

Benutzer- und Referenzhandbuch



Copyright © 1998-2005 Altova GmbH. All rights reserved. Use of this software is governed by and subject to an Altova software license agreement. XMLSpy, StyleVision, Authentic, MapForce and ALTOVA are trademarks (registered in numerous countries) of Altova GmbH.

XML, XSL, XHTML, and W3C are trademarks (registered in numerous countries) of the World Wide Web Consortium; marks of the W3C are registered and held by its host institutions, MIT, INRIA, and Keio. UNICODE and the Unicode Logo are trademarks of Unicode, Inc. This software contains material that is ©1994-1998 Dundas Software Ltd., all rights reserved. The Sentry Spelling Checker Engine - ©2000 Wintertree Software Inc., STLport ©1999,2000 Boris Fomitchev, ©1994 Hewlett-Packard Company, ©1996,97 Silicon Graphics Computer Systems, Inc., ©1997 Moscow Center for SPARC Technology. Scintilla Copyright ©1996-2002 by Neil Hodgson <neilh@scintilla.org>. All other names or trademarks are the property of their respective owners.

ALTOVA®

Altova Authentic 2005 Browser Edition

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

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

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

Published: 2005

© 2005 Altova GmbH 1998-2005 Altova GmbH.

Printed in the United States of America

5 4 3 2 1

ISBN 1-933210-03-6

Inhaltsverzeichnis

1	Einführung	1
1.1	Installation der Authentic 2005 Browser Edition	3
1.2	Herunterladen der Authentic 2005 Browser Edition auf den Server	4
1.3	Voraussetzungen für DB-basierte SPS-Dateien	5
2	Die HTML-Seite für Authentic	8
2.1	Das OBJECT-Element	9
2.2	Das SCRIPT-Element	12
2.3	HTML-Seite: Beispiel 1 (einfach)	13
2.4	HTML-Seite: Beispiel 2 (Tabelle mit bubble-sort sortieren)	15
2.5	HTML-Seite: Beispiel 4 (DB-Referenz)	18
3	Mechanismen	20
3.1	Ereignisse: Connection Point	21
3.2	Ereignisse: Symbolleisten-Schaltfläche	23
3.3	Aufrufen und Ändern von Dokumentinhalt	24
3.4	Bearbeitungsoperationen	25
3.5	Suchen und Ersetzen	26
3.6	Zeilenoperationen	27
3.7	Tastaturkürzel	28
3.8	Textstatus-Schaltflächen	29
3.9	Eingabehilfen	30
3.10	Zusatzpakete	31
3.11	Verwendung von XMLData	33
3.12	DOM und XMLData	37
4	Objekte	42
4.1	Authentic	42
4.1.1	Authentic.ApplyTextState	45
4.1.2	Authentic.attachCallBack	46

4.1.3	AuthenticView	48
4.1.4	Authentic.AutoHideUnusedCommandGroups	49
4.1.5	Authentic.BaseURL	50
4.1.6	Authentic.ClearSelection	51
4.1.7	Authentic.ClearUndoRedo	52
4.1.8	Authentic.ControlInitialized	53
4.1.9	Authentic.CreateChild	54
4.1.10	Authentic.CurrentSelection	55
4.1.11	Authentic.DesignDataLoadObject	56
4.1.12	Authentic.EditClear	57
4.1.13	Authentic.EditCopy	58
4.1.14	Authentic.EditCut	59
4.1.15	Authentic.EditPaste	60
4.1.16	Authentic.EditRedo	61
4.1.17	Authentic.EditSelectAll	62
4.1.18	Authentic.EditUndo	63
4.1.19	Authentic.EnableModifications	64
4.1.20	Authentic.EntryHelperAlignment	65
4.1.21	Authentic.EntryHelpersEnabled	66
4.1.22	Authentic.EntryHelperSize	67
4.1.23	Authentic.EntryHelperWindows	68
4.1.24	Authentic.event	69
4.1.25	Authentic.FindDialog	70
4.1.26	Authentic.FindNext	71
4.1.27	Authentic.GetAllAttributes	72
4.1.28	Authentic.GetAllowedElements	74
4.1.29	Authentic.GetFileVersion	76
4.1.30	Authentic.GetNextVisible	77
4.1.31	Authentic.GetPreviousVisible	78
4.1.32	Authentic.IsEditClearEnabled	79
4.1.33	Authentic.IsEditCopyEnabled	80
4.1.34	Authentic.IsEditCutEnabled	81
4.1.35	Authentic.IsEditPasteEnabled	82
4.1.36	Authentic.IsEditRedoEnabled	83
4.1.37	Authentic.IsEditUndoEnabled	84
4.1.38	Authentic.IsFindNextEnabled	85
4.1.39	Authentic.IsRowAppendEnabled	86
4.1.40	Authentic.IsRowDeleteEnabled	87
4.1.41	Authentic.IsRowDuplicateEnabled	88
4.1.42	Authentic.IsRowInsertEnabled	89
4.1.43	Authentic.IsRowMoveDownEnabled	90

4.1.44	Authentic.IsRowMoveUpEnabled	91
4.1.45	Authentic.IsTextStateApplied	92
4.1.46	Authentic.IsTextStateEnabled	93
4.1.47	Authentic.LoadXML	94
4.1.48	Authentic.MarkUpView	95
4.1.49	Authentic.Modified	96
4.1.50	Authentic.Print	97
4.1.51	Authentic.PrintPreview	98
4.1.52	Authentic.RedrawEntryHelpers	99
4.1.53	Authentic.ReloadToolbars	100
4.1.54	Authentic.ReplaceDialog	101
4.1.55	Authentic.Reset	102
4.1.56	Authentic.RowAppend	103
4.1.57	Authentic.RowDelete	104
4.1.58	Authentic.RowDuplicate	105
4.1.59	Authentic.RowInsert	106
4.1.60	Authentic.RowMoveDown	107
4.1.61	Authentic.RowMoveUp	108
4.1.62	Authentic.Save	109
4.1.63	Authentic.SaveButtonAutoEnable	110
4.1.64	Authentic.SavePOST	111
4.1.65	Authentic.SaveXML	112
4.1.66	Authentic.SchemaLoadObject	113
4.1.67	Authentic.SelectionChanged	114
4.1.68	Authentic.SelectionMoveTabOrder	115
4.1.69	Authentic.SelectionSet	116
4.1.70	Authentic.SetUnmodified	117
4.1.71	Authentic.StartEditing	118
4.1.72	Authentic.StartSpellChecking	119
4.1.73	Authentic.TextStateBmpURL	120
4.1.74	Authentic.TextStateToolbarLine	121
4.1.75	Authentic.ToolbarRows	122
4.1.76	Authentic.ToolbarsEnabled	123
4.1.77	Authentic.ToolbarTooltipsEnabled	124
4.1.78	Authentic.UICommands	125
4.1.79	Authentic.ValidateDocument	126
4.1.80	Authentic.validationBadData	127
4.1.81	Authentic.validationMessage	128
4.1.82	Authentic.XMLDataLoadObject	129
4.1.83	Authentic.XMLDataSaveUrl	130
4.1.84	Authentic.XMLRoot	131

