNames](#)

[XMLElementContents](#)

[XMLAttributeNames](#)

[XMLAttributeContents](#)

[XMLComments](#)

[XMLCDATA](#)

[XMLPI](#)

[XMLRest](#)

[ShowResult](#)

Description

Use this object to configure the search (or replacement) for strings in files. The method [FindInFiles](#) expects a FindInFilesDlg as parameter.

13.3.2.17.1 AdvancedXMLSearch

Property: AdvancedXMLSearch as Boolean

Description

Specifies if the XML search properties ([XMLElementNames](#), [XMLElementContents](#), [XMLAttributeNames](#), [XMLAttributeContents](#), [XMLComments](#), [XMLCDATA](#), [XMLPI](#) and [XMLRest](#)) are considered. The default is false.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

13.3.2.17.2 Application

Property: Application as [Application](#) (read-only)

Description

Access the Authentic Desktop application object.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

13.3.2.17.3 DoReplace

Property: DoReplace as Boolean

Description

Specifies if the matched string is replaced by the string defined in [Replace](#). The default is false.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

13.3.2.17.4 FileExtension

Property: FileExtension as String

Description

Specifies the file filter of the files that should be considered during the search. Multiple file filters must be delimited with a semicolon (eg: *.xml;*.dtd;a*.xsd). Use the wildcards * and ? to define the file filter.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

13.3.2.17.5 Find

Property: Find as String

Description

Specifies the string to search for.

Errors

- 3500 The object is no longer valid.

3501 Invalid address for the return parameter was specified.

13.3.2.17.6 IncludeSubfolders

Property: IncludeSubfolders as Boolean

Description

Specifies if subfolders are searched too. The default is true.

Errors

3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

13.3.2.17.7 MatchCase

Property: MatchCase as Boolean

Description

Specifies if the search is case sensitive. The default is true.

Errors

3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

13.3.2.17.8 MatchWholeWord

Property: MatchWholeWord as Boolean

Description

Specifies whether the whole word or just a part of it must match. The default is false.

Errors

3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

13.3.2.17.9 Parent

Property: Parent as [Dialogs](#) (read-only)

Description

Access the parent of the object.

Errors

3500 The object is no longer valid.
3501 Invalid address for the return parameter was specified.

13.3.2.17.10 RegularExpression

Property: RegularExpression as Boolean

Description

Specifies if [Find](#) contains a regular expression. The default is false.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

13.3.2.17.11 Replace

Property: Replace as String

Description

Specifies the replacement string. The matched string is only replaced if [DoReplace](#) is set true.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

13.3.2.17.12 ReplaceOnDisk

Property: ReplaceOnDisk as Boolean

Description

Specifies if the replacement is done directly on disk. The modified file is not opened. The default is false.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

13.3.2.17.13 SearchInProjectFilesDoExternal

Property: SearchInProjectFilesDoExternal as Boolean

Description

Specifies if the external folders in the open project are searched, when a project search is performed. The default is false.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

13.3.2.17.14 SearchLocation

Property: SearchLocation as [SPYFindInFilesSearchLocation](#)

Description

Specifies the location of the search. The default is spyFindInFiles_Documents.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

13.3.2.17.15 ShowResult

Property: ShowResult as Boolean

Description

Specifies if the result is displayed in the Find in Files output window. The default is false.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

13.3.2.17.16 StartFolder

Property: StartFolder as String

Description

Specifies the folder where the disk search starts.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

13.3.2.17.17 XMLAttributeContents

Property: XMLAttributeContents as Boolean

Description

Specifies if attribute contents are searched when [AdvancedXMLSearch](#) is true. The default is true.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

13.3.2.17.18 XMLAttributeNames

Property: XMLAttributeNames as Boolean

Description

Specifies if attribute names are searched when [AdvancedXMLSearch](#) is true. The default is true.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

13.3.2.17.19 XMLCDATA

Property: XMLCDATA as Boolean

Description

Specifies if CDATA tags are searched when [AdvancedXMLSearch](#) is true. The default is true.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

13.3.2.17.20 XMLComments

Property: XMLComments as Boolean

Description

Specifies if comments are searched when [AdvancedXMLSearch](#) is true. The default is true.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

13.3.2.17.21 XMLElementContents

Property: XMLElementContents as Boolean

Description

Specifies if element contents are searched when [AdvancedXMLSearch](#) is true. The default is true.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

13.3.2.17.22 XMLElementNames

Property: XMLElementNames as Boolean

Description

Specifies if element names are searched when [AdvancedXMLSearch](#) is true. The default is true.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

13.3.2.17.23 XMLPI

Property: XMLPI as Boolean

Description

Specifies if XML processing instructions are searched when [AdvancedXMLSearch](#) is true. The default is true.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

13.3.2.17.24 XMLRest

Property: XMLRest as Boolean

Description

Specifies if the rest of the XML (which is not covered by the other XML search properties) is searched when [AdvancedXMLSearch](#) is true. The default is true.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

13.3.2.18 FindInFilesResult

Properties and Methods

Standard automation properties

[Application](#)
[Parent](#)

[Count](#)
[Item](#)

[Path](#)

[Document](#)

Description

This object represents a file that matched the search criteria. It contains a list of [FindInFilesResultMatch](#) objects that describe the matching position.

13.3.2.18.1 Application

Property: Application as [Application](#) (read-only)

Description

Access the Authentic Desktop application object.

Errors

- 3700 The object is no longer valid.
- 3701 Invalid address for the return parameter was specified.

13.3.2.18.2 Count

Property: Count as long (read-only)

Description

Count of elements in this collection.

13.3.2.18.3 Document

Property: Path as [Document](#) (read-only)

Description

This property returns the [Document](#) object if the matched file is already open in XMLSpy.

Errors

- 3700 The object is no longer valid.
- 3701 Invalid address for the return parameter was specified.

13.3.2.18.4 Item

Method: Item(n as long) as [FindInFilesResultMatch](#)

Description

Gets the element with the index n from this collection. The first item has index 1.

13.3.2.18.5 Parent

Property: Parent as [FindInFilesResults](#) (read-only)

Description

Access the parent of the object.

Errors

- 3700 The object is no longer valid.
- 3701 Invalid address for the return parameter was specified.

13.3.2.18.6 Path

Property: Path as String (read-only)

Description

Returns the path of the file that matched the search criteria.

Errors

- 3700 The object is no longer valid.
- 3701 Invalid address for the return parameter was specified.

13.3.2.19 FindInFilesResultMatch

Properties and Methods

Standard automation properties

[Application](#)

[Parent](#)

[Line](#)

[Position](#)

[Length](#)

[LineText](#)

[Replaced](#)

Description

Contains the exact position in the file of the matched string.

13.3.2.19.1 Application

Property: Application as [Application](#) (read-only)

Description

Access the Authentic Desktop application object.

Errors

- 3800 The object is no longer valid.
- 3801 Invalid address for the return parameter was specified.

13.3.2.19.2 Length

Property: Length as Long (read-only)

Description

Returns the length of the matched string.

Errors

- 3800 The object is no longer valid.
- 3801 Invalid address for the return parameter was specified.

13.3.2.19.3 Line

Property: Line as Long (read-only)

Description

Returns the line number of the match. The line numbering starts with 0.

Errors

- 3800 The object is no longer valid.
- 3801 Invalid address for the return parameter was specified.

13.3.2.19.4 LineText

Property: LineText as String (read-only)

Description

Returns the text of the line.

Errors

- 3800 The object is no longer valid.
- 3801 Invalid address for the return parameter was specified.

13.3.2.19.5 Parent

Property: Parent as [FindInFilesResult](#) (read-only)

Description

Access the parent of the object.

Errors

- 3800 The object is no longer valid.
- 3801 Invalid address for the return parameter was specified.

13.3.2.19.6 Position

Property: Position as Long (read-only)

Description

Returns the start position of the match in the line. The position numbering starts with 0.

Errors

- 3800 The object is no longer valid.
- 3801 Invalid address for the return parameter was specified.

13.3.2.19.7 Replaced

Property: Replaced as Boolean (read-only)

Description

True if the matched string was replaced.

Errors

- 3800 The object is no longer valid.
- 3801 Invalid address for the return parameter was specified.

13.3.2.20 FindInFilesResults

Properties and Methods

Standard automation properties

[Application](#)
[Parent](#)

[Count](#)
[Item](#)

Description

This is the result of the [FindInFiles](#) method. It is a list of [FindInFilesResult](#) objects.

13.3.2.20.1 Application

Property: Application as [Application](#) (read-only)

Description

Access the Authentic Desktop application object.

Errors

- 3600 The object is no longer valid.
- 3601 Invalid address for the return parameter was specified.

13.3.2.20.2 Count

Property: Count as long (read-only)

Description

Count of elements in this collection.

13.3.2.20.3 Item

Method: Item(n as long) as [FindInFilesResult](#)

Description

Gets the element with the index n from this collection. The first item has index 1.

13.3.2.20.4 Parent

Property: Parent as [Application](#) (read-only)

Description

Access the parent of the object.

Errors

- 3600 The object is no longer valid.
- 3601 Invalid address for the return parameter was specified.

13.3.2.21 GenerateSampleXMLDlg

Properties and Methods

Standard automation properties

[Application](#)

[Parent](#)

[NonMandatoryAttributes](#)

[NonMandatoryElements](#)

[RepeatCount](#)

[FillAttributesWithSampleData](#)

[FillElementsWithSampleData](#)

[ContentOfNillableElementsIsNonMandatory](#)

[TryToUseNonAbstractTypes](#)

[SchemaOrDTDAssignment](#)

[LocalNameOfRootElement](#)

[NamespaceURIOfRootElement](#)

[OptionsDialogAction](#)

Properties that are no longer supported

[TakeFirstChoice](#) - obsolete

[FillWithSampleData](#) - obsolete

[Optimize](#) - obsolete

Description

Used to set the parameters for the generation of sample XML instances based on a W3C schema or DTD.

13.3.2.21.1 Application

Property: Application as [Application](#) (read-only)

Description

Access the Authentic Desktop application object.

Errors

2200 The object is no longer valid.

2201 Invalid address for the return parameter was specified.

13.3.2.21.2 ChoiceMode

Property: ChoiceMode as [SPYSampleXMLGenerationChoiceMode](#)

Description

Specifies which elements will be generated.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

13.3.2.21.3 ConsiderSampleValueHints

Property: ConsiderSampleValueHints as Boolean

Description

Selects whether to use [SampleValueHints](#) or not.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

13.3.2.21.4 ContentOfNillableElementsIsNonMandatory

Property: ContentOfNillableElementsIsNonMandatory as Boolean

Description

If true, the contents of elements that are nillable will not be treated as mandatory.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

13.3.2.21.5 FillAttributesWithSampleData

Property: FillAttributesWithSampleData as Boolean

Description

If true, attributes will have sample content.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

13.3.2.21.6 FillElementsWithSampleData

Property: FillElementsWithSampleData as Boolean

Description

If true, elements will have sample content.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

13.3.2.21.7 `FillWithSampleData` – obsolete

Property: `FillWithSampleData` as Boolean

Description

Do no longer access this property. Use [FillAttributesWithSampleData](#) and [FillElementsWithSampleData](#), instead.

Errors

- 0001 The property is no longer accessible.

13.3.2.21.8 `LocalNameOfRootElement`

Property: `LocalNameOfRootElement` as String

Description

Specifies the local name of the root element for the generated sample XML.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

13.3.2.21.9 `NamespaceURIOfRootElement`

Property: `NamespaceURIOfRootElement` as String

Description

Specifies the namespace URI of the root element for the generated sample XML.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

13.3.2.21.10 `NonMandatoryAttributes`

Property: `NonMandatoryAttributes` as Boolean

Description

If `true` attributes which are not mandatory are created in the sample XML instance file.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

13.3.2.21.11 NonMandatoryElements

Property: NonMandatoryElements as Boolean

Description

If `true`, elements which are not mandatory are created in the sample XML instance file.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address was specified for the return parameter.

13.3.2.21.12 Optimization – obsolete

Property: Optimization as [SPYSampleXMLGenerationOptimization](#)

Description

Do not use this property any longer. Use ChoiceMode and NonMandatoryElements.

Errors

- 0001 The property is no longer accessible.

13.3.2.21.13 OptionsDialogAction

Property: OptionsDialogAction as [SPYDialogAction](#)

Description

To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value `spyDialogUserInput(2)`. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use `spyDialogOK(0)`. Default is `spyDialogOK`.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

13.3.2.21.14 Parent

Property: Parent as [Dialogs](#) (read-only)

Description

Access the parent of the object.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

13.3.2.21.15 RepeatCount

Property: RepeatCount as long

Description

Number of elements to create for repeated types.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

13.3.2.21.16 SampleValueHints

Property: SampleValueHints as [SPYSampleXMLGenerationSampleValueHints](#)

Description

Specifies how to select data for the generated sample file.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

13.3.2.21.17 SchemaOrDTDAssignment

Property: SchemaOrDTDAssignment as [SPYSampleXMLGenerationSchemaOrDTDAssignment](#)

Description

Specifies in which way a reference to the related schema or DTD - which is this document - will be generated into the sample XML.

Errors

- 2200 The object is no longer valid.

2201 Invalid address for the return parameter was specified.

13.3.2.21.18 TakeFirstChoice – obsolete

Property: TakeFirstChoice as Boolean

Description

Do no longer use this property.

Errors

0001 The property is no longer accessible.

13.3.2.21.19 TryToUseNonAbstractTypes

Property: TryToUseNonAbstractTypes as Boolean

Description

If true, tries to use a non-abstract type for xsi:type, if element has an abstract type.

Errors

2200 The object is no longer valid.

2201 Invalid address for the return parameter was specified.

13.3.2.22 GridView

Methods

[Deselect](#)

[Select](#)

[SetFocus](#)

Properties

[CurrentFocus](#)

[IsVisible](#)

Description

GridView Class

13.3.2.22.1 Events

13.3.2.22.1.1 *OnBeforeDrag*

Event: OnBeforeDrag() as Boolean

XMLSpy scripting environment - VBScript:

```
Function On_BeforeDrag()  
    ' On_BeforeStartEditing = False ' to prohibit dragging  
End Function
```

XMLSpy scripting environment - JScript:

```
function On_BeforeDrag()  
{  
    // return false; /* to prohibit dragging */  
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (4, ...) // nEventId = 4
```

Description

This event gets fired on an attempt to drag an XMLData element on the grid view. Return *false* to prevent dragging the data element to a different position.

13.3.2.22.1.2 *OnBeforeDrop*

Event: OnBeforeDrop(objXMLData as [XMLData](#)) as Boolean

XMLSpy scripting environment - VBScript:

```
Function On_BeforeDrop(objXMLData)  
    ' On_BeforeStartEditing = False ' to prohibit dropping  
End Function
```

XMLSpy scripting environment - JScript:

```
function On_BeforeDrop(objXMLData)  
{  
    // return false; /* to prohibit dropping */  
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (5, ...) // nEventId = 5
```

Description

This event gets fired on an attempt to drop a previously dragged XMLData element on the grid view. Return *false* to prevent the data element to be moved from its original position to the drop destination position.

13.3.2.22.1.3 *OnBeforeStartEditing*

Event: OnBeforeStartEditing(objXMLData as [XMLData](#), bEditingName as Boolean)as Boolean

XMLSpy scripting environment - VBScript:

```
Function On_BeforeStartEditing(objXMLData, bEditingName)
    ' On_BeforeStartEditing = False ' to prohibit editing the field
End Function
```

XMLSpy scripting environment - JScript:

```
function On_BeforeStartEditing(objXMLData, bEditingName)
{
    // return false; /* to prohibit editing the field */
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (1, ...) // nEventId = 1
```

Description

This event gets fired before the editing mode for a grid cell gets entered. If the parameter *bEditingName* is true, the name part of the element will be edited, if its value is false, the value part will be edited.

13.3.2.22.1.4 *OnEditingFinished*

Event: OnEditingFinished(objXMLData as [XMLData](#), bEditingName as Boolean)

XMLSpy scripting environment - VBScript:

```
Function On_EditingFinished(objXMLData, bEditingName)
End Function
```

XMLSpy scripting environment - JScript:

```
function On_EditingFinished(objXMLData, bEditingName)
{
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (2, ...) // nEventId = 2
```

Description

This event gets fired when the editing mode of a grid cell is exited. The parameter *bEditingName* specifies if the name part of the element has been edited.

13.3.2.22.1.5 *OnFocusChanged*

Event: OnFocusChanged(objXMLData as [XMLData](#), bSetFocus as Boolean, bEditingName as Boolean)

XMLSpy scripting environment - VBScript:

```
Function On_FocusChanged(objXMLData, bSetFocus, bEditingName)
End Function
```

XMLSpy scripting environment - JScript:

```
function On_FocusChanged(objXMLData, bSetFocus, bEditingName)
{
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (3, ...) // nEventId = 3
```

Description

This event gets fired whenever a grid cell receives or loses the cursor focus. If the parameter *bEditingName* is *true*, focus of the name part of the grid element has changed. Otherwise, focus of the value part has changed.

13.3.2.22.2 CurrentFocus

Property: CurrentFocus as [XMLData](#)

Description

Holds the XML element with the current focus. This property is read-only.

13.3.2.22.3 Deselect

Method: Deselect(*pData* as [XMLData](#))

Description

Deselects the element *pData* in the grid view.

13.3.2.22.4 IsVisible

Property: IsVisible as Boolean

Description

True if the grid view is the active view of the document. This property is read-only.

13.3.2.22.5 Select

Method: Select (*pData* as [XMLData](#))

Description

Selects the XML element *pData* in the grid view.

13.3.2.22.6 SetFocus

Method: SetFocus (*pFocusData* as [XMLData](#))

Description

Sets the focus to the element *pFocusData* in the grid view.

13.3.2.23 SchemaDocumentationDlg

Properties and Methods

Standard automation properties

[Application](#)

[Parent](#)

Interaction and visibility properties

[OutputFile](#)

[OutputFileDialogAction](#)

[OptionsDialogAction](#)

[ShowProgressBar](#)

[ShowResult](#)

Document generation options and methods

[OutputFormat](#)

[UseFixedDesign](#)

[SPSFile](#)

[EmbedDiagrams](#)

[DiagramFormat](#)

[MultipleOutputFiles](#)

[EmbedCSSInHTML](#)

[CreateDiagramsFolder](#)

[GenerateRelativeLinks](#)

[IncludeAll](#)

[IncludeIndex](#)

[IncludeGlobalAttributes](#)

[IncludeGlobalElements](#)

[IncludeLocalAttributes](#)

[IncludeLocalElements](#)

[IncludeGroups](#)

[IncludeComplexTypes](#)

[IncludeSimpleTypes](#)

[IncludeAttributeGroups](#)

[IncludeRedefines](#)

[IncludeReferencedSchemas](#)

[AllDetails](#)

[ShowDiagram](#)

[ShowNamespace](#)

[ShowType](#)

[ShowChildren](#)

[ShowUsedBy](#)

[ShowProperties](#)
[ShowSingleFacets](#)
[ShowPatterns](#)
[ShowEnumerations](#)
[ShowAttributes](#)
[ShowIdentityConstraints](#)
[ShowAnnotations](#)
[ShowSourceCode](#)

Description

This object combines all options for schema document generation as they are available through user interface dialog boxes in Authentic Desktop. The document generation options are initialized with the values used during the last generation of schema documentation. However, before using the object you have to set the [SetOutputFile](#) property to a valid file path. Use [OptionsDialogAction](#), [OutputFileDialogAction](#) and [ShowProgressBar](#) to specify the level of user interaction desired. You can use [IncludeAll](#) and [AllDetails](#) to set whole option groups at once or the individual properties to operate on a finer granularity.

13.3.2.23.1 AllDetails

Method: AllDetails (i_bDetailsOn as Boolean)

Description

Use this method to turn all details options on or off.

Errors

2900 The object is no longer valid.

13.3.2.23.2 Application

Property: Application as [Application](#) (read-only)

Description

Access the Authentic Desktop application object.

Errors

2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

13.3.2.23.3 CreateDiagramsFolder

Property: CreateDiagramsFolder as Boolean

Description

Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The default for the first run is false.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.4 DiagramFormat

Property: DiagramFormat as [SPYImageKind](#)

Description

This property specifies the generated diagram image type. This property is not available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is PNG.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.5 EmbedCSSInHTML

Property: EmbedCSSInHTML as Boolean

Description

Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.6 EmbedDiagrams

Property: EmbedDiagrams as Boolean

Description

Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.7 GenerateRelativeLinks

Property: GenerateRelativeLinks as Boolean

Description

Set this property to true, to create relative paths to local files. This property is not available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is false.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.8 IncludeAll

Method: IncludeAll (i_bInclude as Boolean)

Description

Use this method to mark or unmark all include options.

Errors

- 2900 The object is no longer valid.

13.3.2.23.9 IncludeAttributeGroups

Property: IncludeAttributeGroups as Boolean

Description

Set this property to true, to include attribute groups in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.10 IncludeComplexTypes

Property: IncludeComplexTypes as Boolean

Description

Set this property to true, to include complex types in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.11 IncludeGlobalAttributes

Property: IncludeGlobalAttributes as Boolean

Description

Set this property to true, to include global attributes in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.12 IncludeGlobalElements

Property: IncludeGlobalElements as Boolean

Description

Set this property to true, to include global elements in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.13 IncludeGroups

Property: IncludeGroups as Boolean

Description

Set this property to true, to include groups in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.14 IncludeIndex

Property: IncludeIndex as Boolean

Description

Set this property to true, to include an index in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.15 IncludeLocalAttributes

Property: IncludeLocalAttributes as Boolean

Description

Set this property to true, to include local attributes in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.16 IncludeLocalElements

Property: IncludeLocalElements as Boolean

Description

Set this property to true, to include local elements in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.17 IncludeRedefines

Property: IncludeRedefines as Boolean

Description

Set this property to true, to include redefines in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.18 IncludeReferencedSchemas

Property: IncludeReferencedSchemas as Boolean

Description

Set this property to true, to include referenced schemas in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.19 IncludeSimpleTypes

Property: IncludeSimpleTypes as Boolean

Description

Set this property to true, to include simple types in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.20 MultipleOutputFiles

Property: MultipleOutputFiles as Boolean

Description

Set this property to true, to split the documentation files. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is false.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

13.3.2.23.21 OptionsDialogAction

Property: OptionsDialogAction as [SPYDialogAction](#)

Description

To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

13.3.2.23.22 OutputFile

Property: OutputFile as String

Description

Full path and name of the file that will contain the generated documentation. In case of HTML output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to [Document.GenerateSchemaDocumentation](#).

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.23 OutputFileDialogAction

Property: OutputFileDialogAction as [SPYDialogAction](#)

Description

To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in [OutputFile](#) should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

13.3.2.23.24 OutputFormat

Property: OutputFormat as [SPYSchemaDocumentationFormat](#)

Description

Defines the kind of documentation that will be generated: HTML (value=0), MS-Word (value=1), or RTF (value=2). The property gets initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is HTML.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

13.3.2.23.25 Parent

Property: Parent as [Dialogs](#) (read-only)

Description

Access the parent of the object.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.26 ShowAnnotations

Property: ShowAnnotations as Boolean

Description

Set this property to true, to show the annotations to a type definition in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.27 ShowAttributes

Property: ShowAttributes as Boolean

Description

Set this property to true, to show the type definitions attributes in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.28 ShowChildren

Property: ShowChildren as Boolean

Description

Set this property to true, to show the children of a type definition as links in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.29 ShowDiagram

Property: ShowDiagram as Boolean

Description

Set this property to true, to show type definitions as diagrams in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.30 ShowEnumerations

Property: ShowEnumerations as Boolean

Description

Set this property to true, to show the enumerations contained in a type definition in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.31 ShowIdentityConstraints

Property: ShowIdentityConstraints as Boolean

Description

Set this property to true, to show a type definitions identity constraints in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.32 ShowNamespace

Property: ShowNamespace as Boolean

Description

Set this property to true, to show the namespace of type definitions in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.33 ShowPatterns

Property: ShowPatterns as Boolean

Description

Set this property to true, to show the patterns of a type definition in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.34 ShowProgressBar

Property: ShowProgressBar as Boolean

Description

Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.35 ShowProperties

Property: ShowProperties as Boolean

Description

Set this property to true, to show the type definition properties in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.36 ShowResult

Property: ShowResult as Boolean

Description

Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in Authentic Desktop. To show Word documentation, MS-Word will be started. The property gets initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.37 ShowSingleFacets

Property: ShowSingleFacets as Boolean

Description

Set this property to true, to show the facets of a type definition in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.38 ShowSourceCode

Property: ShowSourceCode as Boolean

Description

Set this property to true, to show the XML source code for type definitions in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.39 ShowType

Property: ShowType as Boolean

Description

Set this property to true, to show the type of type definitions in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.40 ShowUsedBy

Property: ShowUsedBy as Boolean

Description

Set this property to true, to show the used-by relation for type definitions in the schema documentation. The property is initialized with the value used during the last call to [Document.GenerateSchemaDocumentation](#). The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.41 SPSFile

Property: SPSFile as String

Description

Full path and name of the SPS file that will be used to generate the documentation.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.23.42 UseFixedDesign

Property: UseFixedDesign as Boolean

Description

Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

13.3.2.24 SpyProject

Methods

[CloseProject](#)

[SaveProject](#)

[SaveProjectAs](#)

Properties

[RootItems](#)

[ProjectFile](#)

Description

SpyProject Class

13.3.2.24.1 CloseProject

Declaration: CloseProject(*bDiscardChanges* as Boolean, *bCloseFiles* as Boolean, *bDialog* as Boolean)

Parameters

bDiscardChanges

Set *bDiscardChanges* to FALSE if you want to save the changes of the open project files and the project.

bCloseFiles

Set *bCloseFiles* to TRUE to close all open project files.

bDialog

Show dialogs for user input.

Description

CloseProject closes the current project.

13.3.2.24.2 ProjectFile

Declaration: ProjectFile as String

Description

Path and filename of the project.

13.3.2.24.3 RootItems

Declaration: RootItems as [SpyProjectItems](#)

Description

Root level of collection of project items.

13.3.2.24.4 SaveProject

Declaration: SaveProject

Description

SaveProject saves the current project.

13.3.2.24.5 SaveProjectAs

Declaration: SaveProjectAs (*strPath* as String, *bDialog* as Boolean)

Parameters

strPath

Full path with file name of new project file.

bDialog

If bDialog is TRUE, a file-dialog will be displayed.

Description

SaveProjectAs stores the project data into a new location.

13.3.2.25 SpyProjectItem

Methods

[Open](#)

Properties

[ChildItems](#)

[ParentItem](#)

[FileExtensions](#)

[ItemType](#)

[Name](#)

[Path](#)

[ValidateWith](#)

[XMLForXSLTransformation](#)

[XSLForXMLTransformation](#)

[XSLTransformationFileExtension](#)

[XSLTransformationFolder](#)

Description

SpyProjectItem Class

13.3.2.25.1 ChildItems

Declaration: ChildItems as [SpyProjectItems](#)

Description

If the item is a folder, ChildItems is the collection of the folder content.

13.3.2.25.2 FileExtensions

Declaration: FileExtensions as String

Description

Used to set the file extensions if the project item is a folder.

13.3.2.25.3 ItemType

Declaration: ItemType as [SPYProjectItemTypes](#)

Description

This property is read-only.

13.3.2.25.4 Name

Declaration: Name as String

Description

Name of the project item. This property is read-only.

13.3.2.25.5 Open

Declaration: Open as [Document](#)

Return Value

The project item opened as document.

Description

Opens the project item.

13.3.2.25.6 ParentItem

Declaration: ParentItem as [SpyProjectItem](#)

Description

Parent item of the current project item. Can be NULL (Nothing) if the project item is a top-level item.

13.3.2.25.7 Path

Declaration: Path as String

Description

Path of project item. This property is read-only.

13.3.2.25.8 ValidateWith

Declaration: ValidateWith as String

Description

Used to set the schema/DTD for validation.

13.3.2.25.9 XMLForXSLTransformation

Declaration: XMLForXSLTransformation as String

Description

Used to set the XML for XSL transformation.

13.3.2.25.10 XSLForXMLTransformation

Declaration: XSLForXMLTransformation as String

Description

Used to set the XSL for XML transformation.

13.3.2.25.11 XSLTransformationFileExtension

Declaration: XSLTransformationFileExtension as String

Description

Used to set the file extension for XSL transformation output files.

13.3.2.25.12 XSLTransformationFolder

Declaration: XSLTransformationFolder as String

Description

Used to set the destination folder for XSL transformation output files.

13.3.2.26 SpyProjectItems

Methods

[AddFile](#)

[AddFolder](#)

[AddURL](#)

[RemoveItem](#)

Properties

[Count](#)

[Item](#)

Description

SpyProjectItems Class

13.3.2.26.1 AddFile

Declaration: AddFile (*strPath* as String)

Parameters

strPath

Full path with file name of new project item

Description

The method adds a new file to the collection of project items.

13.3.2.26.2 AddFolder

Declaration: AddFolder (*strName* as String)

Parameters

strName
Name of the new folder.

Description

The method AddFolder adds a folder with the name *strName* to the collection of project items.

13.3.2.26.3 AddURL

Declaration: AddURL (*strURL* as String, *nURLType* as [SPYURLTypes](#), *strUser* as String, *strPassword* as String, *bSave* as Boolean)

Description

strURL
URL to open as document.

nURLType
Type of document to open. Set to -1 for auto detection.

strUser
Name of the user if required. Can be empty.

strPassword
Password for authentication. Can be empty.

bSave
Save user and password information.

Description

The method adds an URL item to the project collection.

13.3.2.26.4 Count

Declaration: Count as long

Description

This property gets the count of project items in the collection. The property is read-only.

13.3.2.26.5 Item

Declaration: Item (*n* as long) as [SpyProjectItem](#)

Description

Retrieves the n-th element of the collection of project items. The first item has index 1.

13.3.2.26.6 RemoveItem

Declaration: RemoveItem (pItem as [SpyProjectItem](#))

Description

RemoveItem deletes the item pItem from the collection of project items.

13.3.2.27 TextImportExportSettings

Properties for import only

[ImportFile](#)

Properties for export only

[DestinationFolder](#)

[FileExtension](#)

[CommentIncluded](#)

[RemoveDelimiter](#)

[RemoveNewline](#)

Properties for import and export

[HeaderRow](#)

[FieldDelimiter](#)

[EnclosingCharacter](#)

[Encoding](#)

[EncodingByteOrder](#)

Description

TextImportExportSettings contains options common to text import and export functions.

13.3.2.27.1 CommentIncluded

Property: CommentIncluded as Boolean

Description

This property tells whether additional comments are added to the generated text file. Default is true. This property is used only when exporting to text files.

13.3.2.27.2 DestinationFolder

Property: DestinationFolder as String

Description

The property DestinationFolder sets the folder where the created files are saved during text export.

13.3.2.27.3 EnclosingCharacter

Property: EnclosingCharacter as [SPYTextEnclosing](#)

Description

This property defines the character that encloses all field values for import and export. Default is [spyNoEnclosing](#).

13.3.2.27.4 Encoding

Property: Encoding as String

Description

The property Encoding sets the character encoding for the text files for importing and exporting.

13.3.2.27.5 EncodingByteOrder

Property: EncodingByteOrder as [SPYEncodingByteOrder](#)

Description

The property EncodingByteOrder sets the byte order for Unicode characters. Default is [spyNONE](#).

13.3.2.27.6 FieldDelimiter

Property: FieldDelimiter as [SPYTextDelimiters](#)

Description

The property FieldDelimiter defines the delimiter between the fields during import and export. Default is [spyTabulator](#).

13.3.2.27.7 FileExtension

Property: FileExtension as String

Description

This property sets the file extension for files created on text export.

13.3.2.27.8 HeaderRow

Property: HeaderRow as Boolean

Description

The property HeaderRow is used during import and export. Set HeaderRow true on import, if the first line of the text file contains the names of the columns. Set HeaderRow true on export, if the first line in the created text files should contain the name of the columns. Default value is true.

13.3.2.27.9 ImportFile

Property: ImportFile as String

Description

This property is used to set the text file for import. The string has to be a full qualified path.

13.3.2.27.10 RemoveDelimiter

Property: RemoveDelimiter as Boolean

Description

The property RemoveDelimiter defines whether characters in the text that are equal to the delimiter character are removed. Default is false. This property is used only when exporting to text files.

13.3.2.27.11 RemoveNewline

Property: RemoveNewline as Boolean

Description

The property RemoveNewline defines whether newline characters in the text are removed. Default is false. This property is used only when exporting to text files.

13.3.2.28 TextView

Properties and Methods

[Application](#)
[Parent](#)

[LineFromPosition](#)
[PositionFromLine](#)
[LineLength](#)
[SetText](#)
[GetRangeText](#)
[ReplaceText](#)
[MoveCaret](#)
[GoToLineChar](#)
[SelectText](#)
[SelectionStart](#)
[SelectionEnd](#)
[Text](#)
[LineCount](#)
[Length](#)

Description

13.3.2.28.1 Events

13.3.2.28.1.1 *OnBeforeShowSuggestions*

Event: OnBeforeShowSuggestions() as Boolean

Description

This event gets fired before a suggestion window is shown. The [Document](#) property [Suggestions](#) contains a string array that is recommended to the user. It is possible to modify the displayed recommendations during this event. Before doing so you have to assign an empty array to the [Suggestions](#) property. The best location for this is the [OnDocumentOpened](#) event. To prevent the suggestion window to show up return false and true to continue its display.

Examples

Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:

```
Function On_BeforeShowSuggestions()  
End Function
```

XMLSpy scripting environment - JScript:

```
function On_BeforeShowSuggestions()  
{  
}  
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (33, ...)// nEventId = 33
```

13.3.2.28.1.2 *OnChar*

Event: OnChar(nChar as Long, bExistSuggestion as Boolean) as Boolean

Description

This event gets fired on each key stroke. The parameter nChar is the key that was pressed and bExistSuggestions tells whether a Authentic Desktop generated suggestions window is displayed after this key. The [Document](#) property [Suggestions](#) contains a string array that is recommended to the user. It is possible to modify the displayed recommendations during this event. Before doing so you have to assign an empty array to the [Suggestions](#) property. The best location for this is the [OnDocumentOpened](#) event. To prevent the suggestion window to show up return false and true to continue its display. It is also possible to create a new suggestions window when none is provided by Authentic Desktop. Set the [Document](#) property [Suggestions](#) to a string array with your recommendations and return true. This event is fired before the [OnBeforeShowSuggestions](#) event. If you prevent to show the suggestion window by returning false then [OnBeforeShowSuggestions](#) is not fired.

Examples

Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:

```
Function On_Char(nChar, bExistSuggestions)
```

End Function

XMLSpy scripting environment - JScript:

```
function On_Char(nChar, bExistSuggestions)
{
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (35, ...)// nEventId = 35
```

13.3.2.28.2 Application

Property: Application as [Application](#) (read-only)

Description

Access the Authentic Desktop application object.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.28.3 GetRangeText

Method: GetRangeText(nStart as Long, nEnd as Long) as String

Description

Returns the text in the specified range.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.28.4 GoToLineChar

Method: GoToLineChar(nLine as Long, nChar as Long)

Description

Moves the caret to the specified line and character position.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.28.5 Length

Property: Length as Long

Description

Returns the character count of the document.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.28.6 LineCount

Property: LineCount as Long

Description

Returns the number of lines in the document.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.28.7 LineFromPosition

Method: LineFromPosition(nCharPos as Long) as Long

Description

Returns the line number of the character position.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.28.8 LineLength

Method: LineLength(nLine as Long) as Long

Description

Returns the length of the line.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.28.9 MoveCaret

Method: MoveCaret(nDiff as Long)

Description

Moves the caret nDiff characters.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.28.10 Parent

Property: Parent as [Document](#) (read-only)

Description

Access the parent of the object.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.28.11 PositionFromLine

Method: PositionFromLine(nLine as Long) as Long

Description

Returns the start position of the line.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.28.12 ReplaceText

Method: ReplaceText(nPosFrom as Long, nPosTill as Long, sText as String)

Description

Replaces the text in the specified range.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.28.13 SelectionEnd

Property: SelectionEnd as Long

Description

Returns/sets the text selection end position.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.28.14 SelectionStart

Property: SelectionStart as Long

Description

Returns/sets the text selection start position.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.28.15 SelectText

Method: SelectText(nPosFrom as Long, nPosTill as Long)

Description

Selects the text in the specified range.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.28.16 SelText

Property: SelText as String

Description

Returns/sets the selected text.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.28.17 Text

Property: Text as String

Description

Returns/sets the document text.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29 WSDLDocumentationDlg

Properties and Methods

Standard automation properties

[Application](#)

[Parent](#)

Interaction and visibility properties

[GlobalElementsAndTypesOnly](#)

[OptionsDialogAction](#)

[OutputFile](#)

[OutputFileDialogAction](#)

[SeparateSchemaDocument](#)

[ShowProgressBar](#)

[ShowResult](#)

Document generation options and methods

[OutputFormat](#)

[UseFixedDesign](#)

[SPSFile](#)

[EmbedDiagrams](#)

[DiagramFormat](#)

[MultipleOutputFiles](#)

[EmbedCSSInHTML](#)

[CreateDiagramsFolder](#)

[IncludeAll](#)

[IncludeBinding](#)

[IncludeImportedWSDLFiles](#)

[IncludeMessages](#)

[IncludeOverview](#)

[IncludePortType](#)

[IncludeService](#)

[IncludeTypes](#)

[AllDetails](#)

[ShowBindingDiagram](#)

[ShowExtensibility](#)

[ShowMessageParts](#)

[ShowPort](#)

[ShowPortTypeDiagram](#)

[ShowPortTypeOperations](#)

[ShowServiceDiagram](#)

[ShowSourceCode](#)
[ShowTypesDiagram](#)
[ShowUsedBy](#)

Description

This object combines all options for WSDL document generation as they are available through user interface dialog boxes in Authentic Desktop. The document generation options are initialized with the values used during the last generation of WSDL documentation. However, before using the object you have to set the [OutputFile](#) property to a valid file path. Use [OptionsDialogAction](#), [OutputFileDialogAction](#) and [ShowProgressBar](#) to specify the level of user interaction desired. You can use [IncludeAll](#) and [AllDetails](#) to set whole option groups at once or the individual properties to operate on a finer granularity.

13.3.2.29.1 AllDetails

Method: AllDetails (i_bDetailsOn as Boolean)

Description

Use this method to turn all details options on or off.

Errors

4300 The object is no longer valid.

13.3.2.29.2 Application

Property: Application as [Application](#) (read-only)

Description

Access the Authentic Desktop application object.

Errors

3900 The object is no longer valid.
3901 Invalid address for the return parameter was specified.

13.3.2.29.3 CreateDiagramsFolder

Property: CreateDiagramsFolder as Boolean

Description

Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is false.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.4 DiagramFormat

Property: DiagramFormat as [SPYImageKind](#)

Description

This property specifies the generated diagram image type. This property is not available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is PNG.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.5 EmbedCSSInHTML

Property: EmbedCSSInHTML as Boolean

Description

Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.6 EmbedDiagrams

Property: EmbedDiagrams as Boolean

Description

Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.7 GlobalElementsAndTypesOnly

Property: GlobalElementsAndTypesOnly as Boolean

Description

Returns/sets a value indicating whether a full Schema documentation is done or only Global Elements and Types are documented.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.8 IncludeAll

Method: IncludeAll (i_bInclude as Boolean)

Description

Use this method to mark or unmark all include options.

Errors

- 4300 The object is no longer valid.

13.3.2.29.9 IncludeBinding

Property: IncludeBinding as Boolean

Description

Set this property to true, to include bindings in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.10 IncludeImportedWSDLFiles

Property: IncludeImportedWSDLFiles as Boolean

Description

Set this property to true, to include imported WSDL files in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.11 IncludeMessages

Property: IncludeMessages as Boolean

Description

Set this property to true, to include messages in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.12 IncludeOverview

Property: IncludeOverview as Boolean

Description

Set this property to true, to include an overview in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.13 IncludePortType

Property: IncludePortType as Boolean

Description

Set this property to true, to include port types in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.14 IncludeService

Property: IncludeService as Boolean

Description

Set this property to true, to include services in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.15 IncludeTypes

Property: IncludeTypes as Boolean

Description

Set this property to true, to include types in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.16 MultipleOutputFiles

Property: MultipleOutputFiles as Boolean

Description

Set this property to true, to split the documentation files. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is false.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

13.3.2.29.17 OptionsDialogAction

Property: OptionsDialogAction as [SPYDialogAction](#)

Description

To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

13.3.2.29.18 OutputFile

Property: OutputFile as String

Description

Full path and name of the file that will contain the generated documentation. In case of HTML output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to [Document.GenerateWSDLDocumentation](#).

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.19 OutputFileDialogAction

Property: OutputFileDialogAction as [SPYDialogAction](#)

Description

To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in [OutputFile](#) should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

13.3.2.29.20 OutputFormat

Property: OutputFormat as [SPYSchemaDocumentationFormat](#)

Description

Defines the kind of documentation that will be generated: HTML (value=0), MS-Word (value=1), or RTF (value=2). The property gets initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is HTML.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

13.3.2.29.21 Parent

Property: Parent as [Dialogs](#) (read-only)

Description

Access the parent of the object.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.22 `SeparateSchemaDocument`

Property: `SeparateSchemaDocument` as Boolean

Description

Returns/sets a value indicating whether the Schema documentation should be placed in a separate document.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.23 `ShowBindingDiagram`

Property: `ShowBindingDiagram` as Boolean

Description

Set this property to true, to show binding diagrams in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.24 `ShowExtensibility`

Property: `ShowExtensibility` as Boolean

Description

Set this property to true, to show service and binding extensibilities in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.25 ShowMessageParts

Property: ShowMessageParts as Boolean

Description

Set this property to true, to show message parts of messages in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.26 ShowPort

Property: ShowPort as Boolean

Description

Set this property to true, to show service ports in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.27 ShowPortTypeDiagram

Property: ShowPortTypeDiagram as Boolean

Description

Set this property to true, to show port type diagrams in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.28 ShowPortTypeOperations

Property: ShowPortTypeOperations as Boolean

Description

Set this property to true, to show port type operations in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.29 ShowProgressBar

Property: ShowProgressBar as Boolean

Description

Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.30 ShowResult

Property: ShowResult as Boolean

Description

Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in Authentic Desktop. To show Word documentation, MS-Word will be started. The property gets initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.31 ShowServiceDiagram

Property: ShowServiceDiagram as Boolean

Description

Set this property to true, to show service diagrams in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.32 ShowSourceCode

Property: ShowSourceCode as Boolean

Description

Set this property to true, to show source code for the includes in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.33 ShowTypesDiagram

Property: ShowTypesDiagram as Boolean

Description

Set this property to true, to show type diagrams in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.34 ShowUsedBy

Property: ShowUsedBy as Boolean

Description

Set this property to true, to show the used-by relation for types, bindings and messages definitions in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDLDocumentation](#). The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.35 UseFixedDesign

Property: UseFixedDesign as Boolean

Description

Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.29.36 SPSFile

Property: SPSFile as String

Description

Full path and name of the SPS file that will be used to generate the documentation.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.30 WSDL20DocumentationDlg

Properties and Methods

Standard automation properties

[Application](#)

[Parent](#)

Interaction and visibility properties

[GlobalElementsAndTypesOnly](#)

[OptionsDialogAction](#)

[OutputFile](#)

[OutputFileDialogAction](#)

[SeparateSchemaDocument](#)

[ShowProgressBar](#)

[ShowResult](#)

Document generation options and methods

[OutputFormat](#)

[UseFixedDesign](#)

[SPSFile](#)

[EmbedDiagrams](#)

[DiagramFormat](#)

[MultipleOutputFiles](#)

[EmbedCSSInHTML](#)

[CreateDiagramsFolder](#)

[IncludeAll](#)

[IncludeBinding](#)

[IncludeImportedWSDLFiles](#)

[IncludeInterface](#)

[IncludeOverview](#)

[IncludeService](#)

[IncludeTypes](#)

[AllDetails](#)

[ShowBindingDiagram](#)

[ShowExtensibility](#)

[ShowEndpoint](#)

[ShowFault](#)

[ShowInterfaceDiagram](#)

[ShowOperation](#)

[ShowServiceDiagram](#)

[ShowSourceCode](#)

[ShowTypesDiagram](#)

[ShowUsedBy](#)

Description

This object combines all options for WSDL document generation as they are available through user interface dialog boxes in Authentic Desktop. The document generation options are initialized with the values used during the last generation of WSDL documentation. However, before using the object you have to set the [OutputFile](#) property to a valid file path. Use [OptionsDialogAction](#), [OutputFileDialogAction](#) and [ShowProgressBar](#) to specify the level of user interaction desired. You can use [IncludeAll](#) and [AllDetails](#) to set whole option groups at once or the individual properties to operate on a finer granularity.

13.3.2.30.1 AllDetails

Method: AllDetails (i_bDetailsOn as Boolean)

Description

Use this method to turn all details options on or off.

Errors

4300 The object is no longer valid.

13.3.2.30.2 Application

Property: Application as [Application](#) (read-only)

Description

Access the Authentic Desktop application object.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.3 CreateDiagramsFolder

Property: CreateDiagramsFolder as Boolean

Description

Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is false.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.4 DiagramFormat

Property: DiagramFormat as [SPYImageKind](#)

Description

This property specifies the generated diagram image type. This property is not available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is PNG.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.5 EmbedCSSInHTML

Property: EmbedCSSInHTML as Boolean

Description

Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.6 EmbedDiagrams

Property: EmbedDiagrams as Boolean

Description

Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.7 GlobalElementsAndTypesOnly

Property: GlobalElementsAndTypesOnly as Boolean

Description

Returns/sets a value indicating whether a full Schema documentation is done or only Global Elements and Types are documented.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.30.8 IncludeAll

Method: IncludeAll (i_bInclude as Boolean)

Description

Use this method to mark or unmark all include options.

Errors

- 4300 The object is no longer valid.

13.3.2.30.9 IncludeBinding

Property: IncludeBinding as Boolean

Description

Set this property to true, to include bindings in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.10 IncludeImportedWSDLFiles

Property: IncludeImportedWSDLFiles as Boolean

Description

Set this property to true, to include imported WSDL files in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.11 IncludeInterface

Property: IncludeInterface as Boolean

Description

Set this property to true, to include interfaces in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.12 IncludeOverview

Property: IncludeOverview as Boolean

Description

Set this property to true, to include an overview in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.13 IncludeService

Property: IncludeService as Boolean

Description

Set this property to true, to include services in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.14 IncludeTypes

Property: IncludeTypes as Boolean

Description

Set this property to true, to include types in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.15 MultipleOutputFiles

Property: MultipleOutputFiles as Boolean

Description

Set this property to true, to split the documentation files. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is false.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

13.3.2.30.16 OptionsDialogAction

Property: OptionsDialogAction as [SPYDialogAction](#)

Description

To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

13.3.2.30.17 OutputFile

Property: OutputFile as String

Description

Full path and name of the file that will contain the generated documentation. In case of HTML output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to [Document.GenerateWSDL20Documentation](#).

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.18 OutputFileDialogAction

Property: OutputFileDialogAction as [SPYDialogAction](#)

Description

To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in [OutputFile](#) should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

13.3.2.30.19 OutputFormat

Property: OutputFormat as [SPYSchemaDocumentationFormat](#)

Description

Defines the kind of documentation that will be generated: HTML (value=0), MS-Word (value=1), or RTF (value=2). The property gets initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is HTML.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

13.3.2.30.20 Parent

Property: Parent as [Dialogs](#) (read-only)

Description

Access the parent of the object.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.21 SeparateSchemaDocument

Property: SeparateSchemaDocument as Boolean

Description

Returns/sets a value indicating whether the Schema documentation should be placed in a separate document.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

13.3.2.30.22 ShowBindingDiagram

Property: ShowBindingDiagram as Boolean

Description

Set this property to true, to show binding diagrams in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.23 ShowEndpoint

Property: ShowEndpoint as Boolean

Description

Set this property to true, to show service endpoints in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.24 ShowExtensibility

Property: ShowExtensibility as Boolean

Description

Set this property to true, to show service and binding extensibilities in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.25 ShowFault

Property: ShowFault as Boolean

Description

Set this property to true, to show faults in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.26 ShowInterfaceDiagram

Property: ShowInterfaceDiagram as Boolean

Description

Set this property to true, to show interface diagrams in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is true.

Errors

- 4300 The object is no longer valid.

4301 Invalid address for the return parameter was specified.

13.3.2.30.27 ShowOperation

Property: ShowOperation as Boolean

Description

Set this property to true, to show interface and binding operations in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.28 ShowProgressBar

Property: ShowProgressBar as Boolean

Description

Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.29 ShowResult

Property: ShowResult as Boolean

Description

Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in Authentic Desktop. To show Word documentation, MS-Word will be started. The property gets initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.30 ShowServiceDiagram

Property: ShowServiceDiagram as Boolean

Description

Set this property to true, to show service diagrams in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.31 ShowSourceCode

Property: ShowSourceCode as Boolean

Description

Set this property to true, to show source code for the includes in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.32 ShowTypesDiagram

Property: ShowTypesDiagram as Boolean

Description

Set this property to true, to show type diagrams in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.33 ShowUsedBy

Property: ShowUsedBy as Boolean

Description

Set this property to true, to show the used-by relation for types, bindings and messages definitions in the WSDL documentation. The property is initialized with the value used during the last call to [Document.GenerateWSDL20Documentation](#). The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.34 SPSFile

Property: SPSFile as String

Description

Full path and name of the SPS file that will be used to generate the documentation.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.30.35 UseFixedDesign

Property: UseFixedDesign as Boolean

Description

Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

13.3.2.31 XBRLDocumentationDlg

Properties and Methods

Standard automation properties

[Application](#)
[Parent](#)

Interaction and visibility properties

[OptionsDialogAction](#)
[OutputFile](#)
[OutputFileDialogAction](#)
[ShowProgressBar](#)

[ShowResult](#)

Document generation options and methods

[OutputFormat](#)

[UseFixedDesign](#)

[SPSFile](#)

[EmbedDiagrams](#)

[DiagramFormat](#)

[EmbedCSSInHTML](#)

[CreateDiagramsFolder](#)

[IncludeAll](#)

[IncludeOverview](#)

[IncludeNamespacePrefixes](#)

[IncludeGlobalElements](#)

[IncludeDefinitionLinkroles](#)

[IncludePresentationLinkroles](#)

[IncludeCalculationLinkroles](#)

[AllDetails](#)

[ShowDiagram](#)

[ShowSubstitutiongroup](#)

[ShowIemtype](#)

[ShowBalance](#)

[ShowPeriod](#)

[ShowAbstract](#)

[ShowNillable](#)

[ShowLabels](#)

[ShowReferences](#)

[ShowLinkbaseReferences](#)

[ShortQualifiedName](#)

[ShowImportedElements](#)

Description

This object combines all options for XBRL document generation as they are available through user interface dialog boxes in Authentic Desktop. The document generation options are initialized with the values used during the last generation of XBRL documentation. However, before using the object you have to set the [OutputFile](#) property to a valid file path. Use [OptionsDialogAction](#), [OutputFileDialogAction](#) and [ShowProgressBar](#) to specify the level of user interaction desired. You can use [IncludeAll](#) and [AllDetails](#) to set whole option groups at once or the individual properties to operate on a finer granularity.

13.3.2.31.1 AllDetails

Method: AllDetails (i_bDetailsOn as Boolean)

Description

Use this method to turn all details options on or off.

Errors

4400 The object is no longer valid.

13.3.2.31.2 Application

Property: Application as [Application](#) (read-only)

Description

Access the Authentic Desktop application object.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.3 CreateDiagramsFolder

Property: CreateDiagramsFolder as Boolean

Description

Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is false.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.4 DiagramFormat

Property: DiagramFormat as [SPYImageKind](#)

Description

This property specifies the generated diagram image type. This property is not available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is PNG.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.5 EmbedCSSInHTML

Property: EmbedCSSInHTML as Boolean

Description

Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.6 EmbedDiagrams

Property: EmbedDiagrams as Boolean

Description

Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.7 IncludeAll

Method: IncludeAll (i_bInclude as Boolean)

Description

Use this method to mark or unmark all include options.

Errors

- 4400 The object is no longer valid.

13.3.2.31.8 IncludeCalculationLinkroles

Property: IncludeCalculationLinkroles as Boolean

Description

Set this property to true, to include calculation linkroles in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.9 IncludeDefinitionLinkroles

Property: IncludeDefinitionLinkroles as Boolean

Description

Set this property to true, to include definition linkroles in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.10 IncludeGlobalElements

Property: IncludeGlobalElements as Boolean

Description

Set this property to true, to include global elements in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.11 IncludeNamespacePrefixes

Property: IncludeNamespacePrefixes as Boolean

Description

Set this property to true, to include namespace prefixes in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.12 IncludeOverview

Property: IncludeOverview as Boolean

Description

Set this property to true, to include an overview in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.13 IncludePresentationLinkroles

Property: IncludePresentationLinkroles as Boolean

Description

Set this property to true, to include presentation linkroles in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.14 OptionsDialogAction

Property: OptionsDialogAction as [SPYDialogAction](#)

Description

To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

13.3.2.31.15 OutputFile

Property: OutputFile as String

Description

Full path and name of the file that will contain the generated documentation. In case of HTML output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to [Document.GenerateXBRLDocumentation](#).

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.16 OutputFileDialogAction

Property: OutputFileDialogAction as [SPYDialogAction](#)

Description

To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in [OutputFile](#) should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

13.3.2.31.17 OutputFormat

Property: OutputFormat as [SPYSchemaDocumentationFormat](#)

Description

Defines the kind of documentation that will be generated: HTML (value=0), MS-Word (value=1), or RTF (value=2). The property gets initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is HTML.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid value has been used to set the property.
Invalid address for the return parameter was specified.

13.3.2.31.18 Parent

Property: Parent as [Dialogs](#) (read-only)

Description

Access the parent of the object.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.19 ShortQualifiedName

Property: ShortQualifiedName as Boolean

Description

Set this property to true, to use short qualified names in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.20 ShowAbstract

Property: ShowAbstract as Boolean

Description

Set this property to true, to show abstracts in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.21 ShowBalance

Property: ShowBalance as Boolean

Description

Set this property to true, to show balances in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.22 ShowDiagram

Property: ShowDiagram as Boolean

Description

Set this property to true, to show diagrams in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.23 ShowImportedElements

Property: ShowImportedElements as Boolean

Description

Set this property to true, to show imported elements in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.24 ShowItemtype

Property: ShowItemtype as Boolean

Description

Set this property to true, to show item types in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.25 ShowLabels

Property: ShowLabels as Boolean

Description

Set this property to true, to show labels in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.26 ShowLinkbaseReferences

Property: ShowLinkbaseReferences as Boolean

Description

Set this property to true, to show linkbase references in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.27 ShowNillable

Property: ShowNillable as Boolean

Description

Set this property to true, to show nillable properties in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.28 ShowPeriod

Property: ShowPeriod as Boolean

Description

Set this property to true, to show periods in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.29 ShowProgressBar

Property: ShowProgressBar as Boolean

Description

Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.30 ShowReferences

Property: ShowReferences as Boolean

Description

Set this property to true, to show references in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.31 ShowResult

Property: ShowResult as Boolean

Description

Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in Authentic Desktop. To show Word documentation, MS-Word will be started. The property gets initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.32 ShowSubstitutiongroup

Property: ShowSubstitutiongroup as Boolean

Description

Set this property to true, to show substitution groups in the XBRL documentation. The property is initialized with the value used during the last call to [Document.GenerateXBRLDocumentation](#). The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.33 SPSFile

Property: SPSFile as String

Description

Full path and name of the SPS file that will be used to generate the documentation.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.31.34 UseFixedDesign

Property: UseFixedDesign as Boolean

Description

Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

13.3.2.32 XMLData

Properties

[Kind](#)

[Name](#)

[TextValue](#)

[HasChildren](#)

[MayHaveChildren](#)

[Parent](#)

Methods

[GetFirstChild](#)

[GetNextChild](#)
[GetCurrentChild](#)

[InsertChild](#)
[InsertChildAfter](#)
[InsertChildBefore](#)
[AppendChild](#)

[EraseAllChildren](#)
[EraseChild](#)
[EraseCurrentChild](#)

[IsSameNode](#)

[CountChildren](#)
[CountChildrenKind](#)

[GetChild](#)
[GetChildAttribute](#)
[GetChildElement](#)
[GetChildKind](#)
[GetNamespacePrefixForURI](#)

[HasChildrenKind](#)
[SetTextValueXMLEncoded](#)

Description

The XMLData interface provides direct XML-level access to a document. You can read and directly modify the XML representation of the document. However, please, note the following restrictions:

- The XMLData representation is only valid when the document is shown in grid view or authentic view.
- When in authentic view, additional XMLData elements are automatically inserted as parents of each visible document element. Typically this is an XMLData of kind spyXMLDataElement with the [Name](#) property set to 'Text'.
- When you use the XMLData interface while in a different view mode you will not receive errors, but changes are not reflected to the view and might get lost during the next view switch.

Note also:

- Setting a new text value for an XML element is possible if the element does not have non-text children. A text value can be set even if the element has attributes.
- When setting a new text value for an XML element which has more than one text child, the latter will be deleted and replaced by one new text child.
- When reading the text value of an XML element which has more than one text child, only the value of the first text child will be returned.

13.3.2.32.1 AppendChild

Declaration: AppendChild (*pNewData* as [XMLData](#))

Description

AppendChild appends pNewData as last child to the XMLData object.

Errors

- 1500 The XMLData object is no longer valid.
- 1505 Invalid XMLData kind was specified.
- 1506 Invalid address for the return parameter was specified.
- 1507 Element cannot have Children
- 1512 Cyclic insertion - new data element is already part of document
- 1514 Invalid XMLData kind was specified for this position.
- 1900 Document must not be modified

Example

```
Dim objCurrentParent As XMLData
Dim objNewChild As XMLData

Set objNewChild = objSpy.ActiveDocument.CreateChild(spyXMLDataElement)
Set objCurrentParent = objSpy.ActiveDocument.RootElement

objCurrentParent.AppendChild objNewChild

Set objNewChild = Nothing
```

13.3.2.32.2 CountChildren

Declaration: CountChildren as long

Description

CountChildren gets the number of children.

Available with TypeLibrary version 1.5

Errors

- 1500 The XMLData object is no longer valid.

13.3.2.32.3 CountChildrenKind

Declaration: CountChildrenKind (*nKind* as [SPYXMLDataKind](#)) as long

Description

CountChildrenKind gets the number of children of the specific kind.

Available with TypeLibrary version 1.5

Errors

- 1500 The XMLData object is no longer valid.

13.3.2.32.4 EraseAllChildren

Declaration: EraseAllChildren

Description

EraseAllChildren deletes all associated children of the XMLData object.

Errors

- 1500 The XMLData object is no longer valid.
- 1900 Document must not be modified

Example

The sample erases all elements of the active document.

```
Dim objCurrentParent As XMLData

Set objCurrentParent = objSpy.ActiveDocument.RootElement
objCurrentParent.EraseAllChildren
```

13.3.2.32.5 EraseChild

Method: EraseChild (Child as [XMLData](#))

Description

Deletes the given child node.

Errors

- 1500 Invalid object.
- 1506 Invalid input xml
- 1510 Invalid parameter.

13.3.2.32.6 EraseCurrentChild

Declaration: EraseCurrentChild

Description

EraseCurrentChild deletes the current XMLData child object. Before you call EraseCurrentChild you must initialize an internal iterator with [XMLData.GetFirstChild](#). After deleting the current child, EraseCurrentChild increments the internal iterator of the XMLData element. No error is returned when the last child gets erased and the iterator is moved past the end of the child list. The next call to EraseCurrentChild however, will return error 1503.

Errors

- 1500 The XMLData object is no longer valid.
- 1503 No iterator is initialized for this XMLData object, or the iterator points past the last child.
- 1900 Document must not be modified

Examples

```
// -----  
// XMLSpy scripting environment - JScript  
// erase all children of XMLData  
// -----  
// let's get an XMLData element, we assume that the  
// cursor selects the parent of a list in grid view  
var objList = Application.ActiveDocument.GridView.CurrentFocus;  
  
// the following line would be shorter, of course  
// objList.EraseAllChildren ();  
  
// but we want to demonstrate the usage of EraseCurrentChild  
if ((objList != null) && (objList.HasChildren))  
{  
    try  
    {  
        objEle = objList.GetFirstChild(-1);  
        while (objEle != null)  
            objList.EraseCurrentChild();  
            // no need to call GetNextChild  
    }  
    catch (err)  
    {  
        // 1503 - we reached end of child list  
        { if ((err.number & 0xffff) != 1503) throw (err); }  
    }  
}
```

13.3.2.32.7 GetChild

Declaration: GetChild (*position* as long) as [XMLData](#)

Return Value

Returns an XML element as XMLData object.

Description

GetChild() returns a reference to the child at the given index (zero-based).

Available with TypeLibrary version 1.5

Errors

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

13.3.2.32.8 GetChildAttribute

Method: GetChildAttribute (*strName* as string) child as XMLData object (NULL on error)

Description

Retrieves the attribute having the given name.

Errors

- 1500 Invalid object.
- 1510 Invalid parameter.

13.3.2.32.9 GetChildElement

Method: GetChildElement (strName as string, nIndex as long) child as XMLData object (NULL on error)

Description

Retrieves the Nth child element with the given name.

Errors

- 1500 Invalid object.
- 1510 Invalid parameter.

13.3.2.32.10 GetChildKind

Declaration: GetChildKind (*position* as long, *nKind* as [SPYXMLDataKind](#)) as [XMLData](#)

Return Value

Returns an XML element as XMLData object.

Description

GetChildKind() returns a reference to a child of this kind at the given index (zero-based). The position parameter is relative to the number of children of the specified kind and not to all children of the object.

Available with TypeLibrary version 1.5

Errors

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

13.3.2.32.11 GetCurrentChild

Declaration: GetCurrentChild as [XMLData](#)

Return Value

Returns an XML element as XMLData object.

Description

GetCurrentChild gets the current child. Before you call GetCurrentChild you must initialize an internal iterator with [XMLData.GetFirstChild](#).

Errors

- 1500 The XMLData object is no longer valid.
- 1503 No iterator is initialized for this XMLData object.
- 1510 Invalid address for the return parameter was specified.

13.3.2.32.12 GetFirstChild

Declaration: GetFirstChild (*nKind* as [SPYXMLDataKind](#)) as [XMLData](#)

Return Value

Returns an XML element as XMLData object.

Description

GetFirstChild initializes a new iterator and returns the first child. Set *nKind* = -1 to get an iterator for all kinds of children.

REMARK: The iterator is stored inside the XMLData object and gets destroyed when the XMLData object gets destroyed. Be sure to keep a reference to this object as long as you want to use [GetCurrentChild](#), [GetNextChild](#) or [EraseCurrentChild](#).

Errors

- 1500 The XMLData object is no longer valid.
- 1501 Invalid XMLData kind was specified.
- 1504 Element has no children of specified kind.
- 1510 Invalid address for the return parameter was specified.

Example

See the example at [XMLData.GetNextChild](#).

13.3.2.32.13 GetNamespacePrefixForURI

Method: GetNamespacePrefixForURI (*strURI* as string) *strNS* as string

Description

Returns the namespace prefix of the supplied URI.

Errors

- 1500 Invalid object.
- 1510 Invalid parameter.

13.3.2.32.14 GetNextChild

Declaration: GetNextChild as [XMLData](#)

Return Value

Returns an XML element as XMLData object.

Description

GetNextChild steps to the next child of this element. Before you call GetNextChild you must initialize an internal iterator with [XMLData.GetFirstChild](#).

Check for the last child of the element as shown in the sample below.

Errors

- 1500 The XMLData object is no longer valid.
- 1503 No iterator is initialized for this XMLData object.
- 1510 Invalid address for the return parameter was specified.

Examples

```
'-----  
' VBA code snippet - iterate XMLData children  
'-----  
On Error Resume Next  
Set objParent = objSpy.ActiveDocument.RootElement  
  
'get elements of all kinds  
Set objCurrentChild = objParent.GetFirstChild(-1)  
  
Do  
    'do something useful with the child  
  
    'step to next child  
    Set objCurrentChild = objParent.GetNextChild  
Loop Until (Err.Number - vbObjectError = 1503)  
  
// -----  
// XMLSpy scripting environment - JScript  
// iterate through children of XMLData  
// -----  
try  
{  
    var objXMLData = ... // initialize somehow  
    var objChild = objXMLData.GetFirstChild(-1);  
  
    while (true)  
    {  
        // do something usefull with objChild  
  
        objChild = objXMLData.GetNextChild();  
    }  
}  
catch (err)
```

```
{
    if ((err.number & 0xffff) == 1504)
        ; // element has no children
    else if ((err.number & 0xffff) == 1503)
        ; // last child reached
    else
        throw (err);
}
```

13.3.2.32.15 GetTextValueXMLDecoded

Method: GetTextValueXMLDecoded () as string

Description

Gets the decoded text value of the XML.

Errors

- 1500 Invalid object.
- 1510 Invalid parameter.

13.3.2.32.16 HasChildren

Declaration: HasChildren as Boolean

Description

The property is true if the object is the parent of other XMLData objects. This property is read-only.

Errors

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

13.3.2.32.17 HasChildrenKind

Declaration: HasChildrenKind (*nKind* as [SPYXMLDataKind](#)) as Boolean

Description

The method returns true if the object is the parent of other XMLData objects of the specific kind.

Available with TypeLibrary version 1.5

Errors

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

13.3.2.32.18 InsertChild

Declaration: InsertChild (*pNewData* as [XMLData](#))

Description

InsertChild inserts the new child before the current child (see also [XMLData.GetFirstChild](#), [XMLData.GetNextChild](#) to set the current child).

Errors

- 1500 The XMLData object is no longer valid.
- 1503 No iterator is initialized for this XMLData object.
- 1505 Invalid XMLData kind was specified.
- 1506 Invalid address for the return parameter was specified.
- 1507 Element cannot have Children
- 1512 Cyclic insertion - new data element is already part of document
- 1514 Invalid XMLData kind was specified for this position.
- 1900 Document must not be modified

13.3.2.32.19 InsertChildAfter

Method: InsertChildAfter (Node as XMLData, NewData as XMLData)

Description

Inserts a new XML node (supplied with the second parameter) after the specified node (first parameter).

Errors

- 1500 Invalid object.
- 1506 Invalid input xml
- 1507 No children allowed
- 1510 Invalid parameter.
- 1512 Child is already added
- 1514 Invalid kind at position

13.3.2.32.20 InsertChildBefore

Method: InsertChildBefore (Node as XMLData, NewData as XMLData)

Description

Inserts a new XML node (supplied with the second parameter) before the specified node (first parameter).

Errors

- 1500 Invalid object.
- 1506 Invalid input xml
- 1507 No children allowed
- 1510 Invalid parameter.
- 1512 Child is already added

1514 Invalid kind at position

13.3.2.32.21 IsSameNode

Declaration: IsSameNode (*pNodeToCompare* as [XMLData](#)) as Boolean

Description

Returns true if *pNodeToCompare* references the same node as the object itself.

Errors

- 1500 The XMLData object is no longer valid.
- 1506 Invalid address for the return parameter was specified.

13.3.2.32.22 Kind

Declaration: Kind as [SPYXMLDataKind](#)

Description

Kind of this XMLData object. This property is read-only.

Errors

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

13.3.2.32.23 MayHaveChildren

Declaration: MayHaveChildren as Boolean

Description

Indicates whether it is allowed to add children to this XMLData object. This property is read-only.

Errors

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

13.3.2.32.24 Name

Declaration: Name as String

Description

Used to modify and to get the name of the XMLData object.

Errors

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

13.3.2.32.25 Parent

Declaration: Parent as [XMLData](#)

Return value

Parent as XMLData object. Nothing (or NULL) if there is no parent element.

Description

Parent of this element. This property is read-only.

Errors

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

13.3.2.32.26 SetTextValueXMLEncoded

Method: SetTextValueXMLEncoded (*strVal* as [String](#))

Description

Sets the encoded text value of the XML.

Errors

- 1500 Invalid object.
- 1513 Modification not allowed.

13.3.2.32.27 TextValue

Declaration: TextValue as String

Description

Used to modify and to get the text value of this XMLData object.

Errors

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

13.3.3 Enumerations

This is a list of all enumerations used by the Authentic Desktop API. If your scripting environment does not support enumerations use the number-values instead.

13.3.3.1 ENUMApplicationStatus

Description

Enumeration to specify the current Application status.

Possible values:

eApplicationRunning	= 0
eApplicationAfterLicenseCheck	= 1
eApplicationBeforeLicenseCheck	= 2
eApplicationConcurrentLicenseCheckFailed	= 3
eApplicationProcessingCommandLine	= 4

13.3.3.2 SPYAttributeTypeDefinition

Description

Attribute type definition that can be selected for generation of Sample XML.

This type is used with the method [GenerateDTDOrSchema](#) and [GenerateDTDOrSchemaEx](#).

Possible values:

spyMergedGlobal	= 0
spyDistinctGlobal	= 1
spyLocal	= 2

13.3.3.3 SPYAuthenticActions

Description

Actions that can be performed on [AuthenticRange](#) objects.

Possible values:

spyAuthenticInsertAt	= 0
spyAuthenticApply	= 1
spyAuthenticClearSurr	= 2
spyAuthenticAppend	= 3
spyAuthenticInsertBefore	= 4
spyAuthenticRemove	= 5

13.3.3.4 SPYAuthenticDocumentPosition

Description

Relative and absolute positions used for navigating with [AuthenticRange](#) objects.

Possible values:

spyAuthenticDocumentBegin	= 0
spyAuthenticDocumentEnd	= 1
spyAuthenticRangeBegin	= 2
spyAuthenticRangeEnd	= 3

13.3.3.5 SPYAuthenticElementActions

Description

Actions that can be used with the obsolete object `GetAllowedElements` (superseded by [AuthenticRange.CanPerformActionWith](#)).

Possible values:

k_ActionInsertAt	= 0
k_ActionApply	= 1
k_ActionClearSurr	= 2
k_ActionAppend	= 3
k_ActionInsertBefore	= 4
k_ActionRemove	= 5

13.3.3.6 SPYAuthenticElementKind

Description

Enumeration of the different kinds of elements used for navigation and selection within the [AuthenticRange](#) and [AuthenticView](#) objects.

Possible values:

spyAuthenticChar	= 0
spyAuthenticWord	= 1
spyAuthenticLine	= 3
spyAuthenticParagraph	= 4
spyAuthenticTag	= 6
spyAuthenticDocument	= 8
spyAuthenticTable	= 9
spyAuthenticTableRow	= 10
spyAuthenticTableColumn	= 11

13.3.3.7 SPYAuthenticMarkupVisibility

Description

Enumeration values to customize the visibility of markup with [MarkupVisibility](#).

Possible values:

spyAuthenticMarkupHidden	= 0
spyAuthenticMarkupSmall	= 1
spyAuthenticMarkupLarge	= 2
spyAuthenticMarkupMixed	= 3

13.3.3.8 SPYAuthenticToolBarButtonState

Description

Authentic toolbar button states are given by the following enumeration:

Possible values:

authenticToolBarButtonDefault	= 0
authenticToolBarButtonEnabled	= 1
authenticToolBarButtonDisabled	= 2

13.3.3.9 SPYDatabaseKind

Description

Values to select different kinds of databases for import. See [DatabaseConnection.DatabaseKind](#) for its use.

Possible values:

spyDB_Access	= 0
spyDB_SQLServer	= 1
spyDB_Oracle	= 2
spyDB_Sybase	= 3
spyDB_MySQL	= 4
spyDB_DB2	= 5
spyDB_Other	= 6
spyDB_Unspecified	= 7
spyDB_PostgreSQL	= 8
spyDB_iSeries	= 9

13.3.3.10 SPYDialogAction

Description

Values to simulate different interactions on dialogs. See [Dialogs](#) for all dialogs available.

Possible values:

spyDialogOK	= 0	// simulate click on OK button
spyDialogCancel	= 1	// simulate click on Cancel button
spyDialogUserInput	= 2	// show dialog and allow user interaction

13.3.3.11 SPYDOMType

Description

Enumeration values to parameterize generation of C++ code from schema definitions.

Possible values:

spyDOMType_msxml4	= 0	Obsolete
spyDOMType_xerces	= 1	
spyDOMType_xerces3	= 2	
spyDOMType_msxml6	= 3	

spyDOMType_xerces indicates Xerces 2.x usage; spyDOMType_xerces3 indicates Xerces 3.x usage.

13.3.3.12 SPYDTDSchemaFormat

Description

Enumeration to identify the different schema formats.

Possible values:

spyDTD	= 0
spyW3C	= 1

13.3.3.13 SPYEncodingByteOrder

Description

Enumeration values to specify encoding byte ordering for text import and export.

Possible values:

spyNONE	= 0
spyLITTLE_ENDIAN	= 1
spyBIG_ENDIAN	= 2

13.3.3.14 SPYExportNamespace

Description

Enumeration type to configure handling of namespace identifiers during export.

Possible values:

```
spyNoNamespace          = 0
spyReplaceColonWithUnderscore = 1
```

13.3.3.15 SPYFindInFilesSearchLocation

Description

The different locations where a search can be performed. This type is used with the [FindInFilesDlg](#) dialog.

Possible values:

```
spyFindInFiles_Documents    = 0
spyFindInFiles_Project      = 1
spyFindInFiles_Folder       = 2
```

13.3.3.16 SPYFrequentElements

Description

Enumeration value to parameterize schema generation.

Possible values:

```
spyGlobalElements          = 0
spyGlobalComplexType       = 1
```

13.3.3.17 SPYImageKind

Description

Enumeration values to parameterize image type of the generated documentation. These values are used in [SchemaDocumentationDialog.DiagramFormat](#).

Possible values:

```
spyImageType_PNG           = 0
spyImageType_EMF           = 1
```

13.3.3.18 SPYImportColumnsType

Description

Enumeration to specify different Import columns types.

Possible values:

```
spyImportColumns_Element   = 0
spyImportColumns_Attribute = 1
```

13.3.3.19 SPYKeyEvent

Description

Enumeration type to identify the different key events. These events correspond with the equally named windows messages.

Possible values:

spyKeyDown	= 0
spyKeyUp	= 1
spyKeyPressed	= 2

13.3.3.20 SPYKeyStatus

Description

Enumeration type to identify the key status.

Possible values:

spyLeftShiftKeyMask	= 1
spyRightShiftKeyMask	= 2
spyLeftCtrlKeyMask	= 4
spyRightCtrlKeyMask	= 8
spyLeftAltKeyMask	= 16
spyRightAltKeyMask	= 32

13.3.3.21 SPYLibType

Description

Enumeration values to parameterize generation of C++ code from schema definitions.

Possible values:

spyLibType_static	= 0
spyLibType_dll	= 1

13.3.3.22 SPYLoading

Description

Enumeration values to define loading behaviour of URL files.

Possible values:

spyUseCacheProxy	= 0
spyReload	= 1

13.3.3.23 SPYMouseEvent

Description

Enumeration type that defines the mouse status during a mouse event. Use the enumeration values as bitmasks rather than directly comparing with them.

Examples

```
' to check for ctrl-leftbutton-down in VB
If (i_eMouseEvent = (XMLSpyLib.spyLeftButtonDownMask Or XMLSpyLib.spyCtrlKeyDownMask)) Then
    ' react on ctrl-leftbutton-down
End If

' to check for double-click with any button in VBScript
If (((i_eMouseEvent And spyDoubleClickMask) <> 0) Then
    ' react on double-click
End If
```

Possible values:

spyNoButtonMask	= 0	
spyMouseMoveMask	= 1	
spyLeftButtonMask	= 2	
spyMiddleButtonMask	= 4	
spyRightButtonMask	= 8	
spyButtonUpMask	= 16	
spyButtonDownMask	= 32	
spyDoubleClickMask	= 64	
spyShiftKeyDownMask	= 128	
spyCtrlKeyDownMask	= 256	
spyLeftButtonDownMask	= 34	// spyLeftButtonMask spyButtonDownMask
spyMiddleButtonDownMask	= 36	// spyMiddleButtonMask spyButtonDownMask
spyRightButtonDownMask	= 40	// spyRightButtonMask spyButtonDownMask
spyLeftButtonUpMask	= 18	// spyLeftButtonMask spyButtonUpMask
spyMiddleButtonUpMask	= 20	// spyMiddleButtonMask spyButtonUpMask
spyRightButtonUpMask	= 24	// spyRightButtonMask spyButtonUpMask
spyLeftDoubleClickMask	= 66	// spyRightButtonMask spyButtonUpMask
spyMiddleDoubleClickMask	= 68	// spyMiddleButtonMask spyDoubleClickMask
spyRightDoubleClickMask	= 72	// spyRightButtonMask spyDoubleClickMask

13.3.3.24 SPYNumberDateTimeFormat

Description

Enumeration value to configure database connections.

Possible values:

spySystemLocale	= 0
spySchemaCompatible	= 1

13.3.3.25 SPYProgrammingLanguage

Description

Enumeration values to select the programming language for code generation from schema definitions.

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Possible values:

spyUndefinedLanguage	= -1
spyJava	= 0
spyCpp	= 1
spyCSharp	= 2

13.3.3.26 SPYProjectItemTypes

Description

Enumeration values to identify the different elements in project item lists. See [SpyProjectItem.ItemType](#).

Possible values:

spyUnknownItem	= 0
spyFileItem	= 1
spyFolderItem	= 2
spyURLItem	= 3

13.3.3.27 SPYProjectType

Description

Enumeration values to parameterize generation of C# from schema definitions.

Possible values:

spyVisualStudioProject	= 0	Obsolete
spyVisualStudio2003Project	= 1	Obsolete
spyBorlandProject	= 2	Obsolete
spyMonoMakefile	= 3	Obsolete
spyVisualStudio2005Project	= 4	Obsolete
spyVisualStudio2008Project	= 5	Obsolete
spyVisualStudio2010Project	= 6	For C++ code also
spyVisualStudio2013Project	= 7	For C++ code also
spyVisualStudio2015Project	= 8	For C++ code also
spyVisualStudio2017Project	= 9	For C++ code also
spyVisualStudio2019Project	= 10	For C++ code also

13.3.3.28 SpySampleXMLGenerationChoiceMode

Description

This enumeration is used in [GenerateSampleXMLDlg.ChoiceMode](#):

```
spySampleXMLGen_FirstBranch    = 0
spySampleXMLGen_AllBranches    = 1
spySampleXMLGen_ShortestBranch = 2
```

13.3.3.29 SPYSampleXMLGenerationOptimization (Obsolete)

This enumeration is OBSOLETE since v2014.

Description

Specify the elements that will be generated in the Sample XML.
This enumeration is used in [GenerateSampleXMLDlg](#).

Possible values:

```
spySampleXMLGen_Optimized      = 0
spySampleXMLGen_NonMandatoryElements = 1
spySampleXMLGen_Everything     = 2
```

13.3.3.30 SpySampleXMLGenerationSampleValueHints

Description

This enumeration is used in [GenerateSampleXMLDlg.SampleValueHints](#)

```
spySampleXMLGen_FirstFit      = 0
spySampleXMLGen_RandomFit     = 1
spySampleXMLGen_CycleThrough  = 2
```

13.3.3.31 SPYSampleXMLGenerationSchemaOrDTDAssignment

Description

Specifies what kind of reference to the schema/DTD should be added to the generated Sample XML.
This enumeration is used in [GenerateSampleXMLDlg](#).

Possible values:

spySampleXMLGen_AssignRelatively	= 0
spySampleXMLGen_AssignAbsolutely	= 1
spySampleXMLGen_DoNotAssign	= 2

13.3.3.32 SPYSchemaDefKind

Description

Enumeration type to select schema diagram types.

Possible values:

spyKindElement	= 0
spyKindComplexType	= 1
spyKindSimpleType	= 2
spyKindGroup	= 3
spyKindModel	= 4
spyKindAny	= 5
spyKindAttr	= 6
spyKindAttrGroup	= 7
spyKindAttrAny	= 8
spyKindIdentityUnique	= 9
spyKindIdentityKey	= 10
spyKindIdentityKeyRef	= 11
spyKindIdentitySelector	= 12
spyKindIdentityField	= 13
spyKindNotation	= 14
spyKindInclude	= 15
spyKindImport	= 16
spyKindRedefine	= 17
spyKindFacet	= 18
spyKindSchema	= 19
spyKindCount	= 20

13.3.3.33 SPYSchemaDocumentationFormat

Description

Enumeration values to parameterize generation of schema documentation. These values are used in [SchemaDocumentationDialog.OutputFormat](#).

Possible values:

spySchemaDoc_HTML	= 0
spySchemaDoc_MSWord	= 1
spySchemaDoc_RTF	= 2
spySchemaDoc_PDF	= 3

13.3.3.34 SPYSchemaExtensionType

Description

Enumeration to specify different Schema Extension types.

Possible values:

spySchemaExtension_None	= 0
spySchemaExtension_SQL_XML	= 1
spySchemaExtension_MS_SQL_Server	= 2
spySchemaExtension_Oracle	= 3

13.3.3.35 SPYSchemaFormat

Description

Enumeration to specify different Schema Format types.

Possible values:

spySchemaFormat_Hierarchical	= 0
spySchemaFormat_Flat	= 1

13.3.3.36 SPYTextDelimiters

Description

Enumeration values to specify text delimiters for text export.

Possible values:

spyTabulator	= 0
spySemicolon	= 1
spyComma	= 2
spySpace	= 3

13.3.3.37 SPYTextEnclosing

Description

Enumeration value to specify text enclosing characters for text import and export.

Possible values:

spyNoEnclosing	= 0
spySingleQuote	= 1
spyDoubleQuote	= 2

13.3.3.38 SPYTypeDetection

Description

Enumeration to select how type detection works during [GenerateDTDOrSchema](#) and [GenerateDTDOrSchemaEx](#).

Possible values:

spyBestPossible	= 0
spyNumbersOnly	= 1
spyNoDetection	= 2

13.3.3.39 SPYURLTypes

Description

Enumeration to specify different URL types.

Possible values:

spyURLTypeAuto	= -1
spyURLTypeXML	= 0
spyURLTypeDTD	= 1

13.3.3.40 SPYValidateXSDVersion

Description

Enumeration values that select what XSD version to use. The XSD version that is selected depends on both (i) the presence/absence—and, if present, the value—of the `/xs:schema/@vc:minVersion` attribute of the XSD document, and (ii) the value of this enumeration.

`spyValidateXSDVersion_1_0` selects XSD 1.0 if `vc:minVersion` is absent, or is present with any value.
`spyValidateXSDVersion_1_1` selects XSD 1.1 if `vc:minVersion` is absent, or is present with any value.
`spyValidateXSDVersion_AutoDetect` selects XSD 1.1 if `vc:minVersion=1.1`. If the `vc:minVersion` attribute is absent, or is present with a value other than 1.1, then XSD 1.0 is selected.

Possible values

<code>spyValidateXSDVersion_AutoDetect</code>	= 0
<code>spyValidateXSDVersion_1_1</code>	= 1
<code>spyValidateXSDVersion_1_0</code>	= 2

13.3.3.41 SPYValidateErrorFormat

Description

Enumeration values that select the format of the error message.

Possible values

```
spyValidateErrorFormat_Text           = 0
spyValidateErrorFormat_ShortXML      = 1
spyValidateErrorFormat_LongXML       = 2
```

13.3.3.42 SPYViewModes**Description**

Enumeration values that define the different view modes for XML documents. The mode *spyViewAuthentic(4)* identifies the mode that was intermediately called DocEdit mode and is now called Authentic mode. The mode *spyViewJsonSchema* identifies a mode which is mapped to the Schema Design View on the GUI but is distinguished internally.

Possible values:

```
spyViewGrid           = 0
spyViewText           = 1
spyViewBrowser        = 2
spyViewSchema         = 3
spyViewContent        = 4    // obsolete
spyViewAuthentic      = 4
spyViewWSDL           = 5
spyViewZIP            = 6
spyViewEditionInfo    = 7
spyViewXBRL           = 8
spyViewJsonSchema     = 9
```

13.3.3.43 SPYVirtualKeyMask**Description**

Enumeration type for the most frequently used key masks that identify the status of the virtual keys. Use these values as bitmasks rather than directly comparing with them. When necessary, you can create further masks by using the 'logical or' operator.

Examples

```
' VBScript sample: check if ctrl-key is pressed
If ((i_nVirtualKeyStatus And spyCtrlKeyMask) <> 0) Then
    ' ctrl-key is pressed
End If
```

```
' VBScript sample: check if ONLY ctrl-key is pressed
If (i_nVirtualKeyStatus == spyCtrlKeyMask) Then
    ' exactly ctrl-key is pressed
End If
```

```
// JScript sample: check if any of the right virtual keys is pressed
if ((i_nVirtualKeyStatus & (spyRightShiftKeyMask | spyRightCtrlKeyMask | spyRightAltKeyMask)) != 0)
{
```

```

    ; ' right virtual key is pressed
}

```

Possible values:

spyNoVirtualKeyMask	= 0	
spyLeftShiftKeyMask	= 1	
spyRightShiftKeyMask	= 2	
spyLeftCtrlKeyMask	= 4	
spyRightCtrlKeyMask	= 8	
spyLeftAltKeyMask	= 16	
spyRightAltKeyMask	= 32	
spyShiftKeyMask	= 3	// spyLeftShiftKeyMask spyRightShiftKeyMask
spyCtrlKeyMask	= 12	// spyLeftCtrlKeyMask spyRightCtrlKeyMask
spyAltKeyMask	= 48	// spyLeftAltKeyMask spyRightAltKeyMask

13.3.3.44 SPYXMLDataKind

Description

The different types of XMLData elements available for XML documents.

Possible values:

spyXMLDataXMLDocStruct	= 0
spyXMLDataXMLEntityDocStruct	= 1
spyXMLDataDTDDocStruct	= 2
spyXMLDataXML	= 3
spyXMLDataElement	= 4
spyXMLDataAttr	= 5
spyXMLDataText	= 6
spyXMLDataCDATA	= 7
spyXMLDataComment	= 8
spyXMLDataPI	= 9
spyXMLDataDefDoctype	= 10
spyXMLDataDefExternalID	= 11
spyXMLDataDefElement	= 12
spyXMLDataDefAttlist	= 13
spyXMLDataDefEntity	= 14
spyXMLDataDefNotation	= 15
spyXMLDataKindsCount	= 16

13.4 ActiveX Integration

このセクションで説明されている Authentic Desktop ユーザーインターフェイスと ActiveX コントロール機能を使用するアプリケーションに統合することができます。ActiveX 技術により C++、C#、VB.NET、HTML などの統合のために使用される広範囲の言語が有効化されます (HTML 内に統合されている ActiveX コンポーネントは、Microsoft Internet Explorer バージョンと ActiveX をサポートすると共に作動する必要があります)。全てのコンポーネントはフル OLE コントロールです。Java への統合はランタイムクラスにより与えられます。

ActiveX コントロールを、カスタムコードに統合するには、Authentic Desktop Integration Package をインストールしてください (次を参照してください: http://www.altova.com/ide_integration.html)。すべての Authentic Desktop を最初に、そして、Authentic Desktop Integration Package として他の必要条件を言語とプラットフォームに応じてインストールしてください (次を参照してください: [必要条件](#))。

統合の2つの異なるレベル間を柔軟に選択することができます: アプリケーションレベルとドキュメントレベル

アプリケーションレベルの統合は (メニュー、ツールバー、ペインを含む) Authentic Desktop のフルインターフェイスを ActiveX コントロールをカスタムアプリケーションとして埋め込むことを意味します。例えば、多くのシナリオでは、カスタムアプリケーションを、Authentic Desktop グラフィカルユーザーインターフェイスを埋め込む1つのフォームから構成することができます。

ドキュメントレベルの統合は、先このアプローチは、簡単に実装することができますが、カスタム必要条件に従って柔軟的に Authentic Desktop グラフィカルユーザーインターフェイスを構成するためにお適切ではない可能性があります。ドキュメントレベルの統合は、アプリケーションに Authentic Desktop を一つ埋め込むことを意味します。メイン Authentic Desktop コントロールのみではなく、メインドキュメント エディターウィンドウの実装も含まれます。また、オプションで、追加ウィンドウの実装も含まれます。このアプローチは GUI の構成に柔軟性を与えますが、選択された言語により ActiveX コントロールとの高度なインタラクションが必要とされます。

セクション [アプリケーションレベルの統合とドキュメントレベルの統合](#) は、それぞれのレベルの主要なステップに関して説明しています。 [ActiveX 統合のサンプル](#) セクションは、C#、HTML、および、Java のサンプルを提供します。これらのサンプルを確認することにより、正しい選択が直ぐに行うことができます。 [オブジェクトレファレンス](#) セクションは、プロパティとメソッドに加え統合のために使用できるすべての COM オブジェクトについて説明しています。

Authentic Desktop を Visual Studio プラグインとして使用するための情報に関しては、次を参照してください: [Visual Studio 内の Authentic Desktop](#)。

13.4.1 必要条件

Authentic Desktop ActiveX コントロールをカスタムアプリケーションに統合するには、以下が使用中のコンピューターにインストールされている必要があります:

- Authentic Desktop
- Authentic Desktop Integration Package は、次からダウンロードできます:
http://www.altova.com/ide_integration.html

64 ビット ActiveX コントロールを統合するには、Authentic Desktop の64 ビット バージョンと Authentic Desktop Integration Package をインストールする必要があります。Visual Studio を使用して、Microsoft .NET フォームの下で開発されたアプリケーションに関しては、Authentic Desktop の32 ビット と64 ビット バージョンと Authentic Desktop Integration Package に、以下で説明されているとおりにインストールします。

メモ Windows XP と Windows Server 2003 で統合パッケージが正確に作動するためには、再頒布可能パッケージを手動でインストールしてください <https://www.microsoft.com/en-US/download/details.aspx?id=48145>。

Microsoft .NET (C#, VB.NET) with Visual Studio

Authentic Desktop ActiveX コントロールを、Microsoft .NET の下で開発された 32 ビット アプリケーションに統合するには、以下を使用中のコンピュータにインストールしてください。

- Microsoft .NET Framework 4.0 以降
- Visual Studio 2010/2012/2013/2015/2017/2019
- Authentic Desktop 32 ビットと Authentic Desktop Integration Package 32 ビット
- ActiveX コントロールが Visual Studio ツールボックスに追加されている必要があります (次を参照してください: [ActiveX コントロールをツールボックスに追加する](#))。

64 ビット ActiveX コントロールを統合する場合、次の必要条件が上記に加え満たされる必要があります。

- Authentic Desktop 32 ビットと Authentic Desktop Integration Package 32 ビット をインストールしてください (Visual Studio は 32 ビットで作動するため、32 ビット ActiveX コントロールを Visual Studio デザイナーに与える必要があります。)
- Authentic Desktop 64 ビットと Authentic Desktop Integration Package 64 ビット をインストールしてください (実際の 64 ビット ActiveX コントロールをランタイムカスタムアプリケーションに与えます。)
- Visual Studio 内で、64 ビットビルド更新を作成し、この更新を使用してアプリケーションをビルドします。サンプルに関しては、次を参照してください: [サンプル C# ソリューションの実行](#)。

Java

Eclipse 開発環境を使用して、Authentic Desktop ActiveX コントロールを Java アプリケーションに統合するには、次を使用中のコンピュータにインストールしてください。

- Java Runtime Environment (JRE) または Java Development Kit (JDK) 7 以降
- Eclipse
- Authentic Desktop and Authentic Desktop Integration Package

メモ Authentic Desktop ActiveX コントロールの 64 ビット バージョンを作動するには、Eclipse の 64 ビット バージョン、および Authentic Desktop と Authentic Desktop Integration Package の 64 ビット バージョン of を使用してください。

クライアントコンピュータ上での Authentic Desktop 統合とデプロイ

.NET アプリケーションを作成し他のクライアントに配布する場合は、クライアントコンピュータ上に次をインストールしてください。

- Authentic Desktop
- Authentic Desktop Integration Package
- カスタムの統合コード、または アプリケーション。

13.4.2 ActiveX コントロールをツールボックスに追加する

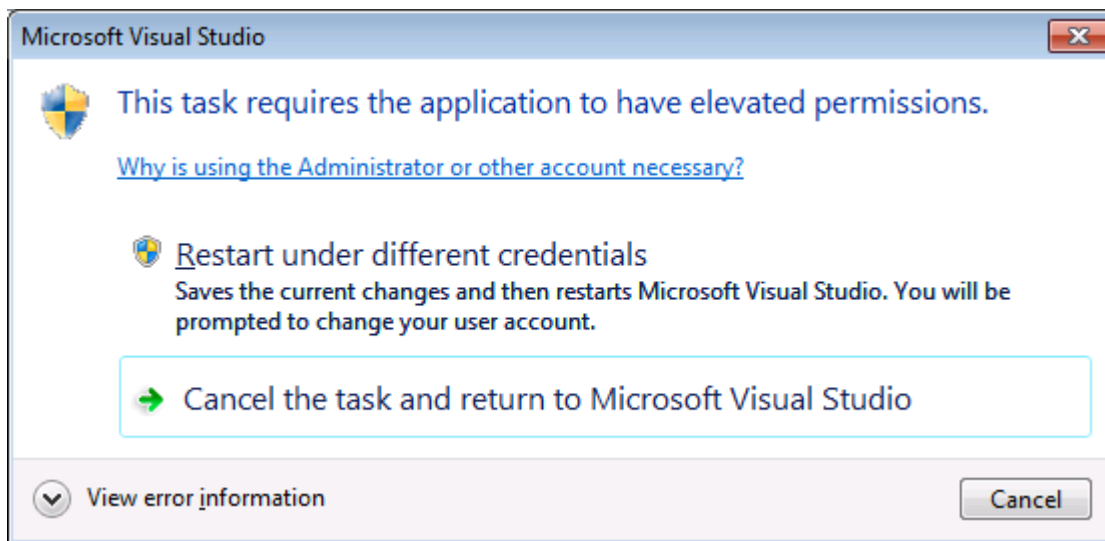
Visual Studio を使用して開発されたアプリケーション内で、Authentic Desktop ActiveX コントロールを使用するには、コントロールが最初に Visual Studio ツールボックスに追加される必要があります。

1. Visual Studio の「ツール」メニューから、「ツールボックスアイテムの選択」をクリックします。

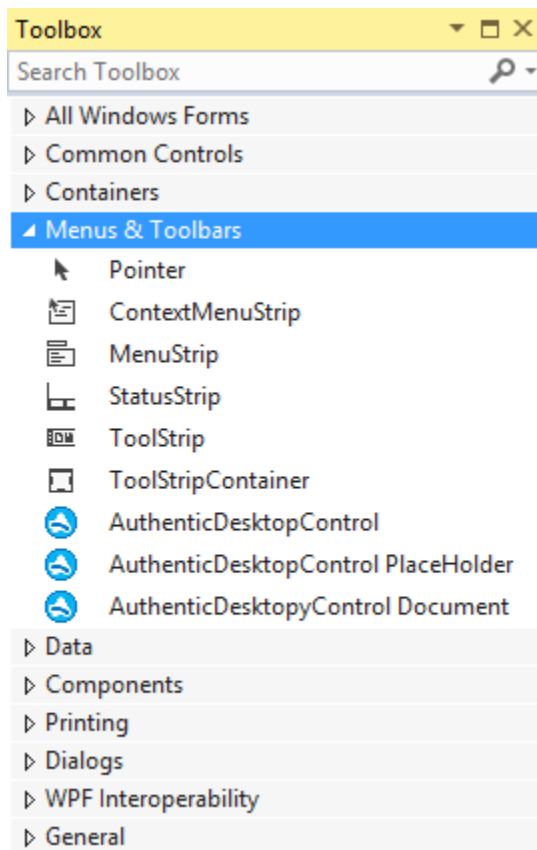
2. 「COM コンポーネント」タブ、から、Authentic DesktopControl, Authentic DesktopControl ドキュメント、および Authentic DesktopControl プレースホルダーの横のチェックボックスを選択します。

上記のコントロールを使用することができない場合、下のステップに従います:

1. 「COM コンポーネント」タブから「参照」をクリックします。 **AuthenticControl.ocx** を Authentic Desktop インストールからフォルダーを選択します。 Authentic Desktop Integration Package をインストールすることを忘れないでください。 それ以外の場合、このファイルを使用することはできません。次を参照してください: [必要条件](#)。
2. を更に高いバージョンと共に再起動するようプロンプトされると、「異なる資格情報下で再起動する」をクリックします。



上記のステップが完了すると、Authentic Desktop ActiveX コントロールは、Visual Studio ツールボックス内で使用できるようになります。



メモ アプリケーションレベルの統合では、**AuthenticDesktopControl** ActiveX コントロールのみが使用されます（次を参照してください：[アプリケーションレベルの統合](#)）。AuthenticDesktopControl ドキュメント と **AuthenticDesktopControl** プレースホルダー コントロールは、ドキュメントレベルの統合のために使用されます（次を参照してください：[ドキュメントレベルの統合](#)）。

13.4.3 アプリケーションレベルの統合

アプリケーションレベルの統合により使用中のアプリケーションのウィンドウに Authentic Desktop のインターフェイスを埋め込むことができます。この種類の統合により、すべてのメニュー、ツールバー、ステータスバー、ドキュメントウィンドウ、および、ヘルパーウィンドウを含む Authentic Desktop のユーザーインターフェイスを使用することができます。アプリケーションのユーザーインターフェイスのカスタム化は Authentic Desktop が提供する機能に限られています。これは、ヘルパーウィンドウの整理とサイズ調整、及びメニューとツールバーのカスタム化が含まれています。

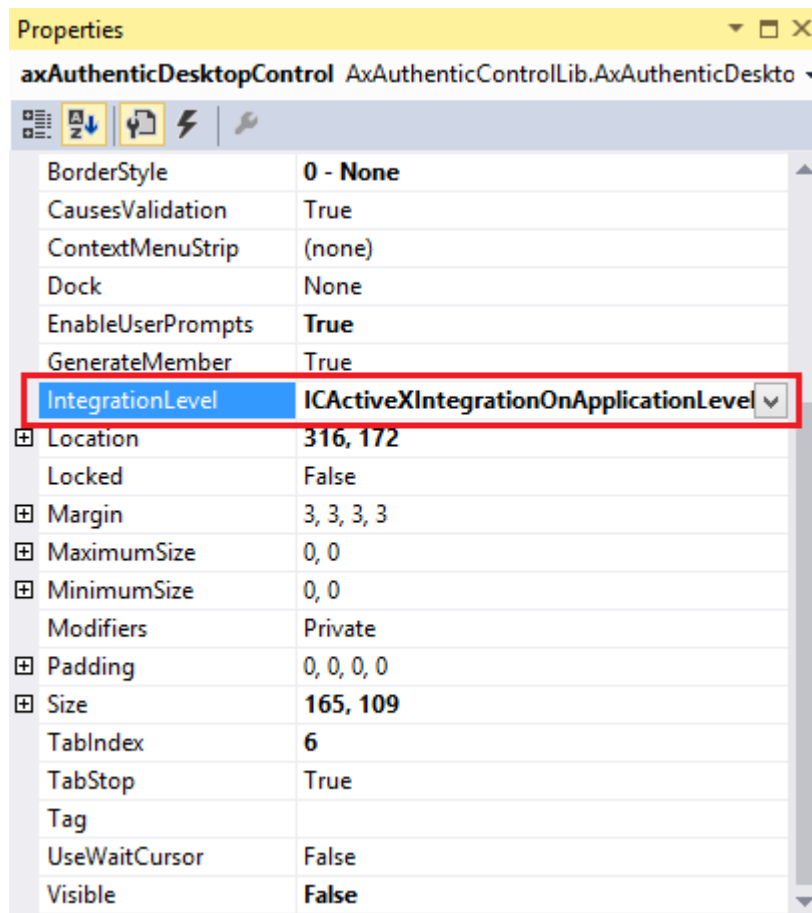
アプリケーションレベルで統合する場合、統合する必要がある唯一の ActiveX コントロールは [AuthenticDesktopControl](#) です。[AuthenticDesktopControlDocument](#) または [AuthenticDesktopControlPlaceholder](#) ActiveX コントロールをインスタンス化、またはアクセス使用とさせていただきます。

初期化、または、Authentic Desktop の振る舞いを自動化するには、[AuthenticDesktopControl](#) のために説明されているプロパティ、メソッド、とイベントを使用してください。Authentic Desktop 機能への複雑なアクセスのために [AuthenticDesktopControl.application](#) を使用することを考慮してください。

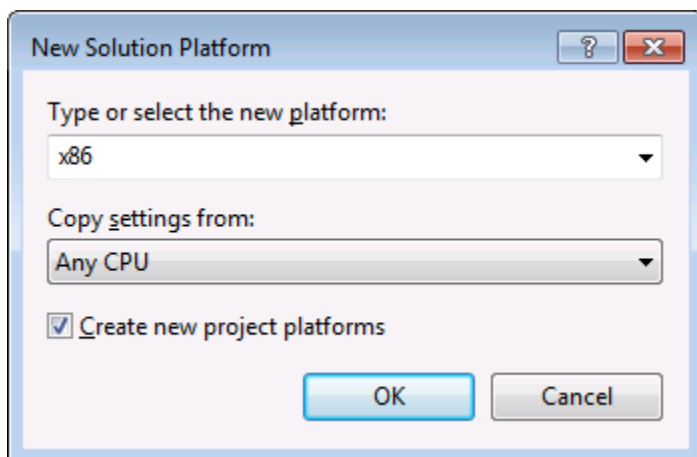
Authentic Desktop アプリケーションをHTML ページに埋め込むサンプルに関しては、次を参照してください: [アプリケーションレベルのHTML 統合](#)。

Visual Studio を使用するC# またはVB.NET でのAuthentic Desktop ActiveX コントロールをアプリケーションレベルで統合する基本的なアプリケーションを作成するステップは、以下のとおりです:

1. すべての必要条件が満たされていることを確認してください(次を参照してください: [必要条件](#))。
2. 新規のVisual Studio Windows Forms プロジェクトを新規の空のフォーム共に作成してください。
3. おざつていない場合、ActiveX コントロールをツールボックスに追加します(次を参照してください: [ActiveX コントロールをツールボックスに追加する](#))。
4. ツールボックスから、**AuthenticDesktopControl** をドラッグして、新規のフォームドロップしてください。
5. フォームで**AuthenticDesktopControl** を選択し、プロパティウィンドウ内で**IntegrationLevel** プロパティを**ICActiveXIntegrationOnApplicationLevel** に設定します。



6. ビルド(x86、x64)するプラットフォームに一致するビルドプラットフォームの更新を作成します。ビルドの更新の作成の方法は以下のとおりです:
 - a. Visual Studio 内のソリューションを右クリックし、「更新 マネージャー」を選択します。
 - b. 「アクティブなソリューションプラットフォーム」から、「新規作成...」を選択し、x86 またはx64 更新を選択します(このサンプルでは x86)。



Visual Studio 内でソリューションをビルドして実行する準備が整うと、ターゲットプラットフォーム(x86、x64)に一致する更新をしようとしてビルドすることを確認してください。

13.4.4 ドキュメントレベルの統合

アプリケーションレベルの統合に比べ、ドキュメントレベルの統合複雑ですが、ActiveX コントロールを使用することによりアプリケーションに Authentic Desktop 機能を柔軟的に埋め込むことができます。このアプローチにより、コードは Authentic Desktop ユーザーインターフェイスの次の部分に部分的にアクセスすることができます:

- ドキュメントの編集ウィンドウ
- プロジェクトウィンドウ
- 情報ウィンドウ
- メッセージウィンドウ
- 入力ヘルパーウィンドウ(要素、属性、エンティティ)
- 出力ウィンドウ

[アプリケーションレベルの統合](#) で説明されているとおり、ActiveX アプリケーションレベルの統合のためには **AuthenticDesktopControl** のみが必要とされる唯一のコントロールです。しかしながら、ActiveX ドキュメントレベルの統合の場合には、機能 Authentic Desktop が次の ActiveX コントロールにより与えられます:

1. [AuthenticDesktopControl](#)
2. [AuthenticDesktopControl ドキュメント](#)
3. [AuthenticDesktopControl フレースホルダ](#)

これらのコントロールは Authentic Desktop のアプリケーション インストールフォルダーで使用することのできる **AuthenticControl.ocx** ファイルにより与えられます。Visual Studio を使用して ActiveX 統合を開発すると、これらのコントロールを Visual Studio ツールボックスに追加する必要があります(次を参照してください: [ActiveX コントロールをツールボックスに追加する](#))。

ドキュメントレベルでの ActiveX コントロールのアプリケーションへの統合の基本のステップは、以下のとおりです:

1. 最初にアプリケーション内で **AuthenticDesktopControl** をインスタンス化します。「**AuthenticDesktopControl** ドキュメント」と **AuthenticDesktopControl フレースホルダ** コントロールへのサポートが有効化されるために、このコントロールの統合は必須です。[IntegrationLevel](#) プロパティを「ICActiveXIntegrationOnDocumentLevel」(または「1」)に設定することは重要です。ユーザーに対してコントロールを非表示にするには、「Visible」プロパティを「False」に設定します。

メモ ドキュメントレベルで統合する場合、未知の結果が生成される可能性があるため、**AuthenticDesktopControl** の「開く」メソッドを使用しないでください。 **AuthenticDesktopControl** ドキュメント と **AuthenticDesktopControl** プレースホルダ の対応するメソッドを代わりに使用して開いてください。

2. **AuthenticDesktopControl** ドキュメント のインスタンスを少なくとも一つアプリケーション内に作成します。このコントロールは、アプリケーションに Authentic Desktop のドキュメントの編集ウィンドウを与え、必要であれば、複数回インスタンス化することができます。

既存のファイルを開くために、メソッド「開く」を使用します。ドキュメントに関連した機能にアクセスするには、メソッドプロパティ「ドキュメント」を介してアクセスすることのできる「パス」と「保存」を使用します。

メモ コントロールは、読み取り専用モードをサポートしません。プロパティ「読み取り専用」の値は無視されます。

3. 任意で、アプリケーションに、アプリケーションで使用することのできるドキュメント ウィンドウ以外の各追加ウィンドウのために **AuthenticDesktopControl** プレースホルダ コントロールを追加することができます。

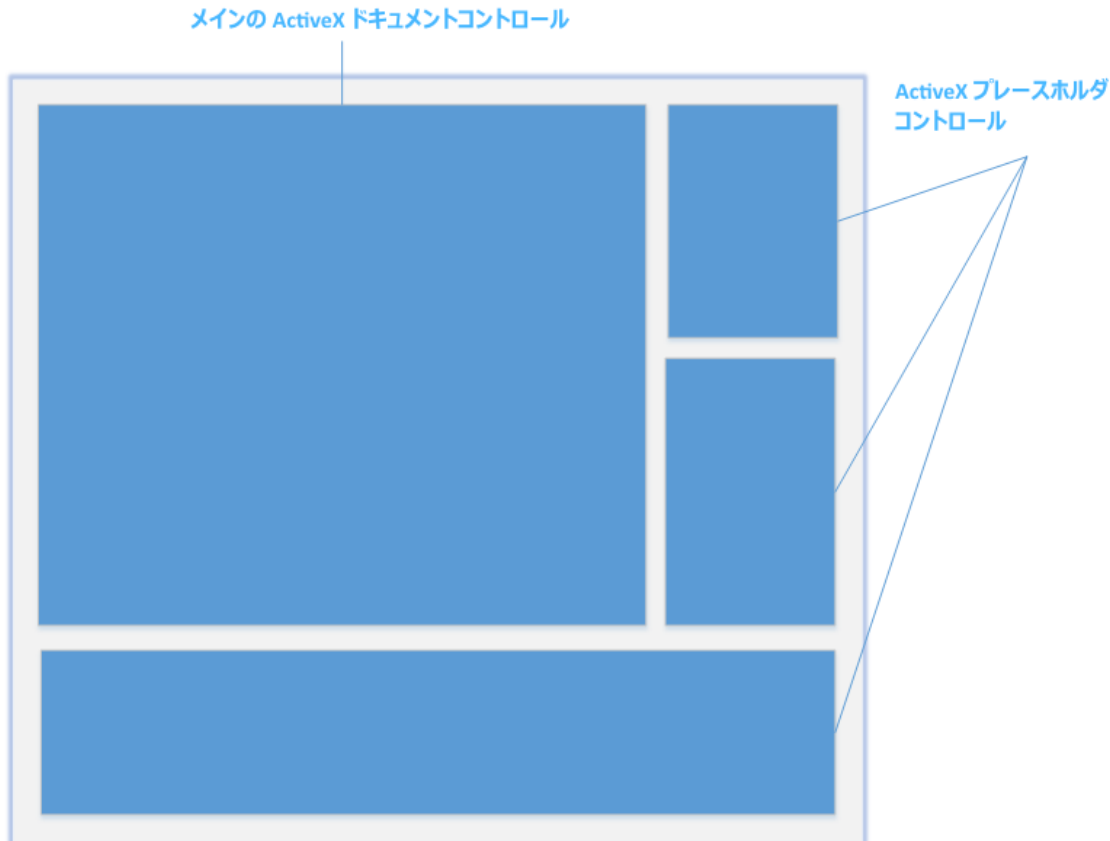
AuthenticDesktopControl プレースホルダ のインスタンスによりアプリケーションウィンドウを選択して Authentic Desktop の追加ウィンドウを埋め込むことができます。ウィンドウの種類（例えば、プロジェクトウィンドウ）はプロパティ「PlaceholderWindowID」により定義することができます。有効なウィンドウの識別子のための場合は、次を参照してください：[AuthenticDesktopControlPlaceholderWindow](#)。

メモ 各ウィンドウ識別子のための **AuthenticDesktopControl** プレースホルダ のみを使用してください。

Authentic Desktop プロジェクトウィンドウを選択するプレースホルダコントロールのために、追加メソッドを使用することができます。「OpenProject」を使用して、Authentic Desktop プロジェクトをロードすることができます。Authentic Desktop 自動化インターフェイスからプロパティプロジェクトとメソッドを使用して、他のプロジェクトに関連したオペレーションを操作します。

例えば、Visual Studio を使用した C# または VB.NET 内では、Authentic Desktop ActiveX コントロールをドキュメントレベルで統合する基本的なアプリケーションの作成ステップは、以下にリストされるものに類似しています。必要に応じて適用は複雑になる可能性があります。しかしながら、ActiveX ドキュメントレベルの統合のための必要条件として、下の命令を理解することは重要です。

1. 新規の空のフォームと共に新規の Visual Studio Windows Forms プロジェクトを作成します。
2. おざりではない場合、ActiveX コントロールをツールボックスに追加します（次を参照してください：[ActiveX コントロールをツールボックスに追加する](#)）。
3. ツールボックスから [AuthenticDesktopControl](#) をドラッグして新規のフォームにドラッグします。
4. **AuthenticDesktopControl** の「IntegrationLevel」プロパティを「ICActiveXIntegrationOnDocumentLevel」と「Visible」プロパティを「False」に設定します。コードから、または「プロパティ」ウィンドウから行うことができます。
5. [AuthenticDesktopControl ドキュメント](#) をツールボックスからフォームにドラッグします。このコントロールは、使用中のアプリケーションに Authentic Desktop のメインウィンドウを与え、ドキュメントのためにサイズを調整することができます。
6. 代わりに、フォームに一つまたは複数の [AuthenticDesktopControl プレースホルダ](#) コントロールを追加することができます（アプリケーションが必要とする各追加ウィンドウの型など。例えば、「プロジェクト」ウィンドウなど）。このような追加プレースホルダコントロールをメインドキュメントコントロールの下、または、左右に通常配置します。例:



7. 各「AuthenticDesktopControl プレースホルダ」コントロールの「PlaceholderWindowID」プロパティを有効なウィンドウ識別子に設定してください。有効な値のリストに関しては、次を参照してください：
[AuthenticDesktopControlPlaceholderWindow](#)。
8. アプリケーションコマンドを以下のように追加します（少なくとも、ドキュメントを開き、保存し、閉じる必要があります）。

Authentic Desktop コマンドをクエリする

ドキュメントレベルで統合する場合は、Authentic Desktop メニューまたはツールバーをアプリケーションで使用することはできません。代わりに、必要とするコマンドを取得し、状態を確認し、プログラマ的に実行することができます。

- 使用することのできるすべてのコマンドを取得するには、AuthenticDesktopControl の [CommandsList](#) プロパティを使用します。
- メニュー構造に整理されているコマンドを取得するには、メイン [Menu](#) プロパティを使用します。
- 表示される順番に整理されているコマンドを取得するには、[Toolbars](#) プロパティを使用します。
- Authentic Desktop コマンドを送信するには、[Exec](#) メソッドを使用します。
- コマンドが現在有効化されているか、または無効化されているかをクエリするために、[QueryStatus](#) メソッドを使用します。

これにより Authentic Desktop コマンドをアプリケーションのメニューとツールバーに柔軟的に統合することができます。

Authentic Desktop のインストールは、Authentic Desktop 内で使用されているコマンドラベルを与えます。GIF 書式内のアイコンのために Authentic Desktop インストールのフォルダー「<ApplicationFolder>\Examples\ActiveX\Images」を参照してください。コマンド名に対応するファイル名は、コマンド [参照](#) セクション内にリストされています。

一般的な注意点

Authentic Desktop の振る舞いを自動化するには、[AuthenticDesktopControl](#)、[AuthenticDesktopControl ドキュメント](#)と[AuthenticDesktopControl プレースホルダ](#)内で説明されているプロパティ、メソッド、とイベントを使用してください。

Authentic Desktop 機能への高度なアクセスに関しては、次のプロパティを考慮してください。

- [AuthenticDesktopControl.Application](#)
- [AuthenticDesktopControl.Document.Document](#)
- [AuthenticDesktopControl.PlaceHolder.Project](#)

これらのプロパティは Authentic Desktop 自動化インターフェイス (AuthenticDesktopAPI) へのアクセスを提供します

メモ ドキュメントを開くには、適切なドキュメントコントロールの [AuthenticDesktopControl.Document.Open](#) または [AuthenticDesktopControl.Document.New](#) を使用してください。プロジェクトを開くには、Authentic Desktop プロジェクトウィンドウを埋め込むプレースホルダコントロール上の [AuthenticDesktopControl.PlaceHolder.OpenProject](#) を常に使用してください。

異なるプログラミング環境内で必要なコントロールをインスタント化しアクセスする方法を説明するサンプルに関しては、次を参照してください：[ActiveX 統合のサンプル](#)。

13.4.5 ActiveX 統合のサンプル

このセクションは、異なるコンテナ環境とプログラム言語を使用する Authentic Desktop ドキュメントレベルの統合のサンプルを含みます (HTML セクションは、追加でアプリケーションレベルの統合のサンプルが含まれます)。全てのサンプルのソースコードは、Authentic Desktop インストール内のフォルダー <ApplicationFolder>\Examples\ActiveX で使用することができます。

13.4.5.1 C#

C# and Visual Studio のための基本の ActiveX 統合サンプルソリューションは、フォルダー <ApplicationFolder>\Examples\ActiveX\C# 内にあります。ソースコードをコンパイルする前に、サンプルを実行して、全ての必要条件が満たされていることを確認してください。次を参照してください：[サンプル C# ソリューションの実行](#)。

13.4.5.1.1 サンプル C# ソリューションの実行

フォルダー <ApplicationFolder>\Examples\ActiveX\C# 内で使用することのできるサンプル Visual Studio ソリューションは、Authentic Desktop ActiveX コントロールの使用法について説明しています。このソリューションをビルドして実行する前に、次のステップに注意してください。

ステップ 1: 必要条件をチェックする

Visual Studio 2010 または以降は、サンプルソリューションを開く必要があります。必要条件の完全なリストは、次を参照してください：[必要条件](#)。

ステップ 2: 書き込みのパーミッションを保有するディレクトリにサンプルをコピーする

Visual Studio を管理者として作動することを回避するには、デフォルトの場所から実行する代わりに、書き込みのパーミッションを持つディレクトリへのソースコードをコピーします。

ステップ 3: すべての必要とされるコントロールプロパティをチェック設定する

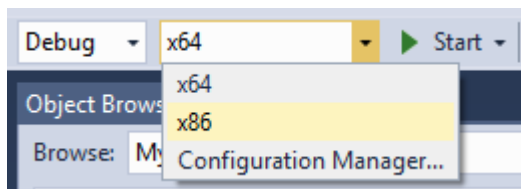
サンプルアプリケーションは、[AuthenticDesktopControlDocument](#) の 1 つのインスタンスと [AuthenticDesktopControlPlaceHolder](#) コントロールの複数のインスタンスを含んでいます。下のテーブルに表示されているとおりこれらのコントロールの次のプロパティが設定されていることを確認してください。

コントロール名	プロパティ	プロパティ値
axAuthenticDesktopControl	IntegrationLevel	ICActiveXIntegrationOnDocumentLevel
axAuthenticDesktopControlHelperWindowEntities	PlaceholderWindowID	2
axAuthenticDesktopControlHelperWindowAttributes	PlaceholderWindowID	1
axAuthenticDesktopControlHelperWindowElements	PlaceholderWindowID	0
axAuthenticDesktopControlHelperWindowInfo	PlaceholderWindowID	18
axAuthenticDesktopControlHelperWindowProject	PlaceholderWindowID	4

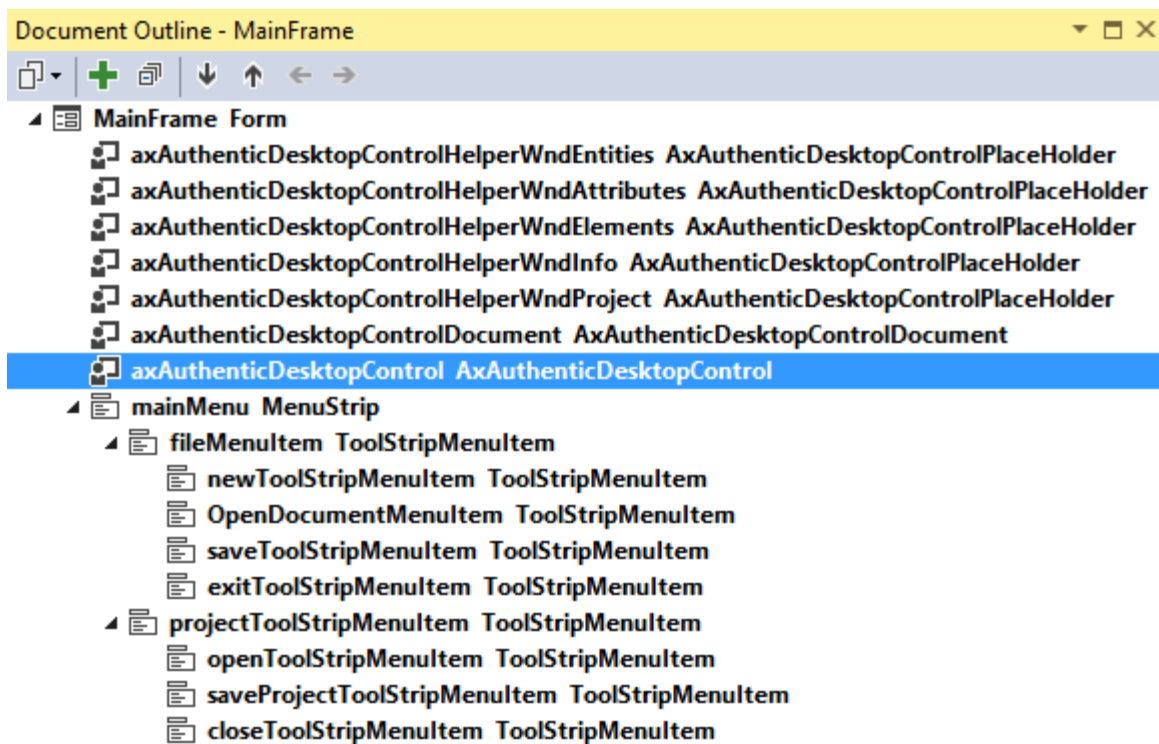
ActiveX コントロールのプロパティをビュー、または設定する方法は、以下のとおりです。

1. デザイナーウィンドウ内で **MDIMain.cs** フォームを開きます。

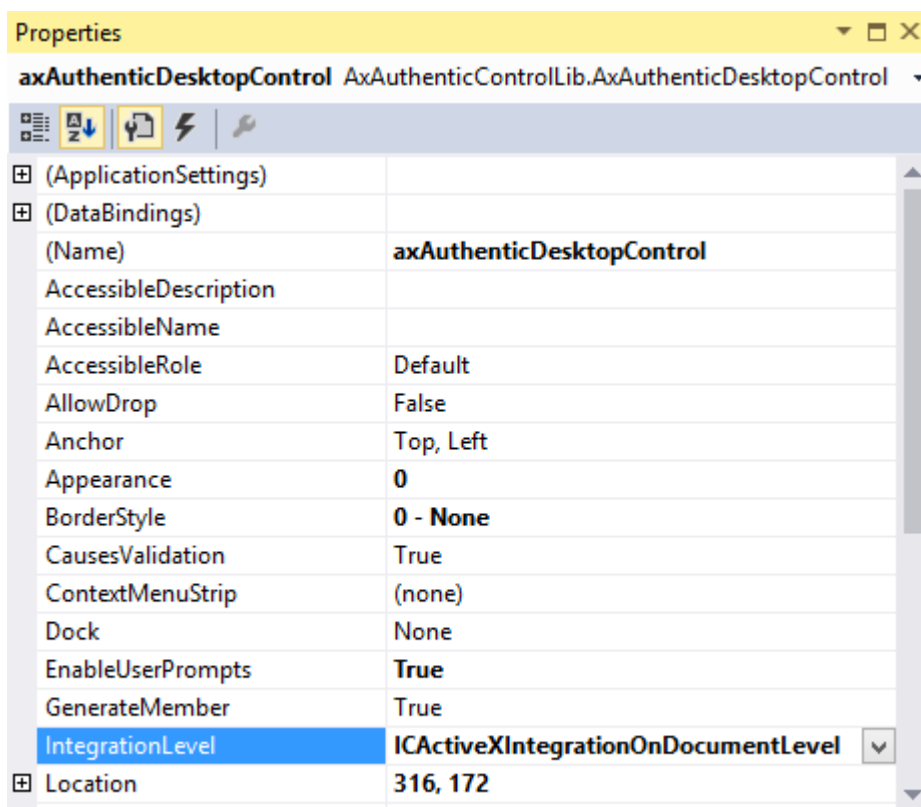
メモ 64 ビット Windows では、デザイナーウィンドウを開く前に Visual Studio ソリューションを「x86」にビルドを更新する必要があります。64 ビットアプリケーションとしてサンプルをビルドする場合は、次を参照してください: [必要条件](#)。



2. Visual Studio のドキュメント概要 ウィンドウを開きます(「表示」メニューから、「他のウィンドウ | ドキュメントの概要」をクリックします)。



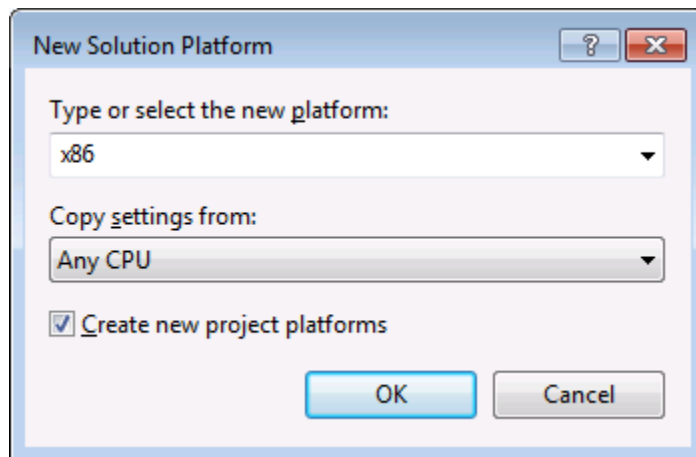
3. ドキュメント 概要 ウィンドウ内のActiveX コントロールをクリックし、プロパティ ウィンドウ内の必要とされるプロパティを編集します。
例



IntegrationLevel

ステップ 4: ビルドプラットフォームを設定する

- ビルド(x86、x64)するプラットフォームに一致するビルドプラットフォームの更新を作成します。ビルドの更新の作成の方法は以下のとおりです:
 - a. Visual Studio 内のソリューションを右クリックし、「更新マネージャー」を選択します。
 - b. 「アクティブなソリューションプラットフォーム」から、「新規作成...」を選択し、x86 または x64 更新を選択します（このサンプルでは、**x86**）。



Visual Studio 内でソリューションをビルドして実行する準備が整うと、ターゲットプラットフォーム(x86、x64)に一致するように更新をしようとしてビルドすることを忘れてください。それ以外の場合、ランタイムエラーが起こる可能性があります。

13.4.5.2 HTML

このセクションのコードリストは、アプリケーションレベルドキュメントレベルでの AuthenticDesktopControl の統合方法を示しています。全てのサンプルのためのソースコードは Authentic Desktop インストールのフォルダー <ApplicationFolder>\Examples\ActiveX\HTML 内にあります。サンプルは、32 ビット アプリケーションとして作動する場合、Internet Explorer 内でのみ作動することができます。

メモ 64 ビット モード上で作動する場合、Internet Explorer 10 未満以降は、ActiveX コントロールをロードしません。

13.4.5.2.1 アプリケーションレベルの HTML 統合

このサンプルは、アプリケーションレベルでの HTML ページへの Authentic Desktop コントロールの簡単な統合について説明しています。統合に関しては、次のセクションで説明されています：

- HTML コード内で AuthenticDesktopControl を開始します。
- ボタンドキュメントをロードするように実装し、コード生成タスクを自動化します。
- アプリケーションイベントのためのアクションを定義する。

このサンプルのためのコードは、Authentic Desktop インストール内の次の場所で使用することができます：

<ApplicationFolder>\Examples\ActiveX\HTML\AuthenticActiveX_ApplicationLevel.htm.

13.4.5.2.1.1 コントロールインスタンスを作成する

HTML Object タグは、AuthenticDesktopControl のインスタンスを作成するために使用されます。Classid は AuthenticDesktopControl です。幅と高さがウィンドウのサイズを指定します。アプリケーションレベルがデフォルトのため、追加のパラメータは必要ありません。

13.4.5.2.1.2 デフォルトのドキュメントを開くボタンを追加する

一部のタスクの自動化のシンプルなサンプルとして、ページボタンを追加します:

```
<input type="button" value="Open Marketing Expenses" onclick="BtnOpenMEFile()">
```

クリックされると、定義済みドキュメントが AuthenticDesktopControl 内で開かれます。異なるインストール上でサンプルを作動するため AuthenticDesktopControl に対して相対的なファイルを検索するためにメソッドを使用します。

13.4.5.2.1.3 カスタムイベントへの接続

サンプルは、プリシールを表示するために AuthenticDesktopControl カスタムイベントのための2つのイベントのコールバックを実装します:

```
<!-- ----- -->
<!-- custom event 'OnDocumentOpened' of AuthenticDesktopControl object -->
<SCRIPT LANGUAGE="javascript">
    function objAuthenticDesktopControl::OnDocumentOpened( objDocument )
    {
        // alert("Document '" + objDocument.Name + "' opened!");
    }
</SCRIPT>

<!-- ----- -->
<!-- custom event 'OnDocumentClosed' of AuthenticDesktopControl object -->
<SCRIPT LANGUAGE="javascript">
    function objAuthenticDesktopControl::OnDocumentClosed( objDocument )
    {
        // alert("Document '" + objDocument.Name + "' closed!");
    }
</SCRIPT>
```

13.4.5.2.2 ドキュメントレベルの HTML 統合

このサンプルはドキュメントレベルでHTML ページに Authentic Desktop コントロールを統合する方法について説明しています。次のピクについて説明されています:

- HTML コード内で AuthenticDesktopControl ActiveX コントロールオブジェクトをインスタンス化します。
- Authentic Desktop ファイルの編集を許可する AuthenticDesktopControlDocument ActiveX コントロールをインスタンス化します。
- AuthenticDesktopControl プロジェクトウィンドウのための AuthenticDesktopControlPlaceHolder をインスタンス化します。
- Authentic Desktop ヘルパーウィンドウをホストするために AuthenticDesktopControlPlaceHolder をインスタンス化します。
- 頻繁に使用される Authentic Desktop コマンドのためのシンプルなカスタムツールバーを作成します。
- Authentic Desktop の COM 自動化インターフェイスを使用するボタンを追加します。
- コマンドボタンを更新するためのイベントハンドラーを使用します。

このサンプルは、Authentic Desktop インストールの <ApplicationFolder>\Examples\ActiveX\HTML\ フォルダ内のファイル Authentic DesktopActiveX_ApplicationLevel.htm 内で使用することができます。

13.4.5.2.2.1 AuthenticDesktopControl をインスタンス化する

AuthenticDesktopControl HTML OBJECT タグは、AuthenticDesktopControl のインスタンスを作成するために使用されます。クラスID は AuthenticDesktopControl です。ユーザーインターフェイスのためのコントロール間ジェネレーターとして使用される幅と高さは 0 に設定されています。統合レベルは OBJECT タグ内のパラメーターとして指定されています。

```
<OBJECT id=""
  width="0"
  height="0"
  VIEWASTEXT>
  <PARAM NAME="IntegrationLevel" VALUE="1">
</OBJECT>
```

13.4.5.2.2.2 エディターウィンドウの作成

HTML OBJECT タグは、編集ウィンドウを埋め込むために使用されます。

```
<OBJECT id="objDoc1"
  width="600"
  height="500"
  VIEWASTEXT>
</OBJECT>
```

13.4.5.2.2.3 プロジェクトウィンドウの作成

HTML OBJECT タグは、AuthenticDesktopControlPlaceHolder ウィンドウを作成するために使用されます。最初の追加カスタムパラメーターは、Authentic Desktop プロジェクトウィンドウを表示するためのプレースホルダを定義します。2番目のパラメーターは、Authentic Desktop インストールで提供されたサンプルプロジェクトの1つをロードします（以下の場所で見つけることができます：[<yourusername>/MyDocuments フォルダ](#)）。

```
<OBJECT id="objProjectWindow"
  width="200"
  height="200"
  VIEWASTEXT>
</OBJECT>
```

13.4.5.2.2.4 ヘルパーウィンドウのためにプレースホルダを作成する

HTML OBJECT タグは、異なる Authentic Desktop ヘルパーウィンドウをホストすることができる AuthenticDesktopControlPlaceHolder ActiveX コントロールをインスタンス化するために使用されます。最初から、ヘルパーウィンドウは表示されません。サンプルファイルを参照してください。

```
<OBJECT id="objEHWindow"
  width="200"
  height="200"
  VIEWASTEXT>
  <PARAM name="PlaceholderWindowID" value="">
```

```
</OBJECT>
```

3つのボタンにより表示される実際のウィンドウを切り替えることができます。JavaScript 実行ボタンをクリックすることにより、プロパティ PlaceholderWindowID を定義されている対応する値に設定します。

```
<SCRIPT ID="Javahandlers" LANGUAGE="javascript">
// -----
// specify which of the windows shall be shown in the placeholder control.
function BtnHelperWindow(i_ePlaceholderWindowID)
{
    objEHWindow.PlaceholderWindowID = i_ePlaceholderWindowID;
}
</SCRIPT>
```

13.4.5.3 Java

Authentic Desktop ActiveX コンポーネントにJava コードからアクセスすることができます。Java 統合は下にリストされるライブラリで提供されています。これらのライブラリは、Authentic Desktop と Authentic Desktop Integration Package の両方をインストールした後、Authentic Desktop インストール内のフォルダー <ApplicationFolder>\Examples\JavaAPI で使用することができます (次を参照してください: [必要条件](#))。

- AltovaAutomation.dll: Altova 自動サーバーのためのJNI ラッパー (Authentic Desktop の32 ビット インストールの場合)
- AltovaAutomation_x64.dll: Altova 自動サーバーのためのJNI ラッパー (Authentic Desktop の64 ビット インストールの場合)
- AltovaAutomation.jar: Altova 自動サーバーにアクセスします
- AuthenticActiveX.jar: Authentic ActiveX インターフェイスをラップするJava クラス
- AuthenticActiveX_JavaDoc.zip: Javadoc インターフェイスのためのヘルプドキュメントを含むJavadoc ファイル

メモ Java ActiveX 統合を使用するには、.dll と .jar ファイルがJava クラス検索パス内に含まれている必要があります。

サンプルJava プロジェクト

サンプルJava プロジェクトは製品のインストールにより提供されます。Java プロジェクト をテストし、使用用途に応じて変更することができます。詳細に関しては、次を参照してください: [サンプルJava プロジェクト](#)。

JavaRules にActiveX コントロール名をマッピングする際のルール

ActiveX コントロールのドキュメントに関しては、次を参照してください: [オブジェクトレファレンス](#)。オブジェクトの名前変換はJava の場合、他の言語と比較して、若干異なります。具体的には、ActiveX コントロールとJava ラッパー間のマッピングのためのルールは、以下のとおりです:

- **クラスとクラス名**
Authentic Desktop ActiveX インターフェイスの全てのコンポーネントのために、コンポーネントの名前を持つJava クラスが存在します。
- **メソッド名**
Javadoc インターフェイス上のメソッド名は、COM インターフェイス上で使用される名前と同じですが、Java 名前変換に準拠する小文字で開始します。COM プロパティにアクセスするには、プロパティ名をプレフィックスする get と set から始まるJava メソッドを使用することができます。ふプロパティ書き込みをサポートしない場合、setter メソッドを使用することまでできません。サンプル AuthenticDesktopControl の IntegrationLevel プロパティ、Java メソッド getIntegrationLevel と setIntegrationLevel を使用することができます。

- **列挙**
ActiveX インターフェイス内で定義される全ての列挙のために、Java 列挙は同じ名前と値を使用して定義されます。
- **イベントとイベントハンドラー**
イベントをサポートする自動化インターフェイス内の全てのインターフェイスのために、同じ名前と'イベント'を使用した Javadoc インターフェイスを使用することができます。単一のイベントのオーバーロードを簡素化するために、全てのイベントのためのデフォルトの実装を使用する Java クラスが提供されます。この Java クラスの名前は、イベント インターフェイスの名前と 'DefaultHandler' です。例
AuthenticDesktopControl:アプリケーションにアクセスするための Java クラス
AuthenticDesktopControlEvents:AuthenticDesktopControl のためのイベント インターフェイス
AuthenticDesktopControlEventsDefaultHandler:AuthenticDesktopControlEvents のためのデフォルトのハンドラー

マッピングルールの例外

上記のルールの例外の一部は、以下のとおりです。これらは下にリストされています:

インターフェイス	Java 名
AuthenticDesktopControlDocument, method New	newDocument
Document, method SetEncoding	setFileEncoding
AuthenticView, method Goto	gotoElement
AuthenticRange, method Goto	gotoElement
AuthenticRange, method Clone	cloneRange

このセクション

このセクションは、Java コードからアクセスすることのできる基本の Authentic Desktop ActiveX 機能の一部を紹介しています。次のサブセクションに整理されています:

- [サンプル Java プロジェクト](#)
- [ActiveX コントロールの作成](#)
- [コントロール内のデータのロード](#)
- [基本のイベントハンドリング](#)
- [メニュー](#)
- [UI アップデートイベントハンドリング](#)
- [XML ツリーの作成](#)

13.4.5.3.1 サンプル Java プロジェクト

Authentic Desktop インストールパッケージには、アプリケーションフォルダー:

<ApplicationFolder>\Examples\ActiveX\Java\ の ActiveX サンプルフォルダーにあるサンプル Java プロジェクトが含まれています。

Java サンプルは、Java を使用して作成された共有されるデスクトップアプリケーション内に AuthenticDesktopControl を統合する方法を示しています。

バッチファイル BuildAndRun.bat を使用してコマンドラインからテストすることができ、また、Eclipse 内からサンプルプロジェクトをコンパイルして実行することもできます。これらのプロセスの使用方法を参照してください。

ファイルリスト

Java サンプルフォルダには、サンプルプロジェクトを動作するために必要な全てのファイルが含まれています。これらのファイルは下記にリストされています。

.classpath	Eclipse プロジェクトヘルパーファイル
.project	Eclipse プロジェクトファイル
AltovaAutomation.dll	Java-COM ブリッジ DLL パート (32ビットインストール用)
AltovaAutomation_x64.dll	Java-COM ブリッジ DLL パート (64ビットインストール用)
AltovaAutomation.jar	Java-COM ブリッジ Java ライブラリパート
AuthenticActiveX.jar	Authentic Desktop ActiveX コントロールの Java クラス
AuthenticActiveX_JavaDoc.zip	Java API のためのヘルプドキュメントを含む Javadoc ファイル
AuthenticDesktopContainer.java	Java サンプルソースコード
AuthenticDesktopContainerEventHandler.java	Java サンプルソースコード
BuildAndRun.bat	コンパイルするバッチファイル および、コマンドラインプロンプトからサンプルコードを実行します。Java 仮想マシンがランタイムとして存在する箇所にフォルダが期待されます。
XMLTreeDialog.java	Java サンプルソースコード

サンプルの内容

サンプルは、AWT フレームワーク内に AuthenticDesktop ドキュメント エディター ウィンドウ、プロジェクト ウィンドウ、情報 ウィンドウ と Authentic 入力ヘルパーを表示します。Authentic のための定義されているファイルメニューを読み取り、同じ構造を持つ AWT メニューを作成します。このメニューまたはプロジェクトを使用して、ドキュメント エディター内でファイルを開き作業することができます。

サンプルを必要に応じて変更することができます。

コードリスト内で次の特定の機能が説明されています。

- [ActiveX コントロールの作成](#): 自動サーバーとして考えられている Authentic Desktop を返します。または、作動中の場合、Authentic Desktop をアクティブ化します。
- [コントロール内のデータのロード](#): Authentic Desktop にインストールされているサンプルドキュメントの1つを検索し開きます。
- [基本のイベントハンドリング](#): すべての開かれているドキュメントのビューをテキストビューに変えます。コードも開かれているドキュメント内での反復の方法を示しています。
- [メニュー](#): アクティブドキュメントを検証し、メッセージボックス内に結果を表示します。コードは出力パラメータの使用法について説明しています。
- [UI アップデートイベントハンドリング](#): Authentic Desktop イベントの処理方法について説明しています。
- [XML ツリーの作成](#): XML ツリーの作成とモーダルアクティブ化について説明しています。

サンプルフォルダーへのパスの更新

与えられているサンプルを作動する前に、**AuthenticDesktopContainer.java** ファイルを編集します。具体的には、Authentic Desktop サンプルファイルが使用中のオペレーティングシステム上で保存されている実際のフォルダーを参照する次のパスを確認します：

```
// Locate samples installed with the product.
final String strExamplesFolder = System.getenv( "USERPROFILE" ) + "\\My Documents\
\Altova\Authentic2021\AuthenticExamples\";
```

コマンドラインからサンプルを実行する

コマンドラインからサンプルを作動します。

1. すべての必要条件が満たされていることを確認してください(次を参照してください: [必要条件](#)).
2. コマンドプロンプトウィンドウを開き、サンプルJava プロジェクト フォルダーに対して現在のディレクトリを変更し、次を入力します:

```
buildAndRun.bat "<Path-to-the-Java-bin-folder>"
```

3. 「Enter」を押します。

AuthenticDesktopContainer.java 内のJava ソースをコンパイルされ、実行されます。

Eclipse 内でサンプルをコンパイルし実行する サンプルJava プロジェクトをEclipse にインポートする

1. すべての必要条件が満たされていることを確認してください(次を参照してください: [必要条件](#)).
2. ファイルメニューから、「インポート」をクリックします。
3. 「ワークスペースで既存のプロジェクトを開く」を選択し、次の場所にあるEclipse プロジェクトファイルを参照します:
<ApplicationFolder>\Examples\ActiveX\Java\。このフォルダーでは書き込みのアクセスがないため、インポートダイアログボックス上の「ワークスペースにプロジェクトをコピーする」チェックボックスを選択することが奨励されます。

サンプルアプリケーションを作動するには、Package Explorer 内のプロジェクトを右クリックし、コマンド「実行 | Java アプリケーション」を選択します。

コード内のコメントを使用するとJava API クラスのためのヘルプを使用することができ、とEclipse 内でJavadoc ビューを有効化するには、メニューコマンド「Window | ビューの表示 | Javadoc」を選択してください。

13.4.5.3.2 ActiveX コントロールの作成

下のコードリストは、ActiveX コントロールの作成方法を示しています。コンストラクターは、Java ラッパーオブジェクトを作成します。カンパシによって派生したオブジェクトを、`wrap` または `wrapAdd` と追加すると、ラップされたActiveX の生成がトリガーされます。

```
01  /**
02   * Authentic Desktop manager control - always needed
03   */
04  public static AuthenticDesktopControl    authenticDesktopControl = null;
05
06  /**
07   * Authentic Desktop document editing control
08   */
09  public static AuthenticDesktopControlDocument    authenticDesktopDocument = null;
10
11  /**
```

```

12  * Tool windows - Authentic Desktop place-holder controls
13  */
14  private static AuthenticDesktopControlPlaceHolder authenticDesktopInfoToolWindow = null;
15  private static AuthenticDesktopControlPlaceHolder authenticDesktopEHElementToolWindow = null;
16  private static AuthenticDesktopControlPlaceHolder authenticDesktopProjectToolWindow = null;
17
18  // Create the Authentic Desktop ActiveX control, The parameter determines that we want
19  // to place document controls and place-holder controls individually.
20  // It gives us full control over the menu, as well.
21  authenticDesktopControl = new AuthenticDesktopControl(
22      ICActiveXIntegrationLevel.ICActiveXIntegrationOnDocumentLevel.getValue() );
23
24  authenticDesktopDocument = new AuthenticDesktopControlDocument();
25  authenticDesktopDocument.setPreferredSize( new Dimension ( 640, 480 ) );
26  frame.add( authenticDesktopDocument, BorderLayout.CENTER );
27
28  // Create a project window and open the sample project in it
29  authenticDesktopProjectToolWindow = new AuthenticDesktopControlPlaceHolder(
30      XMLSpyControlPlaceholderWindow.XMLSpyControlProjectWindowToolWnd.getValue() );
31  authenticDesktopProjectToolWindow.setPreferredSize( new Dimension( 200, 200 ) );

```

13.4.5.3.3 コントロール内のデータのロード

下のコードリストは、ActiveX コントロール内にデータをロードを表示する方法を表示しています。

```

1  // Locate samples installed with the product.
2  final String strExamplesFolder = System.getenv( "USERPROFILE" ) +
3      "\\My Documents\\Altova\\Authentic2021\\AuthenticExamples\\";
4  authenticDesktopProjectToolWindow = new
5  AuthenticDesktopControlPlaceHolder( XMLSpyControlPlaceholderWindow.XMLSpyControlProjectWindowToolWnd
6  .getValue() );

```

13.4.5.3.4 基本のイベントハンドリング

下のコードリストは、基本のイベントの処理方法について表示しています。AuthenticDesktopControl の open メソッド、または、メニューまたはプロジェクトツリーを使用してファイルを開こうとすると、onOpenedOrFocused イベントは、アタッチ済みのイベントハンドラーに送信されます。このイベントのための基本の処理により、Authentic DesktopDocumentControl の open メソッドを呼び出すことによりファイルを開きます。

```

01 // Open the PXF file when button is pressed
02 btnOpenPxf.addActionListener( new ActionListener() {
03     public void actionPerformed(ActionEvent e) {
04         try {
05             authenticDesktopControl.open( strExamplesFolder + "OrgChart.pxf" );
06         } catch (AutomationException el) {
07             el.printStackTrace();
08         }
09     }
10 });
11 public void onOpenedOrFocused( String i_strFileName, boolean i_bOpenWithThisControl,
12 boolean i_bFileAlreadyOpened ) throws AutomationException
13 {
14     // Handle the New/Open events coming from the Project tree or from the menus
15     if ( !i_bFileAlreadyOpened )

```

```

15  {
16  // This is basically an SDI interface, so open the file in the already existing document
control
17  try {
18  AuthenticDesktopContainer.authenticDesktopDocument.open( i_strFileName );
19  AuthenticDesktopContainer.authenticDesktopDocument.requestFocusInWindow();
20  } catch (Exception e) {
21  e.printStackTrace();
22  }
23  }
24  }

```

13.4.5.3.5 メニュー

下のコードリストは、メニューアイテムの作成方法を表示しています。各 Authentic DesktopCommand オブジェクトは、コマンドのIDへのセットされている ActionCommand を持つ、対応する MenuItem オブジェクト、を取得します。全てのメニューアイテムにより生成されているアクションは、(閉じられたいメカニズムを再度解釈するなどの)特定の処理を行う同じ関数により処理されます。または、exec メソッドを呼び出すことにより AuthenticDesktopControl オブジェクトに実行を委任することができます。メニューの作成中に呼びされる menuMap オブジェクトは、後で使用されます(次のセクションを参照してください: [UI アップデートイベントハンドリング](#)).

```

01 // Load the file menu when the button is pressed
02 btnMenu.addActionListener( new ActionListener() {
03     public void actionPerformed(ActionEvent e) {
04         try {
05             // Create the menubar that will be attached to the frame
06             MenuBar mb = new MenuBar();
07             // Load the main menu's first item - the File menu
08             XMLSpyCommand xmlSpyMenu = xmlSpyControl.getMainMenu().getSubCommands().getItem( 0 );
09             // Create Java menu items from the Commands objects
10             Menu fileMenu = new Menu();
11             handlerObject.fillMenu( fileMenu, xmlSpyMenu.getSubCommands() );
12             fileMenu.setLabel( xmlSpyMenu.getLabel().replace( "&", "" ) );
13             mb.add( fileMenu );
14             frame.setMenuBar( mb );
15             frame.validate();
16         } catch (AutomationException e1) {
17             e1.printStackTrace();
18         }
19         // Disable the button when the action has been performed
20         ((AbstractButton) e.getSource()).setEnabled( false );
21     }
22 } );
23 /** * Populates a menu with the commands and submenus contained in an XMLSpyCommands object */
24 public void fillMenu(Menu newMenu, XMLSpyCommands xmlSpyMenu) throws AutomationException
25 {
26     // For each command/submenu in the xmlSpyMenu
27     for ( int i = 0 ; i < xmlSpyMenu.getCount() ; ++i )
28     {
29         XMLSpyCommand xmlSpyCommand = xmlSpyMenu.getItem( i );
30         if ( xmlSpyCommand.getIsSeparator() )
31             newMenu.addSeparator();
32         else
33         {
34             XMLSpyCommands subCommands = xmlSpyCommand.getSubCommands();
35             // Is it a command (leaf), or a submenu?
36             if ( subCommands.isNull() || subCommands.getCount() == 0 )
37             {
38                 // Command -> add it to the menu, set its ActionCommand to its ID and store it in the
menuMap

```



```

39     MenuItem mi = new MenuItem( xmlSpyCommand.getLabel().replace( "&", "" ) );
40     mi.setActionCommand( "" + xmlSpyCommand.getID() );
41     mi.addActionListener( this );
42     newMenu.add( mi );
43     menuMap.put( xmlSpyCommand.getID(), mi );
44 }
45 else
46 {
47     // Submenu -> create submenu and repeat recursively
48     Menu newSubMenu = new Menu();
49     fillMenu( newSubMenu, subCommands );
50     newSubMenu.setLabel( xmlSpyCommand.getLabel().replace( "&", "" ) );
51     newMenu.add( newSubMenu );
52 }
53 }
54 }
55 }
56
57 /**
58  * Action handler for the menu items
59  * Called when the user selects a menu item; the item's action command corresponds to the
60  * command table for XMLSpy
61  */
62 public void actionPerformed((ActionEvent e)
63 {
64     try
65     {
66         int iCmd = Integer.parseInt( e.getActionCommand() );
67         // Handle explicitly the Close commands
68         switch ( iCmd )
69         {
70             case 57602: // Close
71             case 34050: // Close All
72                 AuthenticDesktopContainer.initXmlSpyDocument();
73                 break;
74             default:
75                 AuthenticDesktopContainer.xmlSpyControl.exec( iCmd );
76                 break;
77         }
78     }
79     catch (Exception ex)
80     {
81         ex.printStackTrace();
82     }
83 }

```

13.4.5.3.6 UI アップデートイベントハンドリング

下のコードリストは、UI-Update イベントハンドラーの作成方法を表示しています。

```

01 /**
02  * Call-back from the XMLSpyControl.
03  * Called to enable/disable commands
04  */
05 @Override
06 public void onUpdateCmdUI() throws AutomationException
07 {
08     // A command should be enabled if the result of queryStatus contains the Supported (1) and
09     // Enabled (2) flags
10     for ( java.util.Map.Entry<Integer, MenuItem> pair : menuMap.entrySet() )

```

```
10     pair.getValue().setEnabled( AuthenticDesktopContainer.authenticDesktopControl.queryStatus( pair.getKey() ) > 2 );
11 }
12 /**
13  * Call-back from the XMLSpyControl.
14  * Usually called while enabling/disabling commands due to UI updates
15  */
16 @Override
17 public boolean onIsActiveEditor( String i_strFilePath ) throws AutomationException
18 {
19     try {
20         return
AuthenticDesktopContainer.authenticDesktopDocument.getDocument().getFullName().equalsIgnoreCase( i_strFilePath );
21     } catch ( Exception e ) {
22         return false;
23     }
24 }
```

13.4.5.3.7 XML ツリーの作成

下のリストは、XML データオブジェクトをツリー内ノードとしてロードします。

```
01 // access required XMLSpy Java-COM classes
02 import com.altova.automation.XMLSpy.XMLData;
03
04 // access AWT and Swing components
05 import java.awt.*;
06 import javax.swing.*;
07 import javax.swing.tree.*;
08
09 /**
10  * A simple example of a tree control loading the structure from an XMLData object.
11  * The class receives an XMLData object, loads its nodes in a JTree, and prepares
12  * for modal activation.
13  *
14  * Feel free to modify and extend this sample.
15  *
16  * @author Altova GmbH
17  */
18 class XMLTreeDialog extends JDialog
19 {
20     /**
21      * The tree control
22      */
23     private JTree myTree;
24
25     /**
26      * Root node of the tree control
27      */
28     private DefaultMutableTreeNode top ;
29
30     /**
31      * Constructor that prepares the modal dialog containing the filled tree control
32      * @param xml    The data to be displayed in the tree
33      * @param parent Parent frame
34      */
35     public XMLTreeDialog( XMLData xml, Frame parent )
```

```
36 {
37     // Construct the modal dialog
38     super( parent, "XML tree", true );
39     // Arrange controls in the dialog
40     top = new DefaultMutableTreeNode("root");
41     myTree = new JTree(top);
42     setContentPane( new JScrollPane( myTree ) );
43     // Build up the tree
44     fillTree( top, xml );
45     myTree.expandRow( 0 );
46 }
47
48 /**
49  * Loads the nodes of an XML element under a given tree node
50  * @param node Target tree node
51  * @param elem Source XML element
52  */
53 private void fillTree( DefaultMutableTreeNode node, XMLData elem)
54 {
55     try
56     {
57         // There are several ways to iterate through child elements: either using the
58         // getFirstChild/getNextChild,
59         // or by incrementing an index up to countChildren and calling getChild [as shown below].
60         // If you only want to get children of one kind, you should use
61         // countChildrenKind/getChildKind,
62         // or provide a kind to the getFirstChild before iterating with the getNextChild.
63         int nSize = elem.countChildren();
64         for ( int i = 0 ; i < nSize ; ++i)
65         {
66             // Create a new tree node for each child element, and continue recursively
67             XMLData newElem = elem.getChild(i) ;
68             DefaultMutableTreeNode newNode = new DefaultMutableTreeNode( newElem.getName() ) ;
69             node.add( newNode ) ;
70             fillTree( newNode, newElem ) ;
71         }
72     }
73     catch (Exception e)
74     {
75         e.printStackTrace();
76     }
77 }
```

13.4.6 コマンド レファレンス

このセクションは、Authentic Desktop 内で使用することができる全てのメニューコマンドの名前と識別子をリストしています。各サブセクションは Authentic Desktop コマンドテーブルの対応するトップレベルメニューからのコマンドが以下のとおり整理されています:

- 「メニューアイテム」列は、コマンドのメニューテキストを Authentic Desktop で表示される順番に表示します。コマンドの機能を簡単に識別することができます。
- 「コマンド名」列は、Authentic Desktop インストールディレクトリの **ActiveXImages** フォルダから同じ名前を持つアイコンを取得するために使用される文字列を指定します。
- 「ID」列は、このコマンドを実行または再実行するメソッドの引数として与えられる列の数値識別子を表示します。

コマドを実行するには、[AuthenticDesktopControl.Exec](#) または [AuthenticDesktopControlDocument.Exec](#) メソッドを利用します。コマドのステータスをクエリするには、[AuthenticDesktopControl.QueryStatus](#) または [AuthenticDesktopControlDocument.QueryStatus](#) メソッドを使用します。

インストールされた Authentic Desktop のエディションにより、コマドの一部がサポートされない場合があります。

13.4.6.1 「ファイル」メニュー

「ファイル」メニューには以下のコマドが含まれます:

メニューアイテム	コマド名	ID
新規作成...	ID_FILE_NEW	57600
開く...	ID_FILE_OPEN	57601
再ロード	IDC_FILE_RELOAD	34065
エコード	IDC_ENCODING	34061
閉じる	ID_FILE_CLOSE	57602
全て閉じる	IDC_CLOSE_ALL	34050
アクティブなファイル以外閉じる	IDC_CLOSE_OTHERS	34271
保存	ID_FILE_SAVE	57603
名前を付けて保存	ID_FILE_SAVE_AS	57604
全て保存する	ID_FILE_SAVE_ALL	34208
電子メールで送信する	ID_FILE_SEND_MAIL	57612
印刷	ID_FILE_PRINT	57607
印刷プレビュー	IDC_PRINT_PREVIEW	34104
印刷設定	ID_FILE_PRINT_SETUP	57606
最近使ったファイル	ID_FILE_MRU_FILE1	57616
終了	ID_APP_EXIT	57665

13.4.6.2 「編集」メニュー

「編集」メニューには以下のコマドが含まれます:

メニューアイテム	コマド名	ID
元に戻す	ID_EDIT_UNDO	57643

メニューアイテム	コマンド名	ID
やり直し	ID_EDIT_REDO	57644
切り取り	ID_EDIT_CUT	57635
コピー	ID_EDIT_COPY	57634
貼り付け	ID_EDIT_PASTE	57637
削除	ID_EDIT_CLEAR	57632
全て選択	ID_EDIT_SELECT_ALL	57642
検索	ID_EDIT_FIND	57636
次を検索	ID_EDIT_REPEAT	57640
置換	ID_EDIT_REPLACE	57641

13.4.6.3 「プロジェクト」メニュー

「プロジェクト」メニューには以下のコマンドが含まれます:

メニューアイテム	コマンド名	ID
新規のプロジェクト	IDC_ICPROJECTGUI_NEW	37200
プロジェクトを開く	IDC_ICPROJECTGUI_OPEN	37201
プロジェクトを再ロードする	IDC_ICPROJECTGUI_RELOAD	37202
プロジェクトを閉じる	IDC_ICPROJECTGUI_CLOSE	37203
プロジェクトを保存する	IDC_ICPROJECTGUI_SAVE	37204
プロジェクトに名前を付けて保存する	IDC_ICPROJECTGUI_SAVE_AS	37207
ソース管理を有効化する	ID_SCC_ENABLE	38602
プロジェクトにファイルを追加する	IDC_ICPROJECTGUI_ADD_FILES_TO_PROJECT	37205
グローバルリソースをプロジェクトに追加する	IDC_ICPROJECTGUI_ADD_GLOBAL_RESOURCE_TO_PROJECT	37239
URL をプロジェクトに追加する	IDC_ICPROJECTGUI_ADD_URL_TO_PROJECT	37206
アクティブなファイルをプロジェクトに追加する	IDC_ICPROJECTGUI_ADD_ACTIVE_FILE_TO_PROJECT	37208
アクティブならびに関連するファイルをプロジェクトに追加する	IDC_ICPROJECTGUI_ADD_ACTIVE_AND_RELATED_FILES_TO_PROJECT	37209

メニューアイテム	コマンド名	ID
プロジェクトフォルダーをプロジェクトに追加する	IDC_ICPROJECTGUI_ADD_FOLDER_TO_PROJECT	37210
外部フォルダーをプロジェクトに追加する	IDC_ICPROJECTGUI_ADD_EXT_FOLDER_TO_PROJECT	37211
外部 Web フォルダーをプロジェクトに追加する	IDC_ICPROJECTGUI_ADD_EXT_URL_FOLDER_TO_PROJECT	37212
スクリプト設定...	IDC_PROJECT_SCRIPT_SETTINGS	34136
プロパティ...	IDC_ICPROJECTGUI_PROJECT_PROPERTIES	37223
最近使ったプロジェクト	IDC_ICPROJECTGUI_RECENT	37224

13.4.6.4 「XML」メニュー

「XML」メニューには以下のコマンドが含まれます:

メニューアイテム	コマンド名	ID
整形形式のチェック	IDC_CHECK_WELL_FORM	34049
XML の検証	IDC_VALIDATE	32954

13.4.6.5 「XSL/XQuery」メニュー

「XSL/XQuery」メニューには以下のコマンドが含まれます:

メニューアイテム	コマンド名	ID
XSL 変換	IDC_TRANSFORM_XSL	33006
XSL-FO 変換	IDC_TRANSFORM_XSLFO	33007
XSL パラメーター/XQuery 変数...	IDC_TRANSFORM_XSL_PARAMS	33008

13.4.6.6 「Authentic」メニュー

「Authentic」メニューには以下のコマンドが含まれます:

メニューアイテム	コマンド名	ID
新規ドキュメント...	IDC_AUTHENTIC_NEW_FILE	34036

メニューアイテム	コマンド名	ID
データベースデータの編集...	IDC_AUTHENTIC_EDIT_DB	34035
StyleVision スタイルシート の編集	IDC_EDIT_SPS	34060
XML データの新たな行を選択し編集	IDC_CHANGE_WORKING_DB_XML_CELL	32861
XML 署名...	IDC_AUTHENTICGUI_XMLSIGNATURE	32862
XML エンティティを定義	IDC_DEFINE_ENTITIES	32805
マークアップを隠す	IDC_MARKUP_HIDE	32855
小さなマークアップの表示	IDC_MARKUP_SMALL	32858
大きなマークアップの表示	IDC_MARKUP_LARGE	32856
混合マークアップの表示	IDC_MARKUP_MIXED	32857
太字	IDC_AUTHENTICGUI_RICHEDIT_TOGGLEBOLD	32813
イタリック	IDC_AUTHENTICGUI_RICHEDIT_TOGGLEITALIC	32814
下線	IDC_AUTHENTICGUI_RICHEDIT_TOGGLEUNDERLINE	32815
取り消し線	IDC_AUTHENTICGUI_RICHEDIT_TOGGLESTRIKETHROUGH	32816
前景色	IDC_AUTHENTICGUI_RICHEDIT_COLOR_FOREGROUND	32824
背景色	IDC_AUTHENTICGUI_RICHEDIT_COLOR_BACKGROUND	32830
左寄せ	IDC_AUTHENTICGUI_RICHEDIT_ALIGN_LEFT	32818
	IDC_AUTHENTICGUI_RICHEDIT_ALIGN_CENTER	32819
右寄せ	IDC_AUTHENTICGUI_RICHEDIT_ALIGN_RIGHT	32820
行の追加	IDC_ROW_APPEND	32806
行の挿入	IDC_ROW_INSERT	32809
行の複製	IDC_ROW_DUPLICATE	32808
行を上へ移動	IDC_ROW_MOVE_UP	32811
行を下へ移動	IDC_ROW_MOVE_DOWN	32810

メニューアイテム	コマンド名	ID
行の削除	IDC_ROW_DELETE	32807
HTMLドキュメントの生成	IDC_PXF_GENERATE_HTML	34283
RTFドキュメントの生成	IDC_PXF_GENERATE_RTF	34284
PDFドキュメントの生成	IDC_PXF_GENERATE_PDF	34285
Word 2007+ドキュメントの生成	IDC_PXF_GENERATE_DOCX	34286
信頼できる場所	IDC_TRUSTED_LOCATIONS	34288

13.4.6.7 「表示」メニュー

「表示」メニューには以下のコマンドが含まれます:

メニューアイテム	コマンド名	ID
Authentic ビュー	IDC_VIEW_CONTENT	34177
ブラウザビュー	IDC_VIEW_BROWSER	34176
テキストビュー設定	IDC_TEXTVIEW_SETTINGS	34119

13.4.6.8 「ブラウザ」メニュー

「ブラウザ」メニューには以下のコマンドが含まれます:

メニューアイテム	コマンド名	ID
戻る	IDC_STEP_BACK	32958
進む	IDC_STEP_FORWARD	32957
中止	IDC_BROWSER_STOP	34047
最新の状態で更新	IDC_BROWSER_REFRESH	34046
最大	IDC_BROWSER_FONT_LARGEST	34041
大	IDC_BROWSER_FONT_LARGE	34040
中	IDC_BROWSER_FONT_MEDIUM	34042
小	IDC_BROWSER_FONT_SMALL	34043
最小	IDC_BROWSER_FONT_SMALLEST	34044

13.4.6.9 「ツール」メニュー

「ツール」メニューには以下のコマンドが含まれています。

メニューアイテム	コマンド名	ID
スペル..	IDC_SPELL_CHECK	34154
スペルオプション..	IDC_SPELL_OPTIONS	34155
スクリプトエディター...	ID_SCRIPTFORMEDITOR_EDIT_PROJECT	39666
無し	ID_SCRIPTFORMEDITOR_EXECUTE_MACRO_MENU_UPPDATE	39600
	IDC_TOOLS_ENTRY	34292
グローバルリソース	IDC_GLOBALRESOURCES	37401
	IDC_GLOBALRESOURCES_SUBMENUENTRY_1	37408
カスタマイズ..	IDC_APP_TOOLS_CUSTOMIZE	32959
オプション..	IDC_SETTINGS	34133
	ID_SCRIPTING_MACROITEMS	34249

13.4.6.10 「ヘルプ」メニュー

「ウィンドウ」メニューには以下のコマンドが含まれます。

メニューアイテム	コマンド名	ID
重ねて表示	ID_WINDOW_CASCADE	57650
上下に並べて表示	ID_WINDOW_TILE_HORZ	57651
左右に並べて表示	ID_WINDOW_TILE_VERT	57652
プロジェクトウィンドウ	IDC_PROJECT_WINDOW	34128
情報ウィンドウ	IDC_INFO_WINDOW	34085
入力ヘルパー	IDC_ENTRY_HELPERS	34062
出力ウィンドウ	IDC_OUTPUT_DIALOGBARS	34004
プロジェクトと入力ヘルパー	IDC_PROJECT_ENTRYHELPERS	34006
全てオン/オフ	IDC_ALL_BARS	34031

13.4.6.11 「ヘルプ」メニュー

「ヘルプ」メニューには以下のコマンドが含まれます:

メニューアイテム	コマンド名	ID
目次	IDC_HELP_CONTENTS	32966
インデックス	IDC_HELP_INDEX	32967
検索	IDC_HELP_SEARCH	32969
キーボードマップ	IDC_HELP_KEYMAPDLG	32968
ライセンスの登録	IDC_ACTIVATION	32970
注文フォーム	IDC_OPEN_ORDER_PAGE	32971
登録	IDC_REGISTRATION	32972
最新情報のチェック	IDC_CHECK_FOR_UPDATES	32973
XMLSpy 製品の比較	IDC_PRODUCT_COMPARISON	32955
サポートセンター	IDC_OPEN_SUPPORT_PAGE	32961
Web のFAQ	IDC_SHOW_FAQ	32962
コンポーネントや無料ツールのダウンロード	IDC_OPEN_COMPONENTS_PAGE	32963
インターネット上のAuthentic	IDC_OPEN_XML_SPY_HOME	32964
Authentic トレーニング	ID_HELP_XMLSPYTRAINING	32965
Authentic について	ID_APP_ABOUT	57664

13.4.7 オブジェクトレファレンス

オブジェクト:

[Authentic DesktopCommand](#)
[Authentic DesktopCommands](#)
[AuthenticDesktopControl](#)
[AuthenticDesktopControlDocument](#)
[AuthenticDesktopControlPlaceHolder](#)

標準のAuthentic Desktop 機能へのアクセスを与えるには、**Authentic Desktop** 自動化インターフェイスのオブジェクトにもアクセスすることができます。詳細に関しては、次を参照してください: [AuthenticDesktopControl.Application](#), [AuthenticDesktopControlDocument.Document](#) と [AuthenticDesktopControlPlaceHolder.Project](#)。

13.4.7.1 Authentic DesktopCommand

プロパティ:

[ID](#)

[Label](#)

[Name](#)

[IsSeparator](#)

[ToolTip](#)

[StatusText](#)

[Accelerator](#)

[SubCommands](#)

詳細:

コマンドオブジェクトは次のオプションであることができます: 実行可能なコマンド、コマンドコンテナ(例えば、メニュー、サブメニュー、またはツールバー)、または、メニューセレータ。現在のCommand オブジェクト内に保管される情報の種類を決定するには、ID、IsSeparator、とSubCommands プロパティをクエリしてください。

コマンドオブジェクト	使用可能な場合...
実行可能なコマンド	<ul style="list-style-type: none"> • ID はゼロ以外大きい場合 • IsSeparator がfalse の場合 • SubCommands が空白の場合
コマンドコンテナ	<ul style="list-style-type: none"> • ID はゼロの場合 • IsSeparator がtrue の場合 • SubCommands がCommand オブジェクトを含む場合
セレータ	<ul style="list-style-type: none"> • ID はゼロの場合 • IsSeparator がtrue の場合

13.4.7.1.1 Accelerator

プロパティ: Accelerator をstring として

詳細:

コマンドのために定義されたアクセラータキーを返します。コマンドにアクセラータキーが割り当てられていない場合、このプロパティは、空の文字列を返します。アクセラータキーの文字列の表示は次のフォーマットを取ります:

[ALT+] [CTRL+] [SHIFT+] key

Windows プラットフォームSDK 関数 GetKeyNameText を使用して key が変換された箇所。

13.4.7.1.2 ID

プロパティ: ID をlong として

詳細:

このプロパティは、コマドの一意の識別子を取得します。コマドのID は [Exec](#) を使用して コマドを実行するため、または [QueryStatus](#) を使用してステータスをクエリするため必須です。コマドが他のコマド（例えば、トップレベルのメニュー）のためのエネター、または セレクターの場合、ID は0 です。

13.4.7.1.3 IsSeparator

プロパティ: IsSeparator を [boolean](#) として

詳細:

コマドオブジェクトがメニューセレクターの場合、の場合、プロパティは true を返します。それ以外の場合は、false を返します。次も参照してください: [コマド](#)。

13.4.7.1.4 Label

プロパティ: Label を [string](#) として

詳細:

このプロパティは、Authentic Desktop のグラフィカルなユーザーインターフェイス内に表示されるとおりのコマドのテキストを取得します。コマドがセレクターの場合、「ラベル」は、空の文字列です。このプロパティは、関連したGUI テキストを持たない、一部のツールバーコマドのために空の文字列を返します。

13.4.7.1.5 Name

プロパティ: Name を [string](#) として

詳細:

このプロパティは一意のコマド名を取得します。使用できる箇所、この値はコマドのアイコンファイルを取得するために使用できます。使用することのできるアイコンファイルは、使用中のAuthentic Desktop インストールのフォルダー `<ApplicationFolder>\Examples\ActiveX\Images` 内にあります。

13.4.7.1.6 StatusText

プロパティ: Label を [string](#) として

詳細:

コマドが選択されると、ステータステキストは、Authentic Desktop のステータスバー内で表示されるテキストです。他のコマドのセレクターまたは、コンテナオではないコマドオブジェクトに対してのみ適用することができます。それ以外の場合、プロパティは空の文字列です。

13.4.7.1.7 SubCommands

プロパティ: SubCommands を [Commands](#) として

詳細:

SubCommands プロパティは、現在のコマドのサブコマドである [Command](#) オブジェクトのコレクションを取得します。プロパティは、他のコマド(メニュー、サブメニュー、または ツールバー)のためのエネターであるコマドにのみ適用されます。このようなコンテナコマドは 0 に設定されたID を持ち、および、false に設定された IsSeparator プロパティを持ちます。

13.4.7.1.8 ToolTip

プロパティ: ToolTip を [string](#) として

詳細:

このプロパティは、各コマンドのためのヒントを表示するテキストを取得します。コマンドがツールのヒントのテキストを持たない場合、プロパティは空の文字列を返します。

13.4.7.2 Authentic DesktopCommands

プロパティ:

[Count](#)

[Item](#)

詳細:

AuthenticDesktopControl のコマンドレベルとID へのアクセスを取得するための [Command](#) オブジェクトのコレクションです。これらのコマンドは [Exec](#) メソッドを使用して実行することができます。また、ステータスを [QueryStatus](#) を使用してクエリすることができます。

13.4.7.2.1 Count

プロパティ: Count を [long](#) として

詳細:

このコレクションのレベル上の [Command](#) オブジェクトの数。

13.4.7.2.2 Item

プロパティ: Item (n を [long](#) として) を [Command](#) として

詳細:

このコレクション内のインデックス n を持つコマンドを取得します。インデックスは1 ベースです。

13.4.7.3 AuthenticDesktopControl

プロパティ:

[IntegrationLevel](#)

[Appearance](#)

[Application](#)

[BorderStyle](#)

[CommandsList](#)

[EnableUserPrompts](#)

[MainMenu](#)

[Toolbars](#)

メソッド:[Open](#)[Exec](#)[QueryStatus](#)**イベント:**[OnUpdateCmdUI](#)[OnOpenedOrFocused](#)[OnCloseEditingWindow](#)[OnFileChangedAlert](#)[OnContextChanged](#)[OnDocumentOpened](#)[OnValidationWindowUpdated](#)

このオブジェクトは完全な ActiveX コントロールで、Authentic Desktop ライブラリがアプリケーションレベルモードで使用されている場合、表示されます。

13.4.7.3.1 プロパティ

以下のプロパティが定義されています:

[IntegrationLevel](#)[EnableUserPrompts](#)[Appearance](#)[BorderStyle](#)

コマンドに関連したプロパティ:

[CommandsList](#)[MainMenu](#)[Toolbars](#)

AuthenticDesktopAPI へのアクセス

[Application](#)

13.4.7.3.1.1 Appearance

プロパティ: Appearance を **short** として

ディスパッチ ID: -520

詳細:

0 に対して等しくない値は、コントロールの周りでクライアントエッジを表示します。デフォルトの値は 0 です。

13.4.7.3.1.2 Application

プロパティ: Application を [Application](#) として

ディスパッチID: 1

詳細:

Application プロパティは、完全な Authentic Desktop 自動サーバーAPI の Application オブジェクトへのアクセスを与えます。プロパティ読み取り専用です

13.4.7.3.1.3 BorderStyle

プロパティ: BorderStyle を [short](#) として

ディスパッチID: -504

詳細:

1 の値は、薄いボーダーを持つコントロールを表示します。デフォルトの値は 0 です。

13.4.7.3.1.4 CommandsList

プロパティ: CommandList を [Commands](#) として (読み取り専用)

ディスパッチID: 1004

詳細:

このプロパティは、AuthenticDesktopControl を持つ定義された全てのコマンドのフラットリストを返します。メニュー構造に従い、コマンドを整理するには、[MainMenu](#) を使用します。ツールバーコマンドを取得するには、[Toolbars](#) を使用します。

```
public void GetAllAuthenticCommands()
{
    // Get all commands from the Authentic ActiveX control assigned to the current form
    AuthenticControlLib.XMLSpyCommands commands =
    this.axAuthenticDesktopControl1.CommandList;
    // Loop through all commands
    for (int i = 0; i < commands.Count; i++)
    {
        // Get each command by index and output it to the console
        AuthenticControlLib.XMLSpyCommand cmd =
        axAuthenticDesktopControl1.CommandList[i];
        Console.WriteLine("{0} {1} {2}", cmd.ID, cmd.Name, cmd.Label.Replace("&", ""));
    }
}
```

C# サンプル

13.4.7.3.1.5 EnableUserPrompts

プロパティ: EnableUserPrompts を **boolean** として

ディスパッチID: 1006

詳細:

このプロパティを `false` に設定すると、コントロール内のユーザープロンプトを無効化します。デフォルトの値は `true` です。

13.4.7.3.1.6 IntegrationLevel

プロパティ: IntegrationLevel を [ICActiveXIntegrationLevel](#) として

ディスパッチID: 1000

詳細:

The IntegrationLevel プロパティ determines コントロールのオペレーションモード。次も参照してください: [アプリケーションレベルの統合](#) と [ドキュメントレベルの統合](#)。

メモ AuthenticDesktopControl オブジェクトの作成後すぐに、このプロパティを設定することは重要です。

13.4.7.3.1.7 MainMenu

プロパティ: MainMenu を **Command** として(読み取り専用)

ディスパッチID: 1003

詳細:

このプロパティは、AuthenticDesktopControl メインメニュー内で使用することのできる構造の情報とコマンドを Command オブジェクトとして提供します。オブジェクトは、Authentic Desktop の使用することのできる全てのサブメニューを含んでいます(例「ファイル」、「編集」、「表示」など)。サブメニューオブジェクトにアクセスするには、MainMenu プロパティの SubCommands プロパティを使用します。各サブメニューは Command オブジェクトでもあります。各サブメニューのために、対応する子コマンドセレーターを取得するために SubCommands プロパティを使用して更に反復することができます(このテクニックが使用される可能性があります。例えば、アプリケーションメニューをプログラムの作成するなど)。メニューコマンドの一部は、SubCommands プロパティを持つ他のメニューコマンドのためのエレメンター(「親」として)の役割を果たします。全てのメニューコマンドの構造をプログラムの取得するには、再帰的な関数を作成する必要があります。

```
public void GetAuthenticMenus ()
{
    // Get the main menu from the Authentic ActiveX control assigned to the current
    // form
    AuthenticControlLib.XMLSpyCommand mainMenu =
    this.axAuthenticDesktopControl1.MainMenu;

    // Loop through entries of the main menu (e.g. File, Edit, etc.)
    for (int i = 0; i < mainMenu.SubCommands.Count; i++)
    {
```



```

        AuthenticControlLib.XMLSpyCommand menu = mainMenu.SubCommands[i];
        Console.WriteLine("{0} menu has {1} children items (including separators)",
            menu.Label.Replace("&", ""), menu.SubCommands.Count);
    }
}

```

C# サンプル

13.4.7.3.1.8 Toolbars

プロパティ: Toolbars を [Commands](#) として (読み取り専用)

ディスパッチID: 1005

詳細:

このプロパティは AuthenticDesktopControl ツールバーの構造の情報を Command オブジェクトとして与えます。Command オブジェクトには Authentic Desktop の使用することのできる全てのツールバーが含まれています。ツールバーにアクセスするには Toolbars プロパティの SubCommands プロパティを使用します。各ツールバーは Command オブジェクトでもあります。各ツールバーのためにコマンドを使用するために SubCommands プロパティを更に反復します (例えば、アプリケーションのツールバープログラムの作成のためにこのテクニックは使用される場合があります)。

```

public void GetAuthenticToolbars()
{
    // Get the application toolbars from the Authentic ActiveX control assigned to the
    // current form
    AuthenticControlLib.XMLSpyCommands toolbars =
    this.axAuthenticDesktopControll.Toolbars;

    // Iterate through all toolbars
    for (int i = 0; i < toolbars.Count; i++)
    {
        AuthenticControlLib.XMLSpyCommand toolbar = toolbars[i];
        Console.WriteLine();
        Console.WriteLine("The toolbar \"{0}\" has the following commands:",
            toolbar.Label);

        // Iterate through all commands of this toolbar
        for (int j = 0; j < toolbar.SubCommands.Count; j++)
        {
            AuthenticControlLib.XMLSpyCommand cmd = toolbar.SubCommands[j];
            // Output only command objects that are not separators
            if (!cmd.IsSeparator)
            {
                Console.WriteLine("{0}, {1}, {2}", cmd.ID, cmd.Name, cmd.Label.Replace("&",
                    ""));
            }
        }
    }
}

```

C# サンプル

13.4.7.3.2 メソッド

以下のメソッドが定義されます:

[Open](#)

[Exec](#)

[QueryStatus](#)

13.4.7.3.2.1 Exec

Method: Exec (nCmdID を `long` として を `boolean` として

ディスパッチ ID: 6

詳細:

このメソッドは、ID nCmdID を持つ Authentic Desktop コマンドを呼び出し、コマンドが実行可能な場合、メソッドは `true` を返します。使用可能な全てのコマンドのリストを取得するには、[CommandsList](#) を使用してください。コマンドの状態を取得するには、[QueryStatus](#) を使用してください。

13.4.7.3.2.2 Open

Method: Open (strFilePath を `string` として を `boolean` として

ディスパッチ ID: 5

詳細:

メソッドの結果は、引数 strFilePath 内で与えられる拡張子により異なります。ファイル拡張子が `.sps` の場合、新規ドキュメントが開かれます。ファイル拡張子が `.svp` の場合、対応するプロジェクトが開かれます。異なるファイル拡張子がメソッドに与えられると、コントロールはファイルを新規のコンポーネントとしてアクティブなドキュメントにロードします。

ドキュメントレベルの統合モード内でコントロールを使用する際、このメソッドを使用して、ドキュメントまたはプロジェクトをロードしないでください。代わりに [AuthenticDesktopControlDocument.Open](#) と [AuthenticDesktopControlPlaceHolder.OpenProject](#) を使用してください。

13.4.7.3.2.3 QueryStatus

Method: QueryStatus (nCmdID を `long` として を `long` として

ディスパッチ ID: 7

詳細:

QueryStatus は、nCmdID により指定されたコマンドのチェックされた/チェックが解除された/ステータスを返します。このステータスビットマスクとして返されます。

ビット	値	名前	意味
0	1	サポートされている	コマンドがサポートされている場合設定します。
1	2	有効化されている	コマンドが有効化されている場合設定します (実行可能)。

2 4 チェックされている コマンドがチェックされている場合設定します。

これは QueryStatus が0 コマンドを返すと、ID は、有効な Authentic Desktop コマンドとして認識されないことを意味します。QueryStatus が1 または5 の値を返すとコマンドは無効化されます。

13.4.7.3.3 Events

AuthenticDesktopControl ActiveX コントロールは次の接続ポイントイベントを提供します:

[OnUpdateCmdUI](#)

[OnOpenedOrFocused](#)

[OnCloseEditingWindow](#)

[OnFileChangedAlert](#)

[OnContextChanged](#)

[OnDocumentOpened](#)

[OnValidationWindowUpdated](#)

13.4.7.3.3.1 OnCloseEditingWindow

イベント: OnCloseEditingWindow (i_strFilePath をString として) をboolean として

ディスパッチID: 1002

詳細:

Authentic Desktop が既に開かれているドキュメントを閉じる場合、このイベントはトリガーされます。このイベントへの応答として、クライアントはi_strFilePath に関連するエディターウィンドウを閉じます。このイベントがtrue を返すと、クライアントがドキュメントを閉じたことを指します。特定の処理が必要とされておらず、AuthenticDesktopControl がエディターを閉じようと試み、関連したドキュメントコントロールを破棄する場合、クライアント は、false を返すことができます。

13.4.7.3.3.2 OnContextChanged

イベント: OnContextChanged (i_strContextName をString として、i_bActive をbool として) をbool として

ディスパッチID: 1004

詳細:

このイベントは、is not used in Authentic Desktop

13.4.7.3.3.3 OnDocumentOpened

イベント: OnDocumentOpened (objDocument をDocument として)

ディスパッチID: 1

詳細:

このイベントはドキュメントが開かれる都度トリガーされます。引数 `objDocument` は Authentic Desktop 自動化インターフェイスからの Document オブジェクトであり、ドキュメントの詳細をクエリするために使用することができ、または追加オペレーションを行います。ドキュメントレベルで統合される場合、イベント [AuthenticDesktopControl.Document.OnDocumentOpened](#) を代わりに使用することが奨励されます。

13.4.7.3.3.4 OnFileChangedAlert

イベント: `OnFileChangedAlert` (`i_strFilePath` を String として) を bool として

ディスパッチID: 1001

詳細:

`AuthenticDesktopControl` を使用してファイルがロードされると、このイベントはトリガーされます。他のアプリケーションによりハードディスク上で変更されます。イベントを処理すると、クライアントは `true` を返します。または、Authentic Desktop が通常の方法で処理すると `false` を返します。例えば、ユーザーに再ロードを促すなど。

13.4.7.3.3.5 OnLicenseProblem

イベント: `OnLicenseProblem` (`i_strLicenseProblemText` を String として)

ディスパッチID: 1005

詳細:

`AuthenticDesktopControl` がこのコントロールのための有効なライセンスを使用できないことを検知すると、このイベントはトリガーされます。制限されているユーザーライセンスの場合、コントロールが初期化されると発生する可能性があります。インテグレーターはこのイベントを介して、このコントロールの機能へのアクセスを無効化することができます。このイベントから戻ると、コントロールは機能へのアクセスをブロックします (例、コントロール内の空のウィンドウを表示し、リクエストに対してエラーを返すなど)。

13.4.7.3.3.6 OnOpenedOrFocused

イベント: `OnOpenedOrFocused` (`i_strFilePath` を String として、`i_bOpenWithThisControl` を bool として)

ディスパッチID: 1000

詳細:

アプリケーションレベルで統合する場合、このイベントはクライアントにドキュメントが開かれたこと、または Authentic Desktop によりアクティブに設定されたことを通知します。

ドキュメントレベルで統合する場合、このイベントは、クライアントにドキュメントウィンドウをファイル `i_strFilePath` で開くように命令します。ファイルが既に開かれている場合、対応するドキュメントウィンドウをアクティブなウィンドウとしてください。

`i_bOpenWithThisControl` が `true` の場合、内部アクセスが必要とされるため、ドキュメントは `AuthenticDesktopControl` により開かれる必要があります。それ以外の場合、ファイルは異なるエディターで開くことができます。

13.4.7.3.3.7 OnToolWindowUpdated

イベント: OnToolWindowUpdated (pToolWnd を long として)

ディスパッチID: 1006

詳細:

ツールウィンドウが更新されると、このイベントはトリガーされます。

13.4.7.3.3.8 OnUpdateCmdUI

イベント: OnUpdateCmdUI ()

ディスパッチID: 1003

詳細:

頻繁に呼び出され、[AuthenticDesktopControl.QueryStatus](#) を使用して Authentic Desktop コマンドの状況をチェックする良い機会をインテグレーターに与えます。このコールバック内で long オペーションを実行しないでください。

13.4.7.3.3.9 OnValidationWindowUpdated

イベント: OnValidationWindowUpdated ()

ディスパッチID: 3

詳細:

新しい情報と共に検証出力ウィンドウが更新されると、このイベントはトリガーされます。

13.4.7.4 AuthenticDesktopControlDocument

プロパティ:

[Appearance](#)

[BorderStyle](#)

[Document](#)

[IsModified](#)

[Path](#)

[ReadOnly](#)

メソッド:

[Exec](#)

[New](#)

[Open](#)

[QueryStatus](#)

[Reload](#)

[Save](#)

[SaveAs](#)

イベント:

[OnDocumentOpened](#)

[OnDocumentClosed](#)

[OnModifiedFlagChanged](#)

[OnContextChanged](#)

[OnFileChangedAlert](#)

[OnActivate](#)

AuthenticDesktopControl がドキュメントレベルモード内で統合されている場合、各ドキュメントは、型 AuthenticDesktopControlDocument のオブジェクト内で表示されます。

AuthenticDesktopControlDocument には、1度1つのドキュメントのみが含まれていますが、場合によっては異なるファイルを表示するために使用される場合があります。

このオブジェクトは完全な ActiveX コントロールです。

13.4.7.4.1 プロパティ

以下のプロパティが定義されています。

[ReadOnly](#)

[IsModified](#)

[Path](#)

[Appearance](#)

[BorderStyle](#)

AuthenticDesktopAPI へのアクセス

[Document](#)

13.4.7.4.1.1 Appearance

プロパティ: Appearance を **short** として

ディスパッチID: -520

詳細:

0 に対して等しくない値は、ドキュメントコントロールの周りにクライアントエッジを表示します。デフォルトの値は 0 です。

13.4.7.4.1.2 BorderStyle

プロパティ: BorderStyle を **short** として

ディスパッチID: -504

詳細:

1 の値は、薄いボーダーを持つコントロールを表示します。デフォルトの値は 0 です。

13.4.7.4.1.3 Document

プロパティ: Document をドキュメントとして

ディスパッチID: 1

詳細:

Document プロパティは、Authentic Desktop 自動サーバーAPI のDocument オブジェクト へのアクセスを与えます。このインターフェイスは、コントロール内にコードされるドキュメントと使用することのできる追加機能を与えます。プロパティは読み取り専用です

13.4.7.4.1.4 IsModified

プロパティ: IsModified を **boolean** として(読み取り専用)

ディスパッチID: 1006

詳細:

ドキュメントコンテンツが最後に開かれてから変更された場合、IsModified は **true** を返します。再ロードし、オペレーションを保存します。それ以外の場合は、**false** を返します。

13.4.7.4.1.5 Path

プロパティ: Path を **string** として

ディスパッチID: 1005

詳細:

コントロール内にコードされるドキュメントのフル 宛名に設定、または、取得します。

13.4.7.4.1.6 ReadOnly

プロパティ: ReadOnly を **boolean** として

ディスパッチID: 1007

詳細:

このプロパティを使用すると、ドキュメントの読み取り専用モードをオンとオフに切り替えることができます。ReadOnly が **true** の場合、変更はできません。ご注意ください。

13.4.7.4.2 メソッド

以下のメソッドが定義されています:

ドキュメントハンドリング:

[New](#)

[Open](#)

[Reload](#)

[Save](#)

[SaveAs](#)

コマンドハンドリング:

[Exec](#)

[QueryStatus](#)

13.4.7.4.2.1 Exec

メソッド: Exec (nCmdID を **long** として を **boolean** として

ディスパッチID: 8

詳細:

このメソッドは、ID nCmdID を持つ Authentic Desktop コマンドを呼び出し、コマンドが実行可能な場合、メソッドは true を返します。現在アクティブなドキュメントがアプリケーションで使用できる場合のみ、このメソッドを呼び出すことができます。

メニュー構造に従いコマンドを整理するには、AuthenticDesktopControl の [MainMenu](#) プロパティを使用します。ツールバーコマンドを取得するには、AuthenticDesktopControl の [Toolbars](#) プロパティを使用します。

13.4.7.4.2.2 New

メソッド: New () を **boolean** として

ディスパッチID: 1000

詳細:

このメソッドはコントロール内で新規ドキュメントを初期化します。

13.4.7.4.2.3 Open

メソッド: Open (strFileName を **string** として を **boolean** として

ディスパッチID: 1001

詳細:

Open は、ファイル strFileName を新規のドキュメントとして、コントロールにロードします。

13.4.7.4.2.4 QueryStatus

メソッド: QueryStatus (nCmdID を long として) を long として
ディスパッチ ID: 9

詳細:

QueryStatus は、nCmdID により指定されたコマンドのチェックされた/チェックが解除されたステータスを返します。このステータスはビットマスクとして返されます。

ビット	値	名前	意味
0	1	サポートされている	コマンドがサポートされている場合設定します。
1	2	有効化されている	コマンドが有効化されている場合設定します (実行可能)。
2	4	チェックされている	コマンドがチェックされている場合設定します。

これは、QueryStatus が 0 コマンドを返すと、ID は、有効な Authentic Desktop コマンドとして認識されないことを意味します。QueryStatus が 1 または 5 の値を返すとコマンドは無効化されます。アプリケーション内で現在アクティブなドキュメントが存在する場合、クライアントはドキュメントコントロールの QueryStatus メソッドを呼び出します。

13.4.7.4.2.5 Reload

メソッド: Reload () を boolean として

ディスパッチ ID: 1002

詳細:

Reload ファイルシステムからドキュメントコンテンツを更新する。

13.4.7.4.2.6 Save

メソッド: Save () を boolean として

ディスパッチ ID: 1003

詳細:

Save は、現在のドキュメントをロケーション [Path](#) に保存する。

13.4.7.4.2.7 SaveAs

メソッド: SaveAs (strFileName を string として) を boolean として

ディスパッチ ID: 1004

詳細:

SaveAs は [Path](#) を `strFileName` に設定し、ドキュメントをこの場所に保存します。

13.4.7.4.3 Events

AuthenticDesktopControlDocumentActiveX コントロールは次の接続ポイントイベントを提供します:

[OnDocumentOpened](#)

[OnDocumentClosed](#)

[OnModifiedFlagChanged](#)

[OnContextChanged](#)

[OnFileChangedAlert](#)

[OnActivate](#)

[OnSetEditorTitle](#)

13.4.7.4.3.1 OnActivate

イベント: `OnActivate ()`

ディスパッチID: 1005

詳細:

ドキュメントコントロールが有効化されると、フォーカスのある、ユーザー入力として使用することができるこのイベントがトリガーされます。

13.4.7.4.3.2 OnContextChanged

イベント: `OnContextChanged (i_strContextName をString として、i_bActive をbool として) をbool として`

ディスパッチID: 1004

詳細: なし

13.4.7.4.3.3 OnDocumentClosed

イベント: `OnDocumentClosed (objDocument をDocument として)`

ディスパッチID: 1001

詳細:

このイベントはドキュメントがこのコントロールロードされると閉じられます。引数 `objDocument` は、Authentic Desktop 自動化インターフェイスからの Document オブジェクトです、で慎重に使用してください。

13.4.7.4.3.4 OnDocumentOpened

イベント: OnDocumentOpened (objDocument を Document として)

ディスパッチID: 1000

詳細:

このイベントは、このコントロール内でドキュメントが開かれるとトリガーされます。引数 objDocument は、Authentic Desktop 自動化インターフェイスからの Document オブジェクトです。ドキュメントの詳細をクエリするために使用することができ、または、追加オペレーションを行います。

13.4.7.4.3.5 OnDocumentSaveAs

イベント: OnContextDocumentSaveAs (i_strFileName を String として)

ディスパッチID: 1007

詳細:

このドキュメントが新しい名前でも内部で保存される場合このイベントがトリガーされます。

13.4.7.4.3.6 OnFileChangedAlert

イベント: OnFileChangedAlert () を bool として

ディスパッチID: 1003

詳細:

このドキュメントにロードされると、このイベントがトリガーされます。コントロールは、ハードディスク上で他のアプリケーションに変更されます。イベントを処理すると、クライアントは true を返します。または、Authentic Desktop が通常の方法で処理すると false を返します。例えば、ユーザーに再ロードを促すなど。

13.4.7.4.3.7 OnModifiedFlagChanged

イベント: OnModifiedFlagChanged (i_bIsModified を boolean として)

ディスパッチID: 1002

詳細:

このイベントはドキュメントの状態を、変更された、および、変更されていない状態に切り替えるとトリガーされます。ドキュメントコンテンツが元のコンテンツと異なる場合、パラメーター i_bIsModified is true それ以外の場合は、false を返します。

13.4.7.4.3.8 OnSetEditorTitle

イベント: OnSetEditorTitle ()

ディスパッチID: 1006

詳細:

このイベントは、含まれているドキュメントが内部で名前を与えられると挙げられます。

13.4.7.5 AuthenticDesktopControlPlaceholder

すべてのプロジェクトプレースホルダーウィンドウに使用することのできるプロパティ:

[PlaceholderWindowID](#)

プロジェクトプレースホルダーウィンドウのためのプロパティ:

[Project](#)

プロジェクトプレースホルダーウィンドウのためのメソッド:

[OpenProject](#)

[CloseProject](#)

AuthenticDesktopControlPlaceholder コントロールは、概要、ライブラリまたはプロジェクトウィンドウなどの追加の Authentic Desktop ウィンドウを表示するために使用されます。他の ActiveX コントロールと同様に使用することができ、クライアントアプリケーション内に配置することができます。

13.4.7.5.1 プロパティ

以下のプロパティが定義されています:

[PlaceholderWindowID](#)

AuthenticDesktopAPI へのアクセス

[Project](#)

13.4.7.5.1.1 Label

プロパティ: Label を **String** (読み取り専用)として

ディスパッチID: 1001

詳細:

このプロパティはプレースホルダーのタイトルへのアクセスを与えます。プロパティは読み取り専用です。

13.4.7.5.1.2 PlaceholderWindowID

プロパティ: PlaceholderWindowID を [AuthenticDesktopControlPlaceholderWindow](#) として

ディスパッチID: 1

詳細:

このプロパティを使用すると、コントロールのクライアントエリア内で表示される Authentic Desktop ウィンドウが通知されます。PlaceholderWindowID は [AuthenticDesktopControlPlaceholderWindow](#) 列挙の有効な値に設定することができます。コントロールはすぐに状態を変更し、すぐに新規の Authentic Desktop ウィンドウを表示します。

13.4.7.5.1.3 Project

プロパティ: Project Project として(読み取り専用)

ディスパッチID: 2

詳細:

Project プロパティは、Authentic Desktop 自動サーバーAPI のProject オブジェクトへのアクセスを与えます。このインターフェイスは、コントロールにロードされるプロジェクトに使用される追加機能を与えます。プレースホルダウィンドウが Authentic DesktopXProjectWindow (=3) の値を持つ [PlaceholderWindowID](#) の場合のみ、プロパティは有効なプロジェクトインターフェイスを返します。プロパティは読み取り専用です

13.4.7.5.2 メソッド

以下のメソッドが定義されています:

[OpenProject](#)

[CloseProject](#)

13.4.7.5.2.1 OpenProject

メソッド: OpenProject (strFileName をstring として をboolean として

ディスパッチID: 3

詳細:

OpenProject はstrFileName を新規のプロジェクトとしてコントロールにロードします。プレースホルダウィンドウが XMLSpyXProjectWindow (=3) と異なる [PlaceholderWindowID](#) を持つ場合、メソッドは失敗します。

13.4.7.5.2.2 CloseProject

メソッド: CloseProject ()

ディスパッチID: 4

詳細:

CloseProject はプロジェクトにロードされたコントロールを閉じます。プレースホルダウィンドウが XMLSpyXProjectWindow (=3) と異なる [PlaceholderWindowID](#) を持つ場合、メソッドは失敗します。

13.4.7.5.3 イベント

AuthenticDesktopControlPlaceholder ActiveX コントロールは次の接続ポイントイベントを提供します:

[OnModifiedFlagChanged](#)

13.4.7.5.3.1 OnModifiedFlagChanged

イベント: OnModifiedFlagChanged (i_bIsModified を **boolean** として)

ディスパッチID: 1

詳細:

このイベントは Authentic DesktopXProjectWindow (=3) の [PlaceholderWindowID](#) を持つプレースホルダコントロールのためのみトリガーされます。プロジェクトのコンテンツが変更された、または変更されていない状態の間で変更があった場合、イベントは実行されます。プロジェクトコンテンツが元のコンテンツと異なる場合、パラメータ *i_bIsModified* は *true* を返し、それ以外の場合は *false* を返します。

13.4.7.5.3.2 OnSetLabel

イベント: OnSetLabel (i_strNewLabel を **string** として)

ディスパッチID: 1000

詳細:

プレースホルダウィンドウのタイトルが変更されると、挙げられます。

13.4.7.6 列挙

以下の列挙が定義されています:

[ICActiveXIntegrationLevel](#)

AuthenticDesktopControlPlaceholderWindow

13.4.7.6.1 ICActiveXIntegrationLevel

AuthenticDesktopControl の [IntegrationLevel](#) プロパティのための可能な値。

ICActiveXIntegrationOnApplicationLevel = 0
ICActiveXIntegrationOnDocumentLevel = 1

13.4.7.6.2 AuthenticDesktopControlPlaceholderWindow

この列挙にはサポートされる追加 Authentic Desktop ウィンドウのリストが含まれています。

AuthenticDesktopControlNoToolWnd	=	-1
AuthenticDesktopControlEntryHelperTopToolWnd	=	0
AuthenticDesktopControlEntryHelperMiddleToolWnd	=	1
AuthenticDesktopControlEntryHelperBottomToolWnd	=	2
AuthenticDesktopControlValidatorOutputToolWnd	=	3
AuthenticDesktopControlProjectWindowToolWnd	=	4
AuthenticDesktopControlInfoToolWnd	=	18

14 付録

以下の付録には、Authentic Desktop に関する技術的な情報や、ライセンスに関する重要な情報が取められています。各付録には以下のようにサブセクションが取められています:

技術情報

- OS ならびにメモリの必要条件
- Altova XML パーサー
- Altova XSLT と XQuery エンジン
- Unicode のサポート
- インターネットへの接続

ライセンス情報

- 電子的なソフトウェアの配布
- 知的財産権と著作権
- エンドユーザーライセンス使用許諾契約書

14.1 技術データ

このセクションは、ソフトウェアの技術面に関する役に立つ背景情報を含んでいます。以下のように整理されています：
[OS とメモリ要件](#)[Altova XML バリデーター](#)[Altova XSLT と XQuery エンジン](#)[Unicode のサポート](#)[インターネットの使用](#)

14.1.1 OS とメモリ要件

オペレーティングシステム

Altova ソフトウェアアプリケーションは、以下のプラットフォームで使用されます：
プラットフォーム更新済みの Windows 7 SP1、Windows 8、Windows 10 プラットフォーム更新済みの Windows Server 2008 R2 SP1 および以降 **メモリ**

ソフトウェアが C++ で書かれているため、Java Runtime Environment をダウンロードする必要はなく、Java ベースのアプリケーションに比べ、通常少ないメモリを必要とします。しかしながら、各ドキュメントは完全に解析するため、また、ビューと編集の速度を向上するためにメモリがダウンロードされます。メモリの要件は、ドキュメントのサイズを増やします。

メモリ要件は、制限のない「元に戻す」履歴にも影響を受けます。大きなドキュメントの大きなセクションの切り取り、貼り付け操作を繰り返していると、使用できるメモリがすぐに消費されます。

14.1.2 Altova XML バリデーター

XML ドキュメントを開くと、アプリケーションは、内蔵の XML バリデーターを使用して、指定されている場合、スキーマに対して整形形式をチェック、ツリーとインフォセットを作成します。XML バリデーターは、ドキュメントを編集する際にインテリジェントな編集ヘルプを提供し、発生する検証エラーを表示するために使用されます。

内蔵の XML バリデーターは、W3C の XML スキーマ 1.0 と 1.1 仕様の最終勧告を実装しています。New developments recommended by the W3C XML スキーマ作業グループによる勧告される新しい項目は、XML バリデーターに継続的に組み込まれるため、Altova 製品は最高水準の開発環境を届けることができます。

14.1.3 Altova XSLT と XQuery エンジン

Altova 製品は、Altova XSLT 1.0、2.0、および 3.0 エンジンと Altova XQuery 1.0 と 3.1 エンジンを使用しています。各エンジンのためのドキュメントと実装に固有の振る舞いに関しては、製品で使用されるエンジンの各ドキュメントの付属書（エンジン情報）で確認することができます。

メモ Altova MapForce は、XSLT 1.0、2.0 および XQuery 1.0 エンジンを使用したコードを生成します。

14.1.4 Unicode のサポート

Altova XML 製品は、Unicode を完全にサポートします。XML ドキュメントを編集する際は、ドキュメント内で使用されている Unicode 文字をサポートするフォントが必要です。

フォントの多くは、Unicode 範囲全体の特定のサブセットを含む場合があります。このため、通常は対応する表記システムをターゲットとします。テキストの一部が、文字化けして表示された場合、理由としては、選択されたフォントが必要とする字形を含まない場合があげられます。です

から、特に異なる言語、または異なる言語システムのXMLドキュメントを編集する場合、範囲全体をカバーするフォントを使用することが役に立ちます。典型的なUnicodeフォントは、Windows PCのArial Unicode MSで確認することができます。

アプリケーションフォルダーの/Examplesフォルダー内で、異なる言語システムで表記された次の文章を含むUnicodeUTF-8.htmlというXHTMLファイルを確認してください。

When the world wants to talk, it speaks Unicode Wenn die Welt miteinander spricht, spricht sie Unicode
世界的に話すなら、Unicodeです。XHTMLファイルを開くと、Unicodeの可能性を確認することができ、使用中のPCの使用することのできるフォントによりサポートされている表記システムが表示されます。

14.1.5 インターネットの使用

Altovaアプリケーションは、次の状況でインターネット接続を開始します。

- 登録ダイアログ(「ヘルプ | ソフトウェアのライセンス認証」)内の「評価キーコードをリクエスト」をクリックした場合、登録ダイアログボックス内の3つのフィールドが通常のhttp(ポート80)接続を使用し、サーバーに転送され、無料の評価キーが顧客に通常のSMTP電子メールを使用して送り返されます。
- Altova製品の一部では、インターネットからファイルを開くことができます(「ファイル | 開く | URLに切り替える」)。この場合、ドキュメントは、次のプロトコルメソッドと接続の1つを使用して取得されます: HTTP(通常、ポート80)、FTP(通常、ポート20/21)、HTTPS(通常、ポート443)。HTTPサーバーをポート8080で作動することもできます(URLダイアログ内で、サーバー名とコロンの後にポートを指定します)。
- XMLスキーマ、またはDTDを参照するXMLドキュメントと、URLにより指定されているドキュメントを開くと、参照されているスキーマドキュメントは、HTTP接続(ポート80)またはURLにより指定されている他のプロトコル上のポート2(参照)により抽出されます。XMLファイルが検証されている場合、スキーマドキュメントも抽出されます(オプションダイアログのファイルタブ内の「ツール | オプション」)。アプリケーションに命令している場合、ドキュメントが開かれると検証が自動的に行われる場合もあります。
- WSDLとSOAPを使用するAltovaアプリケーションでは、Webサービスを使用する接続は、WSDLドキュメントにより定義されています。
- XMLSpy内で、「電子メールで送信」コマンドを使用する場合、(「ファイル | 電子メールで送信」)現在選択されている範囲、またはファイルは、ユーザーのマシンにインストールされているMAPIコンプライアント電子メールプログラムにより送信されます。
- ソフトウェアの荒いライセンス認証とLiveUpdateの一部として、Altovaソフトウェア使用許諾書内で更に詳しい説明を確認することができます。

14.2 ライセンス情報

このセクションには以下の内容が含まれています:

- ソフトウェアの配布に関する情報
- ソフトウェアの使用に関する使用許諾契約書

本製品を使用する前に、上記の情報をよくお読みください。ソフトウェアのインストール時に上記のすべての条件に同意したとみなされ、お客様は上記の条件に拘束されることを同意したとみなされます。

Altova ライセンスの内容を確認するには、[Altova Web サイト](#) の [Altova 法的な情報のページ](#) に移動してください。

14.2.1 電子的なソフトウェアの配布

この製品は電子的なソフトウェアの配布により利用することが可能で、この配布方法により、以下のユニークなメリットがあります:

- 購入を決定する前に、無料でソフトウェアを試用することができます。(Note: Altova Mobile Together Designer に対してライセンスを無料で割り当てることができます。)
- Once ソフトウェアの購入を決定した際は、[Altova Web サイト](#) にて注文を行います。すぐにライセンス登録された製品の使用を開始することができます。
- オンラインにて注文を行うと、常に最新のソフトウェアをご利用いただけます。
- 製品パッケージには包括的なヘルプシステムが画面上に表示されます。最新バージョンのユーザーマニュアルは <https://www.altova.com/ja/> 上にあり (i) HTML フォーマットによる閲覧、ならびに(ii) PDF フォーマットのダウンロードと印刷に対応しております。

30 日間の評価期間

この製品をダウンロードした後は、最大で30日の間無料で製品の評価を行うことができます。20日間を超えた頃から、製品がライセンス登録されていないことがソフトウェアにより表示されます。このメッセージはアプリケーションが起動されるたびに表示され、30日間を超えてプログラムを使用するには、キーコードを含むライセンスファイルから提供される製品のライセンスを購入します。ライセンスファイルを製品のソフトウェアアクティベーションダイアログにアップロードして、製品をアンロックします。

<https://shop.altova.com/> でライセンスを購入することができます

組織内でソフトウェアの評価を行う

評価版のソフトウェアを組織内のネットワークにて配布した場合、またはインターネットに接続されていないコンピュータにてソフトウェアを使用する場合、どのような状態でも改変されていないことを条件に、セットアッププログラムでの配布を行うことが可能です。ソフトウェアインストーラーへアクセスした人は、例外なく30日間の評価ライセンスキーコードをリクエストして、試用期間が経過した後は、製品を使い続けるためライセンスの購入を行う必要があります。

14.2.2 ソフトウェアのアクティベーションとライセンスの計測

Altova のソフトウェアアクティベーションの一部として、ソフトウェアにより内部ネットワークまたはインターネットへの接続を行い、インストール時、登録時、Altova により使用されるライセンスサーバーの更新やライセンスの正当性を検証することで、ソフトウェアの不正な使用を防ぎ、顧客サービスを向上するため、ライセンスに関する情報を送信することがあります。アクティベーションにより、オペレーティングシステムやIP アドレス、日付/時刻、ソフトウェアのバージョン、コンピュータの名前などのライセンスに関する情報が、お使いのコンピュータと Altova ライセンスサーバー間でやり取りされます。

お使いの Altova 製品にはライセンス計測モジュールが内蔵されており、エンドユーザー使用許諾契約書の意図しない違反を防ぎます。お使いの製品はシングルユーザーまたはマルチユーザーとしてインストールされており、ライセンス計測モジュールにより、ライセンスされている数を超えたユーザーが同時に製品を使用することが無いことが保証されます。

このライセンス計測技術により、ローカルエリア接続 (LAN) において、別々のコンピュータ間で動作しているアプリケーションインスタンス間の通信が行われます。

シングルライセンス

ライセンス計測プロセスの一部としてアプリケーションが起動すると、ソフトウェアにより短いデータグラムがブロードキャストにより送信され、同一のネットワークセグメントにある他のコンピュータにてプログラムが動作してはいないかのチェックが行われます。応答が無い場合は、アプリケーションの他インスタンスから送信される信号に反応するため、ポートが開かれます。

マルチユーザーライセンス

同一の LAN 内に2つ以上のアプリケーションインスタンスが使用される場合、スタートアップ時に、これらインスタンス間において通信が行われます。これらのインスタンス間にてキーコードのやりとりが行われ、購入された数のライセンスを超えてインスタンスが起動しないよう保証することができます。このようなライセンス計測システムは UNIX やデータベース開発ツールにて広く使用されているもので、Altova ユーザーはリーズナブルな価格にて同時使用マルチユーザーライセンスを購入することができます。

弊社はアプリケーションのデザインも行っており、少数の小さなネットワークパケットを送信することで、ネットワークに対する負荷を最小限に抑えておきます。Altova により使用される 2799 番 TCP/IP ポートは IANA により公式登録されており(詳細は ([IANA Web サイト \(http://www.iana.org/\)](http://www.iana.org/) を参照ください)、弊社のライセンス計測モジュールは既にテストされたものです。

ファイアウォールを使用している場合、2799 番ポートにて Altova 製品が動作しているコンピュータ同士が通信しているのが気づかれるかも知れません。その他の手段によりライセンス使用許諾書の内容が守られることを保証できる限り、組織間の異なるグループにおいてこのようなトラフィックをブロックすることは勿論可能です。

証明書に関するメモ

Altova アプリケーションは HTTPS を介して Altova ライセンスサーバー (link.altova.com) に通信します。この通信のために Altova は登録済みの SSL 証明書を使用します。(例えば、社内 IT 部署または外部エージェントによりこの証明書が置き換えられている場合、使用中の Altova アプリケーションは接続が安全でないことを警告します。Altova アプリケーションを開始するため代替の証明書を使用することができますが、自己責任で行ってください。安全でない接続の警告メッセージが表示されると、証明書の発行元を確認して (Altova 証明書の代替証明書の使用の継続または停止を決定することができる社内 IT チームと相談してください)。

(例えば、クライアントマシンへのまたはクライアントマシンへの通信を監視するため) 自身の証明書の使用が必要な場合 Altova の無料管理ソフトウェアである [Altova LicenseServer](#) を使用中のネットワークにインストールすることが奨励されます。このセットアップでは Altova LicenseServer は Altova との通信のために Altova 証明書の使用を許可しつつクライアントマシンが所属機関の証明書の使用を継続することができます。

14.2.3 Authentic のための Altova エンドユーザー使用許諾契約書

- Authentic のための Altova エンドユーザー使用許諾契約書: <https://www.altova.com/ja/legal/authentic-eula>
- Altova プライバシーポリシー: <https://www.altova.com/ja/privacy>

インデックス

▪

.NET,

- Authentic Desktop スタンドアロンとの違い, 243
- Authentic Desktop との統合, 241

A

ActiveX,

- アプリケーションレベルの統合, 559
- ドキュメントレベルの統合, 561
- 統合の必要条件, 556

ActiveX controls,

- support, 278

ActiveX コントロール,

- Visual Studio ツールボックスへの追加, 557

Altova XML パーサー,

- について, 609

Altova エンジン,

- Altova 製品内で, 609

Altova グローバルリソース,

- グローバルリソース, 87
- グローバルリソース を参照する, 87

Altova サポート, 21**Altova ソフトウェアの注文, 235****Altova ソフトウェアの登録, 235****Altova 製品, 21****API,**

- documentation, 291
- overview, 292

Application,

- ActiveDocument, 323
- AddMacroMenuItem, 323
- AddXSLT_XQParameter, 324
- Application, 324
- ClearMacroMenu, 324
- CurrentProject, 325
- Dialogs, 325
- Documents, 325
- GetDatabaseImportElementList, 326

GetDatabaseSettings, 327

GetDatabaseTables, 327

GetExportSettings, 328

GetTextImportElementList, 328

GetTextImportExportSettings, 329

GetXSLT_XQParameterCount, 329

GetXSLT_XQParameterName, 329

GetXSLT_XQParameterXPath, 330

ImportFromDatabase, 330

ImportFromSchema, 331

ImportFromText, 332

ImportFromWord, 332

NewProject, 333

OnBeforeOpenDocument, 321

OnBeforeOpenProject, 321

OnDocumentOpened, 322

OnProjectOpened, 322

OpenProject, 334

Parent, 334

Quit, 334

ReloadSettings, 335

RemoveXSLT_XQParameter, 335

RunMacro, 335

ScriptingEnvironment, 336

ShowApplication, 336

ShowForm, 337

URLDelete, 338

URLMakeDirectory, 338

WarningNumber, 338

WarningText, 339

ATL,

- plug-in sample files, 281

Authentic Desktop,

- ユーザーマニュアル, 10
- 統合, 556

Authentic Desktop 統合,

- のサンプル, 568, 569

Authentic Desktopコマンド,

- AuthenticDesktopControl, 587
- AuthenticDesktopControl 内, 589

Authentic Integration Package, 242, 245**Authentic View, 61, 187**

- CDATA セクション, 33
- DB データの編集, 186
- GUI の概要, 39
- PFX ファイルから出カドキュメントを生成, 192
- SPS テーブル, 60
- XML テーブル, 61

- Authentic View, 61, 187
 - XML テーブルアイコン, 65
 - XML テーブルの使用, 61
 - XML ドキュメントの印刷, 37
 - XML ドキュメントを開く, 25
 - XML/テキストとして貼り付け, 51
 - エンティティ, 33
 - エンティティの挿入, 36
 - コンテキストメニュー, 27
 - ツールバーアイコン, 41
 - データエントリデバイス, 33
 - データの挿入, 33
 - テーブル (SPS と XML), 60
 - テーブル内, 30
 - テキストの書式設定, 41
 - テキストの範囲, 51
 - ドキュメントの表示, 44
 - ノードの削除, 30
 - ノードの挿入, 30
 - ノードの追加, 30
 - マークアップタグの表示, 27
 - マークアップの表示, 41, 44
 - メインウィンドウ, 44
 - 主要な機能の使用方法, 53
 - 新たな XML ファイルを開く, 185
 - 切り替え, 194
 - 属性値の挿入, 35
 - 特殊文字, 33
 - 入力ヘルパー, 27, 47
 - 要素のクリア, 30
 - 要素の適用, 30
- Authentic View テンプレート, 25
- Authentic スクリプト,
 - セキュリティ設定, 193
 - 信頼された場所, 193
- Authentic スクリプトの信頼された場所, 193
- Authentic メニュー, 185
 - マークアップの表示, 41
 - 動的なテーブルの編集, 41
- AuthenticDataTransfer,
 - dropEffect, 341
 - getData, 341
 - ownDrag, 342
 - type, 342
- AuthenticDesktopControl, 589
 - C# を使用した統合, 564
 - HTML を使用した統合, 569
 - アプリケーションレベルでのサンプル 統合, 568, 569
 - オブジェクトレファレンス, 586
 - ドキュメント, 556
 - ドキュメントレベルでの統合のサンプルサンプル, 564
- AuthenticDesktopControl ドキュメント, 597
- AuthenticDesktopControl プレースホルダ, 604
- AuthenticRange,
 - AppendRow, 347
 - Application, 348
 - CanPerformAction, 348
 - CanPerformActionWith, 348
 - Close, 349
 - CollapsToBegin, 349
 - CollapsToEnd, 349
 - Copy, 349
 - Cut, 350
 - Delete, 350
 - DeleteRow, 350
 - DuplicateRow, 351
 - ExpandTo, 352
 - FirstTextPosition, 352
 - FirstXMLData, 353
 - FirstXMLDataOffset, 354
 - GetElementAttributeNames, 355
 - GetElementAttributeValue, 355
 - GetElementHierarchy, 356
 - GetEntityNames, 356
 - Goto, 357
 - GotoNext, 357
 - GotoNextCursorPosition, 358
 - GotoPrevious, 358
 - GotoPreviousCursorPosition, 359
 - HasElementAttribute, 359
 - InsertEntity, 359
 - InsertRow, 360
 - IsCopyEnabled, 361
 - IsCutEnabled, 361
 - IsDeleteEnabled, 361
 - IsEmpty, 361
 - IsEqual, 362
 - IsFirstRow, 362
 - IsInDynamicTable, 362
 - IsLastRow, 362
 - IsPasteEnabled, 363
 - IsTextStateApplied, 363
 - LastTextPosition, 363
 - LastXMLData, 364
 - LastXMLDataOffset, 365
 - MoveBegin, 366

AuthenticRange,

- MoveEnd, 366
- MoveRowDown, 367
- MoveRowUp, 367
- Parent, 367
- Paste, 367
- PerformAction, 368
- Select, 369
- SelectNext, 369
- SelectPrevious, 370
- SetElementAttributeValue, 371
- SetFromRange, 372
- Text, 372

Authentic View, 388

- Application, 381
- AsXMLString, 381
- DocumentBegin, 382
- DocumentEnd, 383
- Event, 384
- Goto, 385
- IsRedoEnabled, 385
- IsUndoEnabled, 386
- MarkupVisibility, 386
- OnBeforeCopy, 373
- OnBeforeCut, 374
- OnBeforeDelete, 374
- OnBeforeDrop, 375
- OnBeforePaste, 375
- OnDragOver, 376
- OnKeyboardEvent, 377
- OnMouseEvent, 378
- OnSelectionChanged, 378
- Parent, 386
- Print, 386
- Redo, 387
- Selection, 387
- Undo, 388
- WholeDocument, 389
- XMLDataRoot, 389

C**C#,**

- Authentic Desktop の統合, 564

CDATA セクション,

- Authentic View 内に挿入, 56

Class ID,

- Authentic Desktop 統合, 568

CodeGeneratorDlg,

- Application, 390
- CPPSettings_DOMType, 390
- CPPSettings_LibraryType, 392
- CPPSettings_UseMFC, 392
- CSharpSettings_ProjectType, 392
- OutputPath, 393
- OutputPathDialogAction, 393
- OutputResultDialogAction, 393
- Parent, 393
- ProgrammingLanguage, 394
- PropertySheetDialogAction, 394
- TemplateFileName, 394

COM API,

- in Scripting Editor, 261

COM-API,

- documentation, 291

Configure,

- XMLSPY UI, 278

CR&LF, 220**D****DatabaseConnection,**

- ADOConnection, 396
- AsAttributes, 396
- CreateMissingTables, 397
- CreateNew, 397
- DatabaseKind, 397
- ExcludeKeys, 398
- File, 398
- IncludeEmptyElements, 399
- NumberDateTimeFormat, 399
- ODBCConnection, 399
- SQLSelect, 400
- TextFieldLen, 401

DB, 69, 70

- Authentic View 内のクエリ, 69
- Authentic View 内の編集, 69, 74
- Authentic View内のテーブルのナビゲート, 69
- DB クエリ内のパラメーター, 70
- クエリの作成, 70
- 内の Authentic Viewの表示のフィルター, 70

Delete,

- Delete,**
 - Application.URLDelete, 338
- Dialogs,**
 - Application, 401
 - CodeGeneratorDlg, 402
 - DTDSchemaGeneratorDlg, 403
 - FileSelectionDlg, 402
 - GenerateSampleXMLDlg, 403
 - Parent, 402
 - SchemaDocumentationDlg, 403
- directories,**
 - creating with Application.URLMakeDirectory, 338
- Document, 416**
 - Application, 409
 - AssignDTD, 409
 - AssignSchema, 409
 - AssignXSL, 410
 - AssignXSLFO, 410
 - AuthenticView, 410
 - Close, 411
 - ConvertDTDOrSchema, 411
 - CreateChild, 413
 - CreateSchemaDiagram, 414
 - CurrentViewMode, 414
 - DataRoot, 415
 - DocEditView, 415
 - Encoding, 415
 - EndChanges, 416
 - ExecuteXQuery, 416
 - ExportToDatabase, 417
 - ExportToText, 417
 - FullName, 419
 - GenerateDTDOrSchema, 419, 420
 - GenerateProgramCode, 420
 - GenerateSampleXML, 420
 - GenerateSchemaDocumentation, 421
 - GetExportElementList, 423
 - GetPathName, 423
 - GridView, 424
 - IsModified, 424
 - IsValid, 424
 - IsWellFormed, 426
 - Name, 427
 - OnBeforeCloseDocument, 407
 - OnBeforeSaveDocument, 406
 - OnBeforeValidate, 407
 - OnCloseDocument, 408
 - OnViewActivation, 408
 - Path, 427
 - RootElement, 427
 - Save, 428
 - SaveAs, 428
 - Saved, 428
 - SaveInString, 429
 - SaveToURL, 429
 - SetActiveDocument, 429
 - SetEncoding, 430
 - SetExternalsValid, 430
 - SetPathName, 431
 - StartChanges, 431
 - SwitchViewMode, 431
 - Title, 432
 - TransformXSL, 432
 - TransformXSLFO, 433
 - UpdateViews, 433
 - UpdateXMLData, 434
 - XQuery, 416
- Documents,**
 - Count, 435
 - Item, 435
 - NewAuthenticFile, 435
 - NewFile, 436
 - NewFileFromText, 436
 - OpenAuthenticFile, 436
 - OpenFile, 437
 - OpenURL, 437
 - OpenURLDialog, 438
- DTD, 220, 222**
- DTDSchemaGeneratorDlg,**
 - Application, 439
 - AttributeTypeDefinition, 439
 - DTDSchemaFormat, 439
 - FrequentElements, 440
 - GlobalAttributes, 440
 - MaxEnumLength, 440
 - MergeAllEqualNamed, 440
 - OnlyStringEnums, 441
 - OutputPath, 441
 - OutputPathDialogAction, 441
 - Parent, 442
 - ResolveEntities, 442
 - TypeDetection, 442
 - ValueList, 442

E

Eclipse のための Authentic プラグイン,
インストール, 245

Eclipse プラットフォーム,
と Authentic Desktop, 244
と Authentic Integration Package, 245
内の Authentic パースペクティブ, 247

Eclipse 内の Authentic パースペクティブ, 247

ElementList,
Count, 443
Item, 443
RemoveElement, 443

ElementListItem,
ElementKind, 444
FieldCount, 444
Name, 444
RecordCount, 444

Enter キー, 83
Enter キー参照, 83
使用の効果, 83

Enumerations,
SPYAttributeTypeDefinition, 542
SPYAuthenticActions, 542
SPYAuthenticDocumentPosition, 543
SpyAuthenticElementActions, 543
SPYAuthenticElementKind, 543
SPYAuthenticMarkupVisibility, 544
SPYDatabaseKind, 544
SPYDialogAction, 544
SPYDOMType, 545
SPYDTDSchemaFormat, 545
SPYEncodingByteOrder, 545
SPYExportNamespace, 545
SPYFrequentElements, 546
SPYKeyEvent, 547
SPYLibType, 547
SPYLoading, 547
SPYMouseEvent, 548
SPYNumberDateTimeFormat, 548
SPYProgrammingLanguage, 549
SPYProjectItemTypes, 549
SPYProjectType, 549
SPYSampleXMLGenerationOptimization, 550
SPYSampleXMLGenerationSchemaOrDTDAssignment,
550

SPYSchemaDefKind, 551
SPYSchemaDocumentationFormat, 551
SPYTextDelimiters, 552
SPYTextEnclosing, 552
SPYTypeDetection, 553
SPYURLTapes, 553
SPYViewModes, 554
SPYVirtualKeyMask, 554
SPYXMLDataKind, 555

Event, 321, 322, 373, 374, 375, 376, 377, 378, 406,
407, 408, 467, 468

Events, 298

ExportSettings,
CreateKeys, 445
ElementList, 445
EntitiesToText, 445
ExportAllElements, 446
FromAttributes, 446
FromSingleSubElements, 446
FromTextValues, 446
IndependentPrimaryKey, 446
Namespace, 447
SubLevelLimit, 447

F

FileSelectionDlg,
Application, 448
DialogAction, 448
FullName, 448
Parent, 448

G

Generate Sample XML, 542, 550

GenerateSampleXMLDlg,
Application, 461
FillWithSampleData, 463
NonMandatoryAttributes, 463
NonMandatoryElements, 464
Parent, 465
RepeatCount, 465
TakeFirstChoice, 466

GridView,
CurrentFocus, 469

GridView,
 Deselect, 469
 IsVisible, 469
 OnBeforeDrag, 467
 OnBeforeDrop, 467
 OnBeforeStartEditing, 468
 OnEditingFinished, 468
 OnFocusChanged, 468
 Select, 469
 SetFocus, 470
GUI 詳細, 14

H

HTML,
 Authentic Desktop の統合, 569
HTML サンプル,
 AuthenticDesktopControl 統合, 568, 569
 of AuthenticDesktopControl 統合, 568
HTML 出力,
 PXF ファイルからAuthentic View 内に生成, 192

J

Java, 571
JRE,
 Eclipse のための Authentic プラグイン, 245
JScript,
 scripting with Authentic Desktop, 252

L

loading, 437

M

Macros,
 developing, 252, 257
 enabling, 263, 275
 running, 276
Microsoft® SharePoint® Server, 166

MIME, 222
MSXML, 225

O

OASIS,
 XML カタログ, 176
OS,
 Altova 製品のための, 609
Overview,
 of XMLSpy API, 292

P

Parent, 427
PDF 出力,
 PXF ファイルからAuthentic View 内に生成, 192
Plug-in,
 ATL sample files, 281
 registration, 277
 User interface configuration, 278
 XMLSPY, 277
PUBLIC,
 識別子 - カタログ, 176
PXF ファイル,
 Authentic View から出カドキュメントを生成, 192

R

Register,
 plug-in, 277
Return キー,
 Enter キーを参照, 83
RTF 出力,
 PXF ファイルからAuthentic View 内に生成, 192

S

save, 429
schema, 331
SchemaDocumentationDlg,

- SchemaDocumentationDlg,
 - AllDetails, 471
 - Application, 471
 - IncludeAll, 473
 - IncludeAttributeGroups, 473
 - IncludeComplexTypes, 474
 - IncludeGlobalElements, 474
 - IncludeGroups, 474
 - IncludeIndex, 475
 - IncludeLocalElements, 475
 - IncludeRedefines, 476
 - IncludeSimpleTypes, 476
 - OptionsDialogAction, 477
 - OutputFile, 477
 - OutputFileDialogAction, 477
 - OutputFormat, 478
 - Parent, 478
 - ShowAnnotations, 478
 - ShowAttributes, 479
 - ShowChildren, 479
 - ShowConstraints, 480
 - ShowDiagram, 479
 - ShowEnumerations, 479
 - ShowNamespace, 480
 - ShowPatterns, 480
 - ShowProgressBar, 481
 - ShowProperties, 481
 - ShowResult, 481
 - ShowSingleFacets, 482
 - ShowSourceCode, 482
 - ShowType, 482
 - ShowUsedBy, 482
 - Scripting Editor,
 - overview, 252, 254
 - SharePoint® Server, 166
 - Source control,
 - installing a source-control plug-in, 101
 - SPP ファイルの位置, 145
 - SPS,
 - 新たな XML ファイルに関連付ける, 127
 - SPS テーブル,
 - 動的なテーブルの編集, 41
 - SPS テーブル 内の Authentic View,
 - 使用方法, 60
 - SpyProject,
 - CloseProject, 484
 - ProjectFile, 484
 - RootItems, 484
 - SaveProject, 484
 - SaveProjectAs, 484
 - SpyProjectItem,
 - ChildItems, 485
 - FileExtensions, 485
 - ItemType, 485
 - Name, 486
 - Open, 486
 - ParentItem, 486
 - Path, 486
 - ValidateWith, 486
 - XMLForXSLTransformation, 486
 - XSLForXMLTransformation, 487
 - XSLTransformationFileExtension, 487
 - XSLTransformationFolder, 487
 - SpyProjectItems,
 - AddFile, 487
 - AddFolder, 488
 - AddURL, 488
 - Count, 488
 - Item, 488
 - RemoveItem, 489
- ## T
- terminate, 334
 - TextImportExportSettings,
 - DestinationFolder, 489
 - EnclosingCharacter, 490
 - Encoding, 490
 - EncodingByteOrder, 490
 - FieldDelimiter, 490
 - FileExtension, 490
 - HeaderRow, 490
 - ImportFile, 491
- ## U
- UCS-2, 223
 - Unicode のサポート,
 - Altova 製品内で, 609
 - URL, 338, 429, 437, 438
 - メールで送信, 139
 - User interface,

User interface,
 configure using plug-in, 278
 UTF-16, 223

V

VBScript,
 scripting with Authentic Desktop, 252
 Visual Studio,
 Authentic Desktop ActiveX
 コントロールをツールボックスへ追加, 557
 Visual Studio .Net,
 と Authentic Desktop, 241
 と Authentic Desktop の違い, 243
 VS .NET,
 と Authentic Integration Package, 242
 VS .NET のための Authentic プラグイン,
 インストール, 242

W

Web サーバー, 238
 Windows,
 Altova 製品のためのサポート, 609
 Word 2007+ 出力,
 PXF ファイルから Authentic View 内に生成, 192

X

XML,
 Oasis カタログ, 176
 スペルチェッカー, 197
 XML DB,
 新たにデータ行を Authentic View へロード, 187
 新規 XML データ行のロード, 69
 XML DB 内のデータを編集, 187
 XML テーブル内の Authentic View,
 編集のためのアイコン, 65
 XML テーブル内の Authentic View,
 使用方法, 61
 XML ドキュメント,
 Authentic View 内で開く, 25
 XML パーサー,

について, 609

XML メニュー, 175
 XML 署名, 81, 188
 XMLData,
 AppendChild, 531
 EraseAllChildren, 533
 EraseCurrentChild, 533
 GetChild, 534
 GetChildKind, 535
 GetCurrentChild, 535
 GetFirstChild, 536
 GetNextChild, 537
 HasChildren, 538
 HasChildrenKind, 538
 InsertChild, 539
 IsSameNode, 540
 Kind, 540
 MayHaveChildren, 540
 Name, 540
 Parent, 541
 TextValue, 541
 XMLSPY, 126
 plug-in registration, 277
 ヘルプ, 21
 機能, 21
 XMLSpy API,
 documentation, 291
 overview, 292
 XMLSPY plug-in, 277
 XMLSpy コマンドテーブル, 579
 XMLSPY を使用して編集, 220
 XMLSpyLib, 291, 292
 Application, 319
 AuthenticDataTransfer, 341
 AuthenticRange, 345
 AuthenticView, 373
 CodeGeneratorDlg, 389
 DatabaseConnection, 395
 Dialogs, 401
 Document, 404
 Documents, 434
 DTDSchemaGeneratorDlg, 438
 ElementList, 443
 ElementListItem, 444
 ExportSettings, 445
 FileSelectionDlg, 447
 GenerateSampleXMLDlg, 461
 GridView, 466

- XMLSpyLib, 291, 292
 - ProjectItem, 485
 - SchemaDocumentationDlg, 470
 - SpyProject, 483
 - SpyProjectItems, 487
 - TextImportExportSettings, 489
 - XMLData, 530
 - XML-準拠, 222
 - XQuery,
 - Xquery ドキュメントに変数をパスする, 180
 - XQuery プロセッサ,
 - Altova 製品内で, 609
 - XSL/XQuery メニュー, 178
 - XSLT,
 - プロセッサ, 225
 - XSLT パラメーター,
 - インターフェイスを使用してスタイルシートにパスする, 180
 - XSLT プロセッサ,
 - Altova 製品内で, 609
 - XSLT 変換, 178
 - FO への変換, 179
 - PDF への変換, 179
- ## Z
- アイコン,
 - ツールバー/メニューへ追加, 205
 - 大きく表示, 216
 - アクティブな構成,
 - グローバルリソース, 205
 - イメージ書式設定,
 - 内の Authentic View, 82
 - インターネット, 238
 - インターネットの使用,
 - Altova 製品内で, 610
 - ウィンドウ,
 - フロート、ドッキング、タブ, 14
 - プロジェクト, 231, 232
 - 開く, 233
 - 左右に並べて表示, 231
 - 自動非表示, 14
 - 重ねて表示, 231
 - 上下に並べて表示, 231
 - 情報, 232
 - 入力ヘルパー, 232
 - ウィンドウ メニュー, 231
 - エイリアス,
 - グローバルリソース参照, 87
 - エクスプローラー, 222
 - エンコーディング,
 - デフォルト, 223
 - ファイル, 133
 - エンティティ,
 - Authentic View に挿入, 36
 - Authentic View 内に挿入, 56
 - Authentic View 内の定義, 56
 - 内の Authentic View の定義, 79
 - エンティティ入力ヘルパー,
 - Authentic View, 47
 - エンドユーザー使用許諾契約書, 611
 - お気に入り, 234
 - カスタマイズ, 20
 - コンテキストメニュー, 212
 - コンテキストメニューのカスタマイズ, 216
 - ツールバー/メニューコマンド, 205
 - マクロ, 214
 - メニュー, 212
 - カスタムカタログ, 176
 - カスタム辞書, 197
 - カタログ,
 - Oasis XML, 176
 - キーコード,
 - Altova ソフトウェアのための, 235
 - キーボードショートカット, 210
 - キーマップ, 235
 - クエリ,
 - Authentic View内の DB 表示, 70
 - グローバル,
 - 設定, 219
 - グローバル リソース,
 - 定義, 88
 - グローバル リソース XML ファイル, 88
 - グローバルリソース,
 - アクティブな構成, 205
 - ツールバーにて有効にする, 207
 - データベースの型の定義, 95
 - ファイルの型の定義, 89
 - ファイル型とフォルダー型の使用, 97
 - フォルダーの型の定義, 93
 - 構成の変更, 100
 - 使用, 97, 100
 - 定義, 204
 - グローバルリソースの,
 - 構成, 88

- グローバルリソースの構成, 100
- コピーコマンド, 142
- コマンド, 205
 - キーマップにて表示, 235
 - コンテキストメニュー, 212
 - ツールバー/メニューへ追加, 205
 - メニューから削除, 212
 - メニューのリセット, 212
- コマンドライン, 240
- コンテキストメニュー,
 - カスタマイズ, 216
 - コマンド, 212
- サーチ,
 - 検索を参照, 143
- サポートオプション, 21
- サポートセンター, 238
- ショートカット,
 - ツールチップにて表示, 216
 - 割り当て/削除, 210
- スキーマ,
 - 設定, 220
- スクリプト, 226
- スクリプトエディター,
 - 開始, 203
- スクリプト言語, 226
- スタートグループ,
 - 追加(コンテキストメニュー), 216
- ステータスバー, 19
- スプラッシュ画面, 224
- スペルチェッカー,
 - カスタム辞書, 197
- スペルチェックのオプション, 200
- セルの最大幅, 224
- ソースコントロール,
 - チェックアウトを元に戻す, 154
 - プロバイダーの変更, 161
- ソース管理, 227
 - サポートされているプロバイダー, 148
 - ステータスを最新の状態に更新, 161
 - ソース管理に追加, 154
 - チェックアウト, 152
 - ファイルの削除, 155
 - フォルダーの取得, 151
 - プロジェクトを開く, 149
 - プロパティ, 160
 - 共有, 155
 - 差分の表示, 158
 - 最新バージョンの取得, 151
 - 有効化、無効化, 150
 - 履歴の表示, 157
- ソース管理マネージャー, 161
- ソフトウェアのライセンスの認証, 235
- タブ文字, 220
- チェック,
 - スペルチェッカー, 197
- ツール,
 - 外部アプリケーションを参照, 209
- ツールチップ,
 - ショートカットを表示, 216
 - 表示, 216
- ツールバー, 19
 - コマンドを追加, 205
 - ツールバーならびにメニューコマンドのリセット, 207
 - マクロの追加, 214
 - 新規作成, 207
 - 大きなアイコンで表示, 216
 - 有効化/無効化, 207
- ツールメニュー, 197
- データベース,
 - Authentic Viewにて編集, 186
 - DB を参照, 69
- テーブル,
 - Authentic View 内, 30
 - 自動的に生成, 222
 - 動的な(SPS) テーブルの編集, 41
- テーブル 内の Authentic View,
 - SPS (静的と動的) テーブルの使用, 60
 - XML テーブルの編集のためのアイコン, 65
- テーブル内の Authentic View,
 - XML テーブルの使用, 61
- テキスト,
 - Authentic View 内の書式設定, 56
 - Authentic View 内の編集, 56
 - ドキュメント内を検索, 143
 - 検索と置き換え, 143
- テキストの範囲,
 - 内の Authentic View, 51
- テクニカルサポート, 238
- デフォルト,
 - エンコーディング, 223
 - メニュー, 212
- デフォルトエディター, 222
- デフォルトビュー,
 - メインウィンドウにおける設定, 222
- テンプレート,

- テンプレート,
 - Authentic View における XML ドキュメントのテンプレート, 185
- テンプレート XML ファイル,
 - Authentic View 内で開く, 25
- テンプレートファイル,
 - 新規ドキュメントに対して, 127
- ドキュメント,
 - スペルチェッカー, 197
- ドキュメントレベル,
 - XMLSpy の統合のサンプル, 564
- ドッキングウィンドウ, 14
- ドック可能なウィンドウ, 231, 232
- トピック,
 - 目次にて参照, 234
- パーサー,
 - Altova 製品に内蔵の, 609
- パラメーター,
 - DB クエリ内, 70
 - インターフェイスを使用してスタイルシートにパスする, 180
- バリデーター,
 - Altova 製品内で, 609
- ビッグエンディアン, 223
- ビュー,
 - ブラウザー ビュー, 194
- ビューの変更,
 - Authentic View, 41
- ビューメニュー, 194
- ファイル, 220
 - エンコーディング, 133
 - ソース管理に追加, 154
 - タブ, 220
 - デフォルトエンコーディング, 223
 - メールで送信, 139
 - 印刷オプション, 140
 - 開く, 128
 - 開くオプション, 220
 - 最後に使用, 141
 - 新たに作成, 127
 - 閉じる, 134
 - 保存, 134
- ファイル メニュー, 127
- ファイルタイプ, 222
- ファイルの保存,
 - エンコーディング, 133
- ファイル拡張子,
 - カスタマイズ, 176
- フォントサイズ,
 - ブラウザービューにて, 196
- ブックマーク, 234
- ブラウザー, 224
 - ビュー, 194
- ブラウザー ビュー, 195
- ブラウザー メニュー, 195
- ブラウザービュー,
 - フォントサイズ, 196
 - ページの読み込みを中止, 196
 - 最新の状態に更新, 196
 - 進む, 195
 - 別のウィンドウ, 196
 - 戻る, 195
- プラットフォーム,
 - Altova 製品のための, 609
- プログラマーのレファレンス, 250
- プログラム設定, 219
- プロジェクト, 163
 - URL の追加, 162
 - アクティブなファイルを追加, 162, 163
 - グローバルリソースの追加, 162
 - ソース管理に追加, 154
 - ファイルの追加, 162
 - フォルダーの追加, 163
 - プロパティ, 171
 - 開く, 147
 - 外部ウェブフォルダーの追加, 166
 - 外部フォルダーの追加, 163
 - 概要, 145
 - 関連するファイルを追加, 163
 - 再ロード, 148
 - 最近使用されたプロジェクト, 174
 - 新規作成, 147
 - 閉じる, 148
 - 保存, 148
- プロジェクト ウィンドウ, 16
- プロジェクトウィンドウ, 231, 232
 - 表示/非表示の切り替え, 232
- プロジェクトメニュー, 145
- ヘルプ,
 - キーマップ, 235
 - 目次, 234
- ヘルプ メニュー, 234
- ヘルプシステム, 234
- ホットキー, 210
- マークアップ,
 - 内の Authentic View, 41, 44
- マークアップ (Authentic View にて),

- マークアップ (Authentic View にて),
 - 隠す, 191
 - 小さな/大きな/混合マークアップを表示, 191
- マークアップの非表示, 41
- マークアップを隠す, 44
- マークアップを隠す (Authentic View にて), 191
- マクロ,
 - アプリケーションマクロの起動, 204
 - メニュー/ツールバーへ追加, 214
 - 編集ボタン, 216
- メインウィンドウ, 15
- メインウィンドウにおけるドキュメント, 15
- メインカタログ, 176
- メニュー, 205
 - Authentic, 185
 - XML, 175
 - XSL/XQuery, 178
 - ウィンドウ, 231
 - カスタマイズ, 212
 - コマンドの追加/削除, 205
 - コマンドを削除, 212
 - デフォルト/XMLSpy, 212
 - ビュー, 194
 - プロジェクト, 145
 - ヘルプ, 234
 - マクロの追加, 214
 - ルーツ, 197
 - 編集, 142
- メニュー ブラウザー, 195
- メニューバー, 19
- メモリ要件, 609
- やり直しコマンド, 142
- ユーザーインターフェイス 詳細, 14
- ユーザーマニュアル, 10
- ユーザーマニュアル Authentic Desktop, 12
- ユーザーリファレンス, 126
- ライセンス,
 - Altova ソフトウェアのための, 235
 - 情報, 611
- ライセンス計測,
 - Altova 製品にて, 611
- リセット,
 - ショートカット, 210
 - ツールバーならびにメニューコマンド, 207
 - メニューコマンド, 212
- リトルエンディアン, 223
- リンク,
 - Authentic View 内の, 56
- レジストリ,
 - 設定, 219
- 移動ウィンドウ, 14
- 印刷,
 - Authentic View から, 37
- 印刷オプション, 140
- 印刷設定, 140
- 隠す, 231, 232
- 解析,
 - XSLT, 225
- 改行, 220
- 開く,
 - ファイル, 128
- 開くオプション,
 - ファイル, 220
- 外部 XSL プロセッサ, 225
- 外部アプリケーション,
 - ファイルから開く, 209
- 外部で解析されるエンティティ, 222
- 割り当て,
 - ショートカットをコマンドへ割り当て, 210
- 技術データ, 609
- 空要素, 222
- 検索,
 - ドキュメント内のテキスト, 143
 - と置き換え (ドキュメント内のテキストにて), 143
- 検証, 20, 176
- 検証の設定, 220
- 元に戻すコマンド, 142
- 構文による色分け, 222, 224
- 行,
 - 削除 (Authentic View にて), 192
 - 上/下へ移動, 192
 - 挿入 (Authentic View にて), 192
 - 追加 (Authentic View にて), 192
 - 複製 (Authentic View にて), 192
- 混合マークアップ (Authentic View にて), 191
- 再ロード, 220
 - ファイルの変更, 133
- 最後に使用されたファイル,
 - リスト, 141
- 最適な幅, 224
- 削除,
 - コンテキストメニューのコマンドを削除, 212
 - ショートカット, 210
 - ツールバー, 207
 - ツールバーからアイコンを削除, 205
 - ツールバーからコマンドを削除, 205

- 削除,
 - 行 (Authentic View にて), 192
- 削除コマンド, 142
- 視覚的な書式設定,
 - 内の Authentic View, 82
- 試用期間,
 - Altova ソフトウェア製品の試用, 611
- 自動バックアップの設定, 220
- 自動検証, 222
- 自動検証の無効化, 222
- 自動非表示ウィンドウ, 14
- 辞書,
 - スペルチェッカー, 197
 - 既存のものを修正, 197
 - 新たに追加, 197
- 重ねて表示,
 - ウィンドウ, 231
- 出力ウィンドウ,
 - 表示/非表示の切り替え, 232
- 出力フォーマット, 220
- 小さなマークアップ, 44
- 小さなマークアップ (Authentic View), 191
- 小さなマークアップの表示, 41
- 章, 234
- 上/下へ移動,
 - 行 (Authentic View にて), 192
- 情報 ウィンドウ, 232
- 情報ウィンドウ, 18
- 新規ファイル,
 - 作成, 127
- 整形形式のチェック, 175
- 正規表現,
 - 検索文字列にて, 143
- 静的な (SPS) テーブル 内の Authentic View,
 - 使用方法, 60
- 切り取りコマンド, 142
- 切り替え, 231, 232
- 設定, 20, 219
 - スクリプト, 226
- 選択されたノードに対しての XPath, 39
- 全て選択コマンド, 143
- 挿入, 192
- 属性のプレビュー, 224
- 属性値,
 - Authentic View 内に挿入, 35
- 属性入力ヘルパー,
 - Authentic View, 47
- 大きなマークアップ, 44
- 大きなマークアップ (Authentic View にて), 191
- 大きなマークアップの表示, 41
- 置換え,
 - テキスト, 143
 - ドキュメント内のテキスト, 143
- 著作権に関する情報, 611
- 適用, 219
- 貼り付け,
 - XML, 51
 - XML として, 56
 - テキスト, 51
 - テキストとして, 56
- 貼付けコマンド, 142
- 電子メール,
 - ファイルを送信, 139
- 統合,
 - アプリケーション内の Authentic Desktop, 556
- 動的な (SPS) テーブル 内の Authentic View,
 - 使用方法, 60
- 動的なテーブル,
 - 編集, 41
- 内の Authentic View テーブル,
 - 使用方法, 60
- 日付,
 - 手動での変更, 77
- 日付の選択,
 - 内の Authentic View の使用, 76
- 入力ヘルパー, 18, 232
 - 表示/非表示の切り替え, 232
- 背景情報, 609
- 配置,
 - 左右に並べて, 231
 - 上下に並べて, 231
- 配布,
 - Altova ソフトウェア製品, 611
 - Altova ソフトウェア製品の配布, 611
- 非 XML ファイル, 222
- 表示, 231, 232
- 表示方法, 224
- 評価キー,
 - Altova ソフトウェアのための, 235
- 評価機関,
 - Altova ソフトウェア製品, 611
- 複合型マークアップ, 44
- 複合型マークアップの表示, 41
- 複数ユーザー, 220
- 文字セット,
 - エンコーディング, 223

文法, 222

変換,

 XSLT 変換を参照, 179

変更コマンドでの検証, 177

変更を検知, 220

編集,

 マクロボタン, 216

編集メニュー, 142

法的な情報, 611

目次, 234

要素入力ヘルパー,

 Authentic View, 47

隣り合わせ, 224

列挙,

 AuthenticDesktopControl, 606