

# Altova Authentic 2021 Browser Edition

ユーザーマニュアル

## Altova Authentic 2021 Browser Edition 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

<b>1</b>	<b>はじめに</b>	<b>6</b>
1.1	Authentic Browser の利点	7
1.2	しくみ	8
1.3	Authentic Browser のバージョン	9
1.4	このドキュメンテーションについて	11
<b>2</b>	<b>サーバーのセットアップ</b>	<b>12</b>
2.1	IIS: ブラウザーサービスの構成	13
2.2	XSD、XML、と SPS/PXF ファイル	17
2.3	Authentic プラグインのための HTML ページ	19
2.3.1	Enterprise エディションのためのライセンス供与	20
2.3.2	Internet Explorer	21
2.3.3	Firefox	29
2.3.4	依存しないブラウザ	37
2.4	オンデマンドのインストールのための拡張機能パッケージ	41
<b>3</b>	<b>クライアントのセットアップ</b>	<b>42</b>
3.1	ブラウザの必要条件	43
3.2	Authentic Browser プラグイン	44
3.2.1	オンデマンドのインストール	44
3.2.2	拡張機能パッケージを使用した手動のインストール	45
3.2.3	MSI を使用した手動のインストール	45
3.2.4	MSI を使用したプッシュインストール	45
3.2.5	自動更新	46
3.2.6	アンインストール、無効化	46
3.3	IE9 セキュリティの設定	48
3.4	IE10 セキュリティの設定	50

---

<b>4</b>	<b>ユーザー レファレンス</b>	<b>51</b>
4.1	メカニズム.....	52
4.1.1	イベント: 接続ポイント (IE-固有).....	52
4.1.2	イベント: イベントリスナーの追加 (Firefox-固有).....	53
4.1.3	イベント: ツールバーボタン.....	54
4.1.4	イベント: レファレンス.....	54
4.1.5	ドキュメントコンテンツへのアクセスと変更.....	55
4.1.6	編集の操作.....	55
4.1.7	検索と置換.....	56
4.1.8	行の操作.....	56
4.1.9	ショートカット.....	56
4.1.10	テキストの状態ボタン.....	57
4.1.11	入力ヘルパー.....	57
4.1.12	パッケージ.....	57
4.1.13	XMLData の使用.....	63
4.1.14	DOM と XMLData.....	66
4.1.15	Authentic スクリプト.....	69
4.2	オブジェクト.....	71
4.2.1	Authentic.....	71
4.2.2	AuthenticCommand.....	99
4.2.3	AuthenticCommands.....	100
4.2.4	AuthenticContextMenu.....	101
4.2.5	AuthenticDataTransfer.....	102
4.2.6	AuthenticEvent.....	104
4.2.7	AuthenticEventContext.....	109
4.2.8	AuthenticLoadObject.....	112
4.2.9	AuthenticRange.....	112
4.2.10	AuthenticSelection.....	140
4.2.11	AuthenticToolbarButton.....	141
4.2.12	AuthenticToolbarButtons.....	142
4.2.13	AuthenticToolbarRow.....	144
4.2.14	AuthenticToolbarRows.....	145
4.2.15	AuthenticView.....	146

---

4.2.16	AuthenticXMLTableCommands.....	165
4.2.17	XMLData.....	174
4.3	列挙.....	184
4.3.1	SPYAuthenticActions.....	184
4.3.2	SPYAuthenticCommand.....	184
4.3.3	SPYAuthenticCommandGroup.....	185
4.3.4	SPYAuthenticDocumentPosition.....	185
4.3.5	SPYAuthenticElementActions.....	186
4.3.6	SPYAuthenticElementKind.....	186
4.3.7	SPYAuthenticEntryHelperWindows.....	186
4.3.8	SPYAuthenticMarkupVisibility.....	187
4.3.9	SPYAuthenticToolbarAllignment.....	187
4.3.10	SPYAuthenticToolbarButtonState.....	187
4.3.11	SPYXMLDataKind.....	187
<b>5</b>	<b>ASP.NET Web アプリケーション</b>	<b>189</b>
<b>6</b>	<b>ライセンス情報</b>	<b>190</b>
6.1	Authentic のための Altova エンドユーザー使用許諾契約書.....	191
	<b>インデックス</b>	<b>192</b>

## 1 はじめに

Altova® Authentic® 2021 Browser Edition により使用が簡単なワードプロセッサを彷彿させるユーザーインターフェイスを使用して XML と データベースコンテンツを簡単に作成し編集することができます。Browser Edition を簡単に Web ページに埋め込むことができ、ブラウザー内で直接編集することができます。Authentic Browser Edition は Microsoft Internet Explorer と Mozilla Firefox ブラウザーのためにプラグインとして使用することができます。



Authentic Browser は単一の**Enterprise Edition** (ライセンス必須) [信頼済みバージョン](#)と[信用されないバージョン](#)としてご使用いただけます。

## 1.1 Authentic Browser の利点

Authentic Browser XML 編集ソリューションには多くの利点があり、最も重要な利点は以下にリストされています:

- ブラウザーを介して複数のユーザーがXMLドキュメントにアクセスし編集することができます。現在、**Microsoft Internet Explorer 5.5** または以降 と **Mozilla Firefox** はサポートされています。しかしながら、Internet Explorer 10 と 11 の 64ビットバージョンはサポートされません。
- 使用コストが削減され、所属機関全体へのデプロイと、アプリケーションの管理が大幅に簡単になりました。
- XML スキーマと XSLT などのオープンな標準がベースとされています。
- 完全な Unicode 互換性があります。
- 頻繁に使用されている Internet Explorer ブラウザーをベースとしている Altova の Authentic View を使用することができます。Authentic View によりユーザーは基となる XML コードを参照することなく WYSIWYG で XML ファイルを編集することができます。
- Authentic Browser プラグインはブラウザアドオンであるため、追加のソフトウェアをデプロイする必要はありません。
- Authentic Browser は COM インターフェイスは [Authentic](#) オブジェクトにより定義されている ActiveX コントロールです。完全なオブジェクト モデルはこのドキュメンテーションの [ユーザーレファレンス オブジェクト セクション](#) で説明されています。

## 1.2 しゅき

Authentic Browser プロジェクトを実装するには、サーバーは接続済みのサーバーマシン一台と1つまたは複数のクライアントマシンが必要です。

### Authentic Browser Server

Authentic Browser サーバーは次を行うことができます:

- サーバーは編集する Altova Authentic XML ドキュメントに関連するファイルを保管します。これらのファイル以下のとおりです:
  1. 編集可能な XML ドキュメントがベースとする XML スキーマ (XSD)
  2. 編集する XML ドキュメント
  3. Authentic View 内の XML ドキュメントのレイアウトと入力メカニズムを管理する SPS または PXF ファイル
- サーバーは [Authentic プラグインのための HTML ページ](#) を保管します。この HTML ページは Authentic View の編集のためのアクセスポイントです。XML ドキュメントにアクセスするための命令を含んでおり、XML ドキュメントがロードされ編集される Authentic View ウィンドウのためのエンターとして役割を果たします。このページにアクセスするには、URL がクライアントブラウザ内に入力されます。
- Authentic Browser の Enterprise エディションをデプロイする場合、Enterprise ライセンスがサーバー上に保管される必要ががあります。Enterprise ライセンスは、1つまたは複数の指定されているサーバーのために発行されます。
- Authentic Browser プラグインのオンデマンドのインストールが計画されている場合、クライアントマシン上でプラグインをダウンロードしてインストールするための Authentic Browser 拡張機能パッケージ CAB、XPI、およびまたは CRX ファイル) を保管します (それ以外の場合、プラグインは直接クライアントブラウザ内にインストールされます)。

ステップはサーバーを準備する必要ががあり、セクション [サーバーのセットアップ](#) 内で説明されています。

### Authentic Browser クライアント

XML ドキュメントに割り当てられている SPS または PXF ファイルが存在する場合、XML ドキュメントは Altova の Authentic View 内で表示し、編集することができます。以下のようにセットアップされる必要のある Authentic View の編集はクライアントマシン上で実行されます:

- 各クライアントに次のブラウザの1つがインストールされている必要ががあります: Internet Explorer 5.5 または以降 (32ビットと64ビット)、Firefox (32ビット)
- 更に、各クライアントマシンには、Internet Explorer 5.5 または以降がインストールされている必要ががあります。これは (ブラウザウィンドウ内で表示される) Authentic View インターフェイスは Internet Explorer を使用して生成されているからです。
- オンデマンドのインストールが計画されていない限り、Authentic Browser プラグインは直接クライアントブラウザに直接インストールされている必要ががあります。

クライアントを準備するための必要なステップは、セクション [クライアントのセットアップ](#) 内で説明されています。

### Authentic Browser メカニズム

サーバーとクライアントが上記のように準備されると、ユーザーは Authentic プラグインのための HTML ページの URL をクライアントブラウザ内に入力します。クライアントブラウザ内にプラグインが既にインストールされていると、HTML ページは、クライアントブラウザ内のプラグインのオンデマンドのインストールを行う命令を含むことができます。

クライアント上にプラグインがインストールされると、HTML ページ内のコードにより、ブラウザウィンドウ内で Authentic View 編集ウィンドウが開かれます。Authentic View ウィンドウで編集する XML ドキュメントがロードされ、ユーザーは XML ドキュメントを直接編集し、変更を保存することができます。

## 1.3 Authentic Browser のバージョン

Authentic Browser バージョンはつぎの必要条件に従い使用することができます。

- 言語: 英語 (EN)、ドイツ語 (DE)、スペイン語 (ES)、日本語 (JA)
- 信頼されている信頼されていない: 信頼されている 信頼されていない
- クライアントブラウザ: Microsoft Internet Explorer (32ビットと64ビット)、Mozilla Firefox (32ビット)、Google Chrome (32ビット)。次も参照してください: [Browser 必要条件](#)。

各サポートされる言語 (英語、ドイツ語、スペイン語、日本語) のために、個別の信頼されている、および 信頼されていないバージョンが各サポートされるブラウザ (Microsoft Internet Explorer 32ビットと64ビット、および Mozilla Firefox) のために使用することができます。Authentic Browser プラグインの、複数、全てのバージョンをクライアントマシンにインストールすることができます。それぞれブラウザのアドオンマネージャー内の個別のプラグインとして表示されます。

英語 (EN)、ドイツ語 (DE)、スペイン語 (ES)、日本語 (JA) のための異なるバージョンはクラス (CAB ファイルのための ID と (XPI と CRX ファイルのための) MIME の種類) と共に下にリストされています。[Authentic プラグインのためのHTML ページ](#) 内の関連するクラスID または MIME の種類を指定する必要があります。

- CABファイルとそのクラスID (32ビットと64ビット Internet Explore のためのクラスID )

EN	信頼された	B4628728-E3F0-44a2-BEC8-F838555AE780
EN	信頼されていない	A5985EA9-3332-4ddf-AD7F-F6E98BFEEAF94
DE	信頼された	91DDF44A-DFD1-4F47-8EE3-4CBE874584F7
DE	信頼されていない	28A640E8-EAEE-4B5D-BEBE-BFA956081E66
ES	信頼された	23B503E7-269B-45CE-BAB2-22AA97BED8E2
ES	信頼されていない	8AD3EF86-AC1E-4574-8C13-DE5B6CBECEBE
FR	信頼された	1B768F46-A9E8-4a88-91B0-2917FE47612A
FR	信頼されていない	070F4B50-26F7-48cc-9AC1-52D562C1E749
JA	信頼された	5B15DB5A-1720-4264-BB65-70C3F7A860DA
JA	信頼されていない	4B9512D2-A3D3-46e3-82C1-34248BBDCE58

- XPI と CRX ファイルのための MIME の種類

EN	信頼された	application/x-authentic-scriptable-plugin
EN	信頼されていない	application/x-authentic-scriptable-plugin-untrusted
DE	信頼された	application/x-authentic-scriptable-plugin-german

DE	信頼されていない	application/x-authentic-scriptable-plugin-untrusted-german
ES	信頼された	application/x-authentic-scriptable-plugin-spanish
ES	信頼されていない	application/x-authentic-scriptable-plugin-untrusted-spanish
FR	信頼された	application/x-authentic-scriptable-plugin-french
FR	信頼されていない	application/x-authentic-scriptable-plugin-untrusted-french
JA	信頼された	application/x-authentic-scriptable-plugin-japanese
JA	信頼されていない	application/x-authentic-scriptable-plugin-untrusted-japanese

必要に応じて [Altova Web サイト](#) からつおは複数のバージョンをダウンロードすることができます。

多種の Authentic Browser バージョンに関する以下の点に注意してください！

- すべてのバージョン Unicode バージョンで、それぞれは、XML ドキュメント内の複数の文字セットのフルサポートを提供します。Unicode バージョンは次のクライアントワークステーションでサポートされています：プラットフォーム更新済みの Windows 7 SP1、Windows 8、Windows 10。
- 32-bit と 64-bit Internet Explorer バージョンのそれぞれのために個別の CAB ファイルが存在しますが、(32-bit と 64-bit IE ブラウザ) のための 2 つの CAB ファイルのクラス ID は EN/DE/ES/JA と信頼されている信頼されていないバージョンのそれぞれの上のテーブル内でリストされているものと同様です。
- 信頼されているバージョンはローカルファイルへのアクセスを許可しないため「安全なスクリプト」とマークされています。ブラウザベースのシナリオ内で使用することができ、クライアント側のセキュリティ警告を発生させずに呼び出すことができます。
- 信頼されていないバージョンはインターネットへのデプロイ、または、Authentic Browser Edition をアプリケーション内で ActiveX コントロールとして使用することを目的としています。ローカルファイルへのアクセスを適用するため「安全なスクリプト」としてはマークされません。ブラウザウィンドウからのバージョンを使用しようとすると、ユーザーの許可の有無が問われます。You can decide whether ActiveX コントロールが無効化されるか、または、有効化されるかを決定することができます。また、ブラウザが ActiveX コントロールを有効化するためのバージョンの選択をプロンプトすることもできます(詳細に関しては、ブラウザ固有のセクションを参照してください！ [Internet Explorer 9](#))。

メモ ブラウザサービスを構成してインストールする場合は、(例: Microsoft Internet Information Services)、サプライヤーのドキュメンテーションを参照してください。

## 1.4 このドキュメンテーションについて

このドキュメンテーションは Authentic Browser プラグイン ユーザーマニュアルであり、次のセクションに整理されています:

- 概要 セクションでは以下について説明されています: (i) [Authentic Browser の利点](#) (ii) [Authentic Browser のしくみ](#) と (iii) 多種の [Authentic Browser バージョン](#).
- [Authentic プラグインのための HTML ページ](#) の説明と、Authentic Browser 拡張機能/パッケージの仕組みを説明するセクションを含む、Authentic Browser プロジェクトのためのサーバーのセットアップに必要なステップについて説明する [サーバーのセットアップ](#) セクションは [オンデマンドのインストール](#) のために使用することができます。
- [クライアントのセットアップ](#) セクションには、[Authentic Browser プラグインをクライアントブラウザ内にインストールするための](#) 多種の方法を説明するセクションが含まれています。
- メカニズム上の 3 部のレファレンス セクション ([ユーザーレファレンス メカニズム](#)、[ユーザーレファレンス オブジェクト](#) と [ユーザーレファレンス 列挙](#))、オブジェクト、および、Authentic Browser 内の Authentic View を作成しカスタム化するために使用される列挙。
- Authentic Browser プラグインのデプロイを簡単にするために、Visual Studio .NET と完全に統合するための [ASP.NET サーバー管理](#) を適用します。

### 関連するドキュメンテーション

関連する追加の Altova ドキュメンテーションがセット存在します:

- StyleVision Power Stylesheet (SPS) を作成するための Altova StyleVision 製品ドキュメンテーションは、Altova [Web サイト](#) で使用することができます。SPS は XML ドキュメントの Authentic View を管理するファイルです。このドキュメンテーションは XML 編集のための Authentic View インターフェイスを開発する個人に関連します。
- %AUTH-VIEW%> を使用するためのドキュメンテーション。Authentic View を使用する個人は Altova [Web サイト](#) で使用することのできる Authentic View チュートリアルと Authentic Desktop ユーザーマニュアルの使用方法のセクションを参照してください。

## 2 サーバーのセットアップ

Authentic Browser のためのサーバーのセットアップには以下のステップが含まれます。

### ブラウザサービスの構成

[サーバーのブラウザサービスをインストール構成します。](#) Microsoft Internet Information Services (IIS) を使用する場合、インストールのプロセスは、サブディレクトリを持つ Inetpub とデフォルトのディレクトリを作成することにご注意してください。サーバーのルートディレクトリは以下ようになります: //Inetpub/wwwroot。(ブラウザサーバー構成内で ルートディレクトリとして他のディレクトリを指定しない場合、サーバーのIP アドレスを使用してこのディレクトリに達することができます。ブラウザサービスのインストールと構成に関する詳細は、[ASPサーバーのドキュメンテーション](#)を参照してください！

### XSD、XML、とSPS/PXF ファイルのセットアップ

SPS ファイルはAuthentic View 内でXMLドキュメントが編集可能な書式として表示されることを可能にします。XML スキーマ(XSD ファイル) をベースとしており、[Altova のStyleVision 製品](#) 内でデザインされます。SPS ファイルとXSD ファイルと編集するXMLドキュメントは(通常、Authentic Browser サーバー上で全てのクライアントマシンがアクセスすることが可能なネットワークの場所に保管されている必要があります。セクション [XSD、XML、とSPS/PXF ファイル](#)はこのサーバーの準備ステップを詳細に説明しています。

### Authentic プラグインのためのHTML ページの作成

[Authentic プラグインのためのHTML ページ](#) はAuthentic View 編集のためのアクセスポイントです。XMLドキュメントにアクセスするための命令を含んでおり、XMLドキュメントがロードされ編集されるAuthentic View ウィンドウのためのエンターとしての役割を果たします。このページにアクセスするには、URL がクライアントブラウザ内に入力されます。このHTML ページは正確に作成されサーバー上に保存される必要があります。HTML ページの作成方法は、セクション [Authentic プラグインのためのHTML ページ](#)内で説明されています。

### オンデマンドのインストールのための拡張機能/パッケージの保管

Authentic Browser (圧縮されている CAB、XPI、または CRX ファイル) を Altova Web サイトからサーバー上の場所にダウンロードします。Authentic Browser のEnterprise エディションをデプロイする場合、パッケージはEnterprise ライセンスが登録されているサーバー上に保管される必要があります。このファイルを解凍しないでください。Web サイトでは for 各言語バージョン(英語、ドイツ語、スペイン語、および日本語) のためにAuthentic Browser の信頼されている、および、信頼されていないバージョンがそれぞれフォーマット (CAB 32ビット、CAB 64ビット、XPI とCRX) のために存在します。[ファイル書式](#) と選択する [バージョン](#) に関する情報については、このセクションのサブセクションを参照してください。

**メモ** Authentic Browser をインストールする前に、Authentic Browser の前のバージョンが作動していないことを確認してください。それ以外の場合、新規のバージョンが正確に登録されず、インストールの破損を引き起こす可能性があります。このような状態が発生した場合、次を作動して登録してください! regsvr32 C:\Windows\Downloaded Program Files\AuthenticPlugin.dll。(プログラム regsvr32.exe を作動するために管理者の特権が必要なことにご注意ください)。

## 2.1 IIS: ブラウザーサービスの構成

Microsoft Internet Information Services (IIS) 6 は固有サイト(Web サイト またはフォルダー) のための MIME の種類内で定義されているファイル型のみに対して使用することができます。必要とされるファイルの型は、ですから、特定のサイトのための MIME の種類のリストと追加される必要ががあります。

Authentic Browser と作業するためには、次のファイル型が必要とされ追加される必要ががあります:

ファイルの拡張子	MIME の種類	コメント
xsd	text/plain	
sps	text/plain	
pxf	application/x-zip-compressed	
xpi	application/x-xpinstall	Firefox のため

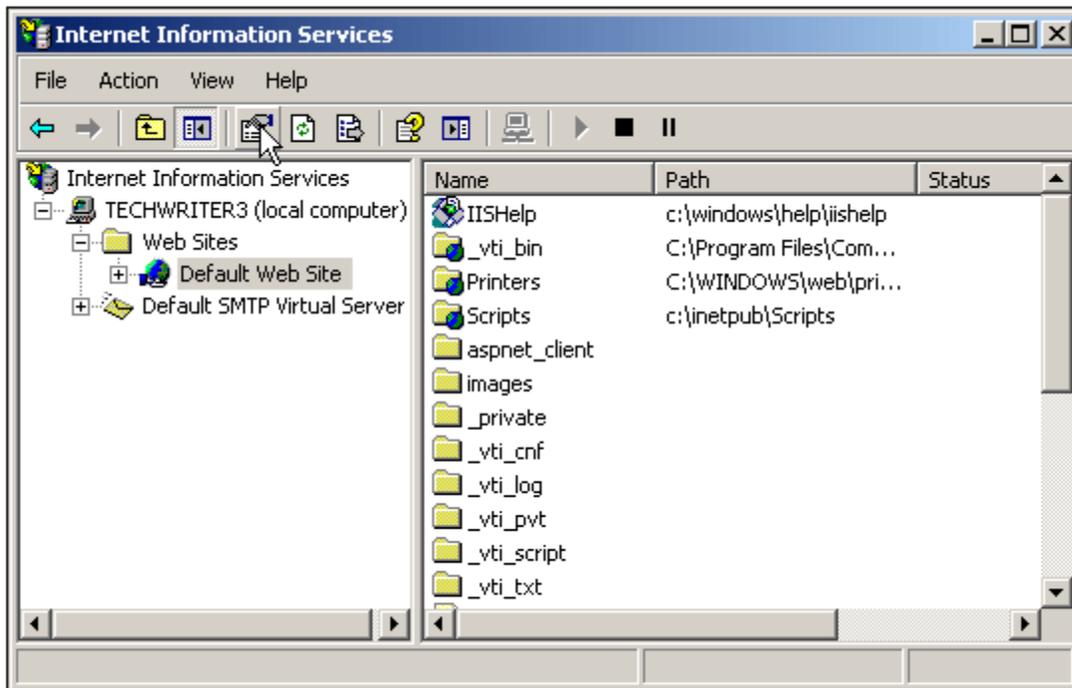
### インターネット情報サービス内のサイトのために MIME の種類を追加する

Windows XP マシンの特定のサイトのための MIME の種類のリストに MIME の種類を追加するには、次を行います。他のサポートされるシステム(プラットフォーム更新済みの Windows 7 SP1、Windows 8、Windows 10) と手順は類似しています。

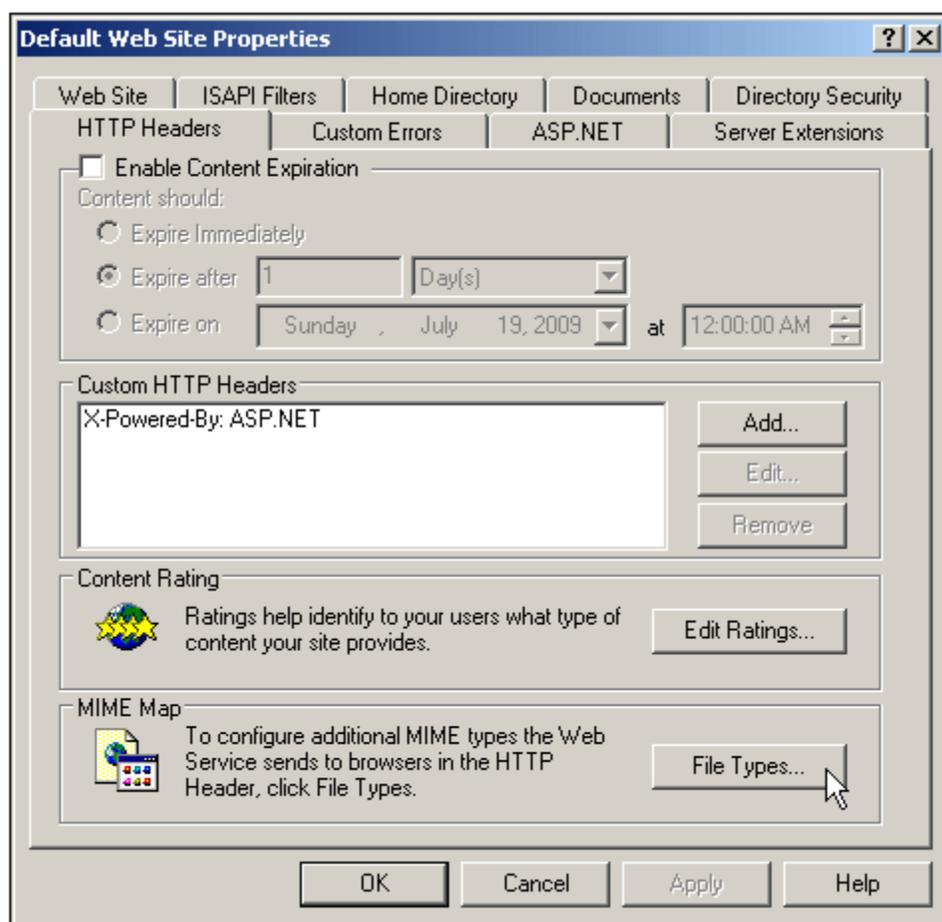
1. コントロールパネルを開き、管理ツールをダブルクリックします。
2. 表示されるフォルダー内でインターネットの情報サービスをダブルクリックします(下のスクリーンショット)。



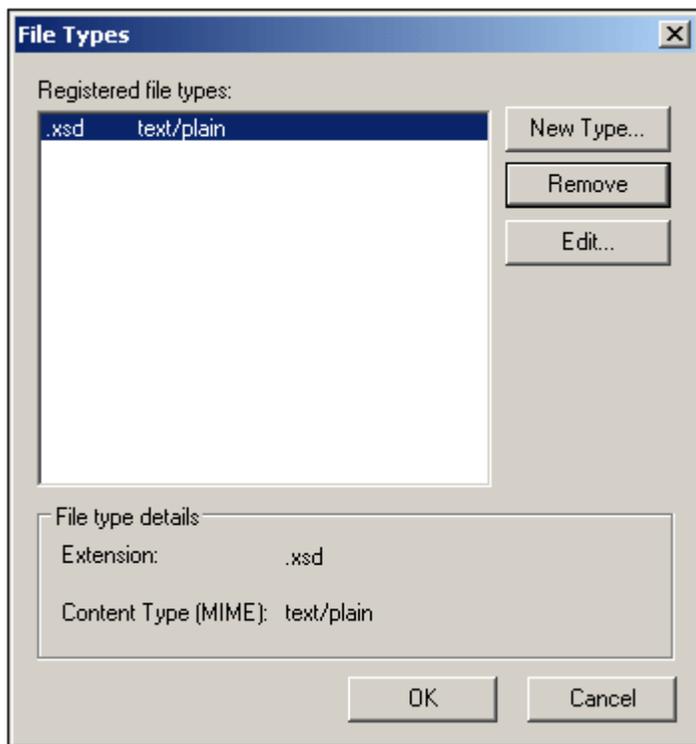
3. 表示されるインターネット情報サービス(IIS) フォルダー内で左側のフォルダーペイン内から必要とされるサイト (Web サイト またはフォルダー) を選択し、(下のスクリーンショット内のカーソルの下) プロパティ アイコンをクリックします、または、コンテキストメニューから右クリックしてアクセスすることのできるプロパティ コマンドをクリックします(下のスクリーンショット)。



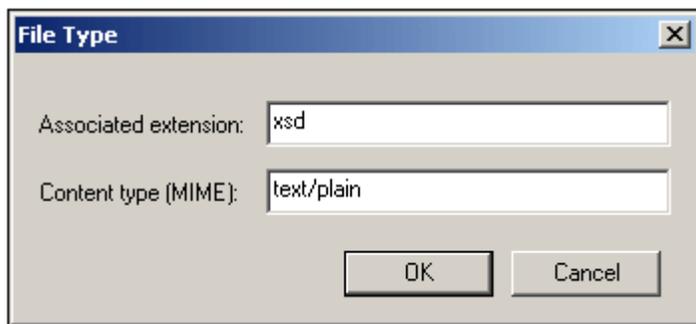
4. プロパティダイアログのHTTP ヘッダータブ内で、MIME Map ペイン内の「ファイル型」ボタンをクリックします（下のスクリーンショットを参照）。



5. ファイルの型 ダイアログ内で「新しい型」ボタンをクリックします(下のスクリーンショット)。



6. ポップアップされるダイアログ内に必要とされる拡張子とMIMEの種類を入力します。次を参照してください！必要とされる対応するMIMEの種類のためは上のテーブルを参照してください。



7. 「OK」を押して確認します。

Web-DAV を使用して内部のサーバー上でフォルダーに遠隔のアクセスをセットアップする方法に関しては、[次を参照してください！ Windows IT プロの記事](#)。

## 2.2 XSD、XML、と SPS/PXF ファイル

Authentic Browser プラグインプロジェクトの実装の主な目的は、Authentic View 書式内でXMLドキュメントをビューし編集することです。この書式はXMLドキュメントのXMLマークアップを隠し、カスタムデザインされたWYSIWYGインターフェイス内でドキュメントの編集を可能にします。Authentic View 内のXMLドキュメントのレイアウトとデータ入力メカニズムはSPSファイル内で定義されています。SPSとXMLドキュメントがベースとしているSPSファイル、XMLファイルとXMLスキーマファイル(XSDファイル)は、Authentic Browser クライアントがアクセスできるようにネットワーク上に保管されます。

### SPS ファイル

SPSファイルは[Altova のStyleVision 製品](#)内で作成されます。XMLファイルがベースとする同じXMLスキーマをベースにしています。StyleVision内では、SPSデザイナーはドラッグアンドドロップとペーレイアウトメカニズムを使用してSPSをデザインすることができます。SPSファイルは、Authentic Viewで表示される際のXMLドキュメントのレイアウトとデータ入力メカニズムを決定します。

[Authentic プラグインのためのHTML ページ](#)内では、XMLファイル、SPSファイル、およびXSDファイルの場所は指定されます。(HTMLページ内の) Authentic View ウィンドウにXMLファイルがロードされ、SPSファイル内で指定されているデザインに従って表示されます。XMLファイルとSPSファイルは両方とも同じXMLスキーマをベースにしています(XMLファイルはスキーマSPSが使用する大きなスキーマの一部をベースにすることもできます)。

**メモ** Authentic Browser が正確に動作するには、SPSファイル内でスキーマがメインスキーマとして割り当てられていることを確認してください。%SVPS%>(SPS)の作成に関する情報は、次を参照してください! [Altova のStyleVision 製品ユーザーマニュアル](#)。StyleVision 製品の説明を [Altova Web サイト](#) で読むことができます。

### XML ファイル

Authentic View ユーザーは[Authentic プラグインのためのHTML ページ](#)にアクセスしてクライアントマシン上のドキュメントを編集することができます。XMLファイルがロードされるAuthentic View ウィンドウがこのページに含まれています。Authentic View ユーザーはクライアントブラウザ内でドキュメントを編集し、XMLドキュメントに変更を戻し保存します。

### XSD ファイル

XSDファイルには2つの目的があります。XMLドキュメントを検証するために使用され、SPSがベースとするジェネリック名ドキュメント構造を提供します。XMLファイルとSPSファイルは両方ともXSDファイルを参照します。ですから、XSDファイルはXMLとSPSファイルにより参照されると正確な場所に保管される必要があります。

### XSD、XML と SPS/PXF ファイルのストレージの場所

XSD、XML、とSPS(またはPXF)ファイルは全てのクライアントマシンによりアクセスすることのできるネットワーク上のロケーションに保存されている必要があります。Authentic Browser サーバー上の場所が最適な場所です。サーバー上の単一のフォルダー内のファイルをロケートし、相対的なパスとして互いの参照を指定することが奨励されます。例えば、XMLファイルをXSDレファレンスを使用して作成し、相対的なパスとして指定します。

### DBベースのSPSに関するメモ

If Authentic Browser を使用してDBベースのStyleVision Power Stylesheet (SPS) を使用するデータベース(DB)を表示、または、編集する場合、データベースに正確に接続することを保証するため次の設定をおこなう必要があります。

**SPS 内の接続情報:** DBへ接続するために必要な全ての情報は、SPS内の接続文字列内に保管されています。StyleVision Power Stylesheet を作成する際にSPS内の接続文字列はStyleVision内で作成されます。MS Access DBへの接続で使用されるメカニズムは他のデータベースのために使用されるメカニズムとは異なります。他のDBのために、ADO接続が使用されます。接続メカニズムのこれらの2つの種類のために行う設定は下で説明されています。

- MS Access DBF のため** MS Access DB のための接続文字列内で、クライアントが正確にデータベースに接続するために、UNC パスが使用される必要があります。StyleVision 内で StyleVision Power Stylesheet がビルドされるとこの UNC パスが指定されます。しかしながら、DB (または、他の祖先フォルダー) を含むフォルダーが共有のためにセットアップされていることを確認してください (Windows XP 内では、共有設定は、フォルダーを右クリックして、「共有とセキュリティ」を選択するとアクセスすることができます)。マシン上では高度なファイル共有を有効化する必要があります (「マイコンピュータ | ツール | フォルダーオプション」シンプルなファイルの共有のチェックを解除します)。クライアント上では必須の設定は存在しません。

**メモ** `servername` がサーバーの名前で、共有されたフォルダーの名前が `sharename` である。接続文字列内で使用される UNC パスの書式: `\\servername\sharename\path\file.mdb` (これは、サーバー上の共有されたフォルダーのために作成する共有設定内で指定されています)。path は DB へのパスで、file.mdb は共有されたフォルダーの共有されたフォルダーまたは子孫フォルダー内の MS Access DB の名前です。

- ADO 接続のため** StyleVision 内で SPS がビルドされる際に、SPS 内の ADO 接続文字列が指定されます。セキュリティ情報を含む必要とされる接続情報が含まれています。StyleVision 内で接続文字列をテストする際に使用されるドライバが使用されていることを確認してください。また、Authentic Browser をホストする全てのクライアントマシンにインストールされていることを確認してください。これにより SPS がクライアントからデータベースに正確に接続することができます。ADO データベース接続はローカルファイルパスのみで作動し、`http://` URL を使用しては作動しないことご注意ください。

### PXF ファイルに関するメモ

XSLT 2.0 を使用する SPS デザインを Power XML Form (PXF) ファイルとして保存することができます。Altova により、PXF ファイル書式は、関連したファイルと共にパッケージ SPS デザインに特別に開発されました。(デザイン内で使用される、スキーマファイル、ソース XML ファイル、イメージファイル、および、ソース XML の出力フォーマットへの変換のための XSLT ファイル)。PXF ファイル書式の利点は、Authentic View 編集のために必要とされる全てのファイル、および、Authentic View からの出力の生成が便利に単一のファイル内でおこなうことができることです。

**メモ** PXF ファイルが Web サーバーに存在する場合、および、Authentic Browser プラグインと使用される場合、サーバーがファイルをブロックしないように確認してください (IIS 管理者パネルを使用して、例えば) PXF (.pxf) ファイルの拡張子のための次の MIME の種類を追加することで行うことができます: `application/x-zip-compressed`。

## 2.3 Authentic プラグインのための HTML ページ

Authentic プラグインのための HTML ページは次の重要な機能を実行します:

1. クライアント上で Authentic プラグインのための HTML ページが最初に開かれると、HTML ページ内のコードは Authentic Browser プラグインがダウンロードされクライアント上にインストールされるようになります。クライアント上にインストールされます。このコードはサーバー上の正確なファイルを正確に識別するように書かれている必要があります。ダウンロードするファイルの書式 (CAB、XPI、または CRX) はクライアントのブラウザにより異なります。Internet Explorer と Firefox 処理コードは異なるため、コードは異なります。
2. HTML ページ内のコードは Authentic View インターフェイスのディメンション、XML、XSD、と SPS ファイルの場所を含むブラウザウィンドウ内の Authentic View インターフェイスをセットアップします。
3. 編集する XML ファイルおよび XML ファイルがベースとするスキーマファイルと SPS を指定します。
4. HTML SCRIPT 要素内で、サブルーチンとイベントの処理のための定義が含まれています。例えば、HTML ページ内のボタンがクリックされると実行されるアクションが指定されます。次を参照してください: [Internet Explorer サンプル1: シンプル](#)と [Firefox サンプル1: シンプル](#)

**メモ** Authentic Browser の Enterprise エディションをデプロイする場合、Enterprise ライセンスが登録されているサーバー上に HTML ページがインストールされる必要があります。

### Authentic Browser の埋め込み

Authentic Browser プラグインを使用する場合は、プラグインを識別するオブジェクトがプラグインをダウンロードする HTML ページ内に埋め込まれている必要があります。次の HTML 要素と属性を使用しておこなうことができます:

- Internet Explorer: `<OBJECT clsid="clsid:<CLSID>" />`
- Firefox: `<embed type="<MIMEType>" />`

使用する CLSID と MIME の種類値に関する詳細は、次を参照してください: [Authentic Browser バージョン](#)。

### このセクション

このセクションのサブセクションは、上リストされる関数かどのように HTML ページ内で実装されるかについて説明しています。これらのセクションはブラウザの型により最初のレベルで整理されています。これは特定のブラウザのために Authentic Browser DLL が異なる方法でダウンロードされるからです:

- [Enterprise エディションのためのライセンス供与](#) では、Authentic Browser の Enterprise エディションのためのライセンスメカニズムについて説明されています。
- [Internet Explorer](#) では、HTML OBJECT 要素を使用して (.cab 拡張子を持つ) CAB ファイル をダウンロードする方法について説明されています。
- [Firefox](#) では、HTML EMBED 要素を使用して、XPI ファイル (拡張子 .xpi) をダウンロードする方法について説明されています。
- [依存しないブラウザ](#) では、リクエストをおこなうブラウザの種類を決定する方法、および、ブラウザのために正確なプラグインバージョンをダウンロードする方法について説明されています。

これらのサブセクション内で、HTML OBJECT、EMBED、と SCRIPT 要素の使用法および Authentic Browser プラグインを呼び出す HTML ページ全体のサンプルに関する説明とサンプルが提供されています。個別のオブジェクトに関する情報に関しては、次を参照してください: [レファレンスセクション内のオブジェクトの説明](#)。

## 2.3.1 Enterprise エディションのためのライセンス供与

Authentic Browser Enterprise Edition のためのライセンスは [Altova Web サイト](#) で購入することができます。

### Enterprise Edition ライセンスのセットアップの概要

Authentic Browser Enterprise Edition のためのライセンス情報を正確にセットアップするためのステップの概要が下のリストに示されています。

- Authentic Browser Enterprise Edition は次の有効なライセンス情報を必要とします: (i) 有効なライセンスのための サーバー名 (ii) ライセンスが登録されている会社名 および (iii) ライセンスキー。この情報は Authentic Browser Enterprise Edition ライセンスを購入する際送信されるライセンス電子メールに送信されます。
- [Authentic プラグインのためのHTML ページ](#) にライセンス情報は存在する必要があります。HTML ページ内で行う方法は下で説明されています (Authentic Browser が参照する固有ライセンスキーファイルは存在しません)。
- [Authentic プラグインのためのHTML ページ](#) 内に与えられるライセンス情報が有効な場合、Authentic Browser ファイル内の Enterprise Edition 機能がアンロックされます。それ以外の場合、機能は Community Edition の機能に制限されています。
- [Authentic プラグインのためのHTML ページ](#) はライセンスが有効なサーバー上に保管されます。

### ライセンスの入力方法に関する情報

3つのライセンスキー/パラメーター (サーバー名、会社名、ライセンスキー) は [Authentic プラグインのためのHTML ページ](#) 内に入力される必要があります。次の方法で行うことができます。

- OBJECT または EMBED 要素のパラメーターの値として。方法は [Internet Explorer](#) と [Firefox](#) のための対応する HTML ページ セクション内で説明されています。
- HTML ページがブラウザに依存しない場合、ライセンス/パラメーターを [ブラウザに依存しないサンプル](#) 内のコードリストで示されるように登録することができます。
- ライセンス/パラメーターは、下に示されるとおり、[ブラウザに依存しないサンプル](#) 内のコードリストを適応するオブジェクトを直接設定することができます。

```
<SCRIPT LANGUAGE=javascript>
// event subscription if running on Firefox
if ( isFirefoxOnWindows() )
{
    objPlugIn.addEventListener("ControlInitialized", InitAuthenticPluginPage, false);
}
</SCRIPT>
<SCRIPT LANGUAGE="javascript" FOR=objPlugIn EVENT="ControlInitialized">
// event subscription if running on Internet Explorer
if ( isIEOnWindows() )
{
    InitAuthenticPluginPage();
}
</SCRIPT>

<SCRIPT type="text/javascript" LANGUAGE="javascript" >
function InitAuthenticPluginPage( )
{
var serverstr='DevAuthBrowTest';
var basedir='Authentic/';
objPlugIn.LicServer = 'DevAuthBrowTest';
objPlugIn.LicCompany = 'Altova';
```

```
objPlugIn.LicKey = 'XXXXXXXXXX';  
objPlugIn.SchemaLoadObject.URL = 'http://' + serverstr + basedir + 'OrgChart.xsd';  
objPlugIn.XMLDataLoadObject.URL = 'http://' + serverstr + basedir + 'OrgChart.xml';  
objPlugIn.DesignDataLoadObject.URL = 'http://' + serverstr + basedir + 'OrgChart.sps';  
objPlugIn.StartEditing();  
}  
</SCRIPT>
```

## 2.3.2 Internet Explorer

Authentic プラグインのための HTML ページが 32 ビット、または 64 ビットの Internet Explorer 内で開かれている場合、次の要素が含まれている必要があります。

- HTML [OBJECT](#) 要素は以下をおこないます (i) Authentic プラグイン (32 ビット、または 64 ビット) のための正確な DLL がサーバーからクライアントにダウンロードされ、(ii) クライアントのブラウザ内の Authentic View ウィンドウのディメンションを指定します。[OBJECT](#) 要素には Authentic Browser プラグインの場所が含まれています。(64 ビット IE 10 と 11 を除く) 32 ビット と 64 ビット Internet Explorer のための使用することのできるバージョン内で Authentic プラグインを使用することができます。正確な Authentic Browser プラグインバージョン (.cab ファイル) をクライアントにダウンロードする必要があります。[ブラウザに依存しないサンプル](#)を参照してください。Internet Explorer の X ビットバージョンを自動的にチェックし、正確な Authentic プラグインをダウンロードします。
- サブルーチンとイベントの処理を定義するための、または複数の HTML [SCRIPT](#) 要素です。SCRIPT 要素は編集する XML ドキュメントと XML スキーマと SPS ファイルがベースとする XML ドキュメントを指定するために使用。

### このセクション

- [OBJECT 要素](#) は HTML [OBJECT](#) 要素が Authentic プラグインのための HTML ページでどのように使用されるかについて説明しています。
- [SCRIPT 要素](#) は HTML [SCRIPT](#) 要素が Authentic プラグインのための HTML ページ内でどのように使用されるかについて説明しています。
- HTML ページ全体のサンプル [IE サンプル1: シンプル](#)と [IE サンプル2: テーブルの並べ替え](#)。

個別のオブジェクトに関する情報に関しては、[レファレンスセクション](#)内の対応する[オブジェクトの説明](#)を参照してください。

メモ Authentic Browser プラグインは Internet Explorer 5.5 または以降 のためにサポートされています。しかしながら、Internet Explorer 10 と 11 の 64 ビットバージョンはサポートされません。

### 2.3.2.1 OBJECT 要素

OBJECT 要素 は次の機能ががあります。

- (id 属性を使用して) Authentic Browser プラグインに名前を与えます
- (classid 属性の値を使用して使用する [Authentic Browser プラグインのバージョン](#) を選択します。
- .CAB ファイルとバージョン番号 (codebase 属性) を指定します。 .CAB ファイルは、 classid 属性 の値である ID と共に COM オブジェクト (Authentic Browser プラグイン) を登録する DLL を含んでいます。通常、Authentic Browser プラグインの 32 ビット と 64 ビットバージョンの両方は、サーバー上に保管されます。各バージョンは異なるファイル名により識別されます。 codebase 属性は必要とされる .CAB ファイルのファイル名を指定します (下の [ロードリスティング参照](#))。 32 ビット と

64ビット .CAB ファイルのクラスID は類似しており、[異なる英語ドイツ語 および信頼されている信頼されていないバージョンのためのテーブル](#)内で説明されているとおりです。

32ビットバージョンのためのファイル名: AuthenticBrowserEdition.CAB

64ビットバージョンのためのファイル名: AuthenticBrowserEdition\_x64.CAB

- (style 属性を使用してクライアントブラウザ内のAuthentic View ウィンドウの定義を指定します。
- パラメーターの数を制限無く指定することができます。

## サンプルOBJECT 要素

サンプルHTML OBJECT 要素は以下のとおりです。信頼されているUnicodeバージョン(classid 属性内で与えられている値と共に)を選択し、クライアントのブラウザ内のAuthentic View ウィンドウのディメンションを600 x 500 ピクセルに設定します。OBJECT 要素が使用することのできる属性とパラメーターは下に説明されています。

**メモ** 下のバージョン番号は現在のバージョンの番号ではありません。詳細に関しては、次を参照してください! [codebase](#)。

### 32ビット Authentic Browser プラグイン

```
<OBJECT id="objPlugIn" style="WIDTH:600px; HEIGHT:500px"
  codeBase="http://yourserver/cabfiles/AuthenticBrowserEdition.CAB#Version=12,3,0,0"
  classid="clsid:B4628728-E3F0-44a2-BEC8-F838555AE780">
  <PARAM NAME="XMLDataURL" VALUE="http://yourserver/OrgChart.xml">
  <PARAM NAME="SPSDataURL" VALUE="http://yourserver/OrgChart.sps">
  <PARAM NAME="SchemaDataURL" VALUE="http://yourserver/OrgChart.xsd">
</OBJECT>
```

### 64ビット Authentic Browser プラグイン

```
<OBJECT id="objPlugIn" style="WIDTH:600px; HEIGHT:500px"
  codeBase="http://yourserver/cabfiles/AuthenticBrowserEdition_x64.CAB#Version=12,3,0,0"
  classid="clsid:B4628728-E3F0-44a2-BEC8-F838555AE780">
  <PARAM NAME="XMLDataURL" VALUE="http://yourserver/OrgChart.xml">
  <PARAM NAME="SPSDataURL" VALUE="http://yourserver/OrgChart.sps">
  <PARAM NAME="SchemaDataURL" VALUE="http://yourserver/OrgChart.xsd">
</OBJECT>
```

#### id

id 属性の値は、スクリプト内で使用される場合、Authentic Browser プラグインオブジェクトの名前として使用されます。例えば、objPlugIn.SchemaLoadObject.URL はスキーマファイルをロードするオブジェクトへの呼び出しです。詳細に関しては、次を参照してください! [SCRIPT 要素](#)。

#### style

これは通常のHTML style 属性であり、クライアントブラウザ内のAuthentic View ウィンドウのディメンションを指定するために使用されます。

#### codebase

codebase 属性は .CAB ファイルの場所を提供します。32ビット Authentic Browser プラグインと64ビット Authentic Browser プラグインのために異なる .CAB ファイルが存在し、それぞれ異なる名前を持つことにご注意ください!

AuthenticBrowserEdition.CAB とAuthenticBrowserEdition\_x64.CAB.

任意の #Version 拡張子の値は、現在サーバー上で使用することのできるコンポーネントのバージョン番号を与えます。クライアントに以前のバージョンがインストールされており、codebase 属性内で更に新しいバージョンが指定されている場合、サーバーから新しいバージョンがインストールされます。#Version 拡張子が指定されていない場合、クライアントから手動でコンポーネントが削除されるまで更新は行われません。コンポーネントの現在のバージョンはコンポーネントの .CAB ファイルの .d11 ファイルのプロパティと共にリストされています（ファイルを右クリックしてプロパティコマンドを選択してください）。

#### classid

32ビットと64ビット .CAB ファイルのクラスID は類似しており、[Authentic Browser バージョン](#)内で説明されているとおりです。

ブラウザプラグインバージョン 5.0 から、Unicode バージョンの classid 値は、バージョン 5.0 より前の Unicode バージョンとは異なります。5.0 前のバージョンからサーバー上で .CAB ファイルを更新する場合、HTML ファイル内の classid 値を変更することを忘れないでください。サーバー上の新規の .CAB ファイルが、.CAB ファイルと同じ CLSID を持つ場合、新規の .CAB ファイルはクライアント上の古いものと自動的に置き換えられないことにご注意してください。

#### パラメーター

次のパラメーターを使用することができます。

##### LicServer

Authentic Browser Enterprise Edition ライセンスキーが有効なサーバーの名前です。Authentic Browser Community Edition のためライセンスキーは必要ありません。

##### LicKey

Authentic Browser Enterprise Edition の使用を検証するためのライセンスキーです。Authentic Browser Community Edition のためライセンスキーは必要ありません。

##### LicCompany

Authentic Browser Enterprise Edition の使用を検証するための会社名です。Authentic Browser Community Edition のためライセンスキーは必要ありません。

##### XMLDataURL

編集される XML ファイルの場所を与える絶対 URL です。信頼されていないバージョンに関しては、フルローカルパスを使用することができます。

##### XMLDataSaveURL

保存される XML ファイルの場所を与える絶対 URL です。信頼されていないバージョンに関しては、フルローカルパスを使用することができます。

##### SPSDataURL

StyleVision Power Stylesheet (.sps ファイル) の場所を与える絶対 URL です。信頼されていないバージョンに関しては、フルローカルパスを使用することができます。

##### SchemaDataURL

関連したスキーマファイルの場所を与える絶対 URL です。信頼されていないバージョンに関しては、フルローカルパスを使用することができます。

##### TextStateBmpURL

テキスト状態アイコンのためのビットマップイメージが保管されるフォルダーです。

##### TextStateToolbarLine

テキスト状態アイコンが配置されるツールバーラインです。デフォルトは 1 です。

##### AutoHideUnusedCommandGroups

使用されていないツールバーコマンドグループが非表示になるかを決定します。デフォルトは True です。

**ToolbarsEnabled**

ツールバーのための一般的なサポートを指定します。デフォルトは True です。

**ToolbarTooltipsEnabled**

ヒントが有効化されるかを指定します。

**HideSaveButton**

True に設定されている場合、デフォルトでは表示されている Authentic ツールバーから保存ボタンを削除します。

**BaseUrl**

相対的なパスと使用されるベース URL を与えます。

**SaveButtonUsePOST**

True に設定されている場合、ドキュメントを保存する際に、HTTP POST コマンドが PUT の代わりに使用されます。

**EntryHelpersEnabled**

True に設定されている場合、Authentic 入力ヘルパーが表示されます。

**EntryHelperSize**

ピクセルで表示された入力ヘルパーウィンドウの幅

**EntryHelperAlignment**

ドキュメント ウィンドウ相対的な入力ヘルパーの場所を指定します。

- 0 = ドキュメントの上の部分にツールバーを配置します
- 1 = ドキュメントの左の部分にツールバーを配置します
- 2 = ドキュメントの下の部分にツールバーを配置します
- 3 = ドキュメントの右の部分にツールバーを配置します

**EntryHelperWindows**

表示される入力ヘルパーサブウィンドウを選択します。

- 1 = 要素
- 2 = 属性
- 4 = エンティティ

全ての組み合わせが許可されています (ビットの確認)

**SaveButtonAutoEnable**

[Authentic.SaveButtonAutoEnable](#) を参照してください

**LoaderSettingsFileURL**

パッケージの管理のための LoaderSettingsFile の URL を与えます。

## 2.3.2.2 SCRIPT 要素

SCRIPT 要素は HTML ファイル内から呼び出すことができるイベントハンドラーとサブルーチンを定義します。

イベントを扱うためのサンプルの SCRIPT は、以下のとおりです:

```
<SCRIPT LANGUAGE="javascript" FOR=objPlugin EVENT="ControlInitialized">  
  objPlugin.SchemaLoadObject.URL = "http://yourserver/OrgChart.xsd"
```

```
objPlugIn.XMLDataLoadObject.URL = "http://yourserver/OrgChart.xml"  
objPlugIn.DesignDataLoadObject.URL = "http://yourserver/OrgChart.sps"  
objPlugIn.StartEditing  
</SCRIPT>
```

サブルーチンを含むスクリプトのサンプルは、以下のとおりです:

```
<SCRIPT ID=clientEventHandlers LANGUAGE=vbscript>  
  Sub BtnOnClick  
    objPlugIn.SchemaLoadObject.URL = "http://yourserver/OrgChart.xsd"  
    objPlugIn.XMLDataLoadObject.URL = "http://yourserver/OrgChart.xml"  
    objPlugIn.DesignDataLoadObject.URL = "http://yourserver/OrgChart.sps"  
    objPlugIn.StartEditing  
  End Sub  
  Sub OnClickFind  
    objPlugIn.FindDialog  
  End Sub  
  Sub BtnOnTestProp  
    If objPlugIn.IsRowInsertEnabled Then  
      MsgBox "true"  
    Else  
      MsgBox "false"  
    End If  
  End Sub  
</SCRIPT>
```

### スクリプト言語

Authentic Browser プラグインは JavaScript と VBScript を使用してテスト済みです。

### イベントの処理

HTML ボディ内の OBJECT 要素の ID 属性の値は、FOR 属性の値として指定されます。呼び出される Authentic Browser プラグイン オブジェクトはこの値である名前を持つ必要があります。

イベントのリストに関しては、次を参照してください! [イベント: レファレンス](#)。

### サブルーチン

HTML ファイル内で定義するイベントのためにサブルーチンを作成することができます。Authentic Browser プラグイン オブジェクト名は HTML ボディ内の OBJECT 要素の ID 属性の値と同じである必要があります。上のサンプル内では、プレフィックスは OBJECT 要素の ID 属性の値である objPlugIn です。

Authentic Browser プラグイン内で使用することのできるメソッド、プロパティ、サブオブジェクトはこのドキュメントの [レファレンスセクション](#)で説明されています。

## 2.3.2.3 IE サンプル 1: シンプル

下の HTML コードは次の機能を持つページを生成します:

- インストールされていない場合、クライアント上に Authentic Browser の信頼されている Unicode バージョンをインストールします。
- ボディは Authentic Browser がロードされる 600px 幅と 500px 高さのウィンドウが含まれています。
- Authentic Browser ウィンドウの下は 4 つのボタンの行です。
- OrgChart.xml の Authentic View がロードされます。
- 「検索と置換」ボタンは検索と置換 ダイアログをそれぞれ表示します

- 「保存」ボタンはサーバーのルートディレクトリ内にある SaveFile.xml という名前のファイルを保存します。
- 「テスト」プロパティボタンはシンプルなプロパティをテストします

このHTML ページがクライアントで開かれると、XML ファイル OrgChart.xml の編集を開始し、編集したファイルを SaveFile.xml として保存することができます。

Authentic Browser が正確に動作するかテストするために、このシンプルな HTML ページを使用することができます。これを行う場合、CAB ファイル、xsd、xml、および sps ファイルとサーバー上の他のリソースをロケートする正確な URL を使用してください。サーバーの一部では大文字と小文字の区別が問題になる場合があります、ファイルを検索する際に問題になる場合がありますので、コード内のファイル名とコマンドの大文字/小文字の指定をチェックしてください。このサンプルを拡張、または変更して、Authentic Browser を使用し、更に複雑なソリューションをビルドすることができます。詳細に関して、次も参照してください！ [OBJECT 要素](#)。

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Minimal XMLSpyDocEditPlugin page</title>

<!-- Script for handling the ControlInitialized event -->
<SCRIPT LANGUAGE="javascript" FOR="objPlugIn" EVENT="ControlInitialized">
  objPlugIn.SchemaLoadObject.URL = "http://yourserver/OrgChart.xsd"
  objPlugIn.XMLDataLoadObject.URL = "http://yourserver/OrgChart.xml"
  objPlugIn.DesignDataLoadObject.URL = "http://yourserver/OrgChart.sps"
  objPlugIn.StartEditing()
</SCRIPT>

<!-- Script with subroutines -->
<SCRIPT ID="clientEventHandlers" LANGUAGE=vbscript>
  Sub OnClickFind
    objPlugIn.FindDialog
  End Sub

  Sub OnClickReplace
    objPlugIn.ReplaceDialog
  End Sub

  Sub BtnOnSave
    objPlugIn.XMLDataSaveUrl = "http://yourserver/SaveFile.xml"
    objPlugIn.Save
  End Sub

  Sub BtnOnTestProp
    If objPlugIn.IsRowInsertEnabled Then
      msgbox "true"
    Else
      msgbox "false"
    End If
  End Sub
</SCRIPT>
</head>

<body>
<!-- Object element has id with value that must be used -->
<!-- as name of Authentic Browser Plug-in objects -->
<!-- Classid selects the Trusted Unicode version -->
<OBJECT id="objPlugIn"
  <!-- CodeBase selects 32-bit CAB file (AuthenticBrowserEdition.CAB) -->
  <!-- or 64-bit CAB file (AuthenticBrowserEdition_x64.CAB) -->
  CodeBase="http://yourserver/AuthenticBrowserEdition.CAB#Version=12,3,0,0"
  <!-- Class Id for 32-bit and 64-bit CAB files is the same -->
  ClassId="clsid:B4628728-E3F0-44a2-BEC8-F838555AE780" width="600" height="500">
</OBJECT>
<p>
<input type="button" value="Find" name="B4" onclick="OnClickFind()">
```

```

<input type="button" value="Replace" name="B5" onclick="OnClickReplace()">
<input type="button" value="Save" name="B6" onclick="BtnOnSave()">
<input type="button" value="Test property" name="B7" onclick="BtnOnTestProp">
</p>
</body>
</html>

```

### 2.3.2.4 IE サンプル 2: テーブルの並べ替え

埋め込まれた JavaScript を持つサンプル HTML ページです。サンプルは使用中のコンピュータにインストールするために Authentic Browser プラグイン (CAB ファイル) を必要とします。サーバーの一部では大文字と小文字の区別が問題となる場合があります。ファイルを検索する際に問題となる場合がありますので、コード内のファイル名とコマンドの大文字/小文字の指定をチェックしてください。

コードは以下を表示します:

- ブラウザープラグインにアクセスする方法。ブラウザープラグインバージョン (信頼されている、または信頼されていない) の CAB ファイルとクラス識別子 (CLSID) を参照するためコードを変更します。
- ブラウザープラグインファイルをロードする方法。サンプルドキュメントを参照するためコードを変更します。
- 簡単なカーソルの配置のためのボタンの実装の方法。
- テーブルの並べ替えなど更に複雑なコマンドの実装の方法。
- SelectionChanged イベントの使用の方法。

詳細に関して、次も参照してください! [OBJECT 要素](#)。

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>test page For Authentic Browser プラグイン</title>

<SCRIPT LANGUAGE="javascript" For="objPlugIn" EVENT="ControlInitialized">
var strSampleRoot = "http://myRoot/myPath/myDocBaseName";
objPlugIn.SchemaLoadObject.URL = strSampleRoot + ".xsd";
objPlugIn.XMLDataLoadObject.URL = strSampleRoot + ".xml";
objPlugIn.DesignDataLoadObject.URL = strSampleRoot + ".sps";
objPlugIn.StartEditing();
</SCRIPT>

<SCRIPT ID="clientEventHandlers" LANGUAGE="javascript">
var objCurrentRange = Null;

Function BtnDocumentBegin() { objPlugIn.AuthenticView.DocumentBegin.Select(); }
Function BtnDocumentEnd() { objPlugIn.AuthenticView.DocumentEnd.Select(); }
Function BtnWholeDocument() { objPlugIn.AuthenticView.WholeDocument.Select(); }
Function BtnSelectNextWord() { objPlugIn.AuthenticView.Selection.SelectNext(1).Select(); }
Function BtnSortDepartmentOnClick()
{
var objCursor = Null;
var objTableStart = Null;
var objBubble = Null;
var strField1 = "";
var strField1 = "";
var nColIndex = 0;
var nRows = 0;

objCursor = objPlugIn.AuthenticView.Selection;
If (objCursor.IsInDynamicTable())
{

```

```

        // calculate current column index
        nColIndex = 0;
        While (True)
        {
            try { objCursor.GotoPrevious(11); }
            catch (err) { break; }
            nColIndex++;
        }

        // GoTo begin of table
        objTableStart = objCursor.ExpandTo(9).CollapsToBegin().Clone();

        // count number of table rows
        nRows = 1;
        While (True)
        {
            try { objTableStart.GotoNext(10); }
            catch (err) { break; }
            nRows++;
        }

        // bubble sort through table
        For (var i = 0; i < nRows - 1; i++) {
            for(var j = 0; j < nRows-i-1; j++) {
                objBubble = objCursor.ExpandTo(9).CollapsToBegin().Clone();
                // Select correct column in jth table row
                objBubble.GotoNext(6).Goto(10,j,2).Goto(11,nColIndex,2).ExpandTo(6);
                strField1 = objBubble.Text;
                strField2 = objBubble.GotoNext(10).Goto(11,nColIndex,2).ExpandTo(6).Text;
                if(strField1 > strField2) {
                    if(!objBubble.MoveRowUp()) {
                        alert("Table row move is not allowed!");
                        return;
                    }
                }
            }
        }
    }
}
</SCRIPT>
</head>

<body>
    <Object id="objPlugIn"
    <!-- CodeBase selects 32-bit CAB file (AuthenticBrowserEdition.CAB) -->
    <!-- or 64-bit Cab file (AuthenticBrowserEdition_x64.CAB) -->
    codeBase="http://myCabfileLocation/AuthenticBrowserEdition.CAB#Version=12,3,0,0"
    <!-- Class Id for 32-bit and 64-bit CAB files is the same -->
    classid="clsid:B4628728-E3F0-44a2-BEC8-F838555AE780"
    width="100%"
    height="80%"
    VIEWASTEXT>
    <PARAM NAME="EntryHelpersEnabled" VALUE="TRUE">
    <PARAM NAME="SaveButtonAutoEnable" VALUE="TRUE">
</Object>
<TABLE>
    <TR>
        <TD><Input Type="button" value="Goto Begin" id="B1" onclick="BtnDocumentBegin()"></TD>
        <TD><Input Type="button" value="Goto End" name="B2" onclick="BtnDocumentEnd()"></TD>
        <TD><Input Type="button" value="Whole Document" name="B3" onclick="BtnWholeDocument()"></TD>
        <TD><Input Type="button" value="Select Next Word" name="B4" onclick="BtnSelectNextWord()"></TD>
    </TR>
</TR>

```

```
<TD><Input Type="button" value="Sort Table by this Column" id="B6" onclick="BtnSortDepartmentOnClick()"></TD>
</TR>
</TABLE>
<TABLE id=SelTable border=1>
  <TR><TD id=SelTable_FirstTextPosition></TD><TD id=SelTable_LastTextPosition></TD></TR>
  <TR><TD id=SelTable_FirstXMLData></TD><TD id=SelTable_FirstXMLDataOffset></TD></TR>
  <TR><TD id=SelTable_LastXMLData></TD><TD id=SelTable_LastXMLDataOffset></TD></TR>
  <TR><TD id=SelTable_Text></TD></TR>
</TABLE>
</body>

<SCRIPT LANGUAGE=javascript For=objPlugIn EVENT=selectionchanged>
var CurrentSelection = Null;
CurrentSelection = objPlugIn.AuthenticView.Selection;
SelTable_FirstTextPosition.innerHTML = CurrentSelection.FirstTextPosition;
SelTable_LastTextPosition.innerHTML = CurrentSelection.LastTextPosition;
SelTable_FirstXMLData.innerHTML = CurrentSelection.FirstXMLData.Parent.Name;
SelTable_FirstXMLDataOffset.innerHTML = CurrentSelection.FirstXMLDataOffset;
SelTable_LastXMLData.innerHTML = CurrentSelection.LastXMLData.Parent.Name;
SelTable_LastXMLDataOffset.innerHTML = CurrentSelection.LastXMLDataOffset;
</SCRIPT>
</html>
```

### 2.3.3 Firefox

Authentic プラグインのための HTML ページが Firefox 内で開かれる場合、次の要素が含まれている必要があります:

- HTML [EMBED](#) 要素は以下を行います: (i) Authentic プラグインのための DLL がサーバーからクライアントにダウンロードされるように、(ii) クライアントのブラウザ内の Authentic View ウィンドウのディメンションを指定します。EMBED 要素は Authentic Browser プラグインの場所を含みます。
- サブルーチンとイベントの処理を定義するための [つまみ](#)は複数の [イベントリスナー](#)です。イベントリスナーは SCRIPT 要素内で構成されます。

#### このセクション

このセクションは次のサブセクションに整理されています:

- [EMBED 要素](#) は HTML [EMBED](#) 要素が Authentic プラグインのための HTML ページでどのように使用されるかについて説明しています。
- [イベントリスナーの追加](#) は [イベントリスナー](#) が Authentic プラグインのための HTML ページ内でどのように使用されるかについて説明しています。
- HTML ページ全体のサンプル: [Firefox サンプル1: シンプル](#)と[Firefox サンプル2: テーブルの並べ替え](#)。

個別のオブジェクトに関する情報に関しては、[レファレンスセクション](#)内の対応する[オブジェクトの説明](#)を参照してください。

#### メモ

- Authentic Browser プラグインは **Firefox** のためにサポートされています。
- および [つまみ](#)は CRX ファイルのための MIME の種類がサーバー上の使用するサイトのための MIME の種類のリスト [ブラウザデータベースに追加されていることを確認してください](#)。

### 2.3.3.1 EMBED 要素

EMBED 要素には次の機能が搭載されています:

- (id 属性を使用して) Authentic Browser プラグインに名前を与えます。
- (type 属性を使用して)は使用する [Authentic Browser プラグインのバージョン](#) を選択します。
- PluginsPage 属性を使用して、(Authentic Browser プラグイン DLL である) XPI ファイルまたは CRX ファイルを指定します。属性の値は XPI または CRX ファイルをロケートするパスです。
- (height と width 属性を使用して) クライアントブラウザ内の Authentic View ウィンドウの定義を指定します。

下はサンプル HTML EMBED 要素です。英語言語の信頼されているバージョンを (type 属性の値と共に) 選択し、クライアントのブラウザ内の Authentic View ウィンドウの幅と高さをそれぞれ 100% と 60% に設定します。

#### EMBED 要素のサンプル

```
<embed
  id="objPlugIn"
  type="application/x-authentic-scriptable-plugin"
  width="100%"
  height="60%"
  PluginsPage="http://your-server-including-path/AuthenticFirefoxPlugin_trusted.xpi"
  [Name-Of-Parameter="Value of Parameter"] />
```

#### id

id 属性の値は、スクリプト内で使用される場合、Authentic Browser プラグインオブジェクトの名前として使用されます。例えば objPlugIn.SchemaLoadObject.URL はスキーマファイルをロードするオブジェクトへの呼び出しです。

#### type

この値は、必要とされる [Authentic Browser バージョン](#) の MIME の種類です。上のロードリストでは、値 application/x-authentic-scriptable-plugin は、英語言語の信頼されているバージョンを識別します。次を参照してください。使用することのできるバージョンのリストと MIME の種類のための [Authentic Browser バージョン](#)。

#### width, height

これらの属性はブラウザウィンドウ内で作成される Authentic View ウィンドウのディメンションを指定します。

#### PluginsPage

この属性の値は Authentic Browser XPI ファイル (Firefox のための) サーバー上の場所を指定します。XPI ファイルをロケートするために URL 内の正確なパスを使用してください。サーバーの一部では、大文字と小文字の区別が問題になる場合がありますので、ファイルを検索する際に問題が発生すると、パスとファイル名の大文字と小文字を確認してください。

#### LicServer

Authentic Browser Enterprise Edition ライセンスキーが有効なサーバーの名前です。

#### LicKey

Authentic Browser Enterprise Edition の使用を検証するためのライセンスキーです。

#### LicCompany

Authentic Browser Enterprise Edition の使用を検証するための会社名です。

## パラメーター

パラメーターは次の使用方法があります(下のリスト参照) :

- **EMBED** 要素の属性値ペアとしてパラメーター名とその値を与えます。例えば、属性値ペア `ToolbarsEnabled="true"` を EMBED 要素に追加することによりパラメーター `ToolbarsEnabled true` の値と共に指定することができます。上のリストを参照してください。
- **PARAM** 要素はサンプル内で示されるようにオブジェクト要素の子として指定することができます:

```
<object id="objPlugIn"
        type="application/x-authentic-scriptable-plugin"
        width="100%"
        height="60%" >
  <param name="ToolbarsEnabled"
        value="true"/>
</object>
```

パラメーターを OBJECT 要素の子として指定する場合、Firefox は OBJECT 要素の `PLUGINSOURCE` 属性を受け入れないため Firefox はサーバー上の Authentic Browser XPI ファイルへの参照を持たず、クライアント上で Authentic Browser プラグインのインストールを開始することができず、という難点が存在します。このため、パラメーターのこのような指定方法はプラグインが既にインストールされているクライアント上でのみ使用することができます。プラグインが既にインストールされているクライアント上でのみこの方法でパラメータを指定することができます。

### XMLDataURL

編集される XML ファイルの場所を与える絶対 URL です。信頼されていないバージョンに関しては、ローカルパスを使用することができます。

### XMLDataSaveURL

保存される XML ファイルの場所を与える絶対 URL です。信頼されていないバージョンに関しては、ローカルパスを使用することができます。

### SPSDataURL

StyleVision Power Stylesheet (.sps ファイル) の場所を与える絶対 URL です。信頼されていないバージョンに関しては、ローカルパスを使用することができます。

### SchemaDataURL

関連したスキーマファイルの場所を与える絶対 URL です。信頼されていないバージョンに関しては、ローカルパスを使用することができます。

### TextStateBmpURL

テキスト状態アイコンのためのビットマップイメージが保管されるフォルダーです。

### TextStateToolbarLine

テキスト状態アイコンが配置されるツールバーラインです。デフォルトは 1 です。

### AutoHideUnusedCommandGroups

使用されていないツールバーコマンドグループが非表示になるかを決定します。デフォルトは True です。

### ToolbarsEnabled

ツールバーのための一般的なサポートを指定します。デフォルトは True です。

### ToolbarTooltipsEnabled

ヒントが有効化されたか否かを指定します。

**HideSaveButton**

True に設定されている場合、デフォルトでは表示されている Authentic ツールバーから保存ボタンを削除します。

**BaseUrl**

相対的なパスと使用されるベース URL を与えます。

**SaveButtonUsePOST**

True に設定されている場合、ドキュメントを保存する際に、HTTP POST コマンドが PUT の代わりに使用されます。

**EntryHelpersEnabled**

True に設定されている場合、Authentic 入力ヘルパーが表示されます。

**EntryHelperSize**

ピクセルで表示された入力ヘルパーウィンドウの幅

**EntryHelperAlignment**

ドキュメント ウィンドウ相対的な入力ヘルパーの場所を指定します。

- 0 = ドキュメントの上の部分にツールバーを配置します
- 1 = ドキュメントの左の部分にツールバーを配置します
- 2 = ドキュメントの下の部分にツールバーを配置します
- 3 = ドキュメントの右の部分にツールバーを配置します

**EntryHelperWindows**

表示される入力ヘルパーサブウィンドウを選択します。

- 1 = 要素
- 2 = 属性
- 4 = エンティティ

全ての組み合わせが許可されています (ビットの確認)

**SaveButtonAutoEnable**

[Authentic.SaveButtonAutoEnable](#) を参照してください

**LoaderSettingsFileURL**

パッケージの管理のための LoaderSettingsFile の URL を与えます。

### 2.3.3.2 イベントリスナーの追加

次の SCRIPT 要素はイベントリスナーを定義し、プラグインオブジェクトと登録します。イベントリスナー関数は指定されたイベントが内部のプラグインをトリガーすると呼び出されます。

```
<SCRIPT LANGUAGE="javascript">
var selCount = 0;
function OnSelectionChanged()
{
    selCount = selCount + 1;
    selectionCounter.value = "SelectionCount = " + selCount;
}
var objPlugIn = document.getElementById('objPlugIn');
objPlugIn.addEventListener("selectionchanged", OnSelectionChanged, false)
</SCRIPT>
```

イベントのリストに関しては、次を参照してください！ [イベント: レファレンス](#)。

### スクリプト言語

Authentic Browser プラグインはJavaScript とVBScript を使用してテスト済みです。

### イベントリスナー

イベントリスナーに関する詳細は、関連する [W3C Recommendation](#) を参照してください。

### Authentic Browser オブジェクトモデル

Authentic Browser プラグイン内で使用することのできるメソッド、プロパティ、及びサブオブジェクトに関しては、このドキュメントの [レファレンスセクション](#) で説明されています。

## 2.3.3.3 Firefox サンプル 1: シンプル

下のHTML コードは次の機能を持つページを生成します:

- インストールされていない場合、クライアント上のFirefox のためにAuthentic Browser の信頼されているバージョンをインストールします。
- ページ内のAuthentic Browser ウィンドウはブラウザウィンドウの100% の幅、および60% の高さで設定されています。
- Authentic Browser ウィンドウの下は4つのボタンの行です。
- OrgChart.xml のAuthentic View がロードされます。
- 「検索と置換」ボタンは検索と置換 ダイアログをそれぞれ表示します
- 「保存」ボタンはサーバーのルートディレクトリ内にあるSaveFile.xml という名前のファイルを保存します。
- 「テスト」プロパティ ボタンはサンプルはプロパティをテストします。

このHTML ページがクライアントで開かれると、XML ファイルOrgChart.xml の編集を開始し、編集したファイルを SaveFile.xml として保存することができます。

Authentic Browser が正確に動作するかテストするために、このシンプルなHTML ページを使用することができます。これを行う場合、XPI ファイル、xsd、xml、およびsps ファイル、およびサーバー上の他のリソースをロードする URL 内のそれぞれのファイルに対して正確なアドレスとパスを使用してください。サーバーの一部では大文字と小文字の区別が問題になる場合があります、ファイルを検索する際問題になる場合がありますので、コード内のファイル名とコマンドの大文字/小文字の指定をチェックしてください。このサンプルを拡張、または変更して、Authentic Browser を使用し、更に複雑なソリューションをビルドすることができます。

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
  <title>Minimal Authentic Browser PlugIn page</title>
</head>
<!-- to disable the fast-back cache in Firefox, define an unload handler -->
<BODY id="bodyId" onunload="Unload()">
  <!-- Embed element has id with value that must be used -->
  <!-- as name of Authentic Browser プラグイン objects -->
  <!-- type selects the Trusted Unicode version -->
  <embed
    id="objPlugIn"
    type="application/x-authentic-scriptable-plugin"
    width="100%"
    height="60%"
    PLUGINSOURCE="http://your-server-including-path/AuthenticFirefoxPlugin_trusted.xpi"
    LicServer="DevAuthBrowTest"
    LicCompany="Altova"
```

```
LicKey="XXXXXXXX" />
<!-- Script with subroutines -->
<SCRIPT LANGUAGE="javascript">
    var objPlugIn = document.getElementById('objPlugIn');
    function OnClickFind()
    {
        objPlugIn.FindDialog();
    }

    function OnClickReplace()
    {
        objPlugIn.ReplaceDialog();
    }

    function BtnOnSave()
    {
        objPlugIn.XMLDataSaveUrl = "http://your-server/Authentic/SaveFile.xml"
        objPlugIn.Save()
    }

    function BtnOnTestProp()
    {
        alert ( objPlugIn.IsRowInsertEnabled );
    }

    function Unload()
    {
    }

    function InitAuthenticPluginPage( )
    {
        var serverstr='your-server/';
        var basedir='Authentic/';
        objPlugIn.SchemaLoadObject.URL = 'http://' + serverstr + basedir + 'OrgChart.xsd';
        objPlugIn.XMLDataLoadObject.URL = 'http://' + serverstr + basedir + 'OrgChart.xml' ;
        objPlugIn.DesignDataLoadObject.URL = 'http://' + serverstr + basedir +
'OrgChart.sps';
        objPlugIn.StartEditing();
    }
    // event subscription if running on Firefox
    objPlugIn.addEventListener("ControlInitialized", InitAuthenticPluginPage, false);
</SCRIPT>
<p>
<input type="button" value="Find" name="B4" onclick="OnClickFind()">
<input type="button" value="Replace" name="B5" onclick="OnClickReplace()">
<input type="button" value="Save" name="B6" onclick="BtnOnSave()">
<input type="button" value="Test property" name="B7" onclick="BtnOnTestProp()">
</p>
</body>
</html>
```

メモ 上のスクリプトにはAuthentic Browser Enterprise Edition を有効化するために必要なライセンス情報が含まれています。

### 2.3.3.4 Firefox サンプル 2: テーブルの並べ替え

埋め込まれた JavaScript を持つサンプル HTML ページです。サンプルは Authentic Browser プラグイン (XPI ファイル) が使用中のコンピュータにインストールされることを必要とします。サーバーの一部では大文字と小文字の区別が問題となる場合があります。ファイルを検索する際に問題となる場合がありますので、コード内のファイル名とコマンドの大文字/小文字の指定をチェックしてください。

コードは以下を表示します:

- ブラウザープラグインにアクセスする方法。ブラウザー プラグインバージョン (信頼されている、または、信頼されていない) の XPI ファイルとクラス識別子 (MIME の種類) を参照するためコードを変更してください。
- ブラウザープラグインファイルをロードする方法。サンプルドキュメントを参照するためのコードの変更。
- 簡単なカーソルの配置のためのボタンの実装の方法。
- テーブルの並べ替えなど更に複雑なコマンドの実装の方法。
- SelectionChanged イベントの使用方法。

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Test Page For Authentic Browser プラグイン</title>

</head>

<!-- to disable the fast-back cache in Firefox, define an unload handler -->
<BODY id="bodyId" onload="Unload()">
  <embed
    id="objPlugin"
    type="application/x-authentic-scriptable-plugin"
    width="100%"
    height="60%"
    PLUGINSOURCE="http://your-server-including-path/AuthenticFirefoxPlugin_trusted.xpi"
    EntryHelpersEnabled="TRUE"
    SaveButtonAutoEnable="TRUE" >
  </embed>
<TABLE>
<SCRIPT LANGUAGE="javascript">
var objCurrentRange = null;
var objPlugin = document.getElementById('objPlugin');

function BtnDocumentBegin() { objPlugin.AuthenticView.DocumentBegin.Select(); }
function BtnDocumentEnd() { objPlugin.AuthenticView.DocumentEnd.Select(); }
function BtnWholeDocument() { objPlugin.AuthenticView.WholeDocument.Select(); }
function BtnSelectNextWord() { objPlugin.AuthenticView.Selection.SelectNext(1).Select(); }

function BtnSortDepartmentOnClick()
{
  var objCursor = null;
  var objTableStart = null;
  var objBubble = null;
  var strField1 = "";
  var strField2 = "";
  var nColIndex = 0;
  var nRows = 0;

  objCursor = objPlugin.AuthenticView.Selection;
  if (objCursor.IsInDynamicTable())
  {
    // calculate current column index
    nColIndex = 0;
    bContinue = true;
    while ( bContinue )
    {
      try { objCursor.GotoPrevious(1); }
      catch (err) { bContinue = false; nColIndex--; }
      nColIndex++;
    }

    // GoTo begin of table
```

```

objTableStart = objCursor.ExpandTo(9).CollapsToBegin().Clone();

// count number of table rows
nRows = 1;
bContinue = true;
while ( bContinue )
{
    try { objTableStart.GotoNext(10); }
    catch (err) { bContinue = false; }
    nRows++;
}

// bubble sort through table
for ( i = 0; i < nRows - 1; i++) {
    for( j = 0; j < nRows-i-1; j++) {
        objBubble = objCursor.ExpandTo(9).CollapsToBegin().Clone();
        // Select correct column in jth table row
        objBubble.GotoNext(6).Goto(10,j,2).Goto(11,nColIndex,2).ExpandTo(6);
        strField1 = objBubble.Text;
        try
        {
            strField2 =
objBubble.GotoNext(10).Goto(11,nColIndex,2).ExpandTo(6).Text;
        }
        catch ( err ) { continue; };
        if(strField1 > strField2) {
            if(!objBubble.MoveRowUp()) {
                alert('Table row move is not allowed!');
                return;
            }
        }
    }
}
}

function InitAuthenticPluginPage( )
{
    var serverstr='your-server/';
    var basedir='Authentic/';
    objPlugIn.SchemaLoadObject.URL = 'http://' + serverstr + basedir + 'OrgChart.xsd';
    objPlugIn.XMLDataLoadObject.URL = 'http://' + serverstr + basedir + 'OrgChart.xml' ;
    objPlugIn.DesignDataLoadObject.URL = 'http://' + serverstr + basedir + 'OrgChart.sps';
    objPlugIn.StartEditing();
}

function Unload()
{
}

function OnSelectionChanged()
{
    var CurrentSelection = null;
    CurrentSelection = objPlugIn.AuthenticView.Selection;
    SelTable_FirstTextPosition.innerHTML = CurrentSelection.FirstTextPosition;
    SelTable_LastTextPosition.innerHTML = CurrentSelection.LastTextPosition;
    SelTable_FirstXMLData.innerHTML = CurrentSelection.FirstXMLData.Parent.Name;
    SelTable_FirstXMLDataOffset.innerHTML = CurrentSelection.FirstXMLDataOffset;
    SelTable_LastXMLData.innerHTML = CurrentSelection.LastXMLData.Parent.Name;
    SelTable_LastXMLDataOffset.innerHTML = CurrentSelection.LastXMLDataOffset;
}

```

```
}  
  
objPlugIn.addListener("selectionchanged", OnSelectionChanged, false)  
// event subscription if running on Firefox  
objPlugIn.addListener("ControlInitialized", InitAuthenticPluginPage, false);  
</SCRIPT>  
<TR>  
<TD><Input Type="button" value="Goto Begin" id="B1" onclick="BtnDocumentBegin()"></TD>  
<TD><Input Type="button" value="Goto End" name="B2" onclick="BtnDocumentEnd()"></TD>  
<TD><Input Type="button" value="Whole Document" name="B3" onclick="BtnWholeDocument()"></TD>  
<TD><Input Type="button" value="Select Next Word" name="B4" onclick="BtnSelectNextWord()"></TD>  
</TR>  
<TR>  
<TD><Input Type="button" value="Sort Table by this column" id="B6"  
onclick="BtnSortDepartmentOnClick()"></TD>  
</TR>  
</TABLE>  
<TABLE id=SelTable border=1>  
<TR><TD id=SelTable_FirstTextPosition></TD><TD id=SelTable_LastTextPosition></TD></TR>  
<TR><TD id=SelTable_FirstXMLData></TD><TD id=SelTable_FirstXMLDataOffset></TD></TR>  
<TR><TD id=SelTable_LastXMLData></TD><TD id=SelTable_LastXMLDataOffset></TD></TR>  
<TR><TD id=SelTable_Text></TD></TR>  
</TABLE>  
</body>  
</html>
```

## 2.3.4 依存しないブラウザ

プロジェクトの一部では、どのブラウザ (Internet Explorer または Firefox) がクライアント上で使用されるか未知の場合があります。このような場合、[Internet Explorer と Firefox のための Authentic Browser バージョン](#) をサーバー上に保管することができます (すなわち、CAB ファイルと XPI ファイルがサーバー上に保管されることを意味します)。HTML ページ内では、HTML ページを開くために使用されたブラウザを決定するスクリプトを挿入することができ、ロードする正確な [Authentic Browser プラグイン](#) が選択されます。

更に、クライアント上で Internet Explorer がブラウザとして使用されている場合、正確な .CAB ファイル (32 ビット、または 64 ビット Internet Explorer) が選択されサーバーからダウンロードされる必要があります。スクリプトは、Internet Explorer は X ビットバージョンのためにテストされ、サーバーからダウンロードされる正確な .CAB ファイルを選択します。

このセクションに次を行う、[サンプルファイル](#) が含まれています。ブラウザを決定し、正確な Authentic Browser バージョンをロードし、いくつかの機能を実行します。個別のオブジェクトに関する情報に関しては、[レファレンスセクション](#) 内の対応する [オブジェクトの説明](#) を参照してください。

### メモ

- Authentic Browser プラグインは Internet Explorer 5.5 または以降、**Mozilla Firefox** のためにサポートされています。
- Firefox 使用法に関しては、XPI ファイルのための MIME の種類がサーバー上の使用するサイトのための MIME の種類のリスト [ブラウザサービス](#) に追加されます。

### 2.3.4.1 ブラウザーに依存しない サンプル

下の HTML コードは次の機能を持つページを生成します:

- クライアント (Internet Explorer、Firefox) にブラウザがインストールされているかチェックし、検出されたブラウザの種類のために Authentic Browser バージョンをインストールします。
- 更に、インストールされたブラウザが Internet Explorer の場合、システムが 32 ビット、または 64 ビットであることをチェックし、(32 ビット、または 64 ビット Internet Explorer のために) [正確な .CAB ファイル](#) が選択されます。
- ページ内の Authentic Browser ウィンドウはブラウザウィンドウの 100% の幅、および 60% の高さで設定されています。
- Authentic Browser ウィンドウの下にはおつのポインタの行が存在します

- 「編集の開始」ボタンはサーバーのルートディレクトリ内にあるOrgChart.xml のAuthentic View をロードします
- 「検索と置換」ボタンは検索と置換 ダイアログをそれぞれ表示します
- 「保存」ボタンはサーバーのルートディレクトリ内にあるSaveFile\_OrgChart.xml という名前のファイルを保存します。
- 「テスト」ボタンはシンプルなブラウザをテストします

このHTML ページがクライアントで開かれると、XML ファイルOrgChart.xml の編集を開始し、編集したファイルを SaveFile\_OrgChart.xml として保存することができます。

Authentic Browser が正確に動作するかテストするために、このシンプルな HTML ページを使用することができます。これを行う場合、XPI ファイル、xsd、xml、およびsps ファイルとサーバー上の他のリソースをロードする URL 内のそれぞれのファイルに対して正確なアドレスとパスを使用してください。サーバーの一部では大文字と小文字の区別が問題になる場合があります、ファイルを検索する際に問題になる場合がありますので、コード内のファイル名とコマンドの大文字/小文字の指定をチェックしてください。このサンプルを拡張、または変更して、Authentic Browser を使用し、更に複雑なリネーションをビルドすることができます。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Orgchart.sps Scriptable Plug-in Test - browser independent</title>
    <script type="text/javascript">
      <!--
        function BtnOnSave() { objPlugIn.Save();}

        function InitAuthenticPluginPage()
        {

          var schema= document.getElementById('xsd');
          var instance=document.getElementById('xml');
          var design=document.getElementById('sps');
          objPlugIn.XMLDataLoadObject.URL =instance.innerHTML;
          objPlugIn.DesignDataLoadObject.URL = design.innerHTML;
          objPlugIn.SchemaLoadObject.URL= schema.innerHTML;
          // alert(schema.innerHTML+" "+instance.innerHTML+" "+design.innerHTML);

          /*
            var serverstr='your-server/';
            var basedir='Authentic/';
            objPlugIn.SchemaLoadObject.URL = 'http://' + serverstr + basedir + 'OrgChart.xsd';
            objPlugIn.XMLDataLoadObject.URL = 'http://' + serverstr + basedir + 'OrgChart.xml' ;
            objPlugIn.DesignDataLoadObject.URL = 'http://' + serverstr + basedir + 'OrgChart.sps';
          */

          objPlugIn.StartEditing();

        }

        function Unload()
        {
        }
      <!-->
    </script>
    <style type="text/css">@page { margin-left:0.60in; margin-right:0.60in; margin-top:0.79in; margin-bottom:0.79in } @media screen { br.altova-page-break { display: none; } } @media print { br.altova-page-break { page-break-before: always; } }
    </style>
  </head>

  <body id="bodyId" onload="Unload()">
    <table border="1">
      <tbody>
```

```

<tr><th><span>DesignLoadURL</span></th><td id="sps">http://your-server/Authentic/Orgchart.sps</td></tr>
<tr><th><span>SchemaLoadURL</span></th><td id="xsd">http://your-server/Authentic/Orgchart.xsd</td></tr>
<tr><th><span>XMLDataLoadURL</span></th><td id="xml">http://your-server/Authentic/Orgchart.xml</td></tr>
<tr><th><span>XMLDataSaveURL</span></th><td id="xmlsave">http://your-
server/Authentic/SaveFile_OrgChart.xml</td></tr>
</tbody>
</table>
<center><h3><span>Authentic Platformindependent プラグイン Enterprise Edition</span></h3></center>
<span>&nbsp;</span>
<center>
<script language="JavaScript" type="text/javascript">
  // return true if the page loads in Firefox
  function isFirefoxOnWindows()
  {
    return ((navigator.userAgent.indexOf('Firef') != -1) && (navigator.userAgent.indexOf('Win') != -1));
  }

  // return true if the page loads in Internet Explorer
  function isIEOnWindows()
  {
    return ((navigator.userAgent.indexOf('MSIE') != -1) && (navigator.userAgent.indexOf('Win') != -1))
  }

  //return true if Browser is 64bit
  function is64bitBrowser()
  {
    return ((navigator.userAgent.indexOf('Win64') != -1)&& (navigator.userAgent.indexOf('x64') != -1))
  }

  //return Codebase for 32 bit or 64 bit
  function getCodeBase()
  {
    if ( is64bitBrowser() ){
      return('CodeBase="http://your-server/AuthenticBrowserEdition_x64.CAB#Version=12,2,0,0" ');
    }
    else {
      return('CodeBase="http://your-server/AuthenticBrowserEdition.CAB#Version=12,2,0,0" ');
    }
  }

  // Create the plugin object instance, according to the browser loading the page
  // -Firefox uses EMBED tag for embedding plugins and supports PLUGINSPPAGE
  // attribute to redirect to an installation file if the plugin is not
  // currently installed;
  // -IE uses <OBJECT> tag for embedding plugins and supports CODEBASE attribute
  // to indicate a .cab file for the installation if the plugin is not
  // currently installed

  function createObject( codebase, clsid)
  {
    if ( isFirefoxOnWindows() )
    {
      document.write ( '<embed ' +
        'id="objPlugin" ' +
        'type="application/x-authentic-scriptable-plugin" ' +
        'width="100%" ' +
        'height="60%" ' + 'PLUGINSPPAGE="http://your-server/Authentic/AuthenticFirefoxPlugin_trusted.xpi" ' +
        'SaveButtonAutoEnable="true" ' +
        'EntryHelpersEnabled="true" ' +
        'LicServer="your-server" ' +
        'LicCompany="Altova" ' +
        'LicKey="XXXXXXXXXX" ' +

```

```

XMLDataSaveUrl="http://your-server/Authentic/SaveFile_OrgChart.xml"> ' +
</embed>');
}
else if ( isIEOnWindows() )
{
document.write ( '<OBJECT ' +
'id="objPlugIn" ' +
getCodeBase() +
'Classid="clsid:B4628728-E3F0-44a2-BEC8-F838555AE780" ' +
'width="100%" ' +
'height="60%" ' +
'> ' +
'<PARAM NAME="XMLDataSaveUrl" VALUE="http://your-server/Authentic/SaveFile_OrgChart.xml"> ' +
'<PARAM NAME="EntryHelpersEnabled" VALUE="TRUE"> ' +
'<PARAM NAME="SaveButtonAutoEnable" VALUE="TRUE"> ' +
'<PARAM NAME="LicServer" VALUE="your-server"> ' +
'<PARAM NAME="LicCompany" VALUE="Altova"> ' +
'<PARAM NAME="LicKey" VALUE="XXXXXXXXXX"> ' +
'</OBJECT>');
}
}

createObject();
// after running createObject the plugin object exists. Initialize the javascript variable to be used in the scripts
var objPlugIn = document.getElementById('objPlugIn');
</script>

<br><br>
<button onclick="objPlugIn.StartEditing()"><span>Start Editing</span></button>
<button onclick="objPlugIn.FindDialog()"><span>Find</span></button>
<button onclick="objPlugIn.ReplaceDialog();"><span>Replace</span></button>
<button onclick="BtnOnSave()"><span>Save</span></button>
<button onclick="alert ( objPlugIn.IsRowInsertEnabled );"><span>Test</span><br></button>
</center>

<script language="javascript" type="text/javascript">
// event subscription if running on Firefox
if ( isFirefoxOnWindows() )
{
objPlugIn.addEventListener("ControlInitialized", InitAuthenticPluginPage, false);
}
</script>

<script event="ControlInitialized" for="objPlugIn" language="javascript" type="text/javascript">
// event subscription if running on Internet Explorer
if ( isIEOnWindows() )
{
InitAuthenticPluginPage();
//if ( isIE64OnWindows() ) alert("IE x64");
}
</script>

</body>
</html>

```

**メモ** 上のスクリプトにはAuthentic Browser Enterprise Edition を有効化するために必要なライセンス情報が含まれています。3つのパラメーターLicServer、LicCompany、およびLicKeyが存在しない場合、Authentic Browser 機能はCommunity Edition の機能のみに制限されます。

## 2.4 オンデマンドのインストールのための拡張機能パッケージ

Authentic Browser プラグインをオンデマンドでインストールする場合、圧縮されている Authentic Browser 拡張機能パッケージ (CAB、XPI、およびまたは CRX ファイル) がサーバー上に保管される必要があります。[Authentic プラグインのための HTML ページ](#) がクライアントブラウザ内で最初にアクセスされると、HTML ページ内の命令により、関連する拡張機能パッケージがサーバーからクライアントにダウンロードされ、圧縮されていないクライアント上にインストールされます。

オンデマンドのインストールのための必要とされる Authentic Browser 拡張機能パッケージは Altova Web サイトからダウンロードすることができます。サーバーに圧縮されている (CAB、XPI、または CRX) ファイルとして保管される必要があります。サーバー上に CAB、XPI、または CRX ファイルを保管するかは、[Authentic プラグインのための HTML ページ](#) を開くクライアント上で使用されるブラウザにより異なります。

Internet Explorer v5.5 または以降 (32 ビット)	CAB ファイル (32 ビット IE のための)
Internet Explorer (64 ビット)	CAB ファイル (64 ビット IE のための)
Firefox	XPI ファイル

クライアントが複数のブラウザまたは任意のブラウザ (Internet Explorer または Firefox) を使用する場合、2 つ、または 3 つの拡張機能パッケージ (CAB、XPI、と CRX ファイル) がサーバー上に保管される必要があります。

クライアントブラウザが未知の場合、[Authentic プラグインのための HTML ページ](#) 内にスクリプトを含み、現在使用されているブラウザを検出することができます。適切なファイル (Internet Explorer のために CAB、Firefox のために XPI) が自動的にサーバーからクライアントにダウンロードされます。このシナリオに関する詳細は、[Authentic プラグインのための HTML ページ | ブラウザーに依存しない](#) を参照してください。

32 ビット IE ブラウザーと 64 ビット IE ブラウザーのために Internet Explorer のための CAB ファイルを使用することができます。CAB ファイル (32 ビット と 64 ビット) の両方の型をサーバーに保管してください。[Authentic プラグインのための HTML ページ](#) はブラウザが 32 ビット、または 64 ビットバージョンであるかを決定するスクリプトを持つことができ、正確な CAB ファイルをダウンロードします。このようなスクリプトの作成方法は、セクション [Authentic プラグインのための HTML ページ | ブラウザーに依存しない](#) 内で説明されています。

### CAB/XPI/CRX ファイルのダウンロードと保管

CAB ファイル、XPI ファイル、およびまたは CRX ファイルは Altova [Web サイト](#) からダウンロードすることができ、サーバー上のロケーションの希望する場所にダウンロードすることができます。Authentic Browser の Enterprise エディションをデプロイする場合、パッケージは Enterprise ライセンスが登録されているサーバー上に保管される必要があります。

インストーラーファイル (CAB、XPI、CRX) の 8 つの種類全ては、圧縮されているファイルの書式です。これらのファイルを解凍しないでください。クライアント上でファイルの抽出とインストールは、[Authentic プラグインのための HTML ページ](#) が最初クライアントにより開かれた際に自動的におこなわれます。サーバー上の CAB/XPI/CRX ファイルの場所は [Authentic プラグインのための HTML ページ](#) 内で指定されています。

### 3 クライアントのセットアップ

クライアントマシンは [Authentic プラグインのためのHTML ページ](#) が開かれ、Altova Authentic XML ページが編集されるマシンです。この機能を使用するには、[Authentic プラグインのためのHTML ページ](#) を開くために使用される Authentic Browser プラグインは、クライアントブラウザ内にアドオンとしてインストールされる必要があります。

[Authentic プラグインのためのHTML ページ](#) がクライアントブラウザ内で開かれると自動的にサーバーから Authentic Browser プラグインをクライアントブラウザにインストールすることができます。または、Authentic Browser プラグインをユーザーに直接インストール、または管理された [MSI ベースのソリューション](#) を使用して、[手動で](#) インストールすることができます。

このセクションでは、これらの機能のためにどのようにクライアントマシンをセットアップするかについて説明されています。次の項目について説明されています：

- Authentic Browser 機能を有効化するための [Internet ブラウザーの必要条件](#)。
- クライアントマシン上のサポートされるブラウザ内のアドオンとしての [Authentic Browser プラグインのインストールの異なるメソッド](#)。Authentic Browser プラグインの [アップグレード](#) と [アンインストール](#) について説明されています。
- [IE9 セキュリティの設定](#) によりプラグインの適切な機能を許可します。
- [IE10 セキュリティの設定](#) によりプラグインの適切な機能を許可します。Internet Explorer 10 内で、IE 10 モード内での Authentic Browser プラグイン使用が許可されるために互換性モードに設定してください。

## 3.1 ブラウザーの必要条件

Altova Authentic XML ページを表示するためのクライアントマシンはプラグインを許可するブラウザを必要とします。ブラウザの多くは [NPAPI アーキテクチャ](#) へのサポートを中断しており、Authentic Browser Edition は Microsoft Internet Explorer 32-bit バージョン 9 と以降のみでサポートされています。他の Internet ブラウザーの使用を希望する場合、プラグインをサポートする古いバージョンを使用する必要があります。

下のリストでは主なブラウザ間での Authentic Browser プラグインのためのサポートレベルに関連する情報が表示されています。

- Microsoft Internet Explorer 32-bit バージョン 9 と以降は Authentic Browser をサポートします。
- Microsoft Internet Explorer 64-bit バージョン 9 は使用することのできる最後の 64-bit バージョンです。これは先新しいバージョンは Microsoft Edge です。
- Microsoft Edge は、ActiveX コントロールをプラグインとしてサポートしないため Authentic Browser をサポートしません。
- Firefox バージョン 26.0 は NPAPI サポートへの最後のオファーです。26.0 までのバージョン (最新バージョンが奨励されます) を Authentic Browser のために使用することができます。
- Google Chrome バージョン 44 は NPAPI サポートへの最後のオファーです。44.0 までのバージョン (最新バージョンが奨励されます) を Authentic Browser のために使用することができます。

**メモ** (ブラウザウィンドウ内に表示される) Authentic View インターフェイスは Internet Explorer を使用して生成されるため、Internet Explorer がインストールされている必要があります。

## 3.2 Authentic Browser プラグイン

このセクションでは Authentic Browser をクライアントマシンにインストールする方法、およびプラグインのアップグレードとアンインストールに関する多種の方法について説明されています。

- [オンデマンドのインストール](#)
- [拡張機能パッケージを使用した手動のインストール](#)
- [MSI を使用した手動のインストール](#)
- [MSI を使用したプッシュインストール](#)
- [自動更新](#)
- [アンインストール、無効化](#)

### 複数のバージョンのインストール

Authentic Browser の複数のバージョンが存在します。サポートされるそれぞれの言語(英語、ドイツ語、スペイン語、日本語)のために個別の信頼されている、および、信頼されていないバージョンがそれぞれのサポートされるブラウザ(Microsoft Internet Explorer 32ビットと64ビット、Mozilla Firefox)のために使用することができます。

Authentic Browser プラグインの1つ、複数、または全てのバージョンをクライアントマシンにインストールすることができます。ブラウザのアドオンマネージャー内ではそれぞれが個別のプラグインとして表示されます。

バージョンに関する詳細に関しては、[Authentic Browser バージョン](#)を参照してください。

### 3.2.1 オンデマンドのインストール

Authentic Browser プラグインのインストールの最善の方法は自動的に必要に応じてインストールすることです。このメカニズムは以下のよう動作します:

1. T拡張機能パッケージ(CAB、XPI、およびまたはCRX ファイル) はサーバー上に保存されます。
2. [Authentic プラグインのためのHTML ページ](#) がクライアントのブラウザ内で開かれると、HTML ページ内の命令は関連する拡張機能パッケージをダウンロードし、Authentic Browser プラグインをクライアントブラウザにインストールします。

HTML ページ内の命令はブラウザにより異なります。

#### *Internet Explorer のための命令*

CAB ファイルの URL を指定するために、OBJECT 要素の CODEBASE 属性を使用します。詳細に関しては、次を参照してください!

#### [Internet Explorer のサンプル](#)

#### *Mozilla Firefox のための命令*

XPI ファイルの URL を指定するために、embed 要素の pluginspage 属性を使用します。詳細に関しては、次を参照してください!

#### [Firefox のサンプル](#)

#### *Mozilla Firefox のための代替メソッド*

拡張機能パッケージをポイントするリンクを追加し、クリックすると、プラグインはインストールされます。例: `<a href="plugin.crx">Click to install Authentic Browser 拡張子</a>`。拡張機能パッケージのサーバー側のセットアップに関する詳細は、セクション [オンデマンドのインストールのための拡張機能パッケージ](#) を参照してください。バージョンに関する情報は、次を参照してください! [Authentic Browser バージョン](#)。

### 3.2.2 拡張機能パッケージを使用した手動のインストール

ブラウザの一部では、Authentic Browser プラグインを手動でインストールすることができます。

#### Internet Explorer

CAB ファイル内に含まれている Internet Explorer アドオンを手動でインストールする方法はありません。MSI を使用した手動のインストールを使用してください。

#### Firefox

Firefox ウィンドウに XPI ファイルをドラッグアンドドロップします。アドオンマネージャー(「ツール | アドオン | 拡張子」) はインストールされた拡張子のリスト内に Authentic Browser プラグインを表示します。

バージョンに関する情報は、次を参照してください。 [Authentic Browser バージョン](#)

### 3.2.3 MSI を使用した手動のインストール

Altova Web サイトから Authentic Browser のための MSI インストーラー (Microsoft Windows Installer) ファイルをダウンロードします。( .exe 拡張子を持つ) このファイルは実行可能なインストーラーファイルで、クライアントマシンに現在インストールされている全てのサポートされるブラウザ (Microsoft Internet Explorer 32 ビット と 64 ビット、Mozilla Firefox) 内にプラグインをインストールします。

#### インストール

クライアントマシンに MSI インストーラーファイルをダウンロードし、ダブルクリックしてインストールを開始します。プラグインがインストールされます (デフォルトでは、フォルダー: C:\Program Files (x86)\Altova\AuthenticBrowserPlugin\... にインストールされます) そして、現在インストールされているサポートされるブラウザにこのステップで統合されます。カスタムインストールダイアログを使用して異なるブラウザのために選択的なインストールを行うことができます。

クライアントマシン上のプログラムの追加/削除 リストによりインストールされたブラウザエディションが表示されます。

#### MSI バージョン

サポートされるそれぞれの言語 (英語、ドイツ語、スペイン語、日本語) のために MSI インストーラーファイルが存在します。信頼されている/信頼されていない型、および、各 32 ビット/64 ビット ブラウザ 型がそれぞれのために存在します。例えば、英語版の信頼されている 32 ビット ブラウザのために MSI インストーラーファイルが一つ存在し、英語版の信頼されている 64 ビット ブラウザのためにもう一つインストーラーファイルが存在し、英語版の信頼されていない 32 ビット ブラウザが存在するなどです。

バージョンに関する詳細に関しては、 [Authentic Browser バージョン](#) を参照してください。

### 3.2.4 MSI を使用したプッシュインストール

新規のソフトウェアをインストールする場所は中央 IT チームとインストーラーにより管理され、管理者フレームワーク内でプッシュアウトされます。また、ネイティブな MSI インストーラー (Microsoft Windows インストーラー) を使用するインストールは役に立つ可能性があります。プラグインのインストールが管理者の権利を必要とする Internet Explorer では掛かり当てはまりません。

Altova Web サイトから Authentic Browser のための MSI インストーラー (Microsoft Windows Installer) ファイルをダウンロードします。( .exe 拡張子を持つ) このファイルは実行可能なインストーラーファイルで、クライアントマシンに現在インストールされている全てのサポートされるブラウザ (Microsoft Internet Explorer 32-bit と 64-bit、Mozilla Firefox) 内にプラグインをインストールします。

## インストール

クライアントマシンに MSI インストーラーファイルを管理者サマードラッグしてダウンロードし、クライアントマシンへのインストールのための管理者のプロセスに含めます。(デフォルトでは、フォルダー: C:\Program Files (x86)\Altova\AuthenticBrowserPlugin\... にインストールされます) そして、現在インストールされているサポートされるブラウザにこのステップで統合されます。カスタムインストールダイアログを使用して異なるブラウザののために選択的なインストールを行うことができます。

通知無しインストールに関しては、コマンドライン入力 /quiet を使用してください。

クライアントマシン上のプログラムの追加/削除リストによりインストールされたブラウザエディションが表示されます。

## MSI バージョン

サポートされるそれぞれの言語 (英語、ドイツ語、スペイン語、日本語) のために MSI インストーラーファイルが存在します。信頼されている/信頼されていない型、および各 32-bit/64-bit ブラウザー型がそれぞれのために存在します。例えば、英語版の信頼されている 32-bit ブラウザーのために MSI インストーラーファイルが一つ存在し、英語版の信頼されている 64-bit ブラウザーのためにもう一つインストーラーファイルが存在し、英語版の信頼されていない 32-bit ブラウザーが存在するなどです。

に関する詳細に関しては、[Authentic Browser バージョン](#) を参照してください。

## 3.2.5 自動更新

Authentic Browser アップグレードとインストールはブラウザにより異なります。

### Internet Explorer

[Authentic プラグインのための HTML ページ](#) 内の object 要素の codebase 属性の値は、サーバー上で使用することのできるプラグインのバージョン番号を示すために後に #Version=... と共に付けます。ユーザーのマシン上にインストールされるプラグインのバージョンが低いバージョンの場合、ユーザーはプラグインを更新するかと問われます。

### Mozilla Firefox

Firefox 拡張子のアップグレードの標準のメカニズムは以下のとおりです。Firefox 拡張子内の install.rdf ファイルはダウンロードサーバー上の update.rdf ファイルへのリンクが含まれています。install.rdf ファイルはユーザーのマシン上にインストールされるバージョンを指定します。update.rdf ファイルはサーバー上でダウンロードのために使用することのできるバージョン 拡張子 ファイルをリストします。更に新しいバージョンを使用することができる場合、ユーザーに通知されます。

これは Authentic Browser のために現在サポートされていません。

## 3.2.6 アンインストール、無効化

MSI を使用してインストールが行われた場合、クライアントマシン上のプログラムの追加/削除リストはインストールされたブラウザエディションを表示します。プログラムの追加/削除リストを使用して製品をアンインストールした場合、プラグインはブラウザと MSI リストから削除されます。

ブラウザ内にプラグインがインストールされていると、ブラウザのアドオンマネージャー内で有効化済みと表示されます。ブラウザのアドオンマネージャーからプラグインを削除すると、プラグインは無効化されますが、ディスク上に保管されます。



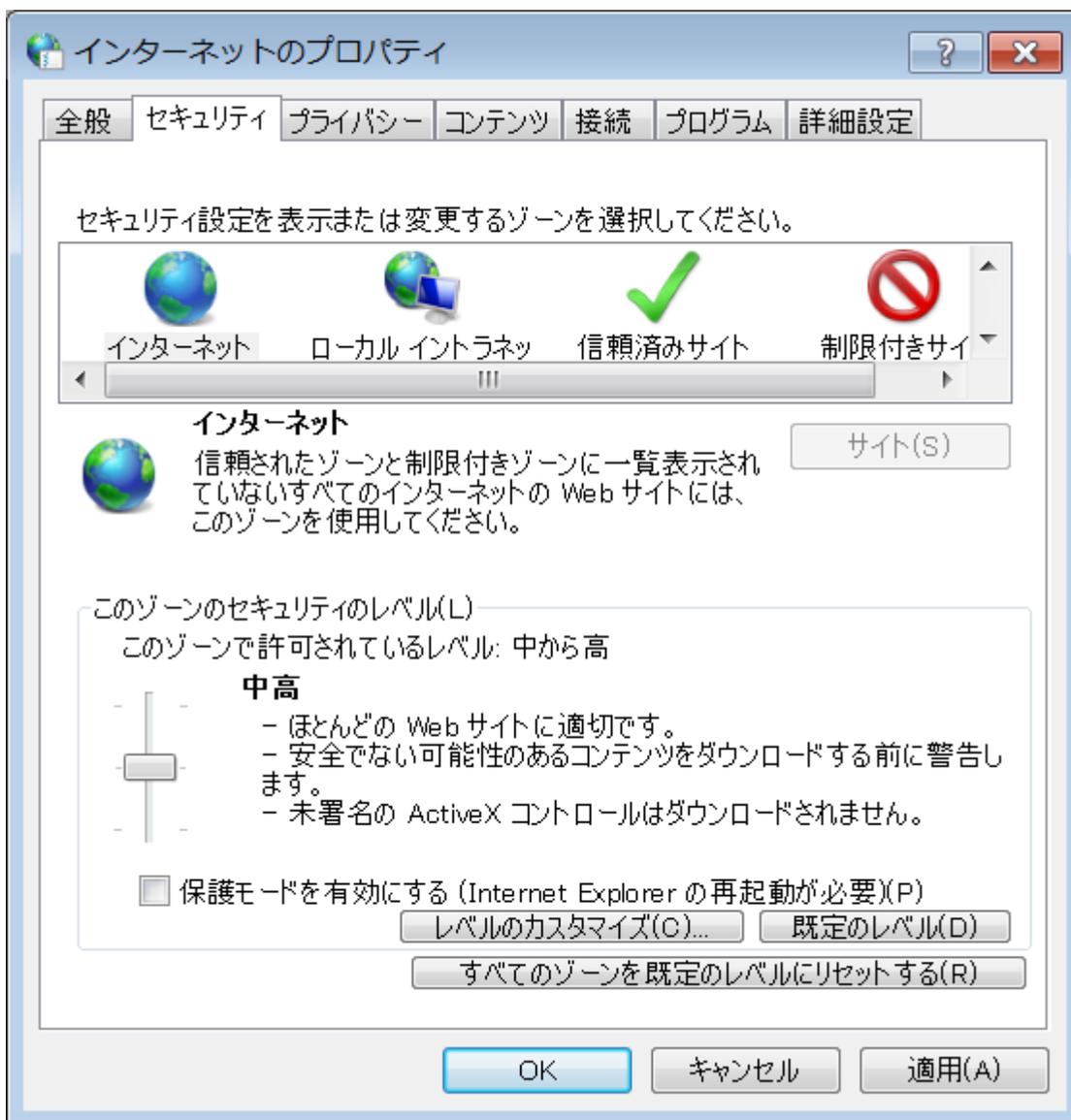
### 3.3 IE9 セキュリティの設定

[Authentic Browser の信頼されていないバージョン](#) が Internet Explorer 9 内で使用されている場合、次のエラーメッセージを受信する可能性があります:

「Internet Explorer は ActiveX コントロールをブロックしているため、このページは正確に表示されていない可能性があります。」

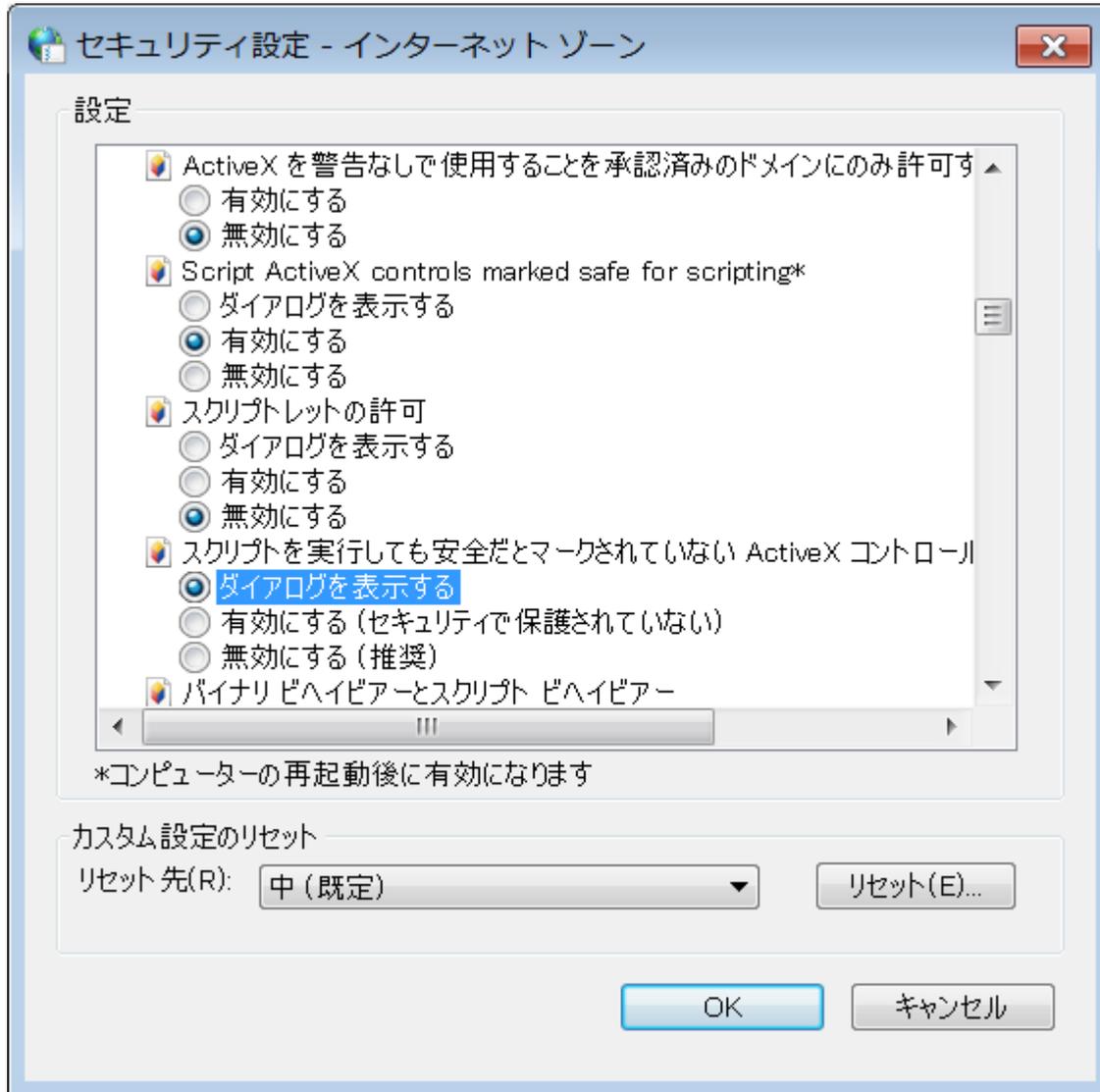
ActiveX コントロールを有効化するには、Internet Explorer のインターネット オプション内の関連するセキュリティのオプションを設定する必要があります。これは以下のとおりです:

1. メニューコマンド「ツール」インターネット オプションをクリックして Internet Explorer のインターネット オプション ダイアログを開きます(下のスクリーンショット)。



2. セキュリティタブを選択し、(インターネットまたはローカルのイントラネットの変更する設定のゾーンを選択します(下のスクリーンショットを参照)。

3. レベルのカスタマイズボタンをクリックします。セキュリティの設定 ダイアログがポップアップされます(下のスクリーンショット)。



4. ActiveX コントロールとプラグイン セクションまでスクロールし、この設定内で、「スクリプトを実行しても安全だとマークされていない ActiveX の初期化と実行」で、「ダイアログを表示する」を選択します。(下のスクリーンショットを参照)。
5. 新規の設定が反映されるように Internet Explorer を再起動します。

以降、[Authentic プラグインのための HTML ページ](#) が ActiveX ファイルのロードを試みると、Internet Explorer はコントロールをロードするかプロンプトします。

## 3.4 IE10 セキュリティの設定

Internet Explorer 10 (IE10) には2つのブラウザーモードが存在します:

- 拡張された保護モードセキュリティで作動するメロスタイル
- 互換性モードのセキュリティで作動するデスクトップ

Authentic Browser プラグインを実行するには、「ツール | オプション | セキュリティ」でIE10 を互換性のあるモードに設定する必要があります。

## 4 ユーザー レファレンス

このセクションには、Authentic Browser [メカニズム](#)、[オブジェクト](#)、および [列挙](#)の説明がなされています。

- [Authentic Browser のメカニズム](#)
- [Authentic Browser のオブジェクト](#)
- [Authentic Browser 列挙](#)

## 4.1 メカニズム

このセクションには、一般的に使用される Authentic Browser **メカニズム**に関する説明が含まれています。

### 4.1.1 イベント: 接続ポイント (IE-固有)

Authentic Browser は HTML ページ上の SCRIPT ブロック内でイベントハンドラーを与える異なる接続ポイントイベントを与えます (次も参照してください: [イベント: レファレンス](#))。

**メモ** このセクション内の説明は、Internet Explorer のみに適用されます。

次のサンプルは ControlInitialized と SelectionChanged イベントのための SCRIPT ブロックを表示しています:

#### ControlInitialized

この接続ポイントイベントはコントロールが作成され初期化されるとすぐトリガーされます。コントロールのための追加のスタートアップスクリプトは、ControlInitialized イベントハンドラー内で処理されます。

```
<SCRIPT LANGUAGE=javascript FOR=objPlugIn EVENT=controlinitialized>
    // add your code here
</SCRIPT>
```

#### SelectionChanged

ビュー内の現在の選択が変更される都度、SelectionChanged イベントはトリガーされます。SCRIPT ブロックを使用して固有のコードを実行します。

```
<SCRIPT LANGUAGE=javascript FOR=objPlugIn EVENT=selectionchanged>
    // add your code here
</SCRIPT>
```

[Authentic.event](#) オブジェクトはこのイベントが発生すると作成されません。イベントハンドラーがスクリプト内で登録されている場合、[Authentic.event](#) オブジェクトのプロパティは前回のイベントから発生した値を含んでいます。

[Authentic.CurrentSelection](#) オブジェクトには有効な情報が含まれています。

### ユーザー仲介無しに SPS、XSD、と XML ファイルをロードする

.sps、.xsd と .XML ファイルをユーザーの仲介なしに HTML ページをロード中に行う場合、ControlInitialized イベントを処理するイベントハンドラーを作成することが奨励されるメソッドです。代わりに、プロパティを 2 番目のサンプルとして使用することもできます。

#### 奨励される

```
<SCRIPT LANGUAGE="javascript" FOR=objPlugIn EVENT="ControlInitialized">
objPlugIn.SchemaLoadObject.URL = "http://yourserver/OrgChart.xsd"
objPlugIn.XMLDataLoadObject.URL = "http://yourserver/OrgChart.xml"
objPlugIn.DesignDataLoadObject.URL = "http://yourserver/OrgChart.sps"
objPlugIn.StartEditing()
</SCRIPT>
```

## または

```
<OBJECT id=objPlugIn style="WIDTH:600px; HEIGHT:500px"
codeBase="http://yourserver/cabfiles/AuthenticBrowserEdition.CAB#Version=12,3,0,0"
classid=clsid:B4628728-E3F0-44a2-BEC8-F838555AE780>
<PARAM NAME="XMLDataURL" VALUE="http://yourserver/OrgChart.xml">
<PARAM NAME="SPSDataURL" VALUE="http://yourserver/OrgChart.sps">
<PARAM NAME="SchemaDataURL" VALUE="http://yourserver/OrgChart.xsd">
</OBJECT>
```

“body” 要素 “onload” イベントを Authentic Browser PlugIn コントロールとして処理するイベントハンドラー内にこれらのファイルをロードすることは、“onload” イベントが実行されてから初期化される可能性があるため、奨励されません。プラグインの場合、メソッドとプロパティを使用することはできず、ファイルはロードされません。詳細に関して、次も参照してください: [OBJECT 要素](#)。

## 推奨されません

```
<SCRIPT LANGUAGE="javascript">
function load () {

objPlugIn.SchemaLoadObject.URL = "http://yourserver/OrgChart.xsd"
objPlugIn.XMLDataLoadObject.URL = "http://yourserver/OrgChart.xml"
objPlugIn.DesignDataLoadObject.URL = "http://yourserver/OrgChart.sps"
objPlugIn.StartEditing()
}
</SCRIPT>

<body onload = "load files">
```

## 4.1.2 イベント: イベントリスナーの追加 (Firefox-固有)

各接続ポイントイベントのために Authentic Browser は `addEventListener` メソッドを使用して、HTML ページ上の SCRIPT ブロックにイベントハンドラーを与えることができます (次も参照してください: [イベント: レファレンス](#))。

**メモ** このセクション内の説明は、Firefox のみに適用されます。

Firefox 内のイベントのためにイベントハンドラーの提供の方法を説明するサンプルは以下のとおりです:

### ***SelectionChanged***

ビュー内の現在の選択が変更される都度、`SelectionChanged` イベントはトリガーされます。SCRIPT ブロックを使用して固有のコードを実行します。

```
<SCRIPT LANGUAGE="javascript">
var selCount = 0;
function OnSelectionChanged()
{
    selCount = selCount + 1;
    selectionCounter.value = "SelectionCount = " + selCount;
}
var objPlugIn = document.getElementById('objPlugIn');
objPlugIn.addEventListener("selectionchanged", OnSelectionChanged, false)
</SCRIPT>
```

[Authentic.event](#) オブジェクトはこのイベントが発生すると作成されません。イベントハンドラーがスクリプト内で登録されている場合、[Authentic.event](#) オブジェクトのプロパティは前回のイベントから発生した値を含んでいます。

[Authentic.CurrentSelection](#) オブジェクトには有効な情報が含まれています。

### 4.1.3 イベント: ツールバーボタン

各ツールバーボタンは変更する必要があるデフォルトの振る舞いがあります。AuthenticCommand イベントにより追加タスクを追加、または、ツールバーボタンのデフォルトの振る舞い全体を再定義することができます。スクリプトは、AuthenticCommand イベントを使用し、ユーザーがツールバーアイコンをクリックする都度通知を受けることができます。

([Authentic.UICommands](#) コレクションから) 全てのコマンドに関連したイベントが存在するとは限りません。ユーザーがクリックしたアイコンを確認するには、スクリプトは対応する [AuthenticCommand](#) オブジェクトに対するレファレンスを含む [AuthenticEvent.srcElement](#) プロパティをチェックする必要があります。

#### サンプル

```
// event handler for OnDocEditCommand
<SCRIPT LANGUAGE=javascript FOR=objPlugIn EVENT=doceditcommand>
  // we are interested in the k_CommandSave button
  if(objPlugIn.event.srcElement.CommandID == 1)
  {
    // instead of the standard HTTP PUT we want to use
    // a HTTP POST
    objPlugIn.SavePOST();

    // no standard execution follows
    objPlugIn.event.cancelBubble = true;
  }
</SCRIPT>
```

#### レファレンス

使用することのできるコマンドに関しては、[AuthenticToolBarButton.CommandID](#) を参照してください。

### 4.1.4 イベント: レファレンス

#### 接続ポイントイベントのリスト

名前	since TypeLib	説明
SelectionChanged	1.0	エディター内の選択が変更されています。 <a href="#">Authentic.CurrentSelection</a> オブジェクトには有効な情報が含まれています。 <a href="#">Authentic.event</a> オブジェクトのプロパティは設定されていません。
Controllinitialized	1.2	コントロールがロードされ初期化されています。 <a href="#">Authentic.event</a> オブジェクトのプロパティは設定されていません。
dragover	1.8	ドラッグオーバーオペレーションが発生しています。
drop	1.8	ドロップオペレーションが発生しました。
keydown	1.8	キーが押されていますが、リリースされていません。

keyup	1.8	キーボードのキーがリリースされました。これはキーストロークに反応する通常のイベントです。
keypressed	1.8	キーボード入力で挙げられます。
mousemove	1.8	マウスポインターが移動されました。
buttonup	1.8	マウスボタンの1つがリリースされました。
buttondown	1.8	マウスボタンの1つが押されました。
contextmenu	1.8	WM_CONTEXTMENU の受信後に送信されます。
editpaste	1.8	貼り付けオペレーションが実行される前に呼び出されました。
editcut	1.8	切り取りオペレーションが実行される前に呼び出されました。
editcopy	1.8	コピーオペレーションが実行される前に呼び出されました。
editclear	1.8	クリアオペレーションが実行される前に呼び出されました。
doceditcommand	1.8	Authentic コマンドの実行時とカスタムコマンドハンドラーの実装時に挙げられます。次も参照してください <a href="#">イベント: Toolbarbuttons</a> 。 <a href="#">Authentic.event</a> オブジェクトのプロパティは設定されていません。
buttdoubleclick	1.8	マウスボタンの1つがダブルクリックされました。

[Authentic.event](#) オブジェクトの説明 プロパティ内で例外について言及されていない場合、イベントハンドラーの呼び出し時に設定されません。

**メモ** これらのイベントハンドラー同期です。すなわち、イベントが発生するとすぐに呼び出されます。

### 4.1.5 ドキュメントコンテンツへのアクセスと変更

ドキュメントコンテンツにアクセスし変更するには、AuthenticRange、AuthenticView とAuthentic オブジェクト（とそのプロパティとメソッド）を使用する必要があります。

AuthenticRange とAuthenticView インターフェイスとAuthentic オブジェクトにより提供されるインターフェイスの間で機能がオーバーラップする一方、AuthenticRange とAuthenticView インターフェイスは優先順位の高いインターフェイスです。Authentic オブジェクトの重複する機能は将来のバージョンで使用できなくなります。

### 4.1.6 編集の操作

Authentic View 内でXML データが表示されると切り取り、コピーおよび張り付けなどの標準の編集の操作を使用して個別の要素を操作することができます。しかしながら、XML データ要素を全て編集することはできません。ですから、最初に編集が可能かテストすることが必要です。使用されるメカニズムは、以下のとおりです：最初に特定の編集オペレーションが有効化されているかチェックします。有効化されている場合はその編集の操作を実行するためのメソッドを呼び出します。ドキュメント内で表示されている全ての要素を自動的に選択するメソッド EditSelectAll のみかテストすることができません。

編集の操作をおこなうプロパティとメソッドのリストは、以下のとおりです。各プロパティはブール値とパラメータを持たないメソッドを返します。

<b>IsEditUndoEnabled</b>	<a href="#">Authentic.EditUndo</a>	編集の操作を元に戻します。
<b>IsEditRedoEnabled</b>	<a href="#">Authentic.EditRedo</a>	編集の操作をやり直します。
<b>IsEditCopyEnabled</b>	<a href="#">Authentic.EditCopy</a>	はグループボードに選択されたテキストをコピーします。
<b>IsEditCutEnabled</b>	<a href="#">Authentic.EditCut</a>	はグループボードに選択されたテキストを切り取ります。
<b>IsEditPasteEnabled</b>	<a href="#">Authentic.EditPaste</a>	現在のカーソルの位置へグループボードのテキストを貼り付けます。

**IsEditClearEnabled** [Authentic.EditClear](#) XMLドキュメントから選択されたテキストをクリアします。

## 4.1.7 検索と置換

[Authentic.FindDialog](#) メソッドはユーザーが検索用語を入力することのできる検索ダイアログボックスを開きます。[Authentic.FindNext](#) メソッドは検索される同じアイテムの次のインスタンスを許可します。[FindNext](#) メソッドはブール値のプロパティ [IsFindNextEnabled](#) を使用してテストすることができます。[FindDialog](#) メソッドの操作は [Authentic.ReplaceDialog](#) メソッドです。この操作は固有のアイテムを検索し、置換ダイアログボックス内にユーザー入力する固有の値と置き換えます。[FindNext](#) メソッドが呼び出されると、次のアイテムが検索され置き換えられます。

## 4.1.8 行の操作

Authentic View 内では、繰り返し要素構造は、各行は繰り返し要素のインスタンスを表す動的なテーブルとして作成されます。動的なテーブルが作成されると、Authentic View ユーザーは行とそのデータを操作することができます。これらの行の操作はスクリプトを使用して実行されます。

外部のスクリプトが行の操作を行う場合、2つのステップにより操作が行われます:

- 最初のステップはカーソルが置かれている行がプロパティを使用するかチェックします。IsRowInsertEnabled などのプロパティが使用され、True または False 値を返します。
- 戻り値が True の場合、必要とされる行メソッドを呼び出すことができます。

行の操作をおこなうプロパティとメソッドのリストは以下のとおりです。各プロパティはブール値とパラメータを持たないメソッドを返します。

<b>IsRowInsertEnabled</b>	<a href="#">Authentic.RowInsert</a>	行の挿入操作です。
<b>IsRowAppendEnabled</b>	<a href="#">Authentic.RowAppend</a>	行操作を追加します。
<b>IsRowDeleteEnabled</b>	<a href="#">Authentic.RowDelete</a>	行操作を削除します。
<b>IsRowMoveUpEnabled</b>	<a href="#">Authentic.RowMoveUp</a>	行を1つ上に移動します。
<b>IsRowMoveDownEnabled</b>	<a href="#">Authentic.RowMoveDown</a>	XML データを一行下に下がります。
<b>IsRowDuplicateEnabled</b>	<a href="#">Authentic.RowDuplicate</a>	現在の行を複製します。

## 4.1.9 ショートカット

Authentic Browser が入力フォーカスを持つ場合、次のショートカットは有効です:

CTRL + P	ドキュメントの印刷
CTRL + Z	元に戻す
CTRL + Y	やり直し
CTRL + X	切り取り
CTRL + C	コピー
CTRL + V	貼り付け
CTRL + A	全て選択
CTRL + F	検索ダイアログボックスを開く
CTRL + H	検索して置換え ダイアログボックスを開く

#### 4.1.10 テキストの状態ボタン

Authentic Browser の Authentic View ツールバーは、テキスト状態アイコンのためのサポートを提供します。定義済みのテキストの書式設定 プロパティを持つ要素を挿入するアイコンです。この機能を使用するには、テキスト状態アイコンとして作成される要素は以下である必要があります:

- スキーマ内のグローバルテンプレートとして宣言されており、
- SPS は必要な定義を含む必要があります ( StyleVision ドキュメンテーションを参照してください)。

プラグインはツールバー内でテキスト状態アイコンまたはカスタムボタンとして使用される .bmp へのパスを必要とします。アイコンのイメージファイルへのこの URL は、 [Authentic.TextStateBmpURL](#) の値として提供されます。

#### 4.1.11 入力ヘルパー

Authentic Browser は XMLSpy 内と同様の入力ヘルパーウィンドウを提供します。Authentic View ユーザーインターフェース内の希望する場所の要素、属性、および、エンティティへのアクセスを与えます。入力ヘルパーはデフォルトでは無効化されており、明示的に有効化されていない限り、表示されません。入力ヘルパーを有効化するには、OBJECT タグ ([Internet Explorer](#)) 内で次の PROPERTYBAG パラメーターを、または、EMBED 要素 ([Firefox](#)) 内の属性として使用してください:

EntryHelpersEnabled	TRUE / FALSE
EntryHelperSize	ピクセルのサイズ
EntryHelperAlignment	<a href="#">SPYAuthenticToolbarAlignment</a> 値を取ります。
EntryHelperWindows	<a href="#">SPYAuthenticEntryHelperWindows</a> 値を取ります。全ての組み合わせが許可されます (ピットの確認)。

OBJECT または EMBED タグ内でパラメーターを使用する代わりに API 内のプロパティとメソッドを使用することができます:

##### プロパティ

- [Authentic.EntryHelpersEnabled](#)
- [Authentic.EntryHelperAlignment](#)
- [Authentic.EntryHelperSize](#)
- [Authentic.EntryHelperWindows](#)

##### メソッド

- [Authentic.RedrawEntryHelpers](#)

#### 4.1.12 パッケージ

Authentic Browser プラグインは、プログラムモジュールの機能性を拡張するパッケージをサポートします。

サーバー上にパッケージは保管され、オンデマンドで Authentic Browser GUI を使用し (クライアントに) ローカルでインストールされ、ユーザーが削除するまで起動時にアクティブ化されます。

現在、Authentic Browser はスペルチェッカーパッケージをサポートしていません。

このセクションは次の部分に整理されています:

- [パッケージ メカニズムの作動を説明する](#) [パッケージとの作業](#)
- [スペルチェッカーパッケージのサーバー上での保管方法、および、使用のためにインストールする方法を説明する](#) [スペルチェッカーパッケージ](#)

#### 4.1.12.1 パッケージとの作業

パッケージはサーバー上に Authentic Browser GUI を使用してオンデマンドで保管されており、ローカルでクライアント上にインストールされています。パッケージがローカルでインストールされていると、ユーザーが削除するまで起動時に有効化されます。

##### パッケージの管理

パッケージの管理は、Authentic Browser ユーザーコントロールに使用することのできる [パッケージ](#) を与える Authentic Browser 機能です。この機能は [パッケージの管理](#) ダイアログを介して使用することができます。

パッケージの管理 機能を有効化するには [Authentic プラグインのためのHTML ページ](#) 内の LoaderSettingsFileURL パラメータを使用する必要があります。このパラメータは [パッケージの管理のためのLoaderSettings ファイルの](#) (アクセスすることのできるローションに存在する) URL を指定します:

- [Internet Explorer のためのHTML ページ](#) 内で、LoaderSettingsFileURL パラメータが OBJECT 要素の子 PARAM 要素として使用されます。

```
<OBJECT>
...
<PARAM NAME="LoaderSettingsFileURL"
VALUE="http://www.server.com/AuthenticFiles/XMLSpyPlugInLoaderSettings.xml"/>
</OBJECT>
```

- [Firefox のためのHTML ページ](#) 内では、LoaderSettingsFileURL パラメータは EMBED 要素の属性として使用されます。

```
<EMBED
...
LoaderSettingsFileURL="http://www.server.com/AuthenticFiles/XMLSpyPlugInLoaderSettings.xml"/>
```

LoaderSettings ファイルは [パッケージの定義と対応する](#) [パッケージ](#) の URL を含んでいます。LoaderSettings ファイルのリストは下に表示されています。

##### LoaderSettings XML ファイルのサンプル

LoaderSettings XML ファイルのドキュメント 要素は loadersettings です ( [下のスティング参照](#) )。この要素は複数の子 package および [または](#) 複数の zippackage 要素を取得することができます。

- package 要素は、古い .pck スペル [パッケージバージョン 2011r3 の前の Authentic Browser バージョン](#) を参照するために使用されます。これらの Authentic Browser バージョンは [以前のバージョン](#) は zippackage 要素を無視します。
- zippackage 要素は、.zip スペル [パッケージ](#) を参照するために使用され、バージョン 2011r3 からのバージョンで Authentic Browser バージョンにより使用されます。これらの Authentic Browser バージョンは package 要素を無視します。

スペル [パッケージ](#) に関する詳細は、セクション [スペルチェッカーパッケージ](#) を参照してください。

```
<?xml version="1.0" encoding="UTF-8"?>
```

## &lt;loadersettings&gt;

```

<zippackage mode="user_demand" category="spelling">
  <packageurl>Portuguese (Brazilian).zip</packageurl>
  <description>Portuguese (BR) language pack.</description>
</zippackage>

<zippackage mode="user_demand" category="spelling">
  <packageurl>Portuguese (Brazilian).zip</packageurl>
  <description>Portuguese (BR) language pack.</description>
</zippackage>

<package mode="user_demand" id="SentrySpellChecker_EAM_only"
  category="spelling" version="1">
  <packageurl>PlugIn/SentrySpellChecker_EAM_only.pck</packageurl>
  <description>Sentry Spellchecker (EN-US)</description>
</package>

<package mode="user_demand" id="SentrySpellChecker_EALL"
  category="spelling" version="1">
  <packageurl>PlugIn/SentrySpellChecker_EALL.pck</packageurl>
  <description>Sentry SpellChecker EN with Legal and Medical.</description>
</package>

```

## &lt;/loadersettings&gt;

次の点に注意してください！

- パッケージの場所は `package` または `zippackage` 要素の `packageurl` 子のエントリとして指定される必要があります。場所の `url` は `LoaderSettings` ファイルを呼び出す HTML ファイルに対して、絶対的、または、相対的であることができます。パッケージはサーバー上の任意の場所に存在することができます。
- (パッケージ 要素を削除して XML ファイル からパッケージを削除すると、クライアント上にインストールするためのパッケージを使用できなくなります。
- `package` 要素の `mode` 属性は、パッケージをインストールする決断への影響のユーザーレベルを指定します。
  1. **user**: ユーザーに Authentic Browser 起動時にパッケージをインストールするか問います。
  2. **user\_demand**: ユーザーにより特別に要求された場合、パッケージをインストールします。ユーザーはこれをスペルチェック、または、パッケージの管理ダイアログを開くためのツールレポートをクリックしてリクエストします。
  3. **force**: ユーザーに通知することなく、パッケージをインストールします。
- `package` の `description` 子のエントリを編集し、パッケージの管理 ダイアログ内の説明テキストとして使用することができます。

## 1度限りのパッケージのインストール

Authentic Browser が更に新しいバージョンに更新されると、クライアント上に以前にインストールされたパッケージを再インストールする必要はありません。既にインストールされているパッケージは、新規のバージョンにより使用されます。クライアント PC 上の Authentic Browser は、サーバー上の `LoaderSettings` ファイルを読み取り、このファイルからパッケージ情報を取得し、パッケージがローカルでインストールされたかを確認します。

**メモ** バージョン 2011r3 または以降 (.zip パッケージ) によりサポートされるスペルパッケージは、前のバージョン (.pck パッケージ) のスペルパッケージとは異なるスペルチェッカーを使用します。from バージョン 2011r3 から以降の Authentic Browser バージョンは .zip スペルパッケージを使用します。Authentic Browser の古いバージョンは .pck スペルパッケージを使用します。

### 4.1.12.2 スペルチェッカーパッケージ

バージョン 2011r3 から、Authentic Browser は Hunspell 辞書を使用しています。

#### インストールのロケーション

クライアントマシン上で、これらの辞書は次のロケーションの Lexicons フォルダ内でロケートすることができます。

*Windows 7/8:* C:\ProgramData\Altova\SpellChecker\Lexicons

*Windows XP:* C:\Documents and Settings\All Users\Application Data\Altova\SharedBetweenVersions\SpellChecker\Lexicons

クライアントマシン上に Altova 製品がインストールされている場合、上記リストされる場所の Lexicons フォルダ内に内蔵の辞書のセットがインストールされます。インストールされた辞書の全ては Altova 製品 (32 ビット、または 64 ビット) の異なるメジャーバージョンを使用するユーザーにより共有することができます。

追加の辞書を [Altova Web サイト](#) からダウンロードすることができます。Hunspell 辞書を他の Web サイトからダウンロードすることもできます。例えば <http://wiki.services.openoffice.org/wiki/dictionaries> または <http://extensions.services.openoffice.org/en/dictionaries> などからダウンロードすることができます。また、Hunspell 辞書は Myspell 辞書をベースとしているため、Myspell 辞書を Lexicons フォルダにインストールすることができます。OpenOffice は辞書ファイルを .oxt ファイルとして圧縮します。.zip に拡張子を変更すると、必要な .aff と .dic ファイルが解凍されます。ユーザーはインストールする辞書ファイルの使用に関するライセンス使用許諾書に同意する必要があります。

**メモ** Altova ソフトウェアに搭載される内蔵辞書の選択に関して、Altova は特定の言語の優先順位を使用しておらず、[MPL](#)、[LGPL](#)、または [BSD ライセンス](#) などの商業ソフトウェアの配布を許可する辞書の状況をベースとしています。他のオープンソース辞書も存在しており、[GPL ライセンス](#) などを含む制限されたライセンス下で配布されています。<https://www.altova.com/ja/dictionaries> に存在する個別のインストーラーの一部としてこれらの辞書の多くを使用することができます。辞書に適用されるライセンスの使用許諾書の内容に同意するかどうか、または使用中のコンピュータ上のソフトウェアの使用に適しているかなどに関してはユーザーが決定します。

#### クライアント上の辞書ファイルと辞書フォルダ構造

クライアントマシン上にインストールされている各辞書は、相互に作動する Hunspell 2つの辞書ファイルにより構成されています: .aff ファイルと .dic ファイル。

すべての言語辞書は上記の場所にある Lexicons フォルダ内にインストールされています。異なる言語辞書はそれぞれ異なるフォルダ内に保管されています: <language name>%<dictionary files>。これらの言語フォルダ名は Authentic Browser 内に表示されるディレクトリ名です。Lexicons フォルダの構成と対応するディレクトリ名は Authentic Browser 内に表示され、以下にリストされるとおりです。

```
Lexicons
|
|-- English (British)           ディレクトリ名: 英語 (英国)
|   |
|   |-- .aff file
|   |-- .dic file
|
|-- English (US)              ディレクトリ名: 英語 (米国)
|   |
|   |-- .aff file
|   |-- .dic file
|
|-- ドイツ語                  ディレクトリ名: ドイツ語
```

```
|          |
|          | -- .aff file
|          |
|          | .dic file
```

### サーバー上の辞書ファイル

言語の辞書はZIP ファイルとしてサーバー上に保存される必要があります。ZIP ファイルがクライアントにダウンロードされ、インストールされると、言語フォルダーがLexicons フォルダー内に作成され（上記のインストールローションを参照してください）また、圧縮されていない .aff と .dic ファイルもこのフォルダー内に保管されます。例えば、ファイルEnglish (British).zip 次のフォルダーを作成します：`<path>\Lexicons\English (British)`。そして、このフォルダー内に .aff と .dic ファイルをインストールします。

また、複数のZIP ファイルを単一のZIP ファイルに圧縮し、この単一のファイルをサーバー上に保管することができます。このZIP ファイルを、クライアント上にダウンロードし、インストールすることができます。ZIP ファイル内部の名前は、辞書フォルダー内のフォルダーの名前として使用することができます。例えば、ファイルEnglish (British).zip とEnglish (US).zip をサーバー上にEnglish.zip という名前のファイルに圧縮することができます。インストール時に、2つのフォルダーがLexicons フォルダー内に作成されます：

`<path>\Lexicons\English (British)` と `<path>\Lexicons\English (US)`。

### 辞書のインストールメカニズム

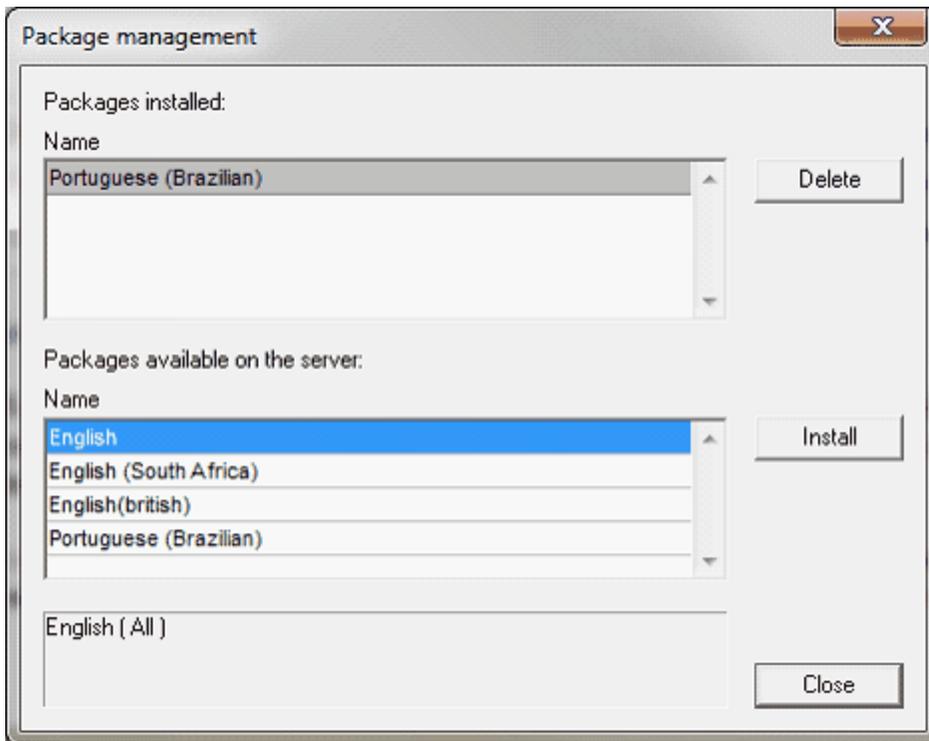
Authentic プラグインのためのHTML ページが開かれると、辞書パッケージをクライアント上に自動的に、またはユーザーのリクエストにより、インストールすることができます。インストールのメカニズムは以下のよう動作します：

1. Authentic Browser 内で開かれるHTML ページは[LoaderSettings ファイル](#)を参照します。
2. [LoaderSettings ファイル](#)は、サーバー上のスペルパッケージを参照します。[LoaderSettings ファイル](#)は、それぞれのスペルパッケージのために、パッケージが自動的にHTML ページがロードされた後にインストールされるか、またはユーザーがインストールについて問うとインストールされるように指定することができます。
3. スペルパッケージは、サーバー上の圧縮されているファイルからクライアントにインストールされます。

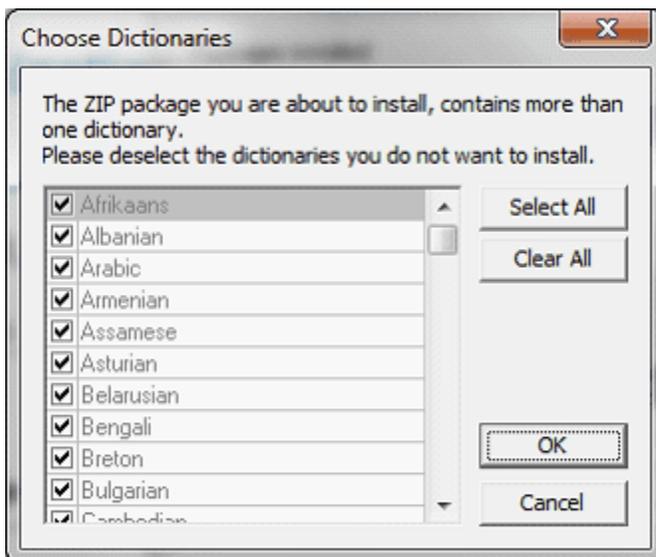
### 辞書のインストール

スペルチェッカーパッケージはサーバー上でZIP (.zip) ファイルとして保管され、Authentic Browser GUI を使用してローカルでインストールすることができます。スペルチェッカーパッケージをインストールするには、Authentic Browser GUI 内のパッケージの管理 アイコンをクリックします。パッケージの管理 ダイアログが表示されます（下のスクリーンショット）。辞書がインストールされておらず、スペルチェッカーアイコンがリンクされている場合、パッケージの管理 ダイアログも表示されます。

パッケージの管理 ダイアログは、サーバー上で使用することのできる全てのパッケージを下のペインでリストして表示します（下のスクリーンショット）。表示される名前は、サーバー上のZIP ファイルの名前です。インストールするファイルを選択し、「インストール」をクリックします。インストールされたパッケージを上側のペイン内で選択し、「削除」をクリックして削除することができます。



インストールするために選択された ZIP ファイルが複数の ZIP パッケージを含む場合、パッケージダイアログの選択がポップアップします（下のスクリーンショット）。このダイアログ内で、インストールするパッケージをチェック、またはインストールを希望しないパッケージのチェックを解除することができます。「OK」をクリックして、チェックされたパッケージをインストールします。このメカニズムにより各言語の辞書をサーバー上に、または単一の ZIP ファイルに保管できるようになります。



パッケージのインストールが、既存のパッケージを上書きする場合、警告メッセージが表示されます。Authentic Browser ユーザーは上書きを続行するか、上書きをキャンセルすることができます。

### 4.1.13 XMLData の使用

XMLData は現在表示されている XML ファイルの要素へのアクセスを与えます。必要なすべての変更を XML 構造の要素に対して行うことができます。XMLData の主な機能は以下のとおりです:

1. 全ての種類の要素の名前と値へのアクセス(例: 要素、属性)。
2. 全ての種類の新規の要素の作成。
3. 新規の要素の挿入と追加。
4. 既存の子要素の削除。

XMLSpy API で既に使用しており、XMLData インターフェイスの使用に詳しい場合は、XML ファイルに新規の要素が挿入、または既存の要素の名前が変更される場合考慮する点があることご注意ください。「新規のXMLData オブジェクトの作成と挿入」「要素の名前と値」を参照してください。

#### XMLData の構造

XMLData インターフェイスを使用する前に、既存のXML ファイルはXMLData 構造内どのようにマップされるかを知る必要があります。考慮する重要な点は、XMLData は属性のためにオブジェクトの個別のプランチを持たないということです。

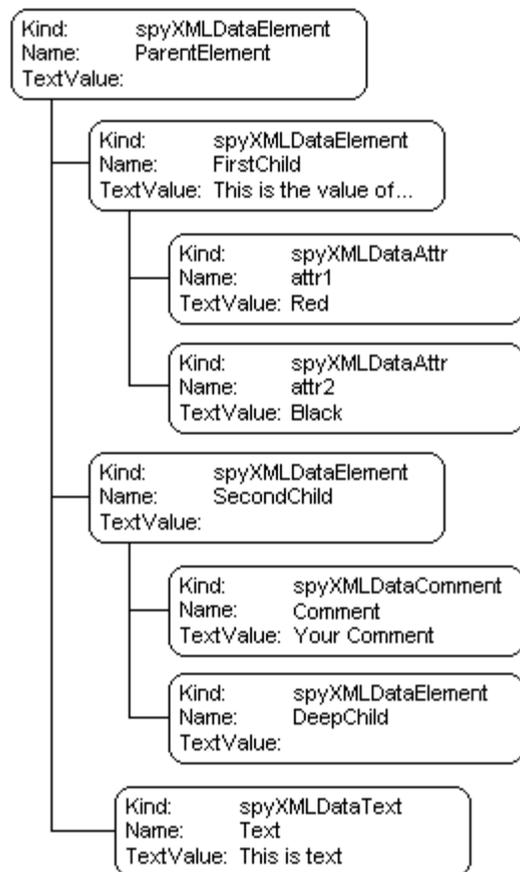
要素の属性は要素の子でもあります。 [XMLData.Kind](#) プロパティは要素の子の異なる型の差分を明確にする機会を与えます。

サンプル

このXMLコード、

```
<ParentElement>
  <FirstChild attr1="Red" attr2="Black">
    This is the value of FirstChild
  </FirstChild>
  <SecondChild>
    <!--Your Comment-->
    </DeepChild>
  </SecondChild>
  This is Text
</ParentElement>
```

は次のXMLData オブジェクト構造にマップされています:



ファイルの内部の全てのXML要素の親はプロパティ [Authentic.XMLRoot](#) です。このXMLData オブジェクトを使用して、構造内の他の全てのXML要素を参照します。

### 要素の名前と値

ToXML 要素の全ての型の名前と値を取得し変更するには、[XMLData.Name](#) と [XMLData.TextValue](#) プロパティを使用します。XMLData オブジェクトの一部と空の要素は関連付けられたテキストの値を持っていません。Authentic View 内の既存のXML要素の名前を変更することは奨励されません。名前はAuthentic View が要素のエントリ名などのように表示するか影響するため重要です。るかを影響します。詳細に関してはStyleVision ドキュメンテーションを参照してください。

### 新規のXMLData オブジェクトの作成と挿入

新規のXML言語エントリの作成には次のステップが必要です:

1. 新規のXMLData オブジェクトを作成します:  
[Authentic.CreateChild](#) メソッドを使用して新規のXMLData オブジェクトを作成します。新規のXML エンティティを挿入する前に名前と値を設定します (ポイント3を参照)。
2. 新規のXMLData オブジェクトのための正確な場所を検索します:  
新規のXMLData オブジェクトを挿入するには、親への参照を最初に取得する必要があります。新規の子が親の最後の子になる場合は、[XMLData.AppendChild](#) メソッドを使用してXMLData オブジェクトを挿入します。  
子オブジェクトのシーケンス内で新規の子がロケットされる場合、[XMLData.GetFirstChild](#) と [XMLData.GetNextChild](#) を使用して子の反復子を新規の子が挿入される前に移動してください。

3. [XMLData.InsertChild](#) を使用して新規の子を挿入します。新規の子は現在の子のすぐ前に挿入されます。Authentic View 内では、作成された要素の追加の子ノードがXML ファイルと追加される場合があります。これは StyleVision を使用して変更することのできるノード設定により異なります。

次のサンプルは、<FirstChild> と <SecondChild> 要素の間3番目の子を追加します:

```
Dim objParent
Dim objChild
Dim objNewChild

Set objNewChild = objPlugIn.CreateChild(spyXMLDataElement)
objNewChild.Name = "OneAndAHalf"

'objParent is set to <ParentElement>
'GetFirstChild(-1) gets all children of the parent element
'and move to <SecondChild>
Set objChild = objParent.GetFirstChild(-1)
Set objChild = objParent.GetNextChild

objParent.InsertChild objNewChild
Set objNewChild = Nothing
```

子要素は区別な順序で挿入されます。他の子要素の後のシーケンスに属性を挿入することは回避してください。これは、属性が他の型の前に来る要素を持つことができず、また、他の要素は型属性の前に来る要素を持つことができないことを意味します。

Authentic View は属性以外のXML 要素のテキストの値を表示するために特別な処理を必要とします。テキスト(またはコンテンツ)は型テキストの追加子ノードの一部である必要があります。このような要素を、ラメーターの値 6 を持つ [Authentic.CreateChild](#) を使用して作成することができます。要素のテキストの値を直接設定する代わりに、子ノードのテキストの値を設定してください。

### 既存のXMLData オブジェクトのコピー

既存のXMLData オブジェクトを同じファイル内の異なる場所に挿入数する場合、XMLData.InsertChild と XMLData.AppendChild メソッドを使用することはできません。これらのメソッドは新規のXMLData オブジェクトのためのみ使用することができます。

InsertChild または AppendChild を使用する代わりに、手動でオブジェクトを階層的にコピーすることができます。JavaScript で書かれている次の機能は、でかかれており、再帰的にXMLData をコピーするために使用します:

```
// this function returns a complete copy of XMLData object
function GetCopy(objXMLData)
{
    var objNew;
    objNew = objPlugIn.CreateChild(objXMLData.Kind);

    objNew.Name = objXMLData.Name;
    objNew.TextValue = objXMLData.TextValue;

    if(objXMLData.HasChildren) {
        var objChild;
        objChild = objXMLData.GetFirstChild(-1);

        while(objChild) {
            try {
                objNew.AppendChild(GetCopy(objChild));
            }
        }
    }
}
```

```

        objChild = objXMLData.GetNextChild();
    }
    catch(e) {
        objChild = null;
    }
}
}

return objNew;
}

```

### XMLData オブジェクトの削除

XMLData は子オブジェクトの削除のための2つのメソッド [XMLData.EraseAllChildren](#) と [XMLData.EraseCurrentChild](#) を与えます。XMLData オブジェクトを削除するには、削除する要素の親へのアクセスが必要です。親 XMLData オブジェクトへの参照を取得するために [XMLData.GetFirstChild](#) と [XMLData.GetNextChild](#) を使用してください。XML 要素の削除方法のサンプルは [XMLData.EraseAllChildren](#) と [XMLData.EraseCurrentChild](#) のメソッドの説明を確認してください。

## 4.1.14 DOM と XMLData

XMLData インターフェイスは、現在のドキュメントのXML 構造へのアクセスをDOM よりも簡単なメソッドで提供します。XMLData インターフェイスは既存の、または新規に作成されたXML データを読み取る、または変更するためのミニマリストのアプローチです。しかしながら、外部ソースからアクセス、または MSXML DOM を実装を希望する場合 DOM ツリーを使用する必要があります。

**ProcessDOMNode()** と **ProcessXMLDataNode()** 関数はXMLData とDOM 間のXML 構造のセグメントを変換します。

**ProcessDOMNode()** 関数を使用します:

- **objNode** に変換するDOM セグメントのルート要素をパスします
- **objCreator** 内の **CreateChild()** メソッドを持つプラグインオブジェクトにパスします

**ProcessXMLDataNode()** 関数を使用する方法

- **objXMLData** 内のXMLData セグメントのルート要素をパス
- **xmldoc** 内のMSXML を使用して作成されたDOMDocument オブジェクトにパスします

```

////////////////////////////////////
// DOM to XMLData conversion

function ProcessDOMNode(objNode,objCreator)
{
    var objRoot;
    objRoot = CreateXMLDataFromDOMNode(objNode,objCreator);

    if(objRoot) {
        if((objNode.nodeValue != null) && (objNode.nodeValue.length > 0))
            objRoot.TextValue = objNode.nodeValue;

        // add attributes
        if(objNode.attributes) {
            var Attribute;
            var oNodeList = objNode.attributes;

```

```
        for(var i = 0;i < oNodeList.length; i++) {
            Attribute = oNodeList.item(i);

            var newNode;
            newNode = ProcessDOMNode(Attribute,objCreator);

            objRoot.AppendChild(newNode);
        }
    }

    if(objNode.hasChildNodes) {
        try {
            // add children
            var Item;
            oNodeList = objNode.childNodes;

            for(var i = 0;i < oNodeList.length; i++) {
                Item = oNodeList.item(i);

                var newNode;
                newNode = ProcessDOMNode(Item,objCreator);

                objRoot.AppendChild(newNode);
            }
        }
        catch(err) {
        }
    }
}

return objRoot;
}

function CreateXMLDataFromDOMNode(objNode,objCreator)
{
    var bSetName = true;
    var bSetValue = true;

    var nKind = 4;

    switch(objNode.nodeType) {
        case 2:nKind = 5;break;
        case 3:nKind = 6;bSetName = false;break;
        case 4:nKind = 7;bSetName = false;break;
        case 8:nKind = 8;bSetName = false;break;
        case 7:nKind = 9;break;
    }

    var objNew = null;
    objNew = objCreator.CreateChild(nKind);

    if(bSetName)
        objNew.Name = objNode.nodeName;
}
```

```
        if(bSetValue && (objNode.nodeValue != null))
            objNew.TextValue = objNode.nodeValue;

        return objNew;
    }

    //////////////////////////////////////
    // XMLData to DOM conversion

function ProcessXMLDataNode(objXMLData,xmlDoc)
{
    var objRoot;
    objRoot = CreateDOMNodeFromXMLData(objXMLData,xmlDoc);

    if(objRoot) {
        if(IsTextNodeEnabled(objRoot) && (objXMLData.TextValue.length > 0))
            objRoot.appendChild(xmlDoc.createTextNode(objXMLData.TextValue));

        if(objXMLData.HasChildren) {
            try {
                var objChild;
                objChild = objXMLData.GetFirstChild(-1);

                while(true) {
                    if(objChild) {
                        var newNode;
                        newNode = ProcessXMLDataNode(objChild,xmlDoc);

                        if(newNode.nodeType == 2) {
                            // child node is an attribute
                            objRoot.attributes.setNamedItem(newNode);
                        }
                        else
                            objRoot.appendChild(newNode);
                    }

                    objChild = objXMLData.GetNextChild();
                }
            }
            catch(err) {
            }
        }

        return objRoot;
    }
}

function CreateDOMNodeFromXMLData(objXMLData,xmlDoc)
{
    switch(objXMLData.Kind) {
        case 4: return xmlDoc.createElement(objXMLData.Name);
        case 5: return xmlDoc.createAttribute(objXMLData.Name);
        case 6: return xmlDoc.createTextNode(objXMLData.TextValue);
        case 7: return xmlDoc.createCDATASection(objXMLData.TextValue);
    }
}
```

```

        case 8: return xmlDoc.createComment(objXMLData.TextValue);
        case 9: return
xmlDoc.createProcessingInstruction(objXMLData.Name, objXMLData.TextValue);
    }

    return xmlDoc.createElement(objXMLData.Name);
}

function IsTextNodeEnabled(objNode)
{
    switch(objNode.nodeType) {
        case 1:
        case 2:
        case 5:
        case 6:
        case 11: return true;
    }

    return false;
}

```

#### 4.1.15 Authentic スクリプト

Authentic スクリプト の機能はSPS デザインに柔軟かつインタラクティブな機能を与えます。StyleVision Enterprise と Professional エディション内でこれらのデザインを作成、または 編集することができます。また、Altova 製品のEnterprise と Professional エディションのAuthentic View で閲覧することができます。

Altova 製品内での機能に対するサポートのリストは、下のテーブルに表示されています。しかしながら、Authentic Browser プラグインの信頼されているバージョン内では、内部スクリプトはセキュリティの事情によりオフに切り替えられています。

Altova 製品	Authentic スクリプトの作成	Authentic スクリプトの有効化済み
StyleVision Enterprise	はい	はい
StyleVision Professional	はい	はい
StyleVision Standard *	いいえ	いいえ
XMLSpy Enterprise	いいえ	はい
XMLSpy Professional	いいえ	はい
XMLSpy Standard	いいえ	いいえ
AuthenticDesktop Community	いいえ	いいえ
AuthenticDesktop Enterprise	いいえ	はい
Authentic Browser プラグイン Community **	いいえ	いいえ
Authentic Browser プラグイン Enterprise 信頼されている ***	いいえ	はい
Authentic Browser プラグイン Enterprise 信頼されていない	いいえ	はい

\* Authentic View なし

\*\* 信頼されている、および、信頼されていないバージョンの両方

\*\*\* スクリプト済みのデザインが表示されています。内部のマクロの実行またはイベントハンドリングは無し。外部イベントが実行されます。

Authentic スクリプトはAltova 製品と同じ振る舞いをします。製品固有のロード または 設定を必要とすることはありません。

### Authentic スクリプトの作動のしくみ

SPS デザインのデザイナーはAuthentic スクリプトを使用してAuthentic ドキュメントを2通りの方法でインタラクティブにします。:

- デザイン要素へのユーザー定義アクション(マクロ)のためのスクリプトにツールバーボタンとコンテキストメニューアイテムを割り当てる方法。
- Authentic View 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する方法。

Authentic ドキュメントをインタラクティブにするために必要な全てのスクリプトは、StyleVision GUI (Enterprise とProfessional エディション) 内で行われます。書式、マクロ、およびイベントハンドラーはStyleVision のスクリプトエディターインターフェイスで作成され、これらのスクリプトはSPS と共に保存されます。StyleVision のデザインビュー内では、保存されたスクリプトはデザイン要素、ツールバーボタン、およびコンテキストメニューに割り当てられます。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 内で開かれると、ドキュメントに更なる柔軟性が与えられ、更にインタラクティブなドキュメントになります(上のテーブル参照)。

### Authentic スクリプトのためのドキュメンテーション

Authentic スクリプトのためのドキュメントは、StyleVision のドキュメント内で使用することができます。[Altova Web サイトの製品ドキュメンテーションページ](#)を介してオンラインで閲覧することもできます。

## 4.2 オブジェクト

このセクションには、多種の Authentic Browser オブジェクトのリストと説明が含まれています。

### 4.2.1 Authentic

次も参照してください

メソッド

[StartEditing](#)

[LoadXML](#)

[Reset](#)

[Save](#)

[SavePOST](#)

[SaveXML](#)

[ValidateDocument](#)

[EditClear](#)

[EditCopy](#)

[EditCut](#)

[EditPaste](#)

[EditRedo](#)

[EditSelectAll](#)

[EditUndo](#)

[RowAppend](#)

[RowDelete](#)

[RowDuplicate](#)

[RowInsert](#)

[RowMoveDown](#)

[RowMoveUp](#)

[FindDialog](#)

[FindNext](#)

[ReplaceDialog](#)

[ApplyTextState](#)

[IsTextStateApplied](#)

[IsTextStateEnabled](#)

[MarkUpView](#)

[Print](#)

[PrintPreview](#)

[CreateChild](#)

[GetAllowedElements](#)

[GetAllAttributes](#)

[GetNextVisible](#)

[GetPreviousVisible](#)

[SelectionMoveTabOrder](#)

[SelectionSet](#)

[ClearSelection](#)

[attachCallBack](#)

[ReloadToolbars](#)

[StartSpellChecking](#)

[GetFileVersion](#)

[RedrawEntryHelpers](#)

[SetUnmodified](#)

[ClearUndoRedo](#)

プロパティ

[AuthenticView](#)

[IsEditClearEnabled](#)

[IsEditCopyEnabled](#)

[IsEditCutEnabled](#)

[IsEditPasteEnabled](#)

[IsEditRedoEnabled](#)

[IsEditUndoEnabled](#)

[IsFindNextEnabled](#)

[IsRowAppendEnabled](#)

[IsRowDeleteEnabled](#)

[IsRowDuplicateEnabled](#)

[IsRowInsertEnabled](#)

[IsRowMoveDownEnabled](#)

[IsRowMoveUpEnabled](#)

[SchemaLoadObject](#)

[XMLDataLoadObject](#)

[DesignDataLoadObject](#)

[XMLDataSaveUrl](#)

[XMLRoot](#)

[CurrentSelection](#)

[event](#)

[validationBadData](#)

[validationMessage](#)

[ToolbarsEnabled](#)

[ToolbarTooltipsEnabled](#)

[AutoHideUnusedCommandGroups](#)

[TextStateToolbarLine](#)

[TextStateBmpURL](#)

[ToolbarRows](#)

[UICommands](#)

[XMLTable](#)

[BaseURL](#)

[EntryHelpersEnabled](#)

[EntryHelperSize](#)

[EntryHelperAlignment](#)

[EntryHelperWindows](#)

[EnableModifications](#)

[Modified](#)

[SaveButtonAutoEnable](#)

イベント

[SelectionChanged](#)

[ControlInitialized](#)

説明

Authentic クラス

#### 4.2.1.1 Authentic.ApplyTextState

次も参照してください:

宣言: `ApplyTextState(elementName` を文字列として

**説明**

パラメーター `elementName` により定義されているテキストの状態を適用、または、削除します。パラメーター `elementName` のための一般的なサンプルは、**厳密** および **斜体** です。

XML ドキュメント内では、サブ要素を含む可能性のあるデータのセグメントが存在します。例えば、次の HTML を考慮してください！

```
<b>fragment</b>
```

HTML タグ `<b>` により文字フラグメントは太字で表示されます。しかしながら、HTML パーサーがタグ `<b>` が太字であると既知であるからです。XML を使用することにより柔軟性が増します。希望するとおり XML タグを定義することができます。XML を使用してテキストの状態を適用することができます。適用されるテキストの状態はスキーマの一部である必要があります。

例えば、`OrgChart.xml`、`OrgChart.sps`、`OrgChart.xsd` サンプル内では、タグ `<strong>` は太字と同じです。太字 マインドを適用するために、**ApplyTextState()** が呼び出されます。行と編集の操作などでは、テキストの状態を適用するためにテストすることが必要です。

次も参照してください！ [IsTextStateEnabled](#) と [IsTextStateApplied](#)。

#### 4.2.1.2 Authentic.attachCallback

##### 使用しないでください

接続ポイントイベントの説明は [ここ](#) で説明されています。

次も参照してください！

**宣言:** `attachCallback(bstrName を文字列として、varCallback を Variant として)`

**説明**

Authentic View はカスタムコールバック関数を使用して処理されるイベントを提供します。全てのイベントハンドラーはパラメーターを取らず、返される値は無視されます。固有のイベントが挙げられる際にこの情報を取得するには、[イベント](#) オブジェクトのプロパティに従って読み取ります。

現在使用することのできるイベントのリスト:

```
ondragover
ondrop
onkeydown
onkeyup
onkeypressed
onmousemove
onbuttonup
onbuttondown
oneditpaste
oneditcut
oneditcopy
```

バージョン 3.0.0.0 から:

```
ondoceditcommand
```

バージョン: 5.3.0.0 から

onbuttondoubleclick

#### JavaScript サンプル

```
// somewhere in your script:
objPlugIn.attachCallBack("ondragover", OnDragOver);
objPlugIn.attachCallBack("ondrop", OnDrop);

// event handlers
function OnDragOver()
{
    if( !objPlugIn.event.dataTransfer.ownDrag &&
        objPlugIn.event.dataTransfer.type == "TEXT")
    {
        objPlugIn.event.dataTransfer.dropEffect = 1;
        objPlugIn.event.cancelBubble = true;
    }
}

// OnDrop() replaces the complete text value of the XML
// element with the selection from the drag operation
function OnDrop()
{
    var objTransfer = objPlugIn.event.dataTransfer;

    if( !objTransfer.ownDrag &&
        (objTransfer.type == "TEXT"))
        objPlugIn.event.srcElement.TextValue = objTransfer.getData();
}
}
```

### 4.2.1.3 AuthenticView

次も参照してください:

**宣言:** [AuthenticView](#) を [AuthenticView](#) として (読み取り専用)

#### 説明

AuthenticView に固有のプロパティとメソッドへのアクセスを与えるオブジェクトを返します。

[AuthenticView](#) は固有の機能の既存のビューとオーバーラップします。AuthenticView の将来のバージョンは固有の機能のすべてのビューを含みます。AuthenticView オブジェクトは将来の実装のために奨励されるインターフェイスです。

#### サンプル

AuthenticView オブジェクトの使用法に関する情報については、[動的テーブルのプルシート](#) を参照してください。

#### 4.2.1.4 Authentic.AutoHideUnusedCommandGroups

**宣言:** `AutoHideUnusedCommandGroups` を **ブール値** として

**説明**

使用されていない ツールバー-CommandGroups は自動的に非表示になり、True が返されます (例: 現在のSPS ファイルがXML テーブルをサポートしない場合、XML - テーブルコマンド)。

デフォルトの値: True

#### 4.2.1.5 Authentic.BaseURL

**宣言:** `BaseURL` を **文字列** として

**説明**

このプロパティはベースURLを相対的なパスに解決するように設定します。

URL が設定されていない場合、現在のXML ファイルのロケーションが使用されます。

#### 4.2.1.6 Authentic.ClearSelection

**宣言:** `ClearSelection()`

**説明**

メソッドは現在の選択をクリアします。CurrentSelection プロパティを使用して選択を取得しようと試みると、ユーザーがノードを選択するまで、または [SelectionSet\(\)](#) への呼び出しが成功するまで失敗します。

現在の選択を持つノードを削除する前に、選択は異なるノードに設定されるか、クリアされる必要があります。

#### 4.2.1.7 Authentic.ClearUndoRedo

**宣言:** `ClearUndoRedo()`

**説明**

メソッドは元に戻す/やり直しメニュー全体をクリアします。

#### 4.2.1.8 Authentic.ControlInitialized

次も参照してください:

**宣言:** `ControlInitialized`

**説明**

コントロールが作成され初期化されるとこのイベントが挙げられます。

次も参照してください! [接続ポイントイベント](#)。

#### 4.2.1.9 Authentic.CreateChild

次も参照してください!

**宣言:** `CreateChild(nKind` を [SPYXMLDataKind](#) として) を [XMLData](#) として

戻り値  
新規のXML ノード

##### 説明

`CreateChild` メソッドは、[XMLData](#) インターフェイスを使用して現在のドキュメントのXML 構造を入力する新規のノードを作成するために使用されます。

次も参照してください! [XMLData.AppendChild](#) と [XMLData.InsertChild](#) 。

#### 4.2.1.10 Authentic.CurrentSelection

次も参照してください!

**宣言:** `CurrentSelection` を [AuthenticSelection](#) として

説明  
プロパティはAuthentic View 内の現在の選択にアクセスを提供します。

下のサンプルコードは現在の選択の複合型のテキストを抽出します:

JavaScript:

```
// somewhere in your script:
GetSelection(objPlugIn.CurrentSelection);

// GetSelection() collects complete text selection
function GetSelection(objSel)
{
    var strText = "";

    var objCurrent = objSel.Start;

    while (!objSel.End.IsSameNode(objCurrent))
    {
        objCurrent = objPlugIn.GetNextVisible(objCurrent);
        strText += objCurrent.TextValue;
    }

    strText += objSel.End.TextValue.substring(0, objSel.EndTextPosition);
}
```

```
        return objSel.Start.TextValue.substr(objSel.StartTextPosition) + strText;
    }
```

#### 4.2.1.11 Authentic.DesignDataLoadObject

次も参照してください:

**宣言:** [DesignDataLoadObject](#) を [AuthenticLoadObject](#) として

##### 説明

DesignDataLoadObject は SPS ドキュメントへの参照が含まれています。SPS ドキュメントは WYSIWYG 編集環境を生成するために使用され、通常 StyleVision により生成されます。

次も参照してください: [SchemaLoadObject](#)。

#### 4.2.1.12 Authentic.EditClear

次も参照してください:

**宣言:** [EditClear](#)

##### 説明

現在の選択をクリアします。

#### 4.2.1.13 Authentic.EditCopy

次も参照してください:

**宣言:** [EditCopy](#)

##### 説明

現在の選択をクリップボードにコピーします。

#### 4.2.1.14 Authentic.EditCut

次も参照してください:

**宣言:** [EditCut](#)

##### 説明

ドキュメントから現在の選択を切り取り、クリップボードに貼り付けます。

#### 4.2.1.15 Authentic.EditPaste

次も参照してください！

宣言: [EditPaste](#)

説明

クリップボードからコンテンツをドキュメントに貼り付けます。

#### 4.2.1.16 Authentic.EditRedo

次も参照してください！

宣言: [EditRedo](#)

説明

最後の元に戻すステップをやり直します。

#### 4.2.1.17 Authentic.EditSelectAll

次も参照してください！

宣言: [EditSelectAll](#)

説明

マウスがドキュメント全体を選択します。

#### 4.2.1.18 Authentic.EditUndo

次も参照してください！

宣言: [EditUndo](#)

説明

最後のアクションを元に戻します。

#### 4.2.1.19 Authentic.EnableModifications

宣言: [EnableModifications](#) を **ブール値** として

**説明**

XML コンテンツに対する Authentic Browser の変更が有効化されている場合、True です。次も参照してください [変更された](#) プロパティ。

デフォルトの値: TRUE

#### 4.2.1.20 Authentic.EntryHelperAlignment

**宣言:** [EntryHelperAlignment](#) を [SPYAuthenticToolbarAlignment](#) として

**説明**

このプロパティを入力ヘルパーの場所を設定するために使用することができます。デフォルトの値は右側を入力ヘルパーを配置する3です。

#### 4.2.1.21 Authentic.EntryHelpersEnabled

**宣言:** [EntryHelpersEnabled](#) を [ブール値](#) として

**説明**

Authentic Browser 入力ヘルパーが有効化されている場合、True です。このプロパティを入力ヘルパーを有効化、または無効化するために使用することができます。

デフォルトの値: FALSE

#### 4.2.1.22 Authentic.EntryHelperSize

**宣言:** [EntryHelperSize](#) を [Integer](#) として

**説明**

このプロパティを入力ヘルパーエリアのサイズをピクセルで設定するために使用することができます。この値が無視される場合、プロパティを1に設定します。

デフォルトの値: -1

#### 4.2.1.23 Authentic.EntryHelperWindows

**宣言:** [EntryHelperWindows](#) を [SPYAuthenticEntryHelperWindows](#) として

**説明**

このプロパティは入力ヘルパーが表示されるかを定義するために使用されます。1つ以上の入力ヘルパーウィンドウを表示するために値が使用されます。デフォルトの値の7は3つの入力ヘルパーすべてを表示します。

#### 4.2.1.24 Authentic.event

次も参照してください！

宣言: `event` を [AuthenticEvent](#) として

##### 説明

イベントプロキシは現在のイベントに関する情報を含むイベントデータオブジェクトをホールドしています。

#### 4.2.1.25 Authentic.FindDialog

次も参照してください！

宣言: [FindDialog](#)

##### 説明

`FindDialog` を表示します。

次も参照してください！ [検索と置換](#)。

#### 4.2.1.26 Authentic.FindNext

次も参照してください！

宣言: [FindNext](#)

##### 説明

メソッドは次の操作を検索します。

次も参照してください！ [検索と置換](#)。

#### 4.2.1.27 Authentic.GetAllAttributes

次も参照してください！

宣言: `GetAllAttributes(pForElement を XMLData として, pElements を Variant として)`

##### 説明

`GetAllAttributes()` は文字列の配列として指定されている要素のために許可された属性を返します。

JavaScript サンプル

```
function GetAllAttributes ()
{
```

```

var arrElements = new Array(1);

var objStart = objPlugIn.CurrentSelection.Start;

var strText;
strText = "Valid attributes at current selection:\n\n";

for(var i = 1;i <= 4;i++)
{
    objPlugIn.GetAllAttributes(objStart, arrElements);
    strText = strText + ListArray(arrElements) + "-----\n";
}

return strText;
}

function ListArray(arrIn)
{
    var strText = "";

    if(typeof(arrIn) == "object")
    {
        for(var i = 0;i <= (arrIn.length - 1);i++)
            strText = strText + arrIn[i] + "\n";
    }

    return strText;
}

```

#### VBScript example:

```

Sub DisplayAllowedAttributes
    dim arrElements()

    dim objStart
    dim objEnd
    set objStart = objPlugIn.CurrentSelection.Start
    set objEnd = objPlugIn.CurrentSelection.End

    dim strText
    strText = "Valid attributes at current selection:" & chr(13) & chr(13)

    dim i

    For i = 1 To 4
        objView.GetAllAttributes objStart, arrElements
        strText = strText & ListArray(arrElements) & "-----" &
chr(13)
    Next

    msgbox strText
End Sub

Function ListArray(arrIn)
    dim strText

```

```
    If IsArray(arrIn) Then
        dim i

        For i = 0 To UBound(arrIn)
            strText = strText & arrIn(i) & chr(13)
        Next
    End If

    ListArray = strText
End Function
```

#### 4.2.1.28 Authentic.GetAllowedElements

次も参照してください！

**宣言:** `GetAllowedElements(nAction を SPYAuthenticElementActions として、pStartElement を XMLData として、pEndElement を XMLData として、pElements を Variant として)`

##### 説明

`GetAllowedElements()` は *nAction* により指定されている異なるアクションのために許可されている要素を返します。

##### JavaScript サンプル

```
function GetAllowed()
{
    var arrElements = new Array(1);

    var objStart = objPlugIn.CurrentSelection.Start;
    var objEnd = objPlugIn.CurrentSelection.End;

    var strText;
    strText = "valid elements at current selection:\n\n";

    for(var i = 0;i <= 4;i++) {
        objPlugIn.GetAllowedElements(i,objStart,objEnd,arrElements);
        strText = strText + ListArray(arrElements) + "-----\n";
    }

    return strText;
}

function ListArray(arrIn)
{
    var strText = "";

    if(typeof(arrIn) == "object") {
        for(var i = 0;i <= (arrIn.length - 1);i++)
            strText = strText + arrIn[i] + "\n";
    }
}
```

```
        return strText;
    }
```

VBScript example:

```
Sub DisplayAllowed
    dim arrElements()

    dim objStart
    dim objEnd
    set objStart = objPlugIn.CurrentSelection.Start
    set objEnd = objPlugIn.CurrentSelection.End

    dim strText
    strText = "Valid elements at current selection:" & chr(13) & chr(13)

    dim i

    For i = 1 To 4
        objView.GetAllowedElements i,objStart,objEnd,arrElements
        strText = strText & ListArray(arrElements) & "-----" &
chr(13)
    Next

    msgbox strText
End Sub

Function ListArray(arrIn)
    dim strText

    If IsArray(arrIn) Then
        dim i

        For i = 0 To UBound(arrIn)
            strText = strText & arrIn(i) & chr(13)
        Next
    End If

    ListArray = strText
End Function
```

#### 4.2.1.29 Authentic.GetFileVersion

次も参照してください:

**宣言:** `GetFileVersion`(*strVersion* を文字列として)

**説明**

メソッドはコンポーネントのバージョンを書式 5.0.0.0. の文字列で返します。

#### 4.2.1.30 Authentic.GetNextVisible

次も参照してください！

宣言: `GetNextVisible(pElement` を [XMLData](#) として ) を [XMLData](#) として

##### 説明

メソッドはドキュメント内で次に視覚性のあるXML要素を取得します。

#### 4.2.1.31 Authentic.GetPreviousVisible

次も参照してください！

宣言: `GetPreviousVisible(pElement` を [XMLData](#) として ) を [XMLData](#) として

##### 説明

メソッドはドキュメント内で前に視覚性のあるXML要素を取得します。

#### 4.2.1.32 Authentic.IsEditClearEnabled

次も参照してください！

宣言: `IsEditClearEnabled` を [ブール値](#) として

##### 説明

[EditClear](#) が可能な場合、True です。

次も参照してください！ [編集の操作](#)。

#### 4.2.1.33 Authentic.IsEditCopyEnabled

次も参照してください！

宣言: `IsEditCopyEnabled` を [ブール値](#) として

##### 説明

グループボードにコピーが可能な場合、True です。

次も参照してください！ [EditCopy](#) と [編集の操作](#)。

#### 4.2.1.34 Authentic.IsEditCutEnabled

次も参照してください！

宣言: `IsEditCutEnabled` を **ブール値** として

説明

[EditCut](#) が現在可能な場合、True です。

次も参照してください！ [編集の操作](#)。

#### 4.2.1.35 Authentic.IsEditPasteEnabled

次も参照してください！

宣言: `IsEditPasteEnabled` を **ブール値** として

説明

[EditPaste](#) が可能な場合、True です。

次も参照してください！ [編集の操作](#)。

#### 4.2.1.36 Authentic.IsEditRedoEnabled

次も参照してください！

宣言: `IsEditRedoEnabled` を **ブール値** として

説明

[EditRedo](#) が現在可能な場合、True です。

次も参照してください！ [編集の操作](#)。

#### 4.2.1.37 Authentic.IsEditUndoEnabled

次も参照してください！

宣言: `IsEditUndoEnabled` を **ブール値** として

説明

[EditUndo](#) が可能な場合、True です。

次も参照してください！ [編集の操作](#)。

#### 4.2.1.38 Authentic.IsFindNextEnabled

次も参照してください！

宣言: `IsFindNextEnabled` を **ブール値** として

##### 説明

`FindNext` が現在可能な場合、`True` を返します。発生がこれ以上存在しない場合、`False` が返されます。

次も参照してください！ [検索と置換](#) と [FindDialog](#)。

#### 4.2.1.39 Authentic.IsRowAppendEnabled

次も参照してください！

宣言: `IsRowAppendEnabled` を **ブール値** として

##### 説明

[RowAppend](#) が可能な場合、`True` です。

次も参照してください！ [行の操作](#)。

#### 4.2.1.40 Authentic.IsRowDeleteEnabled

次も参照してください！

宣言: `IsRowDeleteEnabled` を **ブール値** として

##### 説明

[RowDelete](#) が可能な場合、`True` です。

次も参照してください！ [行の操作](#)。

#### 4.2.1.41 Authentic.IsRowDuplicateEnabled

次も参照してください！

宣言: `IsRowDuplicateEnabled` を **ブール値** として

##### 説明

[RowDuplicate](#) が現在可能な場合、`True` です。

次も参照してください！ [行の操作](#)。

#### 4.2.1.42 Authentic.IsRowInsertEnabled

次も参照してください！

宣言: `IsRowInsertEnabled` を **ブール値** として

説明

`RowInsert` が可能な場合、True です。

次も参照してください！ [行の操作](#)。

#### 4.2.1.43 Authentic.IsRowMoveDownEnabled

次も参照してください！

宣言: `IsRowMoveDownEnabled` を **ブール値** として

説明

`RowMoveDown` が現在可能な場合、True です。

次も参照してください！ [行の操作](#)。

#### 4.2.1.44 Authentic.IsRowMoveUpEnabled

次も参照してください！

宣言: `IsRowMoveUpEnabled` を **ブール値** として

説明

`RowMoveUp` が可能な場合、True です。

次も参照してください！ [行の操作](#)。

#### 4.2.1.45 Authentic.IsTextStateApplied

次も参照してください！

宣言: `IsTextStateApplied`(`elementName` を **文字列** として) を **ブール値** として

説明

テキストの状態が既に適用されたかをチェックします。パラメーター `elementName` のための一般的なサンプルは **厳密および斜体** です。

#### 4.2.1.46 Authentic.IsTextStateEnabled

次も参照してください:

**宣言:** `IsTextStateEnabled`(*elementName* を文字列として) を **ブール値** として

##### 説明

テキストの状態に適用可能かをチェックします。パラメーター *elementName* のための一般的なサンプルは厳密および斜体です。

#### 4.2.1.47 Authentic.LoadXML

次も参照してください:

**宣言:** `LoadXML`(*xmlString* を文字列として)

##### 説明

XML 文字列が適用された現在の XML ドキュメントをロードします。新しいコンテンツがすぐに表示されます。

#### 4.2.1.48 Authentic.MarkUpView

次も参照してください:

**宣言:** `MarkUpView`(*kind* を [SPYAuthenticMarkupVisibility](#) として)

##### 説明

デフォルトでは、HTML テクニックを使用してデフォルトのドキュメントは表意されます。しかし編集タグが表示されることが好ましい場合があります。このメソッドを使用するとマークアップタグの異なる型を表示することが可能になります。

#### 4.2.1.49 Authentic.Modified

**宣言:** `Modified` を **ブール値** として

##### 説明

XML コンテンツが変更されている場合、True です。

このプロパティは読み取り専用です。

#### 4.2.1.50 Authentic.Print

次も参照してください:

**宣言:** `Print`

**説明**

現在編集中のドキュメントを印刷します。

#### 4.2.1.51 Authentic.PrintPreview

次も参照してください！

**宣言:** [PrintPreview](#)

**説明**

現在編集中のプレビュードキュメントを印刷します。

#### 4.2.1.52 Authentic.RedrawEntryHelpers

**宣言:** [RedrawEntryHelpers\(\)](#)

**説明**

RedrawEntryHelpers は [EntryHelpersEnabled](#)、[EntryHelperAlignment](#)、[EntryHelperSize](#) と [EntryHelperWindows](#) プロパティから値を取り、入力ヘルパーウィンドウを描きなおします。

#### 4.2.1.53 Authentic.ReloadToolbars

**宣言:** [ReloadToolbars\(\)](#)

**説明**

ReloadToolbars は [ToolbarRows](#) コレクションを読み取り、ツールバービューを描きなおします。

#### 4.2.1.54 Authentic.ReplaceDialog

次も参照してください！

**宣言:** [ReplaceDialog](#)

**説明**

ReplaceDialog を表示します。

次も参照してください！ [検索と置換](#)

#### 4.2.1.55 Authentic.Reset

使用しないでください

[Authentic.StartEditing](#) を代わりに使用してください。

次も参照してください！

宣言: [Reset](#)

##### 説明

編集されているデータをリセットします。通常 XML、XSL と SPS ドキュメントの新規のセットを編集する前に呼び出されます。

モードビューを変更せず、現在表示されているドキュメントと継続して作業することができます。

#### 4.2.1.56 Authentic.RowAppend

次も参照してください！

宣言: [RowAppend](#)

##### 説明

現在のポジションで行を追加します。

次も参照してください！ [行の操作](#)。

#### 4.2.1.57 Authentic.RowDelete

次も参照してください！

宣言: [RowDelete](#)

##### 説明

現在選択されている行を削除します。

次も参照してください！ [行の操作](#)。

#### 4.2.1.58 Authentic.RowDuplicate

次も参照してください！

宣言: [RowDuplicate](#)

**説明**

メソッドは現在選択されている行を複製します。

次も参照してください！ [行の操作](#)。

#### 4.2.1.59 Authentic.RowInsert

次も参照してください！

**宣言:** [RowInsert](#)

**説明**

現在の選択のすぐ上に新規の行を挿入します。

次も参照してください！ [行の操作](#)。

#### 4.2.1.60 Authentic.RowMoveDown

次も参照してください！

**宣言:** [RowMoveDown](#)

**説明**

現在の行を1つ下に移動します。

次も参照してください！ [行の操作](#)。

#### 4.2.1.61 Authentic.RowMoveUp

次も参照してください！

**宣言:** [RowMoveUp](#)

**説明**

現在の行を1つ上に移動します。

次も参照してください！ [行の操作](#)。

#### 4.2.1.62 Authentic.Save

次も参照してください！

**宣言:** [Save](#)

**説明**

プロパティ [XMLDataSaveUrl](#) により指定される URL にドキュメントを保存します。信頼されていないバージョンに関しては、フルローカルパスを使用することができます。

プラグインは現在表示されている XML ファイルを保存するようにサーバーに HTTP PUT リクエストを送信します。

#### 4.2.1.63 Authentic.SaveButtonAutoEnable

**宣言:** [SaveButtonAutoEnable](#) を **ブール値** として

**説明**

このプロパティが TRUE に設定されている場合、コントロールのツールバー内の保存ボタンの有効化済み/無効化済みの状態はドキュメントの [変更された](#) フラグに従って設定されます。

デフォルトの値は FALSE です。

#### 4.2.1.64 Authentic.SavePOST

次も参照してください！

**宣言:** [SavePOST](#)

**説明**

プロパティ [XMLDataSaveUrl](#) により指定される URL にドキュメントを保存します。信頼されていないバージョンに関しては、フルローカルパスを使用することができます。プラグインは現在表示されている XML ファイルを保存するために、HTTP POST リクエストをサーバーに送信します。

**ファイルが保存されるかチェックする**

Authentic プラグインが 300 の HTTP レスポンスを受信すると、ファイルが保存されていないとみなされ（デフォルトでは）HTTP エラーレスポンスを含む最初のメッセージボックスがポップアップされます。（抑制することのできる）2 番目のメッセージボックスはファイルが保存されていないことを警告します。正確な HTTP レスポンスが保存の試みの成功と失敗に従って保証されるかはアプリケーションの開発者によります。PHP コードのサンプルは以下のようになります：

```
<?php
// suppress error messages to prevent any output
// being generated before headers can be sent
error_reporting (0);
$error = false;
$handle = fopen ( "result.xml", "w+" );
if (! $handle)
    $error = true;
else
{
    if (! fwrite($handle, $HTTP_RAW_POST_DATA))
        $error = true;
    else
        fclose($handle);
}
if ($error)
    header( "HTTP/1.1 500 Server Error" );
?>
```

#### 4.2.1.65 Authentic.SaveXML

次も参照してください！

宣言: [SaveXML](#) を文字列として

戻り値

XML 構造 を文字列として

説明

現在のXML データを呼び出し元へ返される文字列に保存します。

#### 4.2.1.66 Authentic.SchemaLoadObject

次も参照してください！

宣言: [SchemaLoadObject](#) を [AuthenticLoadObject](#) として

説明

SchemaLoadObjectには、現在のXML ファイルのためのXML スキーマドキュメントへの参照が含まれています。スキーマドキュメントは通常 XMLSpy を使用して生成されます。

サンプル

```
objPlugIn.SchemaLoadObject.URL = "http://www.YOURSERVER.com/OrgChart.xsd"  
objPlugIn.XMLDataLoadObject.URL = "http://www.YOURSERVER.com/OrgChart.xml"  
objPlugIn.DesignDataLoadObject.URL = "http://www.YOURSERVER.com/OrgChart.sps"  
objPlugIn.StartEditing
```

上のコードはロードオブジェクトの全てのURL プロパティを設定し、[StartEditing](#) を呼び出してファイルをロードし表示します。信頼されていないバージョンに関しては、フルローカルパスを使用することができます。現在のコンテンツプラグインの状態はクリアされます。

#### 4.2.1.67 Authentic.SelectionChanged

次も参照してください！

宣言: [SelectionChanged](#) を [VT\\_0019](#) として

説明

ユーザーが現在の選択を変更すると、このイベントは挙げられます。

次も参照してください！ [接続ポイントイベント](#)。

#### 4.2.1.68 Authentic.SelectionMoveTabOrder

次も参照してください！

**宣言:** SelectionMoveTabOrder(*bForward* をブール値として、*bTag* をブール値として)

##### 説明

SelectionMoveTabOrder() は現在の選択を前、または 後に移動します。

*bTag* がfalse の場合、テーブル新規のラインの最後のセルに現在の選択が追加されます。

#### 4.2.1.69 Authentic.SelectionSet

次も参照してください！

**宣言:** SelectionSet(*pStartElement* をXMLDataとして、*nStartPos* を long として、*pEndElement* をXMLDataとして、*nEndPos* を long として) をブール値として

##### 説明

Authentic View 内の新規の選択を設定するためにSelectionSet() を使用します。選択が1つの(*pStartElement*) XML 要素である場合、*pEndElement* をNULL (なし) に設定することができます。

#### 4.2.1.70 Authentic.SetUnmodified

**宣言:** SetUnmodified()

##### 説明

このメソッドを呼び出した後、基になるXMLドキュメントのクリーンな状態として元に戻す/やり直し バッファの現在の条件が取られ、[変更された](#) フラグはFALSE に設定されます。

#### 4.2.1.71 Authentic.StartEditing

次も参照してください！

**宣言:** StartEditing をブール値として

##### 戻り値

全てのファイルがロードされ表示される場合、True が返されます。

##### 説明

現在のドキュメントの編集を開始します。ロードオブジェクト [SchemaLoadObject](#)、[DesignDataLoadObject](#) と [XMLDataLoadObject](#) のプロパティを最初に設定することが重要です。

#### 4.2.1.72 Authentic.StartSpellChecking

**宣言:** `StartSpellChecking()`

**説明**

パッケージ「スペルチェックエンジン」が含まれる場合、および、必要とされる辞書が使用可能でアクティブ化されている場合、このコマンドはスペルチェックダイアログを開きます。

#### 4.2.1.73 Authentic.TextStateBmpURL

**宣言:** `TextStateBmpURL` を **文字列** として

**説明**

テキスト状態アイコンのオプティミズドビットマップからの URL が取得されます。URL が指定されていない場合、テキスト状態ボタンアイコンは表示されません。

**サンプル**

```
objPlugIn.TextStateBmpURL = "<http://plugin.xmlspy.com/textstates/>"
```

```
<PARAM NAME="TextStateBmpURL" VALUE="http://plugin.xmlspy.com/textstates/">
```

#### 4.2.1.74 Authentic.TextStateToolBarLine

**宣言:** `TextStateToolBarLine` を **long** として

**説明**

テキスト状態アイコンが置かれるツールバー(行番号)です。テキスト状態アイコンは既存のツールバーに追加、または、新規のツールバーに配置することができます。

デフォルトの値 1

#### 4.2.1.75 Authentic.ToolbarRows

**宣言:** `ToolbarRows` を `AuthenticToolbarRows` として

**説明**

表示される全てのツールバーのコレクションを取得します。ツールバーの削除、追加、または変更の方法に関しては、[AuthenticToolbarRows](#) の説明を参照してください。

#### 4.2.1.76 Authentic.ToolbarsEnabled

**宣言:** `ToolbarsEnabled` を **ブール値** として

**説明**

Authentic Browser ツールバーが有効化されている場合、true を返します。  
このプロパティは全てのツールバーを有効化、または、無効化するために使用することができます。  
デフォルトの値: True

#### 4.2.1.77 Authentic.ToolbarTooltipsEnabled

**宣言:** `ToolbarTooltipsEnabled` を **ブール値** として

**説明**

Authentic Browser ツールバーのためのヒントが有効化されている場合、true です。  
デフォルトの値: True

#### 4.2.1.78 Authentic.UICommands

**宣言:** `UICommands` を `AuthenticCommands` として

**説明**

使用することのできる全てのツールバーコマンドのコレクションを (詳細と共に) 取得します。  
これらのコマンドのために、ボタンを異なるツールバーに配置することができます。  
読み取り専用です。

**サンプル**

使用することのできる全てのコマンド、コマンドグループ、詳細、を取得しメッセージボックス内に表示します。

```
dim str
for each UICommand in objPlugin.UICommands
str = str & UICommand.CommandID & " | " & UICommand.Group & " | " &
UICommand.ShortDescription & chr(13)
next
msgbox str
```

#### 4.2.1.79 Authentic.ValidateDocument

次も参照してください:

**宣言:** `ValidateDocument(showResults` を **ブール値** として) を **ブール値** として

**戻り値**

検証の結果

**説明**

XML スキーマデータに対する正確性のために現在のXML データを検証します。パラメーター `showResults` が `FALSE` の場合、検証エラーは抑制されます。それ以外の場合、検証エラーが表示されます。

#### 4.2.1.80 Authentic.validationBadData

次も参照してください！

**宣言:** `validationBadData` を [XMLData](#) として

##### 説明

このプロパティは最後の検証エラーに関する追加の状況を提供します。ValidateDocument() への呼び出しの後に設定され、NULL、または、エラーを引き起こす XML 要素への参照を持ちます。

#### 4.2.1.81 Authentic.validationMessage

次も参照してください！

**宣言:** `validationMessage` を [文字列](#) として

##### 説明

検証が失敗すると、(ValidateDocument への呼び出しの後) このプロパティはエラーメッセージと共に文字列を保管します。

#### 4.2.1.82 Authentic.XMLDataLoadObject

次も参照してください！

**宣言:** `XMLDataLoadObject` を [AuthenticLoadObject](#) として

##### 説明

XMLDataLoadObject は編集中の XML ドキュメントへの参照を含んでいます。XML ドキュメントは通常 XMLSpy を使用して定義されますが、データベース または他のビジネスプロセスを使用して生成されます。

次も参照してください！ [SchemaLoadObject](#)。

#### 4.2.1.83 Authentic.XMLDataSaveUrl

次も参照してください！

**宣言:** `XMLDataSaveUrl` を [文字列](#) として

##### 説明

XML データが変更されると、URL を使用してサーバーにデータを戻し保存することができます。Authentic.Save メソッドを使用して HTTP PUT を介して XML データを保存する場合、このプロパティは XML データが保存される場所を定義します。Authentic.SavePOST メソッドを使用して HTTP POST を介してデータをポストする場合、このプロパティは POST データを処理するサーバー側のスクリプト/アプリケーションの場所を定義します。信頼されていないバージョンに関しては、フルローカルパスを使用することができます。

次も参照してください！ [Authentic.Save](#) と [Authentic.SavePOST](#) メソッド。

#### 4.2.1.84 Authentic.XMLRoot

次も参照してください！

**宣言:** [XMLRoot](#) を [XMLData](#) として

##### 説明

XMLRoot は現在表示されているXML 構造の親要素です。 [XMLData](#) インターフェイスを使用するとファイルの全部のエレメントにアクセスすることができます。

詳細に関しては、次も参照してください！ [XMLData の使用](#)。

#### 4.2.1.85 Authentic.XMLTable

**宣言:** [XMLTable](#) を [AuthenticXMLTableCommands](#) として

##### 説明

すべてのXML テーブルコマンドのセットを取得します。  
読み取り専用です。

### 4.2.2 AuthenticCommand

メソッド

プロパティ

[CommandID](#)

[Group](#)

[ShortDescription](#)

[Name](#)

#### 4.2.2.1 AuthenticCommand.CommandID

**宣言:** [CommandID](#) を [SPYAuthenticCommand](#) として

##### 説明

コマンドのCommandId を取得します。  
可能な値: [AuthenticToolBarButton](#) を参照してください。  
読み取り専用

#### サンプル

[Authentic.UICommands](#) でのサンプルを参照してください。

### 4.2.2.2 AuthenticCommand.Group

**宣言:** Group を [SPYAuthenticCommandGroup](#) として

#### 説明

コマンドが所属するCommandGroup です。  
読み取り専用

#### サンプル

[Authentic.UICommands](#) でのサンプルを参照してください。

### 4.2.2.3 AuthenticCommand.ShortDescription

**宣言:** ShortDescription を **文字列** として

#### 説明

コマンド (例、ヒト テキスト) などの短い説明。  
読み取り専用

#### サンプル

[Authentic.UICommands](#) でのサンプルを参照してください。

### 4.2.2.4 AuthenticCommand.Name

**宣言:** Name を **文字列** として

#### 説明

## 4.2.3 AuthenticCommands

#### メソッド

[Item](#)

#### プロパティ

[Count](#)

### 4.2.3.1 AuthenticCommands.Count

**宣言:** Count を long として

**説明**

使用することのできる UI コマンドの数。  
読み取り専用

### 4.2.3.2 AuthenticCommands.Item

**宣言:** Item (nPosition を long として) を AuthenticCommand として

**説明**

位置 nPosition のコマンドを取得します。nPosition は1 から開始します。

## 4.2.4 AuthenticContextMenu

コンテキストメニューインターフェイスはユーザーに Authentic 内に表示されるコンテキストメニューをカスタマイズする方法を提供します。インターフェイスはこのセクション内にリストされるメソッドが存在します。

### 4.2.4.1 CountItems

**メソッド:** CountItems ()

**戻り値**

メニューアイテムの番号を返します。

**エラー**

2501 無効なオブジェクトです。

### 4.2.4.2 DeleteItem

**メソッド:** DeleteItem(position を integer として)

**戻り値**

既存のメニューアイテムを削除します。

**エラー**

2501 無効なオブジェクト  
2502 無効なインデックス

### 4.2.4.3 GetItemText

**メソッド:** `GetItemText`(`position` を `integer` として `メニューアイテム名`を文字列として)

**戻り値**

メニューアイテムの名前を取得します。

**エラー**

- 2501 無効なオブジェクト
- 2502 無効なインデックス

### 4.2.4.4 InsertItem

**メソッド:** `InsertItem`(`position` を `integer` として `menu item name` を文字列として `macro name` を文字列として)

**戻り値**

ユーザー定義メニューアイテムを挿入します。メニューアイテムはマクロを開始し、有効なマクロ名が与えられる必要があります。

**エラー**

- 2501 無効なオブジェクト
- 2502 無効なインデックス
- 2503 存在しないマクロ
- 2504 内部エラー

### 4.2.4.5 SetItemText

**メソッド:** `SetItemText`(`position` を `integer` として `menu item name` を文字列として)

**戻り値**

メニューアイテムの名前を設定します。

**エラー**

- 2501 無効なオブジェクト
- 2502 無効なインデックス

## 4.2.5 AuthenticDataTransfer

次も参照してください:

**メソッド**

[getData](#)

**プロパティ**

[dropEffect](#)

## [ownDrag](#) [type](#)

### 説明

AuthenticDataTransfer インターフェイス

#### 4.2.5.1 AuthenticDataTransfer.dropEffect

次も参照してください！

宣言: [dropEffect](#) を [long](#) として

### 説明

デフォルトのイベントハンドラーからのドロップ効果を保管します。この値を変更し、[AuthenticEvent.cancelBubble](#) を TRUE に設定すると、ドロップ効果を設定することができます。

#### 4.2.5.2 AuthenticDataTransfer.getData

次も参照してください！

宣言: [getData](#) を [Variant](#) として

### 説明

[getData](#) は、この dataTransfer オブジェクトに関連する実際のデータを取得します。詳細に関しては、次も参照してください！  
[AuthenticDataTransfer.type](#)

#### 4.2.5.3 AuthenticDataTransfer.ownDrag

次も参照してください！

宣言: [ownDrag](#) を [ブール値](#) として

### 説明

Authentic View の内部から現在のドラッグソースが来る場合、の場合、プロパティは True です。

#### 4.2.5.4 AuthenticDataTransfer.type

次も参照してください！

宣言: [type](#) を [文字列](#) として

### 説明

[AuthenticDataTransfer.getData](#) メソッドを使用して取得するデータの型を持ちます。

現在サポートされているデータ型は、以下のとおりです:

OWN	プラグイン自身からのデータ
TEXT	プレーンテキスト
UNICODETEXT	UNICODE としてのプレーンテキスト
IUNKNOWN	IDataObject インスタンスを参照するIUnknown

## 4.2.6 AuthenticEvent

次も参照してください:

プロパティ

[altKey](#)

[altLeft](#)

[ctrlKey](#)

[ctrlLeft](#)

[shiftKey](#)

[shiftLeft](#)

[keyCode](#)

[repeat](#)

[button](#)

[clientX](#)

[clientY](#)

[dataTransfer](#)

[srcElement](#)

[fromElement](#)

[propertyName](#)

[cancelBubble](#)

[returnValue](#)

[type](#)

説明

AuthenticEvent インターフェイス

#### 4.2.6.1 AuthenticEvent.altKey

次も参照してください！

宣言: altKey を **ブール値** として

##### 説明

右側のALT キーが押されると true です。

#### 4.2.6.2 AuthenticEvent.altLeft

次も参照してください！

宣言: altLeft を **ブール値** として

##### 説明

左側のALT キーが押されると true です。

#### 4.2.6.3 AuthenticEvent.button

次も参照してください！

宣言: button を **long** として

##### 説明

マウスボタンが押されるかを指定します:

- 0 ボタンは押されません。
- 1 左側のボタンが押されます。
- 2 右側のボタンが押されます。
- 3 左右のボタンが押されます。
- 4 中央のボタンが押されます。
- 5 左側と中央のボタンが押されます。
- 6 右側と中央のボタンが押されます。
- 7 3つ全てのボタンが押されます。

onbuttondown と onbuttonup イベントはボタンの値を異なる方法で設定します。onbuttonup イベントは押されたボタンに関わらずリリースイベントのためのボタンのため値を設定します。

#### 4.2.6.4 AuthenticEvent.cancelBubble

次も参照してください！

宣言: cancelBubble を **ブール値** として

**説明**

デフォルトのイベントハンドラーが呼び出されない場合、cancelBubble を True に設定します。

#### 4.2.6.5 AuthenticEvent.clientX

次も参照してください:

**宣言:** clientX を long として

**説明**

クライアント座標内の現在のマウス位置の X 値

#### 4.2.6.6 AuthenticEvent.clientY

次も参照してください:

**宣言:** clientY を long として

**説明**

クライアント座標内の現在のマウス位置の Y 値

#### 4.2.6.7 AuthenticEvent.ctrlKey

次も参照してください:

**宣言:** ctrlKey を ブール値 として

**説明**

右側の CTRL キーが押されると true です。

#### 4.2.6.8 AuthenticEvent.ctrlLeft

次も参照してください:

**宣言:** ctrlLeft を ブール値 として

**説明**

左側の CTRL キーが押されると true です。

#### 4.2.6.9 AuthenticEvent.dataTransfer

次も参照してください！

宣言: `dataTransfer` を `Variant` として

##### 説明

プロパティ `dataTransfer`

#### 4.2.6.10 AuthenticEvent.fromElement

次も参照してください！

宣言: `fromElement` を `Variant` として

##### 説明

このプロパティを設定するイベントは存在しません。

#### 4.2.6.11 AuthenticEvent.keyCode

次も参照してください！

宣言: `keyCode` を `long` として

##### 説明

現在押されているキーのキーコードです。

プロパティは読み取りおよび書き込むことができます。

#### 4.2.6.12 AuthenticEvent.propertyName

次も参照してください！

宣言: `propertyName` を `文字列` として

##### 説明

このプロパティを設定するイベントは存在しません。

#### 4.2.6.13 AuthenticEvent.repeat

次も参照してください！

宣言: `repeat` を **ブール値** として

**説明**

`onkeydown` イベントが繰り返されると True です。

#### 4.2.6.14 `AuthenticEvent.returnValue`

次も参照してください！

宣言: `returnValue` を **Variant** として

**説明**

イベントハンドラーのために戻り値を設定するために `returnValue` を使用します。

#### 4.2.6.15 `AuthenticEvent.shiftKey`

次も参照してください！

宣言: `shiftKey` を **ブール値** として

**説明**

右側の SHIFT キーが押されると true です。

#### 4.2.6.16 `AuthenticEvent.shiftLeft`

次も参照してください！

宣言: `shiftLeft` を **ブール値** として

**説明**

左側の SHIFT キーが押されると true です。

#### 4.2.6.17 `AuthenticEvent.srcElement`

次も参照してください！

宣言: `srcElement` を **Variant** として

**説明**

現在のイベントを実行する要素です。  
これは通常 [XMLData](#) オブジェクトです。

バージョン 3.0.0.0 から

ondoceditcommand イベントから設定されている場合、プロパティは [AuthenticCommand](#) オブジェクトへの参照を持つことができます。

#### 4.2.6.18 AuthenticEvent.type

次も参照してください:

**宣言:** type を文字列として

##### 説明

このプロパティを設定するイベントは存在しません。

### 4.2.7 AuthenticEventContext

EventContext インターフェイスはマクロが実行されるコンテキストの多くのプロパティへのアクセスを提供します。

#### 4.2.7.1 EvaluateXPath

**メソッド:** EvaluateXPath (string 式) を文字列として

##### 戻り値

メソッドはノードのコンテキスト内のイベントがトリガーされ文字列が返される XPath 式を評価します。

##### 説明

EvaluateXPath() 指定されているイベントコンテキストを持つ XPath 式を実行します。結果は文字列として返されます。シーケンスの場合は、スペース文字で区切られた文字列が返されます。

##### エラー

- 2201 無効なオブジェクトです。
- 2202 コンテキストがありません。
- 2209 無効な引数です。
- 2210 内部エラーです。
- 2211 XPath エラーです。

#### 4.2.7.2 GetEventContextType

**メソッド:** GetEventContextType () を EventContextType 列挙として

##### 戻り値

コンテキストノード型を返します。

**説明**

GetEventContextType は、ユーザーマクロがXML ノード内に存在するか、または XPath 動的なアイテムのコンテキスト内に存在するかを決定することを許可します。列挙 AuthenticEventContextType は以下のように定義されています:

```
authenticEventContextXML,
authenticEventContextAtomicItem,
authenticEventContextOther
```

コンテキストが通常のXML ノードの場合、GetXMLNode() 関数はアクセスを与えます(それ以外の場合はNULL を返します)。

**エラー**

- 2201 無効なオブジェクトです。
- 2202 コンテキストがありません。
- 2209 無効なラメーターです。

### 4.2.7.3 GetNormalizedTextValue

**メソッド:** GetNormalizedTextValue() value を文字列として

**戻り値**

現在のノードの値を文字列として返します。

**エラー**

- 2201 無効なオブジェクトです。
- 2202 コンテキストがありません。
- 2203 無効なコンテキストです。
- 2209 無効なラメーターです。

### 4.2.7.4 GetVariableValue

**メソッド:** GetVariableValue(name を文字列として, value を文字列として)

**戻り値**

名前付き変数の値を取得します。

**説明**

GetVariableValue コンテキストの範囲内の変数の値を取得します。

```
nZoom = parseInt( AuthenticView.EventContext.GetVariableValue( 'Zoom' ) );
if ( nZoom > 1 )
{
    AuthenticView.EventContext.SetVariableValue( 'Zoom', nZoom - 1 );
}
```

**エラー**

- 2201 無効なオブジェクトです。
- 2202 コンテキストがありません。
- 2204 範囲内にそのような変数はありません。
- 2205 変数を評価することできません。
- 2206 変数はシーケンスを返します

2209 無効なラメーター

#### 4.2.7.5 GetXMLNode

メソッド: `GetXMLNode()` [XMLData](#) オブジェクト

##### 戻り値

コンテキスト XML ノードまたは NULL が返されます。

##### エラー

- 2201 無効なオブジェクトです。
- 2202 コンテキストがありません。
- 2203 無効なコンテキストです。
- 2209 無効なラメーターです。

#### 4.2.7.6 IsAvailable

メソッド: `IsAvailable()`

##### 戻り値

`EventContext` が設定されている場合、true を返します。

##### エラー

- 2201 無効なオブジェクトです。

#### 4.2.7.7 SetVariableValue

メソッド: `SetVariableValue(name` を文字列として、`value` を文字列として)

##### 戻り値

名前付き変数の値を設定します。

##### 説明

`SetVariableValue` はコンテキストの範囲内の変数の値を設定します。

```
nZoom = parseInt( AuthenticView.EventContext.GetVariableValue( 'Zoom' ) );
if ( nZoom > 1 )
{
    AuthenticView.EventContext.SetVariableValue( 'Zoom', nZoom - 1 );
}
```

##### エラー

- 2201 無効なオブジェクトです。
- 2202 コンテキストがありません。
- 2204 範囲内にそのような変数はありません。
- 2205 変数を評価することはできません。
- 2206 変数バナーケスに戻します
- 2207 読み取り専用の変数です。

2208 変更は許可されていません。

## 4.2.8 AuthenticLoadObject

次も参照してください！

プロパティ

[String](#)

[URL](#)

説明

XMLSpyXMLLoadSave オブジェクトはロードする必要のあるファイルのためにソースを設定するために使用されます。文字列 プロパティを使用して直接、または URL プロパティを使用して外部の場所としてコンテンツを設定することができます。

使用方法に関する詳細については、次を参照してください！ [Authentic.SchemaLoadObject](#)、[Authentic.DesignDataLoadObject](#) と [Authentic.XMLDataLoadObject](#)

### 4.2.8.1 AuthenticLoadObject.String

次も参照してください！

宣言: [String](#) を [文字列](#) として

説明

XML 構造を文字列から設定するためこのプロパティを使用することができます。このプロパティを使用する場合、オブジェクトの URL プロパティは空である必要があります。

### 4.2.8.2 AuthenticLoadObject.URL

次も参照してください！

宣言: [URL](#) を [文字列](#) として

説明

プロパティはロードまたは保存オペレーションのための有効な URL を含んでいる必要があります。現在サポートされている HTTP プロトコールは http、https、ftp および gopher です。

## 4.2.9 AuthenticRange

次も参照してください！

ドキュメントをナビゲートし、固有の箇所を選択するために使用される AuthenticRange のプロパティとメンバは最初のテーブルにリストされています。

## プロパティ

[Application](#)  
[FirstTextPosition](#)  
[FirstXMLData](#)  
[FirstXMLDataOffset](#)  
[LastTextPosition](#)  
[LastXMLData](#)  
[LastXMLDataOffset](#)  
[Parent](#)

## メソッド

[Clone](#)  
[CollapsToBegin](#)  
[CollapsToEnd](#)  
[ExpandTo](#)  
[Goto](#)  
[GotoNext](#)  
[GotoPrevious](#)  
[IsEmpty](#)  
[IsEqual](#)  
[MoveBegin](#)  
[MoveEnd](#)  
[NextCursorPosition](#)  
[PreviousCursorPosition](#)  
[Select](#)  
[SelectNext](#)  
[SelectPrevious](#)  
[SetFromRange](#)

次のテーブルは多くが右/左 マウスメニューで見つけることができるコンテンツの変更メソッドをリストしています。

## プロパティ

[Text](#)

## 編集の操作

[Copy](#)  
[Cut](#)  
[Delete](#)  
[Paste](#)

## 動的なテーブルの操作

[AppendRow](#)  
[DeleteRow](#)  
[DuplicateRow](#)  
[InsertRow](#)  
[IsInDynamicTable](#)  
[MoveRowDown](#)  
[MoveRowUp](#)

次のメソッドは範囲オブジェクトの入力ヘルパーウィンドウのための機能を提供します。

## 入力ヘルパーウィンドウのオペレーション

## 要素

[CanPerformActionWith](#)  
[CanPerformAction](#)  
[PerformAction](#)

## 属性

[GetElementAttributeValue](#)  
[GetElementAttributeNames](#)  
[GetElementHierarchy](#)  
[HasElementAttribute](#)  
[SetElementAttributeValue](#)

## エンティティ

[GetEntityNames](#)  
[InsertEntity](#)

## 説明

AuthenticRange オブジェクトは自動インターフェイスのカーソルの選択です。AuthenticView 内のカーソルの位置をポイントするために、またはドキュメントの箇所を選択するために使用することができます。このセクションで AuthenticRange オブジェクトのために使用することのできるオペレーションは、ユーザーインターフェイスの対応するオペレーションの現在のユーザーインターフェイスによる選択と同様に作動します。任意の数の AuthenticRange オブジェクトを同時に使用できますが、ユーザーインターフェイスではカーソルの選択する1つのみを選択することができます。

最初の範囲オブジェクトが [AuthenticView.Selection](#) を使用する場合は、ユーザーインターフェイス内で現在のカーソルの選択を使用して対応する範囲を取得します。または、簡易の範囲は読み取り専用プロパティ [AuthenticView.DocumentBegin](#)、[AuthenticView.DocumentEnd](#)、と [AuthenticView.WholeDocument](#) を使用してアクセスすることができます。呼び出し内の特定のドキュメントへのナビゲーションを許可する [AuthenticView.Goto](#) が最も柔軟性を持つメソッドです。更に複雑な選択のために、上記をこのページの最初のテーブルにリストされる範囲オブジェクト上の多種のナビゲーションメソッドと組み合わせます。

ドキュメントの箇所を選択する他のメソッドは、範囲オブジェクトのプロパティを使用します。2つのポジションのシステムを使用することができます。任意で結合することができます。

- 絶対的なドキュメントの位置 0 から開始するテキストカーソルの位置を設定し、範囲の開始位置と終了位置のために取得することができます。詳細に関しては [FirstTextPosition](#) と [LastTextPosition](#) を参照してください。このメソッドは複雑な内部計算を必要とし、使用する際はご注意ください。

- この要素内のXMLData 要素とテキストの位置を設定し、範囲の開始位置と終了位置のために取得することができます。詳細に関しては、[FirstXMLData](#)、[FirstXMLDataOffset](#)、[LastXMLData](#)、と[LastXMLDataOffset](#)を参照してください。このメソッドはとても効率的ですが、元になるドキュメント構造の知識を必要とします。ユーザーインターフェイスを介してはアクセスすることのできないXMLData オブジェクトをロケート、および、オペレーションを行うために使用することができます。

ドキュメントコンテンツへの変更は異なるメソッドによりおこなうことができます:

- [Text](#) プロパティにより範囲オブジェクトにより選択されたドキュメントテキストを取得することができます。範囲オブジェクト。設定されると、選択されたドキュメント テキストは新規のテキストと置き換えられます。
- 標準のドキュメントは機能切り取り、コピー、貼り付けと削除を編集します。
- 動的に増やすことのできるテーブルのためのテーブルオペレーション。
- Authentic 入力ヘルパーウィンドウの機能をマップするメソッド。
- 基になるドキュメントのXMLData オブジェクトを直接変更するためのアクセス。

#### 4.2.9.1 AuthenticRange.AppendRow

次も参照してください:

メソッド: [AppendRow](#) () を [ブール値](#) として

##### 説明

範囲の始めが動的なテーブル内に存在する場合、このメソッドは、選択されたテーブルの終わりに新規の行を挿入します。範囲の選択は新規の行の始まりを指すよう変更されます。操作に成功した場合、関数は True を返します。それ以外の場合は False が返されます。

##### エラー

- 2001 Authentic 範囲オブジェクト、または、その関連したビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

##### サンプル

```

'-----
'                                     VBScript
' Append row at end of current dynamically growable table
'-----
Dim objRange
Set objRange = objPlugin.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

```

#### 4.2.9.2 AuthenticRange.Application

次も参照してください:

プロパティ: [Application](#) を [Authentic](#) として(読み取り専用)

##### 説明

アプリケーションオブジェクトにアクセスします。

#### エラー

- 2001 Authentic 範囲オブジェクト、または、その関連したビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

### 4.2.9.3 AuthenticRange.CanPerformAction

次も参照してください！

**メソッド:** `CanPerformAction` (`eAction` を `SPYAuthenticActions` として、`strElementName` を文字列として) を `ブール値` として

#### 説明

`CanPerformAction` と関連するメソッドは Authentic の入力ヘルパーへのアクセスを有効化します。これらの機能は、変更の正確な箇所を知ることなく、ドキュメントコンテンツの簡単かつ整合性のある変更を許可します。範囲の始めオブジェクトは指定されたアクションが実行される次の有効な位置をロケートするために使用されます。場所を検索できる場合、メソッドは True を返します、それ以外の場合、False が返されます。

ヒント: 特定のアクションのために有効な要素名を検索するには、[CanPerformActionWith](#) を使用してください。

#### エラー

- 2001 Authentic 範囲オブジェクト、または、その関連したビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。
- 2007 無効なアクションが指定されています。

#### サンプル

次を参照してください [PerformAction](#)。

### 4.2.9.4 AuthenticRange.CanPerformActionWith

次も参照してください！

**メソッド:** `CanPerformActionWith` (`eAction` を `SPYAuthenticActions` として、`out_arrElementNames` を `Variant` として)

#### 説明

`CanPerformActionWith` と関連するメソッドは、Authentic の入力ヘルパー機能へのアクセスを提供します。これらの機能は、変更の正確な箇所を知ることなく、ドキュメントコンテンツの簡単かつ整合性のある変更を許可します。

このメソッドは指定されたアクションを実行することのできる要素名の配列を返します。

ヒント: アクションを適用するには [PerformAction](#) を使用します。

#### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。
- 2007 無効なアクションが指定されています。

### サンプル

次を参照してください! [PerformAction](#)。

#### 4.2.9.5 AuthenticRange.Clone

次も参照してください!

メソッド: [Clone \(\)](#) を [AuthenticRange](#) として

##### 説明

範囲オブジェクトのコピーを返します。

##### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.9.6 AuthenticRange.CollapsToBegin

次も参照してください!

メソッド: [CollapsToBegin \(\)](#) を [AuthenticRange](#) として

##### 説明

範囲オブジェクトの終わりが終わりを指すように設定します。メソッドは変更された範囲オブジェクトを返します。

##### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.9.7 AuthenticRange.CollapsToEnd

次も参照してください!

メソッド: [CollapsToEnd \(\)](#) を [AuthenticRange](#) として

##### 説明

範囲オブジェクトの始めが終わりを指すように設定します。メソッドは変更された範囲オブジェクトを返します。

##### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.9.8 AuthenticRange.Copy

次も参照してください！

メソッド: `Copy ()` を **ブール値** として

##### 説明

コピーされるドキュメントの一部が範囲に含まれない場合、`False` が返されます。  
テキストの後に `True` を返します。全体が選択された XML 要素の場合、要素も、コピー貼り付けバッファにコピーされます。

##### エラー

- 2001 Authentic 範囲オブジェクト、または、その関連したビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.9.9 AuthenticRange.Cut

次も参照してください！

メソッド: `Cut ()` を **ブール値** として

##### 説明

範囲が削除されないドキュメントの部分を含む場合、`False` が返されます。  
テキストの後に `True` を返します。全体が選択された XML 要素の場合、要素も、ドキュメントから削除され、コピー貼り付けバッファに保存されます。

##### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.9.10 AuthenticRange.Delete

次も参照してください！

メソッド: `Delete ()` を **ブール値** として

##### 説明

範囲が削除されないドキュメントの部分を含む場合、`False` が返されます。  
テキストの後に `True` を返します。全体が選択された XML 要素の場合、要素も、ドキュメントから削除されます。

##### エラー

- 2001 Authentic 範囲オブジェクト、または、その関連したビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.9.11 AuthenticRange.DeleteRow

次も参照してください:

メソッド: `DeleteRow ()` を **ブール値** として

##### 説明

範囲の始めが動的なテーブル内の場合、このメソッドは選択された行を削除します。範囲の選択は削除された行の後の次の要素を指すように変更されます。削除の操作に成功した場合、関数は True を返します。それ以外の場合は False が返されます。

##### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

##### サンプル

```

'-----
'                               VBScript
' Delete selected row from dynamically growing table
'-----

Dim objRange
Set objRange = objPlugin.ActiveDocument.AuthenticView.Selection

' check if we are in a table
If objRange.IsInDynamicTable Then
    objRange.DeleteRow
End If

```

#### 4.2.9.12 AuthenticRange.DuplicateRow

次も参照してください:

メソッド: `DuplicateRow ()` を **ブール値** として

##### 説明

範囲の始めが動的なテーブル内の場合、このメソッドは選択された行の後に現在の行を複製します。範囲の選択は削除された行の後に次の要素を指すように変更されます。複製の操作に成功した場合、関数は True を返します。それ以外の場合は False が返されます。

##### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

##### サンプル

```

'-----
'                               VBScript
' duplicate row in current dynamically growable table
'-----

Dim objRange
Set objRange = objPlugin.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

### 4.2.9.13 AuthenticRange.EvaluateXPath

**メソッド:** `EvaluateXPath` (文字列式) を文字列として

#### 戻り値

メソッドは文字列を返します。

#### 説明

`EvaluateXPath()` は範囲の最初の選択であるコンテキストノードを持つXPath 式を実行します。結果は文字列として返されます。シーケンスの場合は、スペース文字で区切られた文字列が返されます。XML コンテキストノードに関連性がない場合、ユーザーは `AuthenticView.XMLDataRoot` のようなノードを提供することができます。

#### エラー

- 2001 無効なオブジェクト
- 2005 無効なパラメーター
- 2008 内部エラー
- 2202 不足しているコンテキストノード
- 2211 XPath エラー

### 4.2.9.14 AuthenticRange.ExpandTo

次も参照してください:

**メソッド:** `ExpandTo` (`eKind` を `SPYAuthenticElementKind` として) を `AuthenticRange` として

#### 説明

範囲の最初のカーソルの位置である、または、最初のカーソルの位置を含む型 `eKind` の要素全体を選択します。メソッドは変更された範囲オブジェクトを返します。

#### エラー

- 2001 `Authentic` 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2003 拡張の範囲がドキュメントの終わりの外部に存在します。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

### 4.2.9.15 AuthenticRange.FirstTextPosition

次も参照してください:

**プロパティ:** `FirstTextPosition` を `long` として

#### 説明

範囲オブジェクトの左端のテキストの位置インデックスを設定または取得します。このインデックスは常に `LastTextPosition` より小さい、または、等価です。インデックスがドキュメントの始まりの0 から開始し、テキストカーソルの占めることのできる異なる箇所と共に増えます。テキストの位置は1 ずつ増加し、カーソル右キーと同じ効果が起きます。テキストの位置は1 ずつ削減すると子ができ、カーソル左キーと同じ効果が起きます。

現在の [LastTextPosition](#) より大きな値に [FirstTextPosition](#) を設定した場合、[LastTextPosition](#) は新規の [FirstTextPosition](#) に設定されます。

ヒント: カーソルの位置をベースとしたXMLData に対して比較される複雑なオペレーションのため、気をつけてテキストカーソルの位置を使用してください。

#### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトが有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。
- 2006 ドキュメントの外部のテキストの位置が指定されています。

#### サンプル

```
'-----
'                               VBScript
'-----

Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = objPlugin.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 "Ooops!"
End If
```

### 4.2.9.16 AuthenticRange.FirstXMLData

次も参照してください:

プロパティ: [FirstXMLData](#) をXMLData として

#### 説明

範囲により一部、または、全体が選択されている基になるドキュメントの最初のXMLData 要素を設定、または、取得します。選択の正確な開始位置は [FirstXMLDataOffset](#) 属性により定義されます。

[FirstXMLData](#) to 新規のデータオブジェクトを設定すると、[FirstXMLDataOffset](#) が要素内の最初のカーソルの位置に設定されます。カーソルの位置を持つXMLData オブジェクトのみが使用されます。[FirstXMLData](#) / [FirstXMLDataOffset](#) 設定すると、[LastXMLData](#) / [LastXMLDataOffset](#) より大きな位置を選択します。後者の場合は、新規の開始位置に移動されます。

ヒント: [AuthenticRange](#) オブジェクトで使用することができるメソッドが十分ではない場合、基になるXMLドキュメントに直接アクセスし操作するために、[FirstXMLData](#) と [LastXMLData](#) プロパティを使用することができます。

#### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトが有効ではありません。

- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。
- 2008 内部エラー
- 2009 XMLData オブジェクトにアクセスすることができません。

### サンプル

```

'-----
'           VBScript
' show name of currently selected XMLData element
'-----

Dim objAuthenticView
Set objAuthenticView = objPlugin.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

```

#### 4.2.9.17 AuthenticRange.FirstXMLDataOffset

次も参照してください！

プロパティ [FirstXMLDataOffset](#) を long として

#### 説明

範囲の始めのために [FirstXMLData](#) 要素内にカーソルの位置を設定または取得します。オフセットの位置はテキストプロパティより返された文字をベースとし、0 と共に開始します。新規のオフセットを設定する際、-1 を使用して要素内の最後の可能なポジションにオフセットを設定します。次のシナリオの場合、それぞれ特別な点に注意する必要があります。

- エンボックス内のエントリのテキスト書式と CheckBoxes および類似するコントロールはスクリーン上に表示されるものと異なる場合があります。データオフセットはこのテキストをベースにしますが、データオフセットはこのテキストをベースにしますが、2つの有効なオフセットの位置のみが存在します。1つはエントリの始めにもう1つはエントリの終わりに存在します。オフセットをエントリ中に設定しようと試みると、オフセットが終了位置に設定されます。
- XML エンティティのテキスト書式は、スクリーン上の表記とは異なる場合があります。オフセットはこのテキスト書式をベースにします。

If [FirstXMLData](#) / [FirstXMLDataOffset](#) selects 位置 after 現在の [LastXMLData](#) / [LastXMLDataOffset](#), 後者は、新規の開始位置に移動されます。

#### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトが有効ではありません。
- 2005 無効なオフセットが指定されています。  
戻り値/パラメーターのために無効なアドレスが指定されています。

### サンプル

```

'-----
'           VBScript
' Select the complete text of an XMLData element
' using XMLData based selection and ExpandTo
'-----

Dim objAuthenticView
Set objAuthenticView = objPlugin.AuthenticView

' first we use 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

```

#### 4.2.9.18 AuthenticRange.GetElementAttributeNames

次も参照してください！

**メソッド:** `GetElementAttributeNames` (`strElementName` を文字列として、`out_arrAttributeNames` を `Variant` として)

##### 説明

指定された名前を持つ外側の要素のための全ての属性の名前を取得します。要素 / 属性のペアを使用して、メソッド [GetElementAttributeValue](#) と [SetElementAttributeValue](#) を持つ属性の値を設定または取得してください。

##### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 無効な要素名が指定されました。  
戻り値 / パラメーターのために無効なアドレスが指定されています。

##### サンプル

次を参照してください！ [SetElementAttributeValue](#)。

#### 4.2.9.19 AuthenticRange.GetElementAttributeValue

次も参照してください！

**メソッド:** `GetElementAttributeValue` (`strElementName` を文字列として、`strAttributeName` を文字列として、`strValue` を文字列として)

##### 説明

`strElementName` により識別される要素のために `strAttributeName` 内で指定される属性の値を取得します。属性がサポートされ値が割り当てられていない場合、空の文字列が返されます。要素によりサポートされる属性の名前を検索するには、[GetElementAttributeNames](#)、または [HasElementAttribute](#) を使用してください。

##### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。

- 2005 無効な要素名が指定されました。  
無効な属性名が指定されました。  
戻り値/パラメーターのために無効なアドレスが指定されています。

#### サンプル

次を参照してください! [SetElementAttributeValue](#)。

### 4.2.9.20 AuthenticRange.GetElementHierarchy

次も参照してください!

メソッド: [GetElementHierarchy](#) ([out\\_arrElementNames](#) を [Variant](#) として)

#### 説明

現在の選択の親であるすべてのXML 要素の名前を取得します。外部の要素の前にも内部の要素はリストされます。現在の選択が単一のXMLData 要素でない場合、空のリストが返されます。

ドキュメント内のXMLData 要素を一意に識別する範囲オブジェクトを持つ要素の階層の名前です。これらの要素の属性は [GetElementAttributeNames](#) および関連したメソッドにより直接アクセスすることができます。

#### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。  
2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### サンプル

次を参照してください! [SetElementAttributeValue](#)。

### 4.2.9.21 AuthenticRange.GetEntityNames

次も参照してください!

メソッド: [GetEntityNames](#) ([out\\_arrEntityNames](#) を [Variant](#) として)

#### 説明

定義された全てのエンティティの名前を取得します。取得済みのエンティティのリストは現在の選択、または、ロケーションから独立しています。 [InsertEntity](#) 関数を持つ名前の一つを使用します。

#### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。  
2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### サンプル

次を参照してください! [InsertEntity](#)。

#### 4.2.9.22 AuthenticRange.GetVariableValue

メソッド: `GetVariableValue`(`name` を文字列として, `value` を文字列として)

##### 戻り値

名前付き変数の値を取得します。

##### エラー

- 2001 無効なオブジェクトです。
- 2005 無効なノリマーター
- 2202 コンテキストがありません。
- 2204 範囲内にそのような変数はありません。
- 2205 変数を評価することできません。
- 2206 変数はシーケンスを返します

#### 4.2.9.23 AuthenticRange.Goto

次も参照してください！

メソッド: `Goto` (`eKind` を `SPYAuthenticElementKind` として, `nCount` を `long` として, `eFrom` を `SPYAuthenticDocumentPosition` として) を `AuthenticRange` として

##### 説明

型 `eKind` の `nCount` 要素の開始位置をポイントするよう範囲を設定します。開始の位置はノリマーター `eFrom` により定義されています。

`nCount` のための正の値を使用してドキュメントの終了位置をナビゲートします。ドキュメントの始めにナビゲートするために負の値を使用します。メソッドは変更された範囲オブジェクトを返します。

##### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2003 ターゲットがドキュメントの終わりの後に存在します。
- 2004 ターゲットがドキュメントの始まりの前に存在します。
- 2005 無効な要素の種類が指定されています。  
無効な開始位置が指定されています。  
戻り値ノリマーターのために無効なアドレスが指定されています。

#### 4.2.9.24 AuthenticRange.GotoNext

次も参照してください！

メソッド: `GotoNext` (`eKind` を `SPYAuthenticElementKind` として) を `AuthenticRange` として

##### 説明

型 `eKind` の次の要素の開始位置に範囲を設定します。メソッドは変更された範囲オブジェクトを返します。

エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2003 ターゲットがドキュメントの終わりの後に存在します。
- 2005 無効な要素の種類が指定されています。  
戻り値/ラマーターのために無効なアドレスが指定されています。

サンプル

```

-----
'           VBScript
' Scan through the whole document word-by-word
-----

Dim objAuthenticView
Set objAuthenticView = objPlugin.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

```

## 4.2.9.25 AuthenticRange.GotoNextCursorPosition

次も参照してください:

メソッド: [GotoNextCursorPosition \(\)](#) を [AuthenticRange](#) として

## 説明

現在の終了位置の後に次のカーソルの位置に範囲を徹します。変更されたオブジェクトを返します。

エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2003 ターゲットがドキュメントの終わりの後に存在します。
- 2005 戻り値/ラマーターのために無効なアドレスが指定されています。

## 4.2.9.26 AuthenticRange.GotoPrevious

次も参照してください:

メソッド: [GotoPrevious \(eKind\)](#) を [SPYAuthenticElementKind](#) としてを [AuthenticRange](#) として

## 説明

現在範囲の開始位置の前の型 *eKind* の要素の開始位置の範囲を設定します。メソッドは変更された範囲オブジェクトを返します。

エラー

- 2001 Authentic 範囲オブジェクト、または 関連するビューオブジェクトは有効ではありません。
- 2004 ターゲットがドキュメントの始めの前に存在します。
- 2005 無効な要素の種類が指定されています。  
戻り値/ラマーターのために無効なアドレスが指定されています。

サンプル

```

-----
'               VBScript
' Scan through the whole document tag-by-tag
-----

Dim objAuthenticView
Set objAuthenticView = objPlugin.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentEnd
Dim bEndOfDocument
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

```

## 4.2.9.27 AuthenticRange.GotoPreviousCursorPosition

次も参照してください:

メソッド: [GotoPreviousCursorPosition \(\)](#) を [AuthenticRange](#) として

## 説明

カーソルの位置を現在のポジションのすぐ前に範囲を設定します。変更されたオブジェクトを返します。

エラー

- 2001 Authentic 範囲オブジェクト、または 関連するビューオブジェクトは有効ではありません。
- 2004 ターゲットがドキュメントの始まりの前に存在します。
- 2005 戻り値/ラマーターのために無効なアドレスが指定されています。

## 4.2.9.28 AuthenticRange.HasElementAttribute

次も参照してください:

メソッド: [HasElementAttribute](#) ([strElementName](#) を文字列として, [strAttributeName](#) を文字列として) をブール値として

## 説明

名前 [strElementName](#) を使用した外側要素が [strAttributeName](#) により指定されている属性をサポートするかテストします。

エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 無効な要素名が指定されました。  
戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.9.29 AuthenticRange.InsertEntity

次も参照してください！

メソッド: **InsertEntity** (*strEntityName* を文字列として)

説明

指定されたエンティティ範囲の選択を置き換えます。指定されたエンティティは、[GetEntityNames](#) により返されたエンティティ名の1つである必要ががあります。

エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 未知の入力名が指定されています。

サンプル

```

'-----
'                               VBScript
' Insert the first entity in the list of available entities
'-----
Dim objRange
Set objRange = objPlugin.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)
    objRange.Select()
Else
    MsgBox "Sorry, no entities are available for this document"
End If

```

#### 4.2.9.30 AuthenticRange.InsertRow

次も参照してください！

メソッド: **InsertRow** () をブール値として

説明

範囲の始めが動的なテーブル内の場合、このメソッドは現在の行の前へ新規の行を挿入します。範囲の選択は新規に挿入された行の始めを指すように変更されます。入力操作に成功した場合、関数は True を返します。それ以外の場合は False が返されます。

エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。

2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### サンプル

```

'-----
'                               VBScript
' Insert row at beginning of current dynamically growing table
'-----

Dim objRange
Set objRange = objPlugin.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

```

### 4.2.9.31 AuthenticRange.IsCopyEnabled

次も参照してください！

プロパティ: `IsCopyEnabled` をブール値として (読み取り専用)

#### 説明

コピー操作がこの範囲のためにサポートされているかをチェックします。

#### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

### 4.2.9.32 AuthenticRange.IsCutEnabled

次も参照してください！

プロパティ: `IsCutEnabled` をブール値として (読み取り専用)

#### 説明

切り取り操作がこの範囲のためにサポートされているかをチェックします。

#### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

### 4.2.9.33 AuthenticRange.IsDeleteEnabled

次も参照してください！

プロパティ: `IsDeleteEnabled` をブール値として (読み取り専用)

**説明**

削除操作がこの範囲のためにサポートされているかをチェックします。

**エラー**

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.9.34 AuthenticRange.IsEmpty

次も参照してください！

**メソッド:** `IsEmpty ()` を **ブール値** として

**説明**

範囲の最初と最後が同じかをテストします。

**エラー**

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.9.35 AuthenticRange.IsEqual

次も参照してください！

**メソッド:** `IsEqual (objCmpRange を AuthenticRange として)` を **ブール値** として

**説明**

両方の範囲の開始と終了が同じかをテストされます。

**エラー**

- 2001 比較されている2つの範囲オブジェクトの1つが無効です。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.9.36 AuthenticRange.IsFirstRow

次も参照してください！

**プロパティ:** `IsFirstRow ()` を **ブール値** として (読み取り専用)

**説明**

Test if 範囲 is in 最初のテーブルの行. どのテーブルが考慮されるかは範囲によります。選択が単一のテーブルの行を超える場合、このテーブルの埋め込みテーブル内の最初の要素であるかをチェックします。詳細に関してはユーザーマニュアルの入力ヘルパーを参照してください。

**エラー**

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

### 4.2.9.37 AuthenticRange.IsInDynamicTable

次も参照してください！

メソッド: `IsInDynamicTable ()` を **ブール値** として

#### 説明

異なる行の操作をサポートするテーブル内に範囲の始めが存在するかテストします。

#### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

### 4.2.9.38 AuthenticRange.IsLastRow

次も参照してください！

プロパティ: `IsLastRow ()` を **ブール値** として (読み取り専用)

#### 説明

最後のテーブルの行内に範囲が存在するかテストします。どのテーブルが考慮されるかは範囲によります。選択が単一のテーブルの行を超える場合、このテーブルが埋め込みテーブル内の最後の要素であるかをチェックします。詳細に関してはユーザーマニュアルの入力ヘルパーを参照してください。

#### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

### 4.2.9.39 AuthenticRange.IsPasteEnabled

次も参照してください！

プロパティ: `IsPasteEnabled` を **ブール値** として (読み取り専用)

#### 説明

貼り付け操作がこの範囲のためにサポートされているかをチェックします。

#### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

### 4.2.9.40 AuthenticRange.IsSelected

プロパティ: `IsSelected` を **ブール値** として

**説明**

選択が存在する場合、true() を返します。選択の範囲は空であることができます。例: カーソルが設定された場合のみ発生するなど。

#### 4.2.9.41 AuthenticRange.IsTextStateApplied

次も参照してください:

**メソッド:** IsTextStateApplied (*i\_strElementName* を文字列として) をブール値として

**説明**

名前 *i\_strElementName* を持つ XML 要素に選択されたテキストのすべてが埋め込まれるかチェックします。パラメーター *i\_strElementName* の一般的なサンプルは「strong」、「bold」または「italic」です。

**エラー**

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.9.42 AuthenticRange.LastTextPosition

次も参照してください:

**プロパティ:** LastTextPosition を long として

**説明**

範囲オブジェクトの右端のテキストの位置インデックスを設定または取得します。このインデックスは常に [FirstTextPosition](#) より小さい、または、等価です。インデックスはドキュメントの始まりの 0 から開始し、テキストカーソルの占めることのできる異なる箇所と共有増加します。テストの位置は 1 ずつ増加し、カーソル右キーと同じ効果があります。テストの位置は 1 ずつ削減するとことができ、カーソル左キーと同じ効果があります。

LastTextPosition を小さい値に設定すると、現在の [FirstTextPosition](#)、[FirstTextPosition](#) は新規の LastTextPosition に設定されます。

ヒント: カーソルの位置をベースとした XMLData に対して比較される複雑なオペレーションのため、気をつけてテキストカーソルの位置を使用してください。

**エラー**

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトが有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。
- 2006 ドキュメントの外部のテキストの位置が指定されています。

**サンプル**

```

-----
'
'           VBScript
'
-----
Dim objAuthenticView
Set objAuthenticView = objPlugin.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 "Ooops!"
End If

```

#### 4.2.9.43 AuthenticRange.LastXMLData

次も参照してください！

プロパティ: [LastXMLData](#) を [XMLData](#) として

##### 説明

範囲により部分的に、または、完全に選択される基となるドキュメント内でXMLData 要素を設定または取得します。選択範囲の正確な終了位置は、[LastXMLDataOffset](#) 属性により定義されています。

[LastXMLData](#) を新規のデータオブジェクトに設定すると、[LastXMLDataOffset](#) はこの要素内の最後のカーソル位置に設定されます。カーソルの位置を持つXMLData オブジェクトのみを使用することができます。[LastXMLData](#) / [LastXMLDataOffset](#) に設定すると現在の[FirstXMLData](#) / [FirstXMLDataOffset](#) より小さい位置を選択します。後者の場合、新規の終了位置に移動されます。

[AuthenticRange](#) オブジェクトで使用することができるメソッドが十分ではない場合、基となるXMLドキュメントに直接アクセス操作するために、[FirstXMLData](#) と[LastXMLData](#) プロパティを使用することができます。

##### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトが有効ではありません。
- 2005 戻り値/パラメータのために無効なアドレスが指定されています。
- 2008 内部エラー
- 2009 XMLData オブジェクトにアクセスすることができません。

#### 4.2.9.44 AuthenticRange.LastXMLDataOffset

次も参照してください！

プロパティ: [LastXMLDataOffset](#) を [long](#) として

##### 説明

範囲の終わりのために[LastXMLData](#) 要素内にカーソルの位置を設定または取得します。

オフセットの位置はテキストプロパティにより返された文字をベースとし、0 と共に開始します。新規のオフセットを設定する際、-1 を使用して要素内の最後の可能なポジションにオフセットを設定します。次のシナリオの場合、それぞれ特別な点に注意する必要があります:

- エンボボックス内のエントリのテキスト書式とCheckBoxes および類似するコントロールはスクリーン上に表示されるものと異なる場合があります。データオフセットはこのテキストをベースとしますが、2つの有効なオフセットの位置のみが存在します。1つはエントリの始めに、もう1つはエントリの終わりに存在します。オフセットをエントリ中に設定しようと試みると、オフセットが終了位置に設定されます。

- XML エンティティのテキスト書式は、スクリーン上の表記とは異なる場合があります。オフセットはこのテキスト書式をベースにします。

[LastXMLData](#) / [LastXMLDataOffset](#) が [FirstXMLData](#) / [FirstXMLDataOffset](#) の前の位置を選択する場合、後者の場合、新規の終了位置に移動されます。

#### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトが有効ではありません。
- 2005 無効なオフセットが指定されています。  
戻り値/パラメーターのために無効なアドレスが指定されています。

#### サンプル

```

'-----
'          VBScript
' Select the complete text of an XMLData element
' using XMLData based selection and ExpandTo
'-----

Dim objAuthenticView
Set objAuthenticView = objPlugin.AuthenticView

' first we use 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

```

### 4.2.9.45 AuthenticRange.MoveBegin

次も参照してください！

**メソッド:** [MoveBegin](#) (*eKind* を [SPYAuthenticElementKind](#) として、*nCount* を [long](#) として) を [AuthenticRange](#) として

#### 説明

範囲の開始位置を型 *eKind* の *nCount* 要素の開始位置に移動します。現在の範囲の開始オブジェクトからカウントは開始します。

*nCount* のために正の数値を使用してドキュメントの終わりに移動し、負の数値を使用してドキュメントの始まりに移動します。新規の終わりが範囲の始まりより大きい場合意外、範囲の始めは移動されません。この場合、終了位置は新規の開始位置に移動されます。メソッドは変更された範囲オブジェクトを返します。

#### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2003 ターゲットがドキュメントの終わりの後に存在します。

- 2004 ターゲットがドキュメントの始めの前に存在します。
- 2005 無効な要素の種類が指定されています。  
戻り値/ラマーターのために無効なアドレスが指定されています。

#### 4.2.9.46 AuthenticRange.MoveEnd

次も参照してください！

メソッド: `MoveEnd` (`eKind` を `SPYAuthenticElementKind` として、`nCount` を `long` として) を `AuthenticRange` として

##### 説明

Move the 範囲の終了位置を型 `eKind` の `nCount` 要素の始めに移動します。現在の範囲の終了位置オブジェクト。

`nCount` のために正の数値を使用してドキュメントの終わりに移動し、負の数値を使用してドキュメントの始まりに移動します。新規の終わりが範囲の始めより少ない場合意外、範囲の始めは移動されません。この場合、開始位置は新規の終了位置に移動されます。メソッドは変更された範囲オブジェクトを返します。

##### エラー

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2003 ターゲットがドキュメントの終わりの後に存在します。
- 2004 ターゲットがドキュメントの始まりの前に存在します。
- 2005 無効な要素の種類が指定されています。  
戻り値/ラマーターのために無効なアドレスが指定されています。

#### 4.2.9.47 AuthenticRange.MoveRowDown

次も参照してください！

メソッド: `MoveRowDown` () を `ブール値` として

##### 説明

範囲の始めが動的なテーブル内に存在し、このテーブルの最後の行ではない行を選択する場合、このメソッドは下の行と行をスワップします。範囲の選択は行と共に移動しますが、それ以外に変更されません。移動操作に成功した場合、関数は `True` を返します。それ以外の場合は `False` が返されます。

##### エラー

- 2001 Authentic 範囲オブジェクト、または、その関連したビューオブジェクトは有効ではありません。
- 2005 戻り値/ラマーターのために無効なアドレスが指定されています。

#### 4.2.9.48 AuthenticRange.MoveRowUp

次も参照してください！

メソッド: `MoveRowUp` () を `ブール値` として

**説明**

範囲の始めが動的なテーブル内に存在し、このテーブルの最初の行ではない行を選択する場合、このメソッドは上の行と行をスワップします。範囲の選択は行と共に移動しますが、それ以外に変更されません。移動の操作に成功した場合、関数は True を返します。それ以外の場合は False が返されます。

**エラー**

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメータのために無効なアドレスが指定されています。

**サンプル**

次を参照してください! [JScript - 動的テーブルの ヴォルノト](#).

#### 4.2.9.49 AuthenticRange.Parent

次も参照してください!

プロパティ: Parent を [AuthenticView](#) として(読み取り専用)

**説明**

この範囲オブジェクトを所有するビューにアクセスします。

**エラー**

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメータのために無効なアドレスが指定されています。

#### 4.2.9.50 AuthenticRange.Paste

次も参照してください!

メソッド: Paste () を [ブール値](#) として

**説明**

コピー貼り付けバッファが空の場合、またはコンテンツが現在の選択を置き換えられない場合、False を返します。

それ以外の場合、現在の選択を削除し、コピー貼り付けバッファのコンテンツを挿入し、True を返します。

**エラー**

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値/パラメータのために無効なアドレスが指定されています。

#### 4.2.9.51 AuthenticRange.PerformAction

次も参照してください!

メソッド: PerformAction (eAction を [SPYAuthenticActions](#) として、strElementName を [文字列](#) として) を [ブール値](#) として

**説明**

`PerformAction` と関連するメソッドは Authentic の入力ヘルパー機能へのアクセスを提供します。この機能は、変更が発生する正確な位置を知らなくドキュメントのコンテンツを簡単かつ整合的に変更することができます。範囲オブジェクトの最初は、指定されたアクションが実行される次の有効な位置を検索するために使用されます。このようなロケーションが見つからない場合、メソッドは False を返します。それ以外の場合、ドキュメントは変更され、範囲は変更の開始位置をポイントするようになります。

ヒント: 2番目のパラメーターとして、戻ることができる要素名を探るには、[CanPerformActionWith](#) を使用します。

**エラー**

- 2001 Authentic 範囲オブジェクト、または、関連するビューオブジェクトは有効ではありません。
- 2005 戻り値パラメーターのために無効なアドレスが指定されています。
- 2007 無効なアクションが指定されています。

**サンプル**

```

'-----
'           VBScript
' Insert the innermost element
'-----

Dim objRange
Set objRange = objPlugin.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

```

### 4.2.9.52 AuthenticRange.Select

次も参照してください:

メソッド: [Select \(\)](#)

**説明**

この範囲を現在のユーザーインターフェイス選択にします。以下を使用して同じ結果を得ることができます:  
'objRange.Parent.Selection = objRange'

**エラー**

- 2001 Authentic 範囲オブジェクト、または、その関連したビューオブジェクトは有効ではありません。

**サンプル**

```

'-----
'           VBScript
'-----

' set current selection to end of document
objPlugin.objAuthenticView.DocumentEnd.Select()

```

### 4.2.9.53 AuthenticRange.SelectNext

次も参照してください:

メソッド: [SelectNext](#) (*eKind* を *SPYAuthenticElementKind* として) を [AuthenticRange](#) として

#### 説明

現在の範囲の終了位置の後の型 *eKind* の要素を選択します。メソッドは変更された範囲オブジェクトを返します。

#### エラー

- 2001 Authentic 範囲オブジェクト、または関連するビューオブジェクトは有効ではありません。
- 2003 ターゲットがドキュメントの終わりの後に存在します。
- 2005 無効な要素の種類が指定されています。  
戻り値/パラメーターのために無効なアドレスが指定されています。

#### サンプル

```

'-----
'           VBScript
' Scan through the whole document word-by-word
'-----
Dim objAuthenticView
Set objAuthenticView = objPlugin.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

```

### 4.2.9.54 AuthenticRange.SelectPrevious

次も参照してください:

メソッド: [GotoPrevious](#) (*eKind* を *SPYAuthenticElementKind* として) を [AuthenticRange](#) として

#### 説明

現在の範囲の開始の前の型 *eKind* の要素を選択します。メソッドは変更された範囲オブジェクトを返します。

#### エラー

- 2001 Authentic 範囲オブジェクト、または関連するビューオブジェクトは有効ではありません。
- 2004 ターゲットがドキュメントの始まりの前に存在します。
- 2005 無効な要素の種類が指定されています。

戻り値/パラメーターのために無効なアドレスが指定されています。

### サンプル

```

'-----
'      VBScript
' Scan through the whole document tag-by-tag
'-----

Dim objAuthenticView
Set objAuthenticView = objPlugin.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentEnd
Dim bEndOfDocument
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

```

## 4.2.9.55 AuthenticRange.SetElementAttributeValue

次も参照してください！

**メソッド:** `SetElementAttributeValue` (`strElementName` を文字列として, `strAttributeName` を文字列として, `strAttributeValue` を文字列として)

### 説明

`strElementName` と共に識別されている要素のために `strAttributeName` 内で指定されている属性の値を取得します。属性がサポートされ値が割り当てられていない場合、空の文字列が返されます。要素によりサポートされる属性の名前を検索するには、[GetElementAttributeNames](#) または [HasElementAttribute](#) を使用してください。

### エラー

- 2001 Authentic 範囲オブジェクト、または、その関連したビューオブジェクトは有効ではありません。
- 2005 無効な要素名が指定されました。  
無効な属性名が指定されました。  
無効な属性値が指定されました。

### サンプル

```

'-----
'      VBScript
' Get and set element attributes
'-----

Dim objRange
Set objRange = objPlugin.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

```

#### 4.2.9.56 AuthenticRange.SetFromRange

次も参照してください:

**メソッド:** [SetFromRange](#) ([objSrcRange](#) を [AuthenticRange](#) として)

##### 説明

範囲オブジェクトを同じ開始位置と終了位置に [objSrcRange](#) として設定します。

##### エラー

- 2001 2つの範囲オブジェクトの1つが無効です。
- 2005 NULL オブジェクトがソースオブジェクトとして指定されています。

#### 4.2.9.57 AuthenticRange.SetVariableValue

**メソッド:** [SetVariableValue](#)([name](#) を文字列として, [value](#) を文字列として)

##### 戻り値

名前付き変数の値を設定します。

##### エラー

- 2001 無効なオブジェクトです。
- 2002 コンテキストノードが不足しています。
- 2204 範囲内にそのような変数はありません。
- 2205 変数を評価することできません。
- 2206 変数はトークンを返します。
- 2207 読み取り専用の変数です。
- 2208 変更は許可されていません。

### 4.2.9.58 AuthenticRange.Text

次も参照してください！

プロパティ: [Text](#) を [文字列](#) として

#### 説明

範囲オブジェクトにより選択されているテキストコンテキストを設定または取得します。

選択された範囲の開始位置と終了位置の間にテキストカーソルのポジションが存在するため、取得される文字数は、同一である必要がありません。ドキュメント要素の多くは次の要素のカーソルの開始位置とカーソルの終了位置の差分をサポートします。ドロップダウンリストはカーソルの位置の1つのみを管理します、選択することのできる文字列の長さの制限はありません。ラジオボタンとチェックボックスの場合、テキストプロパティの値は対応するXML要素の文字列をホールドします。

範囲が1つ以上の要素を選択する場合、テキストは単一の文字列の結合になります。XMLエンティティは拡張され、'&'は'&amp;'として期待されます。

テキストを空の文字列に設定するには、XML要素を削除せず、[Cut](#)、[Delete](#) または [PerformAction](#) を代わりに使用します。

#### エラー

- 2001 [Authentic](#) 範囲オブジェクト、または、その関連シグネチャーオブジェクトは有効ではありません。
- 2005 戻り値/パラメータのために無効なアドレスが指定されています。

## 4.2.10 AuthenticSelection

次も参照してください！

プロパティ

[Start](#)

[StartTextPosition](#)

[End](#)

[EndTextPosition](#)

### 4.2.10.1 AuthenticSelection.End

次も参照してください！

宣言: [End](#) を [XMLData](#) として

#### 説明

現在の選択が終了するXML要素

#### 4.2.10.2 AuthenticSelection.EndTextPosition

次も参照してください！

宣言: [EndTextPosition](#) を `long` として

説明

選択の範囲が終了する [Authentic.End](#) 内の位置

#### 4.2.10.3 AuthenticSelection.Start

次も参照してください！

宣言: [Start](#) を [XMLData](#) として

説明

現在の選択が開始するXML 要素

#### 4.2.10.4 AuthenticSelection.StartTextPosition

次も参照してください！

宣言: [StartTextPosition](#) を `long` として

説明

選択が開始する [Authentic.Start](#) 内の位置

### 4.2.11 AuthenticToolBarButton

メソッド

プロパティ

[CommandID](#)

#### 4.2.11.1 AuthenticToolBarButton.CommandID

宣言: [CommandID](#) を [SPYAuthenticCommand](#) として

説明

ツールボタンのCommandId。

## 4.2.12 AuthenticToolBarButtons

メソッド

[Item](#)

[NewButton](#)

[NewCustomButton](#)

[NewSeparator](#)

[Remove](#)

プロパティ

[Count](#)

### 4.2.12.1 AuthenticToolBarButtons.Count

宣言: [Count](#) を long として

説明

ツールバー上のボタンの実際の数量を取得します。  
読み取り専用です。

### 4.2.12.2 AuthenticToolBarButtons.Item

宣言: [Item](#)(n を long として) を [AuthenticToolBarButton](#) として

説明

現在のツールバーの n 番目のボタンを取得します。n は 1 から開始します。

サンプル

次のサンプルを確認してください! [AuthenticToolBarButtons.NewButton](#)

### 4.2.12.3 AuthenticToolBarButtons.NewButton

宣言: [NewButton](#)(nPosition を long として、nCommandId を [SPYAuthenticCommand](#) として)

説明

ツールバーのポジション nPosition でコマンド nCommandId のための新規のボタンを挿入します。nPosition は 1 から開始します。

**サンプル**

新規のツールバーが追加され、下に配置し、新規のツールバーボタンを追加します

```
objPlugIn.ToolbarRows.NewRow(3)           // 新規のツールバーが追加されます
                                           (行 3)

set ToolbarRow = objPlugIn.ToolbarRows.Item(3)
set Buttons = ToolbarRow.Buttons
ToolbarRow.Alignment = 2                 // 下へツールバーボタンを配置します
Buttons.NewButton 1, 2                   // 印刷ボタンを追加します
Buttons.NewButton 1, 3                   // プレビューを印刷ボタンを追加しま
                                           す
Buttons.NewSeparator 2                   // セレーターが追加されます
Buttons.NewButton 1, 4                   // 検証ボタンが追加されます
```

`StartEditing` または `ReloadToolbars` が呼び出された場合、変更されたツールバー設定が使用されます。

**4.2.12.4 AuthenticToolbarButtons.NewCustomButton**

**宣言:** `NewCustomButton`(`nPosition` を `long` として, `strName` を 文字列 として, `strTooltip` を 文字列 として, `strBitmapURL` を 文字列 として)

**説明**

ツールバーのポジション `nPosition` で `strName` という名前の新規のカスタムボタンを挿入します。 `nPosition` は 1 から開始します。  
`strTooltip` は ヒントテキストから取得されます。

`strBitmapURL` は新規のボタンのために表示されるビットマップの場所です。この URL は `TextStateBmpURL` プロパティを使用して設定された URL に相対的です。

**サンプル**

「`MyFunction`」という名前のカスタムボタンがツールバーに挿入されたと仮定します。`doceditcommand` の次のイベントハンドラーはユーザーがボタンをクリックするとどのように反応するか、およびボタンが有効化 / 無効化された状態を表示しています。

```
<SCRIPT LANGUAGE=javascript FOR=objPlugIn EVENT=doceditcommand>
  // event.type is set to "command" if the user clicked the button
  if(objPlugIn.event.type == "command")
  {
    if(objPlugIn.event.srcElement.Name == "MyFunction")
      window.alert("You pressed my custom button!");
  }

  // event.type is set to "update" if the button state must be set
  if(objPlugIn.event.type == "update")
  {
    if(objPlugIn.event.srcElement.Name == "MyFunction")
    {
      // we enable the button if only one element is selected

if(objPlugIn.CurrentSelection.Start.IsSameNode(objPlugIn.CurrentSelection.End))
      objPlugIn.event.returnValue = 1;
    else
```

```
        objPlugin.event.returnValue = 0;

        objPlugin.event.cancelBubble = true;
    }
}
</SCRIPT>
```

#### 4.2.12.5 AuthenticToolBarButtons.NewSeparator

**宣言:** `NewSeparator(nPosition)` を `long` として

##### 説明

ツールバーのポジション `nPosition` で識別子を挿入します。 `nPosition` は 1 から開始します。

##### サンプル

次のサンプルを確認してください! [AuthenticToolBarButtons.NewButton](#)

#### 4.2.12.6 AuthenticToolBarButtons.Remove

**宣言:** `Remove(nPosition)` を `long` として

##### 説明

ツールバーのポジション `nPosition` でボタンまたはセパレーターを削除します。 `nPosition` は 1 から開始します。

### 4.2.13 AuthenticToolBarRow

#### メソッド

## [Alignment](#)

#### プロパティ

## [Buttons](#)

#### 4.2.13.1 AuthenticToolBarRowAlignment

**宣言:** `Alignment(nAlign)` を `SPYAuthenticToolBarAllignment` として

##### 説明

プラグイン内のツールバーの配置を取得または配置します。

##### サンプル

全てのツールバーを下の部分に配置します:

```
for each ToolbarRow in objPlugin.ToolbarRows
```

```
        ToolbarRow.Alignment = 2
next
```

### 4.2.13.2 AuthenticToolbarRowButtons

**宣言:** ボタン を [AuthenticToolbarButtons](#) として

#### 説明

ツールバーのボタンを全て取得します。

#### サンプル

次のサンプルを確認してください! [AuthenticToolbarButtons.NewButton](#)

## 4.2.14 AuthenticToolbarRows

#### メソッド

[Item](#)

[RemoveRow](#)

[NewRow](#)

#### プロパティ

[Count](#)

### 4.2.14.1 AuthenticToolbarRows.Count

**Declaration:** Count を long として

#### 説明

定義されたツールバーの実際の数量を取得します。  
読み取り専用です。

### 4.2.14.2 AuthenticToolbarRows.Item

**宣言:** Item(nPosition を long として) を [AuthenticToolbarRow](#) として

#### 説明

ポジション nPosition でツールバーを取得します。nPosition は1 から開始します。  
読み取り専用です。

#### サンプル

次のサンプルを確認してください! [AuthenticToolbarButtons.NewButton](#)

### 4.2.14.3 AuthenticToolBarRows.RemoveRow

**宣言:** `RemoveRow(nPosition)` を `long` として

#### 説明

ユーザーインターフェイスから `nPosition` でツールバーを削除します。 `nPosition` は1 から開始します。

### 4.2.14.4 AuthenticToolBarRows.NewRow

**宣言:** `NewRow(nPosition)` を `long` として

#### 説明

新規のツールバー行を作成し挿入します。 `nPosition` は1 から開始します。

#### サンプル

次のサンプルを確認してください! [AuthenticToolBarButtons.NewButton](#)

## 4.2.15 AuthenticView

次も参照してください!

プロパティ	メソッド	イベント
<a href="#">Application</a>	<a href="#">Goto</a>	
<a href="#">DocumentBegin</a>	<a href="#">Print</a>	
<a href="#">DocumentEnd</a>	<a href="#">Redo</a>	
<a href="#">MarkupVisibility</a>	<a href="#">Undo</a>	
<a href="#">Parent</a>	<a href="#">UpdateXMLInstanceEntities</a>	
<a href="#">Selection</a>		
<a href="#">WholeDocument</a>		

#### 説明

`AuthenticView` その子オブジェクト [AuthenticRange](#) は簡単かつ整合性のあるドキュメントコンテンツの編集 `Authentic View` のためのインターフェイスを提供します。 `Authentic` オブジェクト内の既存のメソッドとプロパティの機能のオーバーラップは意図的であり、 `Authentic` オブジェクトのコンテンツの変更機能が古くなる可能性があります。このため新規の `AuthenticView` インターフェイスの使用強く奨励されます。

`AuthenticView` は印刷などの固有の機能、マルチレベルの元に戻す / 実行、および、現在のカーソルの選択、または、ポジションへの簡単なアクセスを提供します。

`AuthenticView` は型 [AuthenticRange](#) のオブジェクトを使用し、ドキュメント内のナビゲーションを簡単にし、テキスト要素の柔軟かつロジカルな選択をおこなうことができます。簡単な選択のために [DocumentBegin](#)、 [DocumentEnd](#)、または [WholeDocument](#) を使用し、更に複雑な選択のためには [Goto](#) メソッドを使用します。ドキュメント範囲に対して相対的にナビゲートするには、 [AuthenticRange](#) オブジェクトのメソッドとプロパティを参照してください。

## 4.2.15.1 Events

### 4.2.15.1.1 OnBeforeCopy

イベント: `OnBeforeCopy()` をブール値として

#### XMLSpy スクリプト環境 - VBScript

```
Function On_AuthenticBeforeCopy()  
    ' On_AuthenticBeforeCopy = False ' to disable operation  
End Function
```

#### XMLSpy スクリプト環境 - JScript

```
function On_AuthenticBeforeCopy()  
{  
    // return false; /* to disable operation */  
}
```

#### XMLSpy IDE プラグイン

```
IXMLSpyPlugIn.OnEvent (21, ...) // nEventId = 21
```

#### 説明

このイベントは、ドキュメント上でコピー操作の前トリガーされます。コピー操作を許可する場合は、True を返します。(または、何も返しません)。コピーを無効化する場合、False が返されます。

### 4.2.15.1.2 OnBeforeCut

イベント: `OnBeforeCut()` をブール値として

#### XMLSpy スクリプト環境 - VBScript

```
Function On_AuthenticBeforeCut()  
    ' On_AuthenticBeforeCut = False ' to disable operation  
End Function
```

#### XMLSpy スクリプト環境 - JScript

```
function On_AuthenticBeforeCut()  
{  
    // return false; /* to disable operation */  
}
```

#### XMLSpy IDE プラグイン

```
IXMLSpyPlugIn.OnEvent (20, ...) // nEventId = 20
```

#### 説明

このイベントはドキュメント上で切り取り操作前トリガーされます。切り取り操作を許可するために True を返します (または何も返しません)。操作を無効化するために false を返します。

## 4.2.15.1.3 OnBeforeDelete

イベント: `OnBeforeDelete()` をブール値として

XMLSpy スクリプト環境 - VBScript

```
Function On_AuthenticBeforeDelete()
    ' On_AuthenticBeforeDelete = False ' to disable operation
End Function
```

XMLSpy スクリプト環境 - JScript

```
function On_AuthenticBeforeDelete()
{
    // return false; /* to disable operation */
}
```

XMLSpy IDE プラグイン

```
IXMLSpyPlugIn.OnEvent (22, ...) // nEventId = 22
```

## 説明

このイベントはドキュメント上で削除の操作前トリガーされます。削除操作を許可するために True を返します (または何も返しません)。操作を無効化するために false を返します。

## 4.2.15.1.4 OnBeforeDrop

イベント: `OnBeforeDrop` (`i_nXPos` long として、`i_nYPos` long として、`i_ipRange` を `AuthenticRange` として、`i_ipData` を `cancelBoolean` として)

XMLSpy スクリプト環境 - VBScript

```
Function On_AuthenticBeforeDrop(nXPos, nYPos, objRange, objData)
    ' On_AuthenticBeforeDrop = False ' to disable operation
End Function
```

XMLSpy スクリプト環境 - JScript

```
function On_AuthenticBeforeDrop(nXPos, nYPos, objRange, objData)
{
    // return false; /* to disable operation */
}
```

XMLSpy IDE プラグイン

```
IXMLSpyPlugIn.OnEvent (11, ...) // nEventId = 11
```

## 説明

このイベントはアプリケーションウィンドウ内で以前にドラッグされたオブジェクトがドロップされるトリガーされます。イベントに関連する全ての情報はパラメータに伝達されます。

イベントが発生する際に、最初の2つのパラメータはマウスの場所を指定します。1パラメータ `objRange` はマウスの場所の下要素を選択する範囲オブジェクトに伝達します。このパラメータの値は NULL である可能性があります。範囲オブジェクトにアクセスする前に確認してください。1パラメータ `objData` によりドラッグされたレイールオブジェクトの情報へのアクセスが許可されます。

ドロップオペレーションをキャンセルするために False を返します。通常の操作を続行するために True を返します (または、何も返しません)。

サンプル

```

' VB code snippet - connecting to object level events
'-----
' access XMLSpy (without checking for any errors)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' this is the event callback routine connected to the OnBeforeDrop
' event of object objView
Private Function objView_OnBeforeDrop(ByVal i_nXPos As Long, ByVal i_nYPos As Long,
    ByVal i_ipRange As IAuthenticRange,
    ByVal i_ipData As IAuthenticDataTransfer) As Boolean

    If (Not i_ipRange Is Nothing) Then
        MsgBox ("Dropping on content is prohibited");
        Return False;
    Else
        Return True;
    End If
End Function

' use VBA keyword WithEvents to connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continue here with something useful ...
' and serve the windows message loop

```

## 4.2.15.1.5 OnBeforePaste

イベント: `OnBeforePaste` (`objData` パラメータ型として、`strType` を文字列として) をブール値として

XMLSpy スクリプト環境 - VBScript

```

Function On_AuthenticBeforePaste(objData, strType)
    ' On_AuthenticBeforePaste = False ' to disable operation
End Function

```

XMLSpy スクリプト環境 - JScript

```

function On_AuthenticBeforePaste(objData, strType)
{
    // return false; /* to disable operation */
}

```

XMLSpy IDE プラグイン

```
IXMLSpyPlugIn.OnEvent (19, ...) // nEventId = 19
```

## 説明

このイベントはドキュメント上で貼り付けの操作前にトリガーされます。パラメータ `strType` は "TEXT"、"UNICODETEXT" または "UNKNOWN" の一つです。最初の二つの場合、`objData` には解析されるオブジェクトの文字列の表記が含まれています。後者の場合、`objData` には IUnknown COM インターフェイスへのポインターが含まれています。貼り付け操作を許可するには True を返します。(または、何も返しません) 操作を無効化するために False が返されます。

#### 4.2.15.1.6 OnBeforeSave

イベント: `OnBeforeSave` (SaveAs フラグ) をブール値として

**説明:** `OnBeforeSave` は以下をおこなう機会を与えます。例: 既存のXMLドキュメントが上書きされる場合警告、または、固有の環境が満たされない場合ドキュメントを読み取り専用にするように警告します。ファイルダイアログが表示される前にイベントは実行されます (StyleVision 内でSPS デザインを保存される際ではなく、XMLドキュメントを保存する際にイベントは実行されることにご注意してください)。

#### 4.2.15.1.7 OnDragOver

イベント: `OnDragOver` (`nXPos` long として、`nYPos` long として、`eMouseEvent` を `SPYMouseEvent` として、`objRange` を `AuthenticRange` として、`objData` を `AuthenticDataTransfer` として) をブール値として

##### XMLSpy スクリプト環境 - VBScript

```
Function On_AuthenticDragOver(nXPos, nYPos, eMouseEvent, objRange, objData)
    ' On_AuthenticDragOver = False ' to disable operation
End Function
```

##### XMLSpy スクリプト環境 - JScript

```
function On_AuthenticDragOver(nXPos, nYPos, eMouseEvent, objRange, objData)
{
    // return false; /* to disable operation */
}
```

##### XMLSpy IDE プラグイン

```
IXMLSpyPlugIn.OnEvent (10, ...) // nEventId = 10
```

##### 説明

このイベントはアプリケーションウィンドウにAuthentic View の外のオブジェクトがマウスオーバーによりドラッグされるとトリガーされます。イベントに関連する全ての情報はパラメータに渡されます。

イベントが発生する際に、最初の3つのパラメータはマウスの場所、マウスボタンの状態と仮想キーの状態を指定します。1パラメータ `objRange` はマウスの場所の下要素を選択する範囲オブジェクトに渡します。このパラメータの値はNULLである可能性があります。範囲オブジェクトにアクセスする前に確認してください。1パラメータ `objData` によりドラッグされるレイ尔オブジェクトの情報へのアクセスが許可されます。

ドラッグオペレーションをキャンセルするためにFalse を返します。通常の操作を続行するためにTrue を返します(または、何も返しません)。

##### サンプル

```
'-----
' VB code snippet - connecting to object level events
'-----
' access XMLSpy (without checking for any errors)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' this is the event callback routine connected to the OnDragOver
' event of object objView
Private Function objView_OnDragOver(ByVal i_nXPos As Long, ByVal i_nYPos As Long,
    ByVal i_eMouseEvent As SPYMouseEvent,
    ByVal i_ipRange As IAuthenticRange,
    ByVal i_ipData As IAuthenticDataTransfer) As Boolean
```

```

If (((i_eMouseEvent And spyShiftKeyDownMask) <> 0) And
    (Not i_ipRange Is Nothing)) Then
    MsgBox ("Floating over element " & i_ipRange.FirstXMLData.Parent.Name);
End If

Return True;
End Function

' use VBA keyword WithEvents to connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continue here with something useful ...
' and serve the windows message loop

```

#### 4.2.15.1.8 OnKeyboardEvent

**イベント:** `OnKeyboardEvent` (`eKeyEvent` を `SPYKeyEvent` として、`nKeyCode` long として、`nVirtualKeyStatus` long として) をブール値として

##### XMLSpy スクリプト環境 - VBScript

```

Function On_AuthenticKeyboardEvent(eKeyEvent, nKeyCode, nVirtualKeyStatus)
    ' On_AuthenticKeyboardEvent = True ' to cancel bubbling of event
End Function

```

##### XMLSpy スクリプト環境 - JScript

```

function On_AuthenticKeyboardEvent(eKeyEvent, nKeyCode, nVirtualKeyStatus)
{
    // return false; /* to cancel bubbling of event */
}

```

##### XMLSpy IDE プラグイン

```

IXMLSpyPlugIn.OnEvent (30, ...) // nEventId = 30

```

##### 説明

このイベントは `WM_KEYDOWN`、`WM_KEYUP` および `WM_CHAR` ウィンドウメッセージのためにトリガーされます。

実際のメッセージ型は `eKeyEvent` / パラメーター内で使用することができます。仮想キーの状態はパラメーター `nVirtualKeyStatus` 内で結合されます。異なるキーまたはその組合せのために、列挙データ型 `SPYVirtualKeyMask` 内で定義されているビットマスクを使用します。

注釈 XMLSpy のスクリプト環境とIDE プラグインからの次のイベントはサポートされますが、このイベントでは古いため使用されません

```

On_AuthenticKeyUp()           IXMLSpyPlugIn.OnEvent (13, ...) // nEventId = 13
On_AuthenticKeyDown()        IXMLSpyPlugIn.OnEvent (12, ...) // nEventId = 12
On_AuthenticKeyPressed()     IXMLSpyPlugIn.OnEvent (14, ...) // nEventId = 14

```

##### サンプル

```

'-----
' VB code snippet - connecting to object level events
'-----
' access XMLSpy (without checking for any errors)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' this is the event callback routine connected to the OnKeyboard

```

```

' event of object objView
Private Function objView_OnKeyboardEvent(ByVal i_keyEvent As Long, ByVal io_pnKeyCode As Long, ByVal
i_nVirtualKeyStatus As Long) As Boolean
    If ((i_keyEvent = XMLSpyLib.spyKeyUp) And ((i_nVirtualKeyStatus And XMLSpyLib.spyCtrlKeyMask) <> 0)) Then
        MsgBox ("Ctrl " & io_pnKeyCode & " pressed")
        objView_OnKeyboardEvent = True
    Else
        objView_OnKeyboardEvent = False
    End If
End Function

' use VBA keyword WithEvents to connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continue here with something useful ...
' and serve the windows message loop

```

#### 4.2.15.1.9 OnLoad

イベント: `OnLoad ()`

説明: 下のサンプルで示されているように、`OnLoad` は、例えば、`AuthenticView` 機能機能の一部を制約するために使用されます:

```

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 );
}

```

サンプル内では、小さなマークアップ、大きなマークアップ、複合型マークアップの状態ツールバーボタンはボタン識別子の助けにより操作されま  
す。次を参照してください: [完全なリスト](#)。

#### 4.2.15.1.10 OnMouseEvent

イベント: `OnMouseEvent` (`nXPos` long とて、`nYPos` long とて、`eMouseEvent` を `SPYMouseEvent` とて、`objRange` を `AuthenticRange` とて) をブール値とて

##### XMLSpy スクリプト環境 - VBScript

```

Function On_AuthenticMouseEvent(nXPos, nYPos, eMouseEvent, objRange)
    ' On_AuthenticMouseEvent = True ' to cancel bubbling of event
End Function

```

##### XMLSpy スクリプト環境 - JScript

```

function On_AuthenticMouseEvent(nXPos, nYPos, eMouseEvent, objRange)

```

```
{
    // return false; /* to cancel bubbling of event */
}
```

### XMLSpy IDE プラグイン

```
IXMLSpyPlugIn.OnEvent (31, ...) // nEventId = 31
```

#### 説明

このイベントは各マウスの動きとマウスボタンウインドウメッセージのためトリガーされます。

実際のメッセージ型とマウスボタンの状態は、eMouseEvent / 参照範囲内で使用することができます。列挙データ型 SPYMouseEvent 内で定義されているビットマスクを使用して、異なるメッセージ、ボタンの状態、その組合せをテストします。

参照範囲 objRange は現在のマウスカーソルの位置で見つけられドキュメントの一部を識別します。範囲オブジェクトは常にドキュメントの完全なタグを選択します (更に正確なポジションメカニズムを使用できるようになると、この機能は将来のバージョンで変更される可能性があります)。現在のポジションでドキュメントの選択できる部分が存在しない場合、範囲オブジェクト NULL です。

注釈 XMLSpy のスクリプト環境と IDE プラグインからの次のイベントはサポートされますが、このイベントでは古いため使用されません

```
On_AuthenticMouseMove()           IXMLSpyPlugIn.OnEvent (15, ...) // nEventId = 15
On_AuthenticButtonUp()           IXMLSpyPlugIn.OnEvent (16, ...) // nEventId = 16
On_AuthenticButtonDown()        IXMLSpyPlugIn.OnEvent (17, ...) // nEventId = 17
On_AuthenticButtonDoubleClick()  IXMLSpyPlugIn.OnEvent (24, ...) // nEventId = 24
```

#### サンプル

```
'-----
' VB code snippet - connecting to object level events
'-----
' access XMLSpy (without checking for any errors)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' this is the event callback routine connected to the OnMouseEvent
' event of object objView. If you click with the left mouse button
' while pressing a control key, the current selection will be set
' to the tag below the current mouse cursor position
Private Function objView_OnMouseEvent(ByVal i_nXPos As Long, ByVal i_nYPos As Long, ByVal i_eMouseEvent As
XMLSpyLib.SPYMouseEvent, ByVal i_pRange As XMLSpyLib.IAuthenticRange) As Boolean
    If (i_eMouseEvent = (XMLSpyLib.spyLeftButtonDownMask Or XMLSpyLib.spyCtrlKeyDownMask)) Then
        On Error Resume Next
        i_pRange.Select
        objView_OnMouseEvent = True
    Else
        objView_OnMouseEvent = False
    End If
End Function

' use VBA keyword WithEvents to connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continue here with something useful ...
' and serve the windows message loop
```

#### 4.2.15.1.11 OnSelectionChanged

イベント: OnSelectionChanged (objNewSelection を AuthenticRange として)

XMLSpy スクリプト環境 - VBScript

```
Function On_AuthenticSelectionChanged (objNewSelection)
End Function
```

XMLSpy スクリプト環境 - JScript

```
function On_AuthenticSelectionChanged (objNewSelection)
{
}
```

XMLSpy IDE プラグイン

```
IXMLSpyPlugin.OnEvent (23, ...) // nEventId = 23
```

**説明**

このイベントはユーザーインターフェイス内の選択が変更されるとトリガーされます。

サンプル

```
'-----
' VB code snippet - connecting to object level events
'-----
' access XMLSpy (without checking for any errors)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' this is the event callback routine connected to the OnSelectionChanged
' event of object objView
Private Sub objView_OnSelectionChanged (ByVal i_ipNewRange As XMLSpyLib.IAuthenticRange)
MsgBox ("new selection: " & i_ipNewRange.Text)
End Sub

' use VBA keyword WithEvents to connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continue here with something useful ...
' and serve the windows message loop
```

## 4.2.15.1.12 OnToolBarButtonClicked

**イベント:** OnToolBarButtonClicked (ボタン識別子)

**説明:** OnToolBarButtonClicked はツールバーボタンがユーザーによりクリックされると実行されます。パラメーターボタン識別子はクリックするボタンを決定する手助けをします。定義済みボタン識別子のリストは以下のとおりです。

- 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

カスタム ボタン ユーザーのために自身の識別子が追加される場合があります。識別子の一意性はチェックされないため、この点をユーザーが注意する必要があります。Set/GetToolBarState() COM API 呼び出し内でボタンを識別するために同じ識別子を使用することができます。異なるボタンのためにコードを追加することにより、ユーザーは、自身のメソッドをテーブルの操作のために追加するなど、AuthenticView ツールバーの振る舞いを完全に定義することができるようになります。

#### 4.2.15.1.13 OnToolBarButtonExecuted

イベント: [OnToolBarButtonExecuted](#) (ボタン 識別子)

**説明:** OnToolBarButtonClicked はツールバーボタンがユーザーによりクリックされると実行されます。1 パラメーター ボタン 識別子はクリックするボタンを決定する手助けをします。 [定義済みボタン 識別子のリスト](#)を参照してください。

下のサンプルで示されているように、OnToolBarButtonExecuted はツールバーアクションが実行された後実行されます。更新コードを追加するなど、とても役立ちます。:

```
//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();
}

```

この場合 UpdateOwnToolbarButtonStates はグローバル宣言内で定義されたユーザー関数です。

#### 4.2.15.1.14 OnUserAddedXMLNode

イベント: [OnUserAddedXMLNode](#) (XML ノード)

説明: ユーザーがXML ノードをプライマリアクションとして追加すると、[OnUserAddedXMLNode](#) は実行されます。

- ハイパーリンクを自動的に追加します ([OnUserAddedXMLNode.sps](#) のサンプルを参照してください)
- コンテキスト メニューアイテムの挿入、の後に挿入、の前に挿入
- 行の追加、行の削除ツールボタン
- 要素 入力ヘルパー ([StyleVision](#) の外部) 内の後に挿入、前に挿入アクション

イベントは以下の場合実行されません。行の複製時、ノードが例: COM API などを使用して 外部的に追加された時、適用時 (例: テキスト状態アイコン)、または、XML テーブルオペレーション または DB オペレーション内。

ユーザーに追加されたXML ノードを操作する機会を与えるために追加されたイベント/パラメーターはXML ノード オブジェクトです。イベントハンドラーの複雑なサンプルは、[OnUserAddedXMLNode.sps](#) ファイル内にあります (プロジェクト ウィンドウ内のサンプルプロジェクトの [Authentic/Scripting](#) フォルダー)。

#### 4.2.15.2 AuthenticView.Application

次も参照してください:

プロパティ: [Application](#) を [Authentic](#) として(読み取り専用)

説明

アプリケーションオブジェクトにアクセスします。

エラー

- 2000 [AuthenticView](#) オブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.15.3 AuthenticView.AsXMLString

次も参照してください:

プロパティ: [AsXMLString](#) を文字列として

**説明**

ドキュメントコンテンツXML 文字列を返す、または、設定します。新規の値(コンテンツ)を設定しても、使用中のスキーマファイルまたはSPS ファイルは変更されません。新規のXMLString が実際のスキーマファイルと一致しない場合、エラー2011 が返されます。

**エラー**

- 2000 AuthenticView オブジェクトは有効ではありません。
- 2011 AsXMLString は現在のスキーマファイルのための有効なXMLではない値のために設定されています。

#### 4.2.15.4 AuthenticView.ContextMenu

**プロパティ:** ContextMenu を ContextMenu として

**説明**

プロパティ ContextMenu はコンテキストメニューをカスタマイズするためのアクセスを提供します。イベントハンドラー OnContextMenuActivated 内で操作を行う適切な場所です。

**エラー**

- 2000 無効なオブジェクトです。
- 2005 無効なパラメーターです。

#### 4.2.15.5 AuthenticView.CreateXMLNode

**メソッド:** CreateXMLNode (nKind を SPYXMLDataKind として) を XMLData として

**戻り値**

メソッドは新規のXMLData オブジェクトを返します。

**説明**

新規のXMLData オブジェクトを作成するには、CreateXMLNode () メソッドを使用します。次も参照してください: XMLData の使用。

**エラー**

- 2000 無効なオブジェクトです。
- 2012 XML ノードを作成することができません。

#### 4.2.15.6 AuthenticView.DisableAttributeEntryHelper

**メソッド:** DisableAttributeEntryHelper ()

**説明**

XMLSpy、Authentic Desktop と Authentic Browser プラグイン内で DisableAttributeEntryHelper () は属性入力ヘルパーを無効化します。

**エラー**

- 2000 無効なオブジェクトです。

#### 4.2.15.7 AuthenticView.DisableElementEntryHelper

メソッド: `DisableElementEntryHelper()`

##### 説明

XMLSpy、Authentic Desktop と Authentic Browser プラグイン内で `DisableElementEntryHelper()` は要素入力ヘルパーを無効化します。

##### エラー

2000 無効なオブジェクトです。

#### 4.2.15.8 AuthenticView.DisableEntityEntryHelper

メソッド: `DisableEntityEntryHelper()`

##### 説明

XMLSpy、Authentic Desktop と Authentic Browser プラグイン内で `DisableEntityEntryHelper()` はエンティティ入力ヘルパーを無効化します。

##### エラー

2000 無効なオブジェクトです。

#### 4.2.15.9 AuthenticView.DocumentBegin

次も参照してください:

プロパティ: `DocumentBegin` を [AuthenticRange](#) として(読み取り専用)

##### 説明

ドキュメントの始めを指す範囲オブジェクトを抽出します。

##### エラー

2000 AuthenticView オブジェクトは有効ではありません。  
2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.15.10 AuthenticView.DocumentEnd

次も参照してください:

プロパティ: `DocumentEnd` を [AuthenticRange](#) として(読み取り専用)

##### 説明

ドキュメントの終わりを指す範囲オブジェクトを抽出します。

エラー

- 2000 AuthenticView オブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.15.11 AuthenticView.DoNotPerformStandardAction

メソッド: [DoNotPerformStandardAction](#) ()

説明

`DoNotPerformStandardAction()` はマクロのための プルのキャンセルとしての役割を果たします。マクロの完了後、更なる実行を停止します。

エラー

- 2000 無効なオブジェクトです。

#### 4.2.15.12 AuthenticView.EvaluateXPath

メソッド: [EvaluateXPath](#) ([XMLData](#), 文字列式) を文字列として

戻り値

メソッドは文字列を返します。

説明

`EvaluateXPath()` 指定されているXML コンテキストノードを持つXPath 式を実行します。結果は文字列として返されます。シーケンスの場合は、スペース文字で区切られた文字列が返されます。

エラー

- 2000 無効なオブジェクトです。
- 2005 無効なパラメーターです。
- 2008 内部エラーです。
- 2013 XPath エラー。

#### 4.2.15.13 AuthenticView.Event

次も参照してください:

プロパティ: [Event](#) を `AuthenticEvent` として(読み取り専用)

説明

このプロパティは、最後のイベントのパラメーターへ `OldAuthenticView.event` と同様にアクセスを与えます。スクリプト環境と外部クライアントのために全てのイベントを使用することができるため、この `Event` プロパティはIDE プラグイン内でのみ使用されるべきです。

エラー

- 2000 AuthenticView オブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.15.14 AuthenticView.EventContext

**プロパティ:** `EventContext` を `EventContext` として

##### 説明

`EventContext` プロパティはマクロコンテキストの実行へのアクセスを与えます。詳細に関しては、`EventContext` インターフェイスを参照してください。

##### エラー

2000 無効なオブジェクトです。

#### 4.2.15.15 AuthenticView.GetToolBarButtonState

**メソッド:** `GetToolBarButtonState` (`ButtonIdentifier` を `string` として) を `AuthenticToolBarButtonState` として

##### 戻り値

メソッド returns `AuthenticToolBarButtonState`

##### 説明

`GetToolBarButtonState` はツールバーボタンの状況をクエリ、ユーザーがボタンを有効化、または無効化し、ボタン識別子を使用して識別します ([上のリスト参照](#))。1つの使用法はツールバーボタンを永久に無効化することです。ドキュメント内の選択が変更されるとツールボタンが定期的に更新されるため、他の使用法は `OnSelectionChanged` イベントハンドラー内に `SetToolBarButtonState` を配置します。

[リストされた列挙](#)によりツールボタンの状態が与えられます。

デフォルトの状態はボタンの有効化/無効化は `AuthenticView` により管理されています。ユーザーがボタンの状態を有効化、または無効化すると、ユーザーが変更しない限り、ボタンはそのままの状態です。

##### エラー

2000 無効なオブジェクトです。  
2005 無効なパラメーターです。  
2008 内部エラーです。  
2014 無効なボタン識別子です。

#### 4.2.15.16 AuthenticView.Goto

次も参照してください

**メソッド:** `Goto` (`eKind` を `SPYAuthenticElementKind` として、`nCount` を `long` として、`eFrom` を `SPYAuthenticDocumentPosition` として) を `AuthenticRange` として

##### 説明

型 `eKind` の `nCount` 要素の開始位置をポイントする範囲オブジェクトを取得します。パラメーター `eFrom` により開始の位置が定義されます。 `nCount` のために正の値を使用し、ドキュメントの終了位置にナビゲートします。ドキュメントの始めに向かいナビゲートするには負の値を使用します。

エラー

- 2000 AuthenticView オブジェクトは有効ではありません。
- 2003 ターゲットがドキュメントの終わりの後に存在します。
- 2004 ターゲットはドキュメントの始めの前に存在します。
- 2005 無効な要素の種類が指定されています。  
ドキュメント のポジションはspyAuthenticDocumentBegin または  
spyAuthenticDocumentEnd ではありません。  
戻り値/ラマーターのために無効なアドレスが指定されています。

サンプル

```
Dim objAuthenticView
Set objAuthenticView = objPlugin.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
```

## 4.2.15.17 AuthenticView.IsRedoEnabled

次も参照してください！

**プロパティ:** IsRedoEnabled ブール値として(読み取り専用)

説明

やり直しステップとやり直し が可能な場合、True です。

エラー

- 2000 AuthenticView オブジェクトは有効ではありません。
- 2005 戻り値/ラマーターのために無効なアドレスが指定されています。

## 4.2.15.18 AuthenticView.IsUndoEnabled

次も参照してください！

**プロパティ:** IsUndoEnabled ブール値として(読み取り専用)

説明

元に戻す ステップを使用することができ、元に戻す 操作が可能な場合、True が返されます。

エラー

- 2000 AuthenticView オブジェクトは有効ではありません。
- 2005 戻り値/ラマーターのために無効なアドレスが指定されています。

#### 4.2.15.19 AuthenticView.MarkupVisibility

次も参照してください！

プロパティ: `MarkupVisibility` を `SPYAuthenticMarkupVisibility` として

##### 説明

マークアップの現在の可視性を設定または取得します。

##### エラー

- 2000 AuthenticView オブジェクトは有効ではありません。
- 2005 無効な列挙の値が指定されました。  
戻り値/ラメーターのために無効なアドレスが指定されています。

#### 4.2.15.20 AuthenticView.Parent

次も参照してください！

プロパティ: `Parent` を `Authentic` として(読み取り専用)

##### 説明

アプリケーションにアクセスします。

##### エラー

- 2000 AuthenticView オブジェクトは有効ではありません。
- 2005 戻り値/ラメーターのために無効なアドレスが指定されています。

#### 4.2.15.21 AuthenticView.Print

次も参照してください！

メソッド: `Print` (`bWithPreview` をブール値として, `bPromptUser` をブール値として)

##### 説明

このビュー内で表示されるドキュメントを印刷します。`bWithPreview` が `True` に設定されている場合、印刷プレビューダイアログがポップアップします。`bPromptUser` が `True` に設定されている場合、印刷ダイアログがポップアップします。両方のラメーターが `False` に設定されている場合、ドキュメント 更なるユーザーインタラクション無しで印刷されます。

##### エラー

- 2000 AuthenticView オブジェクトは有効ではありません。

#### 4.2.15.22 AuthenticView.Redo

次も参照してください！

メソッド: [Redo \(\)](#) を [ブール値](#) として

##### 説明

最後の元に戻すコマンドにより元に戻された変更をやり直します。

##### エラー

- 2000 AuthenticView オブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.15.23 AuthenticView.Selection

次も参照してください！

プロパティ: [Selection](#) を [AuthenticRange](#) として

##### 説明

ユーザーインターフェイス内の現在のテキストの選択を設定または取得します。

##### エラー

- 2000 AuthenticView オブジェクトは有効ではありません。
- 2002 カーソルのセグメントはアクティブではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

##### サンプル

```

'-----
'           VBScript
'-----
Dim objAuthenticView
Set objAuthenticView = objPlugin.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

```

#### 4.2.15.24 AuthenticView.SetToolBarButtonState

メソッド: [SetToolBarButtonState](#) ([ButtonIdentifier](#) を文字列として、[AuthenticToolBarButtonState](#) 状態)

**説明**

SetToolBarButtonState はツールバーボタンの状況をクエリ、ユーザーがボタンを有効化、または無効化し、ボタン識別子を使用して識別します ([上のリスト参照](#))。1つの使用法はツールバーボタンを永久に無効化することです。ドキュメント内の選択が変更されるとツールボタンが定期的に更新されるため、他の使用法は OnSelectionChanged イベントハンドラー内に SetToolBarButtonState を配置します。

[リストされた列挙](#)によりツールボタンの状態が与えられます。

デフォルトの状態はボタンの有効化/無効化は AuthenticView により管理されています。ユーザーがボタンの状態を有効化、または無効化すると、ユーザーが変更しない限り、ボタンはそのままの状態です。

**エラー**

- 2000 無効なオブジェクトです。
- 2008 内部エラーです。
- 2014 無効なボタン識別子です。

### 4.2.15.25 AuthenticView.Undo

次も参照してください！

メソッド: [Undo \(\)](#) を [ブール値](#) として

**説明**

このビュー内からドキュメントに追加されている最後の変更を元に戻します。

**エラー**

- 2000 AuthenticView オブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

### 4.2.15.26 AuthenticView.UpdateXMLInstanceEntities

次も参照してください！

メソッド: [UpdateXMLInstanceEntities \(\)](#)

**説明**

宣言されたエンティティの内部表記が更新され、入力ヘルパーがリフィルされます。更に、バリデータが再ロードされ、XML ファイルが正確に検証されます。これによりスキーマファイルが再ロードされる可能性があります。

**エラー**

メソッドはエラーを決して返しません。

**サンプル**

```
// -----
//           JavaScript
// -----
var objDocType;
objDocType = objPlugin.XMLRoot.GetFirstChild(10);

if(objDocType)
{
```

```
var objEntity = objPlugin.CreateChild(14);
objEntity.Name = "child";
objEntity.TextValue = "SYSTEM ¥"child.xml¥";
objDocType.AppendChild(objEntity);

objPlugin.AuthenticView.UpdateXMLInstanceEntities();
}
```

#### 4.2.15.27 AuthenticView.WholeDocument

次も参照してください！

プロパティ: [WholeDocument](#) を [AuthenticRange](#) として (読み取り専用)

##### 説明

ドキュメント全体を選択する範囲オブジェクトを抽出します。

##### エラー

- 2000 AuthenticView オブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.15.28 AuthenticView.XMLDataRoot

次も参照してください！

プロパティ: [XMLDataRoot](#) を [XMLData](#) として (読み取り専用)

##### 説明

現在のドキュメントのトップレベルのXMLData 要素を返す、または、設定します。この要素は通常ドキュメント構造を説明し、型 spyXMLDataXMLDocStruct、spyXMLDataXMLEntityDocStruct または spyXMLDataDTDDocStruct です。

##### エラー

- 2000 AuthenticView オブジェクトは有効ではありません。
- 2005 戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.16 AuthenticXMLTableCommands

##### メソッド

[Insert](#)

[Delete](#)

[AppendRow](#)

[InsertRow](#)

[DeleteRow](#)

[AppendColumn](#)  
[InsertColumn](#)  
[DeleteColumn](#)  
[JoinLeft](#)  
[JoinRight](#)  
[JoinUp](#)  
[JoinDown](#)  
[SplitHorizontal](#)  
[SplitVertical](#)  
[AlignVerticalTop](#)  
[AlignVerticalCenter](#)  
[AlignVerticalBottom](#)  
[AlignHorizontalLeft](#)  
[AlignHorizontalRight](#)  
[AlignHorizontalCenter](#)  
[AlignHorizontalJustify](#)  
[EditProperties](#)

プロパティ

[MayInsert](#)  
[MayDelete](#)  
[MayAppendRow](#)  
[MayInsertRow](#)  
[MayDeleteRow](#)  
[MayAppendCol](#)  
[MayInsertCol](#)  
[MayDeleteCol](#)  
[MayJoinLeft](#)  
[MayJoinRight](#)  
[MayJoinUp](#)  
[MayJoinDown](#)

[MaySplitHorizontal](#)  
[MaySplitVertical](#)  
[MayAlignVertical](#)  
[MayAlignHorizontal](#)  
[MayEditProperties](#)

#### 4.2.16.1 AuthenticXMLTableCommands.AlignHorizontalCenter

**宣言:** [AlignHorizontalCenter\(\)](#)

**説明**

垂直方向の配置の属性を「中央」に設定します、または、以前「中央」に設定されていた場合属性をクリアします。

#### 4.2.16.2 AuthenticXMLTableCommands.AlignHorizontalJustify

**宣言:** [AlignHorizontalJustify\(\)](#)

**説明**

垂直方向の配置の属性を「両端揃え」に設定します、または、以前「両端揃え」に設定されていた場合属性をクリアします。

#### 4.2.16.3 AuthenticXMLTableCommands.AlignHorizontalLeft

**宣言:** [AlignHorizontalLeft\(\)](#)

**説明**

垂直方向の配置の属性を「左側」に設定します、または、以前「左側」に設定されていた場合属性をクリアします。

#### 4.2.16.4 AuthenticXMLTableCommands.AlignHorizontalRight

**宣言:** [AlignHorizontalRight\(\)](#)

**説明**

垂直方向の配置の属性を「右側」に設定します、または、以前「右側」に設定されていた場合属性をクリアします。

#### 4.2.16.5 AuthenticXMLTableCommands.AlignVerticalBottom

**宣言:** [AlignVerticalBottom\(\)](#)

**説明**

垂直方向の配置の属性を「下部」に設定します、または、以前に「下部」に設定されていた場合属性をクリアします。

#### 4.2.16.6 AuthenticXMLTableCommands.AlignVerticalCenter

**宣言:** [AlignVerticalCenter\(\)](#)

**説明**

垂直方向の配置の属性を「中央」に設定します、または、以前に「中央」に設定されていた場合属性をクリアします。

#### 4.2.16.7 AuthenticXMLTableCommands.AlignVerticalTop

**宣言:** [AlignVerticalTop\(\)](#)

**説明**

垂直方向の配置の属性を「上部」に設定します、または、以前に「上部」に設定されていた場合属性をクリアします。

#### 4.2.16.8 AuthenticXMLTableCommands.AppendCol

**宣言:** [AppendCol\(\)](#)

**説明**

現在のテーブルに列を追加します。

#### 4.2.16.9 AuthenticXMLTableCommands.AppendRow

**宣言:** [AppendRow\(\)](#)

**説明**

現在のテーブルに行を追加します。

#### 4.2.16.10 AuthenticXMLTableCommands.Delete

**宣言:** [Delete\(\)](#)

**説明**

ユーザーがダイアログ メソッドを確認すると、現在選択されているテーブルが削除されます。

#### 4.2.16.11 AuthenticXMLTableCommands.DeleteCol

**宣言:** `DeleteCol()`

**説明**

メソッドは現在選択されている列を削除します。  
列が一列しか残されておらず、ユーザーがダイアログを確認すると、テーブル全体が削除されます。

#### 4.2.16.12 AuthenticXMLTableCommands.DeleteRow

**宣言:** `DeleteRow()`

**説明**

メソッドは現在選択されている行を削除します。  
列が一列しか残されておらず、ユーザーがダイアログを確認すると、テーブル全体が削除されます。

#### 4.2.16.13 AuthenticXMLTableCommands.EditProperties

**宣言:** `EditProperties()`

**説明**

選択されたテーブルセルのためプロパティダイアログを表示します。

#### 4.2.16.14 AuthenticXMLTableCommands.Insert

**宣言:** `Insert()`

**説明**

ダイアログを表示し、既存のXMLに新規のテーブルを挿入します。

#### 4.2.16.15 AuthenticXMLTableCommands.InsertCol

**宣言:** `InsertCol()`

**説明**

現在選択されている列の前に新規の列が挿入されます。

#### 4.2.16.16 AuthenticXMLTableCommands.InsertRow

**宣言:** `InsertRow()`

**説明**

現在選択されている行の前に行を挿入します。

#### 4.2.16.17 AuthenticXMLTableCommands.JoinDown

**宣言:** `JoinDown()`

**説明**

現在のセルをすぐ下のセルにジョインします。

#### 4.2.16.18 AuthenticXMLTableCommands.JoinLeft

**宣言:** `JoinLeft()`

**説明**

現在のセルをすぐ左のセルにジョインします。

#### 4.2.16.19 AuthenticXMLTableCommands.JoinRight

**宣言:** `JoinRight()`

**説明**

現在のセルをすぐ右のセルにジョインします。

#### 4.2.16.20 AuthenticXMLTableCommands.JoinUp

**宣言:** `JoinUp()`

**説明**

現在のセルをすぐ上のセルにジョインします。

#### 4.2.16.21 AuthenticXMLTableCommands.MayAlignHorizontal

**宣言:** `MayAlignHorizontal` を **ブール値** として

**説明**

現在のセルのために水平方向の配置の属性を設定することができる場合、true です。

#### 4.2.16.22 AuthenticXMLTableCommands.MayAlignVertical

**宣言:** `MayAlignVertical` を **ブール値** として

**説明**

現在のセルのために垂直方向の配置の属性を設定することができる場合、true です。

#### 4.2.16.23 AuthenticXMLTableCommands.MayAppendCol

**宣言:** `MayAppendCol` を **ブール値** として

**説明**

列を追加することができる場合、True です。

#### 4.2.16.24 AuthenticXMLTableCommands.MayAppendRow

**宣言:** `MayAppendRow` を **ブール値** として

**説明**

行の追加が可能な場合、true です。

#### 4.2.16.25 AuthenticXMLTableCommands.MayDelete

**宣言:** `MayDelete` を **ブール値** として

**説明**

テーブルが選択されている場合、True です。

#### 4.2.16.26 AuthenticXMLTableCommands.MayDeleteCol

**宣言:** `MayDeleteCol` を **ブール値** として

**説明**

現在の列を削除できる場合、True です。

#### 4.2.16.27 AuthenticXMLTableCommands.MayDeleteRow

**宣言:** `MayDeleteRow` を **ブール値** として

**説明**

現在の行を削除できる場合、True です。

#### 4.2.16.28 AuthenticXMLTableCommands.MayEditProperties

**宣言:** `MayEditProperties` を **ブール値** として

**説明**

選択されたテーブルセルのためプロパティダイアログが使用できる場合、プロパティは True です。

#### 4.2.16.29 AuthenticXMLTableCommands.MayInsert

**宣言:** `MayInsert` を **ブール値** として

**説明**

現在の選択箇所での新規のテーブルの挿入が可能な場合、プロパティは True です。

#### 4.2.16.30 AuthenticXMLTableCommands.MayInsertCol

**宣言:** `MayInsertCol` を **ブール値** として

**説明**

列を挿入することができる場合、True です。

#### 4.2.16.31 AuthenticXMLTableCommands.MayInsertRow

**宣言:** `MayInsertRow` を **ブール値** として

**説明**

行を挿入することができる場合、True です。

#### 4.2.16.32 AuthenticXMLTableCommands.MayJoinDown

**宣言:** `MayJoinDown` を **ブール値** として

**説明**

現在選択されているセルのため下方向のジョイン操作が可能な場合、プロパティは True です。

#### 4.2.16.33 AuthenticXMLTableCommands.MayJoinLeft

**宣言:** `MayJoinLeft` を **ブール値** として

**説明**

現在選択されているセルのため左方向のジョイン操作が可能な場合、プロパティは True です。

#### 4.2.16.34 AuthenticXMLTableCommands.MayJoinRight

**宣言:** `MayJoinRight` を **ブール値** として

**説明**

現在選択されているセルのため右方向のジョイン操作が可能な場合、プロパティは True です。

#### 4.2.16.35 AuthenticXMLTableCommands.MayJoinUp

**宣言:** `MayJoinUp` を **ブール値** として

**説明**

現在選択されているセルのため上方向のジョイン操作が可能な場合、プロパティは True です。

#### 4.2.16.36 AuthenticXMLTableCommands.MaySplitHorizontal

**宣言:** `MaySplitHorizontal` を **ブール値** として

**説明**

セルを水平に分割できる場合、True です。

#### 4.2.16.37 AuthenticXMLTableCommands.MaySplitVertical

**宣言:** `MaySplitVertical` を **ブール値** として

**説明**

セルを垂直に分割できる場合、True です。

#### 4.2.16.38 AuthenticXMLTableCommands.SplitHorizontal

**宣言:** [SplitHorizontal\(\)](#)

**説明**

メソッドは現在のセルを水平に分割します。

#### 4.2.16.39 AuthenticXMLTableCommands.SplitVertical

**宣言:** [SplitVertical\(\)](#)

**説明**

メソッドは現在のセルを垂直に分割します。

### 4.2.17 XMLData

次も参照してください

**メソッド**

[InsertChild](#)

[AppendChild](#)

[EraseAllChildren](#)

[EraseCurrentChild](#)

[GetCurrentChild](#)

[GetFirstChild](#)

[GetNextChild](#)

[GetChild](#)

[GetChildKind](#)

[IsSameNode](#)

[HasChildrenKind](#)

[CountChildren](#)

[CountChildrenKind](#)

**プロパティ**

[Name](#)

[TextValue](#)

[HasChildren](#)

[MayHaveChildren](#)

[Kind](#)

[Parent](#)

### 説明

XMLData インターフェイスを使用して現在表示されているXMLのエレメントを操作することができます。このインターフェイスは内部のプラグインとXMLSpy 自身を使用する実装の軽量のCOM カウンターパートです。

新規のXMLData オブジェクトを作成する場合は、プラグイン インターフェイスのCreateChild() メソッドを使用します。

## 4.2.17.1 XMLData.AppendChild

次も参照してください！

**宣言:** `AppendChild(pNewData` を [XMLData](#) として

### 説明

AppendChild はpNewData をXMLData オブジェクトの最後の子として追加します。次も参照してください！「[XMLData の使用](#)」。

### サンプル

```
Dim objCurrentParent
Dim objNewChild

Set objNewChild = objPlugIn.CreateChild(spyXMLDataElement)
Set objCurrentParent = objPlugIn.XMLRoot

objCurrentParent.AppendChild objNewChild

Set objNewChild = Nothing
```

## 4.2.17.2 XMLData.CountChildren

次も参照してください！

**宣言:** `CountChildren` long として

### 説明

CountChildren は子の数量を取得します。

TypeLibrary バージョン 1.6 を使用すると使用できるようになります。

### エラー

1500 XMLData オブジェクトは有効ではありません。

## 4.2.17.3 XMLData.CountChildrenKind

次も参照してください！

**宣言:** [CountChildrenKind](#) (*nKind* を [SPYXMLDataKind](#) として long として

#### 説明

[CountChildrenKind](#) は固有の種類の子の数量を取得します。

TypeLibrary バージョン 1.6 を使用すると使用できるようになります。

#### エラー

1500 XMLData オブジェクトは有効ではありません。

### 4.2.17.4 XMLData.EraseAllChildren

次も参照してください:

**宣言:** [EraseAllChildren](#)

#### 説明

[EraseAllChildren](#) はXMLData オブジェクトの関連した子を全て削除します。

#### サンプル

サンプルはアクティブなドキュメントの全ての要素を削除します。

```
Dim objCurrentParent

Set objCurrentParent = objPlugIn.XMLRoot
objCurrentParent.EraseAllChildren
```

### 4.2.17.5 XMLData.EraseChild

**メソッド:** [EraseChild](#) (子ノードと新規のノードを [XMLData](#) として

#### 説明

指定されている子ノードを削除します。

#### エラー

1500 無効なオブジェクトです。  
1506 無効な入力 XML  
1510 無効なパラメーターです。

### 4.2.17.6 XMLData.EraseCurrentChild

次も参照してください:

**宣言:** [EraseCurrentChild](#)

## 説明

EraseCurrentChild は現在のXMLData 子オブジェクトを削除します。EraseCurrentChild を呼び出す前に [XMLData.GetChild](#) を使用して内部の反復子を初期化する必要があります。

## サンプル

このJavaScript サンプルは、名前 "EraseMe" を持つ全ての要素を削除します。同じループ内で子要素をステップスルーするために EraseCurrentChild と GetNextChild を呼び出すことができることをコードは表示しています。

```
function DeleteXMLElements(objXMLData)
{
    if(objXMLData == null)
        return;

    if(objXMLData.HasChildren) {
        var objChild;
        objChild = objXMLData.GetFirstChild(-1);

        while(objChild) {
            DeleteXMLElements(objChild);

            try {
                if(objChild.Name == "EraseMe")
                    objXMLData.EraseCurrentChild();

                objChild = objXMLData.GetNextChild();
            }
            catch(Err) {
                objChild = null;
            }
        }
    }
}
```

### 4.2.17.7 XMLData.GetChild

次も参照してください！

**宣言:** `GetChild` (*position* long として を [XMLData](#) として

#### 戻り値

XML 要素を XMLData オブジェクトとして返します。

#### 説明

`GetChild()` は指定されているインデックス(ゼロベース)で子への参照を返します。

TypeLibrary バージョン 1.6 を使用すると使用できるようになります。

#### エラー

- 1500 XMLData オブジェクトは有効ではありません。
- 1510 戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.17.8 XMLData.GetChildAttribute

**メソッド:** `GetChildAttribute` (Name を文字列として XMLData オブジェクトとして(エラー時はNULL))

##### 説明

指定されている名前を持つ属性を抽出します。

##### エラー

- 1500 無効なオブジェクトです。
- 1510 無効なパラメータです。

#### 4.2.17.9 XMLData.GetChildElement

**メソッド:** `GetChildElement` (Name を文字列として、整数としてのポジション) XMLData オブジェクトとして(エラー時はNULL)

##### 説明

指定されている名前を持つNth 子要素を抽出します。

##### エラー

- 1500 無効なオブジェクトです。
- 1510 無効なパラメータです。

#### 4.2.17.10 XMLData.GetChildKind

次も参照してください！

**宣言:** `GetChildKind` (*position* long として, *nKind* を [SPYXMLDataKind](#) として) を [XMLData](#) として

##### 戻り値

XML 要素を XMLData オブジェクトとして返します。

##### 説明

`GetChildKind()` は、指定されているインデックス(ゼロベースの)でこの型の子への参照を返します。ポジションパラメータは全てのオブジェクトの子でなく、特定の型の子の数量に対して相対的です。

TypeLibrary バージョン 1.6 を使用すると使用できるようになります。

##### エラー

- 1500 XMLData オブジェクトは有効ではありません。
- 1510 戻り値パラメータのために無効なアドレスが指定されています。

#### 4.2.17.11 XMLData.GetCurrentChild

次も参照してください！

**宣言:** `GetCurrentChild` を [XMLData](#) として

**戻り値**

XML 要素を [XMLData](#) オブジェクトとして返します。

**説明**

`GetCurrentChild` は現在の子を取得します。`GetCurrentChild` を呼び出す前に [XMLData.GetFirstChild](#) を使用して内部反復子を初期化する必要があります。

#### 4.2.17.12 XMLData.GetFirstChild

次も参照してください！

**宣言:** `GetFirstChild(nKind)` を [SPYXMLDataKind](#) として を [XMLData](#) として

**戻り値**

XML 要素を [XMLData](#) オブジェクトとして返します。

**説明**

`GetFirstChild` は新規の反復子を初期化し最初の子を返します。`nKind = -1` に設定し、この全ての型の反復子を取得します。

**サンプル**

次のサンプルを確認してください！ [XMLData.GetNextChild](#).

#### 4.2.17.13 XMLData.GetNamespacePrefixForURI

**メソッド:** `GetNamespacePrefixForURI` (URI を文字列として) Prefix を文字列として

**説明**

与えられた URI の名前空間プレフィックスを返します。

**エラー**

- 1500 無効なオブジェクトです。
- 1510 無効なパラメーターです。

#### 4.2.17.14 XMLData.GetNextChild

次も参照してください！

**宣言:** `GetNextChild` を `XMLData` として

#### 戻り値

XML 要素を `XMLData` オブジェクトとして返します。

#### 説明

`GetNextChild` は、この要素の次の子に移動します。 `GetNextChild` を呼び出す前に `XMLData.GetFirstChild` を使用して内部の反復子を初期化します。

下のサンプル内の要素の最後の子をチェックします。

#### サンプル

```
On Error Resume Next
Set objParent = objPlugIn.XMLRoot

'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)
```

### 4.2.17.15 XMLData.GetTextValueXMLDecoded

**メソッド:** `GetTextValueXMLDecoded` を文字列として

#### 説明

XML のデコードされたテキストの値を取得します。

#### エラー

- 1500 無効なオブジェクトです。
- 1510 無効なパラメーターです。

### 4.2.17.16 XMLData.HasChildren

次も参照してください:

**宣言:** `HasChildren` を `ブール値` として

#### 説明

他の `XMLData` オブジェクトの親がオブジェクトの場合、プロパティは `True` です。

このプロパティは読み取り専用です。

#### 4.2.17.17 XMLData.HasChildrenKind

次も参照してください！

**宣言:** `HasChildrenKind` (`nKind` を [SPYXMLDataKind](#) として ブール値として

##### 説明

オブジェクトが固有の型の親の場合、メソッドは True を返します。

TypeLibrary バージョン 1.6 を使用すると使用できるようになります。

##### エラー

- 1500 XMLData オブジェクトは有効ではありません。
- 1510 戻り値/パラメーターのために無効なアドレスが指定されています。

#### 4.2.17.18 XMLData.InsertChild

次も参照してください！

**宣言:** `InsertChild`(`pNewData` を [XMLData](#) として

##### 説明

`InsertChild` は現在の子の前に新規の子を挿入します (次も参照してください！現在の子を設定するための [XMLData.GetFirstChild](#)、[XMLData.GetNextChild](#))。

#### 4.2.17.19 XMLData.InsertChildAfter

**メソッド:** `InsertChildAfter` (子ノードと新規ノードを [XMLData](#) として

##### 説明

新規のXMLノードをあるノードの後に挿入します

##### エラー

- 1500 無効なオブジェクトです。
- 1506 無効な入力 XML
- 1507 子は許可されていません。
- 1510 無効なパラメーターです。
- 1512 子は既に追加されています。
- 1514 ポジションでは無効な型

#### 4.2.17.20 XMLData.InsertChildBefore

**メソッド:** `InsertChildBefore` (子ノードと新規ノードを [XMLData](#) として

**説明**

新規のXML ノードをあるノードの前には挿入します

**エラー**

- 1500 無効なオブジェクトです。
- 1506 無効な入力 XML
- 1507 子は許可されていません。
- 1510 無効なラメーターです。
- 1512 子は既に追加されています。
- 1514 ポジションでは無効な型

#### 4.2.17.21 XMLData.IsSameNode

次も参照してください！

**宣言:** `IsSameNode(pNodeToCompare)` を [XMLData](#) としてを **ブール値** として

**説明**

`pNodeToCompare` がオブジェクト自身と同じノードを参照する場合、True が返されます。

#### 4.2.17.22 XMLData.Kind

次も参照してください！

**宣言:** `Kind` を [SPYXMLDataKind](#) として

**説明**

このXMLData オブジェクトの種類です。

このプロパティは読み取り専用です。

#### 4.2.17.23 XMLData.MayHaveChildren

次も参照してください！

**宣言:** `MayHaveChildren` を **ブール値** として

**説明**

このXMLData オブジェクトへの子の追加が許可されているかを通知します。

このプロパティは読み取り専用です。

#### 4.2.17.24 XMLData.Name

次も参照してください！

宣言: `Name` を `文字列` として

##### 説明

XMLData オブジェクトの名前を変更し取得するために使用されます。

#### 4.2.17.25 XMLData.Parent

次も参照してください！

宣言: `Parent` を `XMLData` として

##### 戻り値

親 XMLData オブジェクト

親要素が存在しない場合、何も返されません(または NULL が返されます)。

##### 説明

この要素の親です。

このプロパティは読み取り専用です。

#### 4.2.17.26 XMLData.SetTextValueXMLDecoded

メソッド: `SetTextValueXMLDecoded` (文字列の値)

##### 説明

XML のエンコードされたテキストの値を設定します。

##### エラー

1500 無効なオブジェクトです。

1513 変更は許可されていません。

#### 4.2.17.27 XMLData.TextValue

次も参照してください！

宣言: `TextValue` を `文字列` として

##### 説明

このXMLData オブジェクトのテキストの値を変更し取得するために使用されます。次も参照してください！「[XMLData の使用](#)」。

## 4.3 列挙

このセクションには、Authentic Browser 列挙のリストと説明が含まれています。

### 4.3.1 SPYAuthenticActions

#### 説明

[AuthenticRange](#) オブジェクト上で実行することのできるアクションです。

#### 可能な値:

spyAuthenticInsertAt	= 0
spyAuthenticApply	= 1
spyAuthenticClearSurr	= 2
spyAuthenticAppend	= 3
spyAuthenticInsertBefore	= 4
spyAuthenticRemove	= 5

### 4.3.2 SPYAuthenticCommand

#### 説明

使用することのできる全てのコマンドの列挙です。

#### 可能な値:

```
// CommandGroupMain
  k_CommandSeparator = 0
  k_CommandSave      = 1
  k_CommandPrint     = 2
  k_CommandPrintPreview = 3
  k_CommandValidate  = 4
  k_CommandUndo      = 5
  k_CommandRedo      = 6

// CommandGroupEdit
  k_CommandEditCut      = 7
  k_CommandEditCopy    = 8
  k_CommandEditPaste   = 9
  k_CommandEditFind    = 10
  k_CommandEditRepeat  = 11
  k_CommandEditReplace = 12

// CommandGroupMarkup
  k_CommandMarkupHide  = 13
  k_CommandMarkupLarge = 14

// CommandGroupRow
  k_CommandRowAppend   = 15
  k_CommandRowInsert  = 16
```

```

k_CommandRowDuplicate = 17
k_CommandRowMoveUp   = 18
k_CommandRowMoveDown = 19
k_CommandRowDelete   = 20

// CommandGroupXMLTables
k_CommandXMLTableInsert      = 21
k_CommandXMLTableDelete     = 22
k_CommandXMLTableAppendRow  = 23
k_CommandXMLTableInsertRow  = 24
k_CommandXMLTableDeleteRow  = 25
k_CommandXMLTableAppendCol  = 26
k_CommandXMLTableInsertCol  = 27
k_CommandXMLTableDeleteCol  = 28
k_CommandXMLTableJoinRight  = 29
k_CommandXMLTableJoinLeft   = 30
k_CommandXMLTableJoinUp     = 31
k_CommandXMLTableJoinDown   = 32
k_CommandXMLTableSplitHorz  = 33
k_CommandXMLTableSplitVert  = 34
k_CommandXMLTableVAlignTop  = 35
k_CommandXMLTableVAlignCenter = 36
k_CommandXMLTableVAlignBottom = 37
k_CommandXMLTableAlignLeft  = 38
k_CommandXMLTableAlignCenter = 39
k_CommandXMLTableAlignRight = 40
k_CommandXMLTableAlignJustify = 41
k_CommandXMLTableEditProperties = 42

// since TypeLib 1.2
k_CommandCheckSpelling      = 43
k_CommandAbout              = 44
k_CommandPackageManagement  = 45

```

### 4.3.3 SPYAuthenticCommandGroup

#### 説明

コマンドが所属することのできるグループです。

#### 可能な値:

```

k_CommandGroupMain      = 0
k_CommandGroupEdit     = 1
k_CommandGroupMarkup    = 2
k_CommandGroupRow      = 3
k_CommandGroupXMLTables = 4

```

### 4.3.4 SPYAuthenticDocumentPosition

#### 説明

相対的および絶対的な位置を [AuthenticRange](#) オブジェクトを使用してナビゲートするために使用されます。

可能な値

```

spyAuthenticDocumentBegin = 0
spyAuthenticDocumentEnd   = 1
spyAuthenticRangeBegin    = 2
spyAuthenticRangeEnd      = 3

```

### 4.3.5 SPYAuthenticElementActions

**説明**

[GetAllowedElements](#) と使用することのできるアクション

可能な値

```

k_ActionInsertAt           = 0
k_ActionApply              = 1
k_ActionClearSurr         = 2
k_ActionAppend             = 3
k_ActionInsertBefore       = 4
k_ActionRemove            = 5

```

### 4.3.6 SPYAuthenticElementKind

**説明**

[AuthenticRange](#) と [AuthenticView](#) オブジェクト内のナビゲーションと選択のために使用される要素の異なる種類の列挙です。

可能な値

```

spyAuthenticChar           = 0
spyAuthenticWord           = 1
spyAuthenticLine           = 3
spyAuthenticParagraph      = 4
spyAuthenticTag            = 6
spyAuthenticDocument       = 8
spyAuthenticTable          = 9
spyAuthenticTableRow       = 10
spyAuthenticTableColumn    = 11

```

### 4.3.7 SPYAuthenticEntryHelperWindows

**説明**

入力ヘルパーウィンドウで使用可能な ID です。

可能な値

```

k_Elements                 = 1
k_Attributes                = 2
k_Entities                  = 4

```

### 4.3.8 SPYAuthenticMarkupVisibility

#### 説明

マークアップの可視性をカスタマイズするための列挙値です。

#### 可能な値

spyAuthenticMarkupHidden	= 0
spyAuthenticMarkupSmall	= 1
spyAuthenticMarkupLarge	= 2
spyAuthenticMarkupMixed	= 3

### 4.3.9 SPYAuthenticToolbarAlignment

#### 説明

ツールバーの配置を指定するための値です。

#### 可能な値

k_ToolbarAlignTop	= 0
k_ToolbarAlignLeft	= 1
k_ToolbarAlignBottom	= 2
k_ToolbarAlignRight	= 3

### 4.3.10 SPYAuthenticToolbarButtonState

#### 説明

Authentic ツールバーボタンの状態は次の列挙により与えられます。

#### 可能な値

authenticToolbarButtonDefault	= 0
authenticToolbarButtonEnabled	= 1
authenticToolbarButtonDisabled	= 2

### 4.3.11 SPYXMLDataKind

#### 説明

XML ドキュメントのために使用することができるXMLData 要素の異なる型。

#### 可能な値

spyXMLDataXMLDocStruct	= 0
spyXMLDataXMLEntityDocStruct	= 1
spyXMLDataDTDDocStruct	= 2
spyXMLDataXML	= 3
spyXMLDataElement	= 4

---

spyXMLDataAttr	= 5
spyXMLDataText	= 6
spyXMLDataCDATA	= 7
spyXMLDataComment	= 8
spyXMLDataP	= 9
spyXMLDataDefDoctype	= 10
spyXMLDataDefExternalID	= 11
spyXMLDataDefElement	= 12
spyXMLDataDefAttlist	= 13
spyXMLDataDefEntity	= 14
spyXMLDataDefNotation	= 15
spyXMLDataKindsCount	= 16

## 5 ASP.NET Web アプリケーション

Altova® Authentic® Browser Edition のデPLOYを簡素化するために、Visual Studio .NET へ完全に統合するASP.NET サーバー管理、および、Web 書式にAuthentic Browser をドラッグアンドドロップする機能がASP.NET デベロッパーに提供されます。

### ASP.NET サーバー管理の使用方法

1. [Authentic Browser プラグインのためのASP.NET サーバー管理](#) を Altova Web サイトからダウンロードします。
2. Visual Studio を開始し、Visual Studio .NET 内のアセンブリ(ダウンロード済みパッケージ) をツールボックスに追加して登録します。これは「ツール | ツールボックスアイテムの選択」コマンドを選択し、and, in ツールボックスアイテムダイアログ内で、.NET フレームワークコンポーネントを選択し、(Altova:Authentic.WebControls.dll) アセンブリを参照することによりおこなうことができます。
3. AuthenticDocumentView コントロールはツールボックスの全般ペインに表示されます。デザインビューはこのコントロールをドラッグし、必要に応じてサイズ調整または変更します(コントロールはクライアント側のWeb ページ Authentic Browser を呼び出します)。
4. プラグインに関連したファイル(スキーマ、SPS、XML、イメージファイルなど) をプロジェクトに追加します。Solution Explorer を使用して新規のフォルダーを Visual Studio プロジェクトに追加して、プラグインに関連したファイルに新規のプロジェクトフォルダーを追加することもできます。
5. プラグインに関連したファイルのそれぞれに対して、次のプロパティを設定します: ビルドアクション = コンテンツと出力ディレクトリにコピーする = 新しいバージョンが存在する場合、はい。
6. AuthenticDocumentView コントロールのため次のプロパティを設定します: (i) 信頼されている信頼されていないバージョンを選択し、(ii) SPSに関連したリソースを追加します (SchemaDataURL、SPSDataURL、XMLDataURL) (iii) Enterprise エディションの場合、ライセンス情報を追加します。
7. IIS バージョン 7.0 または以降に関しては、SPS ファイルのための静的なコンテンツレンドラーを追加します。web.config ファイルに以下のラインを追加します:

```
<system.webServer>
  <staticContent>
    <mimeMap fileExtension=".sps" mimeType="text/xml" />
  </staticContent>
</system.webServer>
```

### サンプルASP.NET プロジェクト

QuickStartApp1 と呼ばれる Visual Studio 2008 のためのサンプルASP.NET Web アプリケーションプロジェクトはASP.NET サーバー管理 パッケージと共に含まれています。圧縮されていないパッケージのサンプルフォルダーで見つけることができます。

## 6 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。

## 6.1 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

## A

- AuthenticRange オブジェクト, 55
- Authentic, 91
  - ApplyTextState, 73
  - attachCallBack, 74
  - ControlIntialized, 76
  - CreateChild, 77
  - CurrentSelection, 77
  - DesignDataLoadObject, 78
  - EditClear, 78
  - EditCopy, 78
  - EditCut, 78
  - EditPaste, 79
  - EditRedo, 79
  - EditSelectAll, 79
  - EditUndo, 79
  - event, 81
  - FindDialog, 81
  - FindNext, 81
  - GetAllAttributes, 81
  - GetAllowedElements, 83
  - GetFileVersion, 84
  - GetNextVisible, 85
  - GetPreviousVisible, 85
  - IsEditClearEnabled, 85
  - IsEditCopyEnabled, 85
  - IsEditCutEnabled, 86
  - IsEditPasteEnabled, 86
  - IsEditRedoEnabled, 86
  - IsEditUndoEnabled, 86
  - IsFindNextEnabled, 87
  - IsRowAppendEnabled, 87
  - IsRowDeleteEnabled, 87
  - IsRowDuplicateEnabled, 87
  - IsRowInsertEnabled, 88
  - IsRowMoveDownEnabled, 88
  - IsRowMoveUpEnabled, 88
  - IsTextStateApplied, 88
  - IsTextStateEnabled, 89
  - LoadXML, 89
  - MarkupView, 89
  - Print, 89
  - PrintPreview, 90
  - ReplaceDialog, 90
  - Reset, 91
  - RowAppend, 91
  - RowDelete, 91
  - RowInsert, 92
  - RowMoveDown, 92
  - RowMoveUp, 92
  - Save, 92
  - SavePOST, 93
  - SaveXML, 94
  - SchemaLoadObject, 94
  - SelectionChanged, 94
  - SelectionMoveTabOrder, 95
  - SelectionSet, 95
  - StartEditing, 95
  - ValidateDocument, 97
  - validationBadData, 98
  - validationMessage, 98
  - XMLDataLoadObject, 98
  - XMLDataSaveUrl, 98
  - XMLRoot, 99
- Authentic Browser, 71
  - MIME 型, 9, 30
  - イベントの処理, 24
  - クラス ID, 9, 21
  - サーバーにダウンロードされたファイル, 9, 12, 41
  - サブルーチン, 24
  - と DB-ベースの SPS, 17
  - ネットワークのセットアップ, 8
  - バージョン, 9, 21, 30
  - プラグイン ファイル, 9, 12, 41
  - 概要, 8
  - 利点, 7
- Authentic RowDuplicate, 91
- Authentic オブジェクト, 55
- AuthenticDataTransfer,
  - dropEffect, 103
  - getData, 103
  - ownDrag, 103
  - 型, 103
- AuthenticEvent,
  - altKey, 105
  - altLeft, 105
  - button, 105
  - cancelBubble, 105
  - clientX, 106

**AuthenticEvent,**

clientY, 106  
ctrlKey, 106  
ctrlLeft, 106  
dataTransfer, 107  
fromElement, 107  
keyCode, 107  
propertyName, 107  
repeat, 107  
returnValue, 108  
shiftKey, 108  
shiftLeft, 108  
srcElement, 108  
type, 109

**AuthenticRange, 112**

AppendRow, 114  
Application, 114  
CanPerformAction, 115  
CanPerformActionWith, 115  
Close, 116  
CollapsToBegin, 116  
CollapsToEnd, 116  
Copy, 117  
Cut, 117  
Delete, 117  
DeleteRow, 118  
DuplicateRow, 118  
ExpandTo, 119  
FirstTextPosition, 119  
FirstXMLData, 120  
FirstXMLDataOffset, 121  
GetElementAttributeNames, 122  
GetElementAttributeValue, 122  
GetElementHierarchy, 123  
GetEntityNames, 123  
Goto, 124  
GotoNext, 124  
GotoNextCursorPosition, 125  
GotoPrevious, 125  
GotoPreviousCursorPosition, 126  
HasElementAttribute, 126  
InsertEntity, 127  
InsertRow, 127  
IsEmpty, 129  
IsEqual, 129  
IsInDynamicTable, 130  
LastTextPosition, 131  
LastXMLData, 132

LastXMLDataOffset, 132

MoveBegin, 133  
MoveEnd, 134  
MoveRowDown, 134  
MoveRowUp, 134  
Parent, 135  
Paste, 135  
PerformAction, 135  
Select, 136  
SelectNext, 137  
SelectPrevious, 137  
SetElementAttributeValue, 138  
SetFromRange, 139  
Text, 140

**AuthenticView, 75, 146, 164**

Application, 156  
DocumentBegin, 158  
DocumentEnd, 158  
Goto, 160  
MarkupVisibility, 162  
Parent, 162  
Print, 162  
Redo, 163  
Selection, 163  
Undo, 164  
WholeDocument, 165

**AuthenticView オブジェクト, 55****C**

CAB ファイル, 9, 12, 41  
CODEBASE 属性, 21  
ControlInitialized, 52, 76

**D**

DB-ベースの SPS,  
Authentic Browser のための必要条件, 17  
DOM,  
と XMLData, 66

## E

EMBED 要素,  
IE の HTML ページ, 30  
Enterprise Edition のためのリリース, 20

## F

Firefox のための HTML ページ,  
EMBED 要素, 30  
sorting-a-table サンプル, 34  
イベントリスナーの追加, 32  
シンプルなサンプル, 33  
概要, 29

## H

HTML ページ,  
概要, 19

## I

IE または Firefox のための HTML ページ,  
サンプルファイル, 37  
概要, 37  
IE のための HTML ページ,  
OBJECT 要素, 21  
SCRIPT 要素, 24  
シンプルなサンプル, 25  
テーブルの並べ替えサンプル, 27  
概要, 21

## M

MIME 型, 9, 30

## O

OBJECT 要素,  
IE の HTML ページ, 21

## P

Packages, 57

## S

SCRIPT 要素,  
IE の HTML ページ, 24  
Selection changed, 53  
selectionchanged, 52  
Shortcut keys., 56  
Spell-checking, 57

## X

XMLData, 63  
  GetChild, 177  
  GetChildKind, 178  
  HasChildrenKind, 181  
  IsSameNode, 182  
  と DOM, 66  
XMLSpyLib,  
  AuthenticDataTransfer, 102  
  AuthenticEvent, 104  
  XMLSpyXMLData, 174  
XMLSPYPLUGINLib,  
  Authentic, 71  
  XMLSpyXMLLoadSave, 112  
XMLSpyXMLData,  
  AppendChild, 175  
  EraseAllChildren, 176  
  EraseCurrentChild, 176  
  GetCurrentChild, 179  
  GetFirstChild, 179  
  GetNextChild, 179

XMLSpyXMLData,  
  HasChildren, 180  
  InsertChild, 181  
  Kind, 182  
  MayHaveChildren, 182  
  Name, 183  
  Parent, 183  
  TextValue, 183  
XMLSpyXMLLoadSave,  
  String, 112  
  URL, 112  
XPI ファイル, 9, 12, 41

入力ヘルパー, 57  
配布,  
  Altova ソフトウェア製品, 190  
評価機関,  
  Altova ソフトウェア製品, 190  
編集の操作, 55  
法的な情報, 190

## Z

イベント,  
  レファレンス, 54  
イベントの処理, 24  
イベントハンドラー, 52, 54  
イベントリスナー (firefox), 53  
エンドユーザー使用許諾契約書, 190  
クラス ID, 9, 21  
コンテンツ,  
  アクセスと変更, 55  
サーバーのセットアップ, 12  
サーバーのためのインターネット情報サービス, 13  
サーバーのためのブラウザサービス, 13  
サブルーチン, 24  
システムの必要条件, 8  
ツールバーボタン,  
  振る舞いの変更, 54  
テキストの状態ボタン, 57  
ドキュメントコンテンツ,  
  アクセスと変更, 55  
ネットワークのセットアップ, 8  
ユーザー レファレンス, 51  
ライセンス,  
  情報, 190  
レファレンス,  
  イベント, 54  
検索と置換, 56  
行オペレーション, 56  
接続ポイント イベント, 52  
置換, 56  
著作権に関する情報, 190  
動的なテーブル, 56