4.1.85	Authentic.XMLTable	132
4.2	AuthenticCommand	133
4.2.1	AuthenticCommand.CommandID	134
4.2.2	AuthenticCommand.Group	135
4.2.3	AuthenticCommand.ShortDescription	136
4.2.4	AuthenticCommand.Name	137
4.3	AuthenticCommands	138
4.3.1	AuthenticCommands.Count	139
4.3.2	AuthenticCommands.Item	140
4.4	AuthenticDataTransfer	141
4.4.1	AuthenticDataTransfer.dropEffect	142
4.4.2	AuthenticDataTransfer.getData	143
4.4.3	AuthenticDataTransfer.ownDrag	144
4.4.4	AuthenticDataTransfer.type	145
4.5	AuthenticEvent	146
4.5.1	AuthenticEvent.altKey	147
4.5.2	AuthenticEvent.altLeft	148
4.5.3	AuthenticEvent.button	149
4.5.4	AuthenticEvent.cancelBubble	150
4.5.5	AuthenticEvent.clientX	151
4.5.6	AuthenticEvent.clientY	152
4.5.7	AuthenticEvent.ctrlKey	153
4.5.8	AuthenticEvent.ctrlLeft	154
4.5.9	AuthenticEvent.dataTransfer	155
4.5.10	AuthenticEvent.fromElement	156
4.5.11	AuthenticEvent.keyCode	157
4.5.12	AuthenticEvent.propertyName	158
4.5.13	AuthenticEvent.repeat	159
4.5.14	AuthenticEvent.returnValue	160
4.5.15	AuthenticEvent.shiftKey	161
4.5.16	AuthenticEvent.shiftLeft	162
4.5.17	AuthenticEvent.srcElement	163
4.5.18	AuthenticEvent.type	164
4.6	AuthenticLoadObject	165
4.6.1	AuthenticLoadObject.String	166
4.6.2	AuthenticLoadObject.URL	167
4.7	AuthenticRange	168
4.7.1	AuthenticRange.AppendRow	171
4.7.2	AuthenticRange.Application	172
4.7.3	AuthenticRange.CanPerformAction	173

4.7.4	AuthenticRange.CanPerformActionWith	174
4.7.5	AuthenticRange.Clone	175
4.7.6	AuthenticRange.CollapsToBegin	176
4.7.7	AuthenticRange.CollapsToEnd	177
4.7.8	AuthenticRange.Copy	178
4.7.9	AuthenticRange.Cut	179
4.7.10	AuthenticRange.Delete	180
4.7.11	AuthenticRange.DeleteRow	181
4.7.12	AuthenticRange.DuplicateRow	182
4.7.13	AuthenticRange.ExpandTo	183
4.7.14	AuthenticRange.FirstTextPosition	184
4.7.15	AuthenticRange.FirstXMLData	185
4.7.16	AuthenticRange.FirstXMLDataOffset	186
4.7.17	AuthenticRange.GetElementAttributeNames	188
4.7.18	AuthenticRange.GetElementAttributeValue	189
4.7.19	AuthenticRange.GetElementHierarchy	190
4.7.20	AuthenticRange.GetEntityNames	191
4.7.21	AuthenticRange.Goto	192
4.7.22	AuthenticRange.GotoNext	193
4.7.23	AuthenticRange.GotoNextCursorPosition	194
4.7.24	AuthenticRange.GotoPrevious	195
4.7.25	AuthenticRange.GotoPreviousCursorPosition	196
4.7.26	AuthenticRange.HasElementAttribute	197
4.7.27	AuthenticRange.InsertEntity	198
4.7.28	AuthenticRange.InsertRow	199
4.7.29	AuthenticRange.IsEmpty	200
4.7.30	AuthenticRange.IsEqual	201
4.7.31	AuthenticRange.IsInDynamicTable	202
4.7.32	AuthenticRange.LastTextPosition	203
4.7.33	AuthenticRange.LastXMLData	204
4.7.34	AuthenticRange.LastXMLDataOffset	205
4.7.35	AuthenticRange.MoveBegin	207
4.7.36	AuthenticRange.MoveEnd	208
4.7.37	AuthenticRange.MoveRowDown	209
4.7.38	AuthenticRange.MoveRowUp	210
4.7.39	AuthenticRange.Parent	211
4.7.40	AuthenticRange.Paste	212
4.7.41	AuthenticRange.PerformAction	213
4.7.42	AuthenticRange.Select	214
4.7.43	AuthenticRange.SelectNext	215
4.7.44	AuthenticRange.SelectPrevious	216

4.7.45	AuthenticRange.SetElementAttributeValue	217
4.7.46	AuthenticRange.SetFromRange	218
4.7.47	AuthenticRange.Text	219
4.8	AuthenticSelection	220
4.8.1	AuthenticSelection.End	221
4.8.2	AuthenticSelection.EndTextPosition	222
4.8.3	AuthenticSelection.Start	223
4.8.4	AuthenticSelection.StartTextPosition	224
4.9	AuthenticToolBarButton	225
4.9.1	AuthenticToolBarButton.CommandID	226
4.10	AuthenticToolBarButtons	227
4.10.1	AuthenticToolBarButtons.Count	228
4.10.2	AuthenticToolBarButtons.Item	229
4.10.3	AuthenticToolBarButtons.NewButton	230
4.10.4	AuthenticToolBarButtons.NewCustomButton	231
4.10.5	AuthenticToolBarButtons.NewSeparator	232
4.10.6	AuthenticToolBarButtons.Remove	233
4.11	AuthenticToolBarRow	234
4.11.1	AuthenticToolBarRowAlignment	235
4.11.2	AuthenticToolBarRowButtons	236
4.12	AuthenticToolBarRows	237
4.12.1	AuthenticToolBarRows.Count	238
4.12.2	AuthenticToolBarRows.Item	239
4.12.3	AuthenticToolBarRows.RemoveRow	240
4.12.4	AuthenticToolBarRows.NewRow	241
4.13	AuthenticView	242
4.13.1	AuthenticView.Application	243
4.13.2	AuthenticView.DocumentBegin	244
4.13.3	AuthenticView.DocumentEnd	245
4.13.4	AuthenticView.Goto	246
4.13.5	AuthenticView.MarkupVisibility	247
4.13.6	AuthenticView.Parent	248
4.13.7	AuthenticView.Print	249
4.13.8	AuthenticView.Redo	250
4.13.9	AuthenticView.Selection	251
4.13.10	AuthenticView.Undo	252
4.13.11	AuthenticView.UpdateXMLInstanceEntities	253
4.13.12	AuthenticView.WholeDocument	254
4.14	AuthenticXMLTableCommands	255
4.14.1	AuthenticXMLTableCommands.AlignHorizontalCenter	256

4.14.2	AuthenticXMLTableCommands.AlignHorizontalJustify	257
4.14.3	AuthenticXMLTableCommands.AlignHorizontalLeft	258
4.14.4	AuthenticXMLTableCommands.AlignHorizontalRight	259
4.14.5	AuthenticXMLTableCommands.AlignVerticalBottom	260
4.14.6	AuthenticXMLTableCommands.AlignVerticalCenter	261
4.14.7	AuthenticXMLTableCommands.AlignVerticalTop	262
4.14.8	AuthenticXMLTableCommands.AppendCol	263
4.14.9	AuthenticXMLTableCommands.AppendRow	264
4.14.10	AuthenticXMLTableCommands.Delete	265
4.14.11	AuthenticXMLTableCommands.DeleteCol	266
4.14.12	AuthenticXMLTableCommands.DeleteRow	267
4.14.13	AuthenticXMLTableCommands.EditProperties	268
4.14.14	AuthenticXMLTableCommands.Insert	269
4.14.15	AuthenticXMLTableCommands.InsertCol	270
4.14.16	AuthenticXMLTableCommands.InsertRow	271
4.14.17	AuthenticXMLTableCommands.JoinDown	272
4.14.18	AuthenticXMLTableCommands.JoinLeft	273
4.14.19	AuthenticXMLTableCommands.JoinRight	274
4.14.20	AuthenticXMLTableCommands.JoinUp	275
4.14.21	AuthenticXMLTableCommands.MayAlignHorizontal	276
4.14.22	AuthenticXMLTableCommands.MayAlignVertical	277
4.14.23	AuthenticXMLTableCommands.MayAppendCol	278
4.14.24	AuthenticXMLTableCommands.MayAppendRow	279
4.14.25	AuthenticXMLTableCommands.MayDelete	280
4.14.26	AuthenticXMLTableCommands.MayDeleteCol	281
4.14.27	AuthenticXMLTableCommands.MayDeleteRow	282
4.14.28	AuthenticXMLTableCommands.MayEditProperties	283
4.14.29	AuthenticXMLTableCommands.MayInsert	284
4.14.30	AuthenticXMLTableCommands.MayInsertCol	285
4.14.31	AuthenticXMLTableCommands.MayInsertRow	286
4.14.32	AuthenticXMLTableCommands.MayJoinDown	287
4.14.33	AuthenticXMLTableCommands.MayJoinLeft	288
4.14.34	AuthenticXMLTableCommands.MayJoinRight	289
4.14.35	AuthenticXMLTableCommands.MayJoinUp	290
4.14.36	AuthenticXMLTableCommands.MaySplitHorizontal	291
4.14.37	AuthenticXMLTableCommands.MaySplitVertical	292
4.14.38	AuthenticXMLTableCommands.SplitHorizontal	293
4.14.39	AuthenticXMLTableCommands.SplitVertical	294
4.15	XMLData	295
4.15.1	XMLData.AppendChild	296
4.15.2	XMLData.EraseAllChildren	297

4.15.3	XMLData.EraseCurrentChild	298
4.15.4	XMLData.GetCurrentChild	299
4.15.5	XMLData.GetFirstChild	300
4.15.6	XMLData.GetNextChild	301
4.15.7	XMLData.HasChildren	302
4.15.8	XMLData.InsertChild	303
4.15.9	XMLData.IsSameNode	304
4.15.10	XMLData.Kind	305
4.15.11	XMLData.MayHaveChildren	306
4.15.12	XMLData.Name	307
4.15.13	XMLData.Parent	308
4.15.14	XMLData.TextValue	309

5 Enumerations 312

5.1	SPYXMLDataKind	312
5.2	SPYAuthenticElementActions	313
5.3	SPYAuthenticCommand	314
5.4	SPYAuthenticCommandGroup	316
5.5	SPYAuthenticToolbarAllignment	317
5.6	SPYAuthenticEntryHelperWindows	318
5.7	SPYAuthenticElementKind	319
5.8	SPYAuthenticActions	320
5.9	SPYAuthenticDocumentPosition	321
5.10	SPYAuthenticMarkupVisibility	322

6 Altova Developer License Agreement 324

Index

Kapitel 1

Einführung

Einführung

Mit der **Altova® Authentic® 2005 Browser Edition** können Benutzer XML-Dokumente bearbeiten, die auf **StyleVision Power Stylesheets (.sps-Dateien)** basieren, die mit **StyleVision** erstellt wurden.

Die Authentic 2005 Browser Edition ist eine einzigartige Lösung, die es Ihnen gestattet, XML-Inhalte direkt von einem beliebigen Desktop (Client) aus zu bearbeiten, vorausgesetzt Sie haben **Internet Explorer 5.5 oder höher** installiert. Das Installationspaket des Authentic 2005 Browser Edition Plug-In ist auf einem Server gespeichert und wird auf einem Client installiert, wenn auf dem Client eine HTML-Seite geöffnet wird, die ein OBJECT-Element enthält, das das Authentic Browser Plug-In als Objekt definiert. Das XML-Dokument wird über die Browser-Oberfläche des Client bearbeitet. Auf diese Art kann ein XML-Dokument von verschiedenen Benutzern über deren Browser aufgerufen werden. Dank dieser Strategie lässt sich die Anwendung problemlos im gesamten Unternehmen verwenden und warten - und dies bei gleichzeitiger Verringerung der Investitionskosten.

Diese XML-Editierlösung basiert auf offenen Standards wie z.B. XML-Schema und XSLT und ist vollständig Unicode-kompatibel. Dabei wird die Altova Authentic-Ansicht als Benutzeroberfläche innerhalb einer HTML-Seite verwendet. Über die Authentic-Ansicht können Sie XML-Dokumente wie bei einem Textverarbeitungsprogramm bearbeiten, d.h. der zugrunde liegende Code wird nicht angezeigt. Sie benötigen außerdem keine zusätzliche Software, da das Plug-In in den Internet Explorer integriert wird.

Die Authentic 2005 Browser Edition ist ein ActiveX-Control, wobei die COM-Schnittstelle durch das [Authentic](#)-Objekt definiert ist. Im Abschnitt "Objekte" dieser Dokumentation finden Sie eine Beschreibung des kompletten Objektmodells.

Implementieren der Authentic 2005 Browser Edition Lösung

Um die Authentic 2005 Browser Edition Lösung zu implementieren, gehen Sie folgendermaßen vor:

- Laden Sie die Authentic 2005 Browser Edition von der Altova-Website herunter und legen Sie sie auf Ihrem Server ab.
- Erstellen Sie das XML-Schema, das StyleVision Power Stylesheet (SPS) und die XML- (oder XML-Vorlagen-) Dateien. In der (mit Altova StyleVision 2005 erstellten) SPS-Datei wird definiert, wie die XML-Datei in der Authentic-Ansicht angezeigt werden soll und wie die Daten in die XML-Datei eingegeben werden. Die XML-Datei (bzw. eine XML-Vorlagendatei) ist die XML-Datei, die mit der Authentic 2005 Browser Edition editiert werden soll. Nähere Informationen zum Erstellen der SPS-Datei und wie man diese mit einer XML-Datei verknüpft, finden Sie in der Dokumentation zu StyleVision 2005.
- Bei StyleVision Power Stylesheets, die auf einer Datenbank basieren, müssen Sie in der SPS, auf dem Server und auf den Clients einige Einstellungen vornehmen. Diese Einstellungen sind unter "[Voraussetzungen für DB-basierte SPS-Dateien](#)" näher beschrieben.
- Erstellen Sie die [HTML-Datei](#), die das [Authentic](#)-Objekt enthält.

Hinweis: Wenn der Benutzer eine HTML-Datei, die ein Authentic-Objekt enthält, zum ersten Mal öffnet, wird das Authentic Browser Plug-In vom Server heruntergeladen und als Plug-In zur Browser-Applikation des Client installiert. Es sind keine weiteren Installationsschritte erforderlich.

In dieser Dokumentation

- wird erklärt, wie Sie die Authentic 2005 Browser Edition von der Altova-Website

- herunteladen und installieren,
- wie Sie die HTML-Seite erstellen, die der Client öffnen muss, um auf das XML-Dokument zugreifen zu können.
- finden Sie einen Referenzteil zu den Objekten, mit denen die Authentic-Ansicht in der Authentic 2005 Browser Edition erstellt und angepasst wird.

Eine Dokumentation sowie ein Tutorial zur Erstellung des StyleVision Power Stylesheet finden Sie im **Hilfe**-Menü von StyleVision 2005.

1 Installation der Authentic 2005 Browser Edition

Um die Authentic 2005 Browser Edition auf Ihrem Server zu installieren, gehen Sie folgendermaßen vor:

1. Installieren und konfigurieren Sie den Browser-Dienst Ihres Servers. Wenn Sie Internet Information Services (IIS) von Microsoft verwenden, beachten Sie, dass bei der Installation ein Standardverzeichnis mit dem Namen `Inetpub` sowie Unterverzeichnisse angelegt werden. Das Root-Verzeichnis des Servers wäre dann: `//Inetpub/wwwroot`. Dies ist das Verzeichnis, das Sie mit der IP-Adresse des Servers erreichen, wenn Sie (in Ihrer Browser Service-Konfiguration) kein anderes als das Root-Verzeichnis angeben. Nähere Informationen zum Installieren und Konfigurieren Ihres Browser-Diensts finden Sie in der Dokumentation Ihres Anbieters.
2. Laden Sie die Authentic 2005 Browser Edition (ein gezippte `.cab`-Datei) in einen beliebigen Ordner auf dem Server herunter. Entpacken Sie die `.cab`-Datei nicht.
3. Testen Sie Ihre Installation durch Erstellen einer HTML-Datei wie der Beispieldatei unter [HTML-Seite: Beispiel 1 \(einfach\)](#). (Vergessen Sie nicht, die erforderliche `xsd`-, `xml`- und `sps`-Datei in die in der HTML-Datei referenzierten Ordner zu kopieren.)

Wenn ein Client auf die HTML-Datei zugreift, wird das Authentic Browser Plug-In auf dem Client installiert und die Authentic-Seite wird im Browser des Client geladen.

2 Herunterladen der Authentic 2005 Browser Edition auf den Server

Das Authentic Browser Plug-In ist auf dem Server in Form einer gezippten (.cab)-Datei gespeichert und wird von dieser Datei aus auf dem Client installiert, wenn der Client auf eine HTML-Datei zugreift, für die das Browser-Plug-In benötigt wird. Die .cab-Datei muss von der Altova Website heruntergeladen werden. Sie können die .cab-Datei in einen beliebigen Ordner auf Ihrem Server herunterladen. **Entpacken Sie die .cab-Datei nicht!** Die Datei wird automatisch extrahiert und auf dem Client installiert, wenn der Client zum ersten Mal auf die HTML-Datei zugreift, die das Browser Plug-In-Objekt enthält. In der HTML-Datei ist der Pfad zur .cab-Datei im codebase-Attribut des HTML OBJECT-Elements definiert.

Auf der Altova-Website stehen die folgenden Versionen der Authentic 2005 Browser Edition zum kostenlosen Download bereit:

- Trusted Unicode
- Untrusted Unicode

Je nach Bedarf (siehe unten) können Sie eine oder mehrere dieser Versionen herunterladen.

Die einzelnen Versionen der Authentic 2005 Browser Edition haben folgende Merkmale:

- Die Unicode-Version bietet vollständige Unterstützung mehrerer Zeichensätze im XML-Dokument. Voraussetzung für die Verwendung der Unicode-Version ist die Verwendung von Windows NT, 2000 oder XP auf dem Client-Rechner.
- Da Unicode unter Windows 95, 98 und ME nicht unterstützt wird, kann die Unicode-Version auf diesen Betriebssystemen nicht verwendet werden.
- Die Trusted-Version gestattet keinen Zugriff auf lokale Dateien und ist daher als "sicher für das Skripting" gekennzeichnet. Sie kann in einem Browser-basierten Szenario eingesetzt und von jeder Webseite aus aufgerufen werden, ohne auf Seiten des Client Sicherheitswarnungen zu verursachen.
- Die Untrusted-Version ist für die Verwendung im Intranet gedacht oder zur Verwendung der Authentic Browser Edition als ActiveX-Control in Ihrer Anwendung. Sie bietet Zugriff auf lokale Dateien und ist daher **nicht** als "sicher für das Skripting" gekennzeichnet. Wenn Sie versuchen, diese Version innerhalb eines Browser-Fensters zu verwenden, wird der Benutzer um Erlaubnis gefragt.

Jede Version hat eine andere Class ID (CLSID). Die CLSIDs werden im <object>-Element des HTML-Dokuments (Näheres dazu siehe [Das OBJECT-Element](#)) verwendet, um die richtige Version auf den Client zu laden.

Hinweis: Um Ihren Browser-Dienst (z.B. Internet Information Services) zu installieren, lesen Sie bitte die Dokumentation des jeweiligen Anbieters.

3 Voraussetzungen für DB-basierte SPS-Dateien

Falls Sie die Authentic 2005 Browser Edition verwenden wollen, um Datenbanken (DB) über ein Datenbank-basiertes StyleVision Power Stylesheet (SPS) aufzurufen oder zu editieren, müssen Sie die folgenden Einstellungen vornehmen, um sicherzustellen, dass der Client eine Verbindung zur Datenbank aufbauen kann. Beispiel-HMTL-Code finden Sie unter [HTML-Seite: Beispiel 4 \(DB-Referenz\)](#).

Connection Info im SPS

Alle für die Herstellung der Verbindung zur Datenbank erforderlichen Informationen sind in einem Connection String im SPS gespeichert. Der Connection String im SPS wird bei der Erstellung des StyleVision Power Stylesheet in StyleVision erstellt. Das Verfahren zur Herstellung einer Verbindung zu MS Access Datenbank unterscheidet sich von den für andere Datenbanken verwendeten Verfahren. Für andere Datenbanken wird eine ADO-Verbindung verwendet. Im Folgenden sind die Einstellungen beschrieben, die Sie für diese beiden Verbindungsarten vornehmen müssen.

Für MS Access-Datenbanken

Im Connection String für MS Access Datenbanken muss ein UNC-Pfad verwendet werden, damit der Client eine Verbindung zur Datenbank aufbauen kann. Dieser UNC-Pfad wird beim Erstellen des StyleVision Power Stylesheet in StyleVision definiert. Sie müssen allerdings sicherstellen, dass der Ordner, in dem die Datenbank gespeichert ist (oder ein übergeordneter Ordner) freigegeben wurde. (In Windows 2000 und Windows XP können Sie diese Einstellungen durch Rechtsklick auf den Ordner und Auswahl des Befehls "**Freigabe und Sicherheit...**" vornehmen). Außerdem müssen Sie auf diesem Rechner die erweiterte Einstellung zur Dateifreigabe aktivieren (Klicken Sie auf **Arbeitsplatz | Extras | Ordneroptionen** und deaktivieren Sie das Kontrollkästchen "einfache Datenfreigabe verwenden"). Auf den Clients sind keine speziellen Einstellungen erforderlich.

Hinweis: Das Format für den im Connection String angegebenen UNC-Pfad ist:

`\\servername\sharename\path\file.mdb`, wobei `servername` der Name des Servers ist, `sharename` der Name des freigegebenen Ordners (der in den Freigabeeinstellungen für den jeweiligen Ordner auf dem Server definiert wurde), `path` der Pfad zur Datenbank und `file.mdb` der Name der MS Access Datenbank im freigegebenen Ordner, bzw. in einem Unterordner dieses Ordners.

Für ADO-Verbindungen

Der ADO Connection String im SPS wird beim Erstellen des SPS in StyleVision definiert. Er enthält die erforderlichen Connection-Informationen sowie die Sicherheitsinformationen. Sie müssen sicherstellen, dass der beim Testen des Connection String in StyleVision verwendete Treiber auch auf allen Client-Rechnern vorhanden ist, auf denen die Authentic 2005 Browser Edition installiert ist, damit das SPS eine Verbindung zwischen Datenbank und Client aufbauen kann.

Kapitel 2

Die HTML-Seite für Authentic

Die HTML-Seite für Authentic

Damit ein Client ein XML-Dokument über das Authentic 2005 Browser Edition Plug-In editieren kann, muss die HTML-Seite die folgenden Elemente enthalten:

- ein HTML `OBJECT`-Element, das (i) die DLL für das Plug-In auf den Client lädt, (ii) die Größe des Authentic-Fensters im Browser des Client definiert und (iii) eine beliebige Anzahl optionaler Parameter deklariert.
- eines oder mehrere HTML `SCRIPT`-Elemente. Ein `SCRIPT`-Element definiert die verschiedenen Subroutinen, die in der HTML-Seite und bei der Ereignisverarbeitung verwendet werden.

In diesem Abschnitt wird beschrieben, wie das HTML [OBJECT](#)- und das HTML [SCRIPT](#)-Element zu verwenden sind. Außerdem enthält dieser Abschnitt ein [Beispiel für eine HTML-Seite](#), die das Authentic 2005 Browser Edition Plug-In aufruft. Informationen zu den einzelnen Objekten finden Sie unter den jeweiligen Objektbeschreibungen im Referenzteil.

1 Das OBJECT-Element

Das OBJECT-Element hat die folgenden Funktionen:

- Es gibt dem Authentic Browser Plug-In einen Namen (über das `id`-Attribut).
- Es wählt aus, welche Version des Authentic Browser Plug-In verwendet werden soll (über den Wert des `classid`-Attributs).
- Es definiert eine `.cab` Datei und eine Versionsnummer (`codebase`-Attribut). Die `.cab`-Datei enthält eine DLL, die ein COM-Objekt (das Authentic Browser Plug-In) mit einer ID registriert, die der Wert des `classid`-Attributs ist.
- Es definiert (über das `style`-Attribut) die Größe des Authentic-Fensters im Browser des Client.
- Es kann eine beliebige Anzahl von Parametern definieren.

Hier sehen Sie ein Beispiel für ein HTML OBJECT-Element. Es wählt die Trusted Unicode-Version aus (über den im `classid`-Attribut definierten Wert) und legt als Größe des Authentic-Fensters im Browser des Client 600 x 500 Pixel fest. Im Folgenden sind alle Attribute und Parameter beschrieben, die dem OBJECT-Element zur Verfügung stehen. (Beachten Sie, dass die unten angegebene Versionsnummer möglicherweise nicht die Nummer der aktuellen Version ist. Nähere Informationen dazu siehe [codebase](#).)

```
<OBJECT id=objPlugIn style="WIDTH:600px; HEIGHT:500px"
  codeBase="http://yourserver/cabfiles/AuthenticBrowserEdition_de.CAB#Version
=7,0,3,0"
  classid=clsid:91DDF44A-DFD1-4F47-8EE3-4CBE874584F7>
  <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

Der Wert des `id`-Attributs wird als Name der Authentic Browser Plug-In Objekte verwendet, wenn diese in Skripts verwendet werden. So wird z.B. mit `objPlugIn.SchemaLoadObject.URL` das Objekt aufgerufen, das die Schema-Datei lädt. Nähere Informationen dazu siehe "[Das SCRIPT-Element](#)".

style

Dies ist das normale HTML `style`-Attribut. Es dient zur Angabe der Größe des Authentic-Fensters im Browser des Client.

codebase

Das `codebase`-Attribut gibt den Pfad zur `.cab`-Datei an.

Der Wert der optionalen `#Version`-Erweiterung gibt die Versionsnummer der Komponente an, die derzeit auf dem Server zur Verfügung steht. Falls der Client eine frühere Version verwendet, wird die neuere Version vom Server installiert. Wenn die `#Version`-Erweiterung nicht angegeben wurde, erfolgt keine Aktualisierung, es sei denn, die Komponente wurde manuell vom Client entfernt. Die **Versionsnummer der Komponente** ist in der `.osd`-Datei der `.cab`-Datei der Komponente angegeben.

classid

Ab Version 5.0 des Browser Plug-In ist der `classid`-Wert für Unicode-Versionen ab Version 5.0 ein anderer als der früherer Unicode-Versionen. Wenn Sie daher die Unicode `.cab`-Datei auf Ihrem Server von einer Version vor Version 5.0 aktualisieren, stellen Sie sicher, dass Sie die `classid`-Werte in Ihren HTML-Dateien ändern. Beachten Sie: Wenn eine neue `.cab`-Datei auf dem Server dieselbe CLSID hat wie eine auf dem Client bereits installierte `.cab`-Datei, wird die

alte .cab-Datei auf dem Client nicht automatisch durch die neue ersetzt. Die müssen die zuvor installierte .cab-Datei entfernen, bevor Sie die neue .cab-Datei herunterladen.

Bedenken Sie bei der Auswahl der Version, die auf den Client heruntergeladen werden soll, dass die Unicode-Version auf älteren Betriebssysteme möglicherweise nicht korrekt installiert werden kann.

Im Folgenden finden Sie die CLSID-Werte der aktuellen deutschen Versionen:

Trusted Unicode:

`clsid:91DDF44A-DFD1-4F47-8EE3-4CBE874584F7`

Untrusted Unicode:

`clsid:28A640E8-EAEE-4B5D-BEBE-BFA956081E66`

Parameter

Es können die folgenden Parameter verwendet werden.

XMLDataURL

Eine absolute URL, die den Speicherort der zu editierenden XML-Datei angibt. Für Untrusted Versionen können Sie auch einen vollständigen lokalen Pfad angeben.

XMLDataSaveURL

Eine absolute URL, die den Ordner angibt, in dem die XML-Datei gespeichert werden soll. Für Untrusted Versionen können Sie auch einen vollständigen lokalen Pfad angeben.

SPSDataURL

Eine absolute URL, die den Pfad zum StyleVision Power Stylesheet (.sps-Datei) angibt. Für Untrusted Versionen können Sie auch einen vollständigen lokalen Pfad angeben.

SchemaDataURL

Eine absolute URL, die den Pfad zur verknüpften Schema-Datei angibt. Für Untrusted Versionen können Sie auch einen vollständigen lokalen Pfad angeben.

TextStateBmpURL

Der Ordner, in dem Bitmap-Grafiken für Textstatus-Symbole gespeichert werden sollen.

TextStateToolbarLine

Die Zeile der Symbolleiste, in der Textstatus-Symbole platziert werden sollen. Die Standardeinstellung ist 1.

AutoHideUnusedCommandGroups

Gibt an, ob Gruppen nicht verwendeter Symbolleistenbefehle ausgeblendet werden sollen. Die Standardeinstellung ist True.

ToolbarsEnabled

Gibt die allgemeine Unterstützung für Symbolleisten an. Die Standardeinstellung ist True.

ToolbarTooltipsEnabled

Gibt an, ob Tooltips aktiviert oder deaktiviert sind.

HideSaveButton

Bei Einstellung "True" wird die standardmäßig eingeblendete Schaltfläche "Speichern" aus der Authentic-Symbolleiste entfernt.

BaseURL

Gibt die Basis-URL für relative Pfade an.

SaveButtonUsePOST

Bei Einstellung "True" wird beim Speichern des Dokuments der Befehl HTTP POST verwendet, anstelle von PUT.

EntryHelpersEnabled

Bei Einstellung "True" werden die Authentic-Eingabehilfen eingeblendet.

EntryHelperSize

Breite der Eingabehilfenfenster in Pixeln.

EntryHelperAlignment

Definiert die Position der Eingabehilfen im Dokumentfenster.

- 0 = Symbolleiste am oberen Rand des Dokuments anzeigen
- 1 = Symbolleiste am linken Rand des Dokuments anzeigen
- 2 = Symbolleiste am unteren Rand des Dokuments anzeigen
- 3 = Symbolleiste am rechten Rand des Dokuments anzeigen

EntryHelperWindows

Gibt an, welche der Eingabehilfenfenster eingeblendet werden sollen.

- 1 = Elemente
- 2 = Attribute
- 4 = Entities

Zulässig ist jede Kombination (bit-check)

SaveButtonAutoEnable

Siehe [Authentic.SaveButtonAutoEnable](#)

LoaderSettingsFileURL

Gibt die URL der `LoaderSettingsFile` für die Paketverwaltung an.

Die folgenden Parameter werden nicht mehr verwendet:

Ab Version 6.0.3.1 der Authentic 2005 Browser Edition sind keine Lizenzen mehr erforderlich.

LicServer

Der Name des Servers, auf dem sich der Lizenzkey befindet (falls der Lizenzkey über eine Datei zur Verfügung gestellt wird). Siehe auch [Lizenzinformationen](#).

LicFile

Der Name der Datei, in der der Lizenzkey gespeichert ist. Siehe auch [Lizenzinformationen](#).

LicKey

Der Lizenzkey zur Überprüfung der Verwendung der Authentic Browser Edition. Siehe auch [Lizenzinformationen](#).

LicCompany

Der Name des Unternehmens (zur Überprüfung der Verwendung der Authentic Browser Edition). Siehe auch [Lizenzinformationen](#).

LicUser

Der Name des Benutzers (zur Überprüfung der Verwendung der Authentic Browser Edition). Siehe auch [Lizenzinformationen](#).

2 Das SCRIPT-Element

Die `SCRIPT`-Elemente definieren die Ereignishandler und Subroutinen, die von innerhalb der HTML-Datei aus aufgerufen werden können.

Im Folgenden sehen Sie ein Beispiel für ein Skript zur Ereignisbehandlung:

```
<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>
```

Hier sehen Sie ein Beispiel für ein Skript mit Subroutinen:

```
<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>
```

LANGUAGE

Das Authentic 2005 Browser Edition Plug-In wurde mit JavaScript and VBScript getestet.

Ereignisbehandlung

Der Wert des `ID`-Attributs des `OBJECT`-Elements im HTML-Textkörper wird als der Wert des `FOR`-Attributs definiert. Authentic Browser Plug-In-Objekte, die aufgerufen werden, müssen einen Namen haben, der diesem Wert entspricht.

Subroutinen

Für jedes Ereignis, das Sie in der HTML-Datei definieren möchten, können Subroutinen erstellt werden. Der Objektname für das Authentic Browser Plug-In muss mit dem Wert des `ID`-Attributs des `OBJECT`-Elements im HTML-Textkörper identisch sein. Im oben gezeigten Beispiel ist das Präfix `objPlugIn`. Dieses Präfix muss der Wert des `ID`-Attributs des `OBJECT`-Elements sein.

Die im Authentic Browser Plug-In verfügbaren Methoden, Eigenschaften und Sub-Objekte werden im Referenzteil dieser Dokumentation beschrieben.

3 HTML-Seite: Beispiel 1 (einfach)

Der unten gezeigte HTML-Code generiert eine Seite, die die folgenden Funktionen hat:

- Sie installiert die Trusted Unicode-Version der Authentic 2005 Browser Edition am Client, falls dies nicht bereits geschehen ist.
- Innerhalb des body-Tags ist ein Fenster von der Größe 600px mal 500px definiert, in das die Authentic 2005 Browser Edition geladen wird.
- Unterhalb des Authentic 2005 Browser Edition-Fensters befindet sich eine Reihe mit fünf Schaltflächen
- Die Schaltfläche **Bearbeitung starten** dient zum Laden der Authentic-Ansicht von `OrgChart.xml`, welche sich im Root-Verzeichnis Ihres Servers befindet.
- Die Schaltflächen **Suchen** und **Ersetzen** dienen zum Aufrufen des Such- bzw. Ersetzungsdialogfelds.
- Die Schaltfläche **Speichern** dient zum Speichern der Änderungen in einer Datei mit dem Namen `SaveFile.xml`, die im Root-Verzeichnis des Servers abgelegt ist.
- Die Schaltfläche **Eigenschaft testen** dient zum Testen einer einfachen Eigenschaft.

Wenn diese HTML-Seite auf dem Client geöffnet wird, kann der Benutzer die XML-Datei `OrgChart.xml` bearbeiten und die bearbeitete Datei als `saveFile.xml` speichern.

Anhand dieser einfachen HTML-Seite können Sie testen, ob die Authentic 2005 Browser Edition korrekt funktioniert. Stellen Sie dabei sicher, dass in den URLs für die `.cab`-Datei sowie die `.xsd`-, `.xml`- und die `.sps`-Datei und alle anderen auf dem Server gespeicherten Ressourcen die richtige IP-Adresse und der korrekte Pfad angegeben sind. Sie können dieses Beispiel erweitern und ändern, um komplexere Lösungen mit Hilfe der Authentic 2005 Browser Edition zu erstellen.

```
<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="http://yourserver/AuthenticBrowserEdition.cab#Version=7,0,1,0"
    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>
```

4 HTML-Seite: Beispiel 2 (Tabelle mit bubble-sort sortieren)

Dies ist eine HTML-Beispieldatei mit einem eingebetteten JScript. Für das Beispiel muss das Authentic Browser Plug-In (CAB-Datei und Lizenzdatei) auf Ihrem Computer installiert werden. Nähere Informationen finden Sie auf der Altova-Homepage.

Der Code gibt an:

- wie das Browser Plug-In aufgerufen werden soll. Passen Sie den Code bitte an und geben Sie den Pfad zu Ihrer CAB-Datei, Lizenzdatei und dem Class Identifier Ihrer Browser Plug-In-Version an (trusted oder untrusted).
- wie eine Datei in das Browser Plug-In geladen werden soll. Passen Sie bitte den Code an und geben Sie den Pfad zu Ihrem Beispieldokument an.
- wie Schaltflächen für einfache Cursor-Positionierungen implementiert werden.
- wie komplexere Befehle wie z.B. die Sortierung von Tabellen implementiert werden.
- wie das 'SelectionChanged'-Ereignis verwendet wird.

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
  <title>test page For Authentic Browser Plug-in</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(1); }
          catch (err) { break; }
          nColIndex++;
        }
      }
    }
  </SCRIPT>
</head>
<body>
  <table border="1">
    <tr>
      <td>1</td>
      <td>2</td>
      <td>3</td>
      <td>4</td>
      <td>5</td>
      <td>6</td>
      <td>7</td>
      <td>8</td>
      <td>9</td>
      <td>10</td>
      <td>11</td>
      <td>12</td>
      <td>13</td>
      <td>14</td>
      <td>15</td>
      <td>16</td>
      <td>17</td>
      <td>18</td>
      <td>19</td>
      <td>20</td>
      <td>21</td>
      <td>22</td>
      <td>23</td>
      <td>24</td>
      <td>25</td>
      <td>26</td>
      <td>27</td>
      <td>28</td>
      <td>29</td>
      <td>30</td>
      <td>31</td>
      <td>32</td>
      <td>33</td>
      <td>34</td>
      <td>35</td>
      <td>36</td>
      <td>37</td>
      <td>38</td>
      <td>39</td>
      <td>40</td>
      <td>41</td>
      <td>42</td>
      <td>43</td>
      <td>44</td>
      <td>45</td>
      <td>46</td>
      <td>47</td>
      <td>48</td>
      <td>49</td>
      <td>50</td>
      <td>51</td>
      <td>52</td>
      <td>53</td>
      <td>54</td>
      <td>55</td>
      <td>56</td>
      <td>57</td>
      <td>58</td>
      <td>59</td>
      <td>60</td>
      <td>61</td>
      <td>62</td>
      <td>63</td>
      <td>64</td>
      <td>65</td>
      <td>66</td>
      <td>67</td>
      <td>68</td>
      <td>69</td>
      <td>70</td>
      <td>71</td>
      <td>72</td>
      <td>73</td>
      <td>74</td>
      <td>75</td>
      <td>76</td>
      <td>77</td>
      <td>78</td>
      <td>79</td>
      <td>80</td>
      <td>81</td>
      <td>82</td>
      <td>83</td>
      <td>84</td>
      <td>85</td>
      <td>86</td>
      <td>87</td>
      <td>88</td>
      <td>89</td>
      <td>90</td>
      <td>91</td>
      <td>92</td>
      <td>93</td>
      <td>94</td>
      <td>95</td>
      <td>96</td>
      <td>97</td>
      <td>98</td>
      <td>99</td>
      <td>100</td>
    </tr>
  </table>
  <div style="text-align: center; margin-top: 10px;">
    <input type="button" value="Sortieren" />
  </div>
</body>
</html>
```

```

// 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++)
{
    // Select correct column in first table row
    objBubble = objCursor.ExpandTo(9).CollapsToBegin().Clone();
    objBubble.Goto (11, nColIndex, 2).ExpandTo(6);

    // bubble
    For (var j = 0; j < nRows - i - 1; j++)
    {
        strField1 = objBubble.Text;
        strField2 = objBubble.GotoNext(10).Goto (11, nColIndex,
            2).ExpandTo(6).Text;
        If (strField1 > strField2)
        {
            objBubble.MoveRowUp();
            objBubble.GotoNext(10).Goto (11, nColIndex, 2).ExpandTo(6);
        }
    }
}
}
}
</SCRIPT>
</head>

<body>
<Object id="objPlugIn"
    codeBase="http://myCabfileLocation/AuthenticBrowserEdition.cab#Version=7,0,1,0"
    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>
<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>
```

5 HTML-Seite: Beispiel 4 (DB-Referenz)

Von der unten gezeigten HTML-Seite wird ein auf einer Datenbank basierendes StyleVision Power Stylesheet (SPS) geladen. Alle für die Verbindung zur Datenbank benötigten Informationen sind im SPS in einem Connection String gespeichert. Die HTML-Datei darf keine Referenz auf eine Schema-Datei oder eine XML-Datei enthalten. Die Schema-Datei und die XML-Datei werden beim Herstellen der Datenbank-Verbindung automatisch anhand der Datenbank generiert. Im unten gezeigten Beispiel wird die Trusted Unicode-Version der Authentic 2005 Browser Edition verwendet.

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
    <title>Product Development Featurelist</title>
    <!-- Script for handling the ControlInitialized event -->
    <SCRIPT LANGUAGE="javascript" FOR=objPlugIn EVENT="ControlInitialized">
      serverstr = "dev37";
      objPlugIn.DesignDataLoadObject.URL = "http://" + serverstr +
"/DBTest/DBSample.sps";
      objPlugIn.StartEditing();
    </SCRIPT>
  </head>
  <body>
    <OBJECT id=objPlugIn
      Classid=clsid:B4628728-E3F0-44a2-BEC8-F838555AE780 width="100%" height="100%">
    </OBJECT>
  </body>
</html>
```

Nähere Informationen zu Connection Strings finden Sie unter "[Voraussetzungen für DB-basierte SPS-Dateien](#)".

Kapitel 3

Mechanismen

Mechanismen

Dieser Abschnitt enthält eine Beschreibung einiger der am häufigsten verwendeten Authentic 2005 Browser Edition-Mechanismen.

1 Ereignisse: Connection Point

In der Authentic 2005 Browser Edition stehen zwei Connection Point-Ereignisse zur Verfügung (`ControlInitialized` und `SelectionChanged`), für die Sie Ereignishandler in den SCRIPT-Blöcken auf Ihrer HTML-Seite bereitstellen können. Die Connection Point-Ereignisse können mit der [Authentic.attachCallBack](#)-Methode nicht außer Kraft gesetzt werden.

ControlInitialized

Dieses Connection Point-Ereignis wird sofort nach Erstellung und Initialisierung des Controls ausgelöst. Ein zusätzliches Initialisierungsskript für das Control kann innerhalb der `ControlInitialized`-Ereignisbehandlungsroutine abgearbeitet werden.

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

SelectionChanged

Das `SelectionChanged`-Ereignis wird jedes Mal ausgelöst, wenn die aktuelle Auswahl in der Ansicht geändert wird. Verwenden Sie einen SCRIPT-Block, um den jeweiligen Code für das Ereignis auszuführen.

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

Bitte beachten Sie, dass das [Authentic.event](#)-Objekt nicht befüllt wird, wenn dieses Ereignis vorkommt. Wenn in Ihrem Skript Ereignishandler registriert sind, enthalten die Eigenschaften des [Authentic.event](#)-Objekts Werte aus dem letzten Ereignis, das aufgetreten ist.

Das [Authentic.CurrentSelection](#)-Objekt enthält nun gültige Informationen.

Laden von SPS-, XSD- und XML-Dateien ohne Zutun des Benutzers

Wenn die `.sps`-, `.xsd`- und die `.xml`-Datei beim Laden der HTML-Seite ohne Zutun des Benutzers geladen werden sollen, ist die beste Methode, einen Ereignishandler zu erzeugen, der das `ControlInitialized`-Event verarbeitet. Alternativ dazu können Sie auch ein `PropertyBag` verwenden (siehe zweites Beispiel):

Empfohlene Methode:

```
<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>
```

oder

```
<OBJECT id=objPlugIn style="WIDTH:600px; HEIGHT:500px"
codeBase="http://yourserver/cabfiles/AuthenticBrowserEdition.CAB#Version=7,0,1,0"
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>
```

Es ist nicht ratsam, diese Dateien in einen Ereignishandler zu laden, der das "onload"-Ereignis des "body"-Elements abarbeitet, da das Authentic Browser Plug-In-Control möglicherweise erst initialisiert wird, nachdem das "onload"-Ereignis ausgelöst wurde. In diesem Fall stehen die Methoden und Eigenschaften des Plug-In nicht zur Verfügung und die Dateien werden nicht geladen.

Nicht empfohlen:

```
<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">
```

2 Ereignisse: Symbolleisten-Schaltfläche

Jede Symbolleisten-Schaltfläche hat ein Standardverhalten, das unter Umständen geändert werden muss. Mit Hilfe des `AuthenticCommand`-Ereignisses können Sie zusätzliche Aufgaben hinzufügen oder das Standardverhalten eines solchen Schaltflächenbefehls völlig neu definieren. In Skripten kann das `AuthenticCommand`-Ereignis dazu verwendet werden, um jedes Mal, wenn der Benutzer auf eine Symbolleisten-Schaltfläche klickt, eine Benachrichtigung zu erhalten.

Beachten Sie bitte, dass jeder Befehl (aus der [Authentic.UICommands](#)-Sammlung) mit einem bestimmten Ereignis verknüpft ist. Um festzustellen, auf welches Symbol der Benutzer geklickt hat, muss das Skript die `AuthenticEvent.srcElement`-Eigenschaft überprüfen, die eine Referenz auf das entsprechende `AuthenticCommand`-Objekt enthält.

Beispiel

```
// event handler for OnDocEditCommand
function OnCommand()
{
    // 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;
    }
}
// somewhere in your script
function MyInit()
{
    objPlugIn.attachCallback("OnDocEditCommand", OnCommand);
}
```

Referenz

Welche Befehle zur Verfügung stehen, finden Sie unter [AuthenticToolBarButton.CommandID](#).

3 Aufrufen und Ändern von Dokumentinhalt

Zum Aufrufen und Ändern von Dokumentinhalt können Sie das `AuthenticRange`-, `AuthenticView`- und das `Authentic`-Objekt (und ihre Eigenschaften und Methoden) verwenden.

Wo sich die Funktionen der `AuthenticRange`- und der `AuthenticView`-Schnittstelle einerseits und der vom `Authentic`-Objekt bereitgestellten Schnittstelle auf der anderen Seite überlappen, sollte man der `AuthenticRange`- und der `AuthenticView`-Schnittstelle den Vorzug geben. Diese überlappenden Funktionalitäten des `Authentic`-Objekts werden allmählich eingestellt und in zukünftigen Versionen nicht mehr zur Verfügung stehen.

4 Bearbeitungsoperationen

Wenn XML-Daten in der Authentic-Ansicht angezeigt werden, können Sie einzelne Elemente mit Hilfe der Standardbearbeitungsoperationen Ausschneiden, Kopieren und Einfügen manipulieren. Es dürfen jedoch nicht alle XML-Datenelemente editiert werden. Sie müssen daher zuerst testen, ob eine Bearbeitung möglich ist. Dabei wird folgendermaßen vorgegangen: Überprüfen Sie zuerst, ob die jeweilige Bearbeitungsoperation aktiviert ist. Ist dies der Fall, rufen Sie die entsprechende Editier-Methode auf. Die einzige Methode, für die kein Test verfügbar ist, ist die `EditSelectAll`-Methode, die automatisch alle im Dokument angezeigten Elemente auswählt.

Nachstehend sehen Sie eine Liste von Eigenschaften und Methoden, die Bearbeitungsoperationen durchführen. Jede Eigenschaft gibt einen Booleschen Wert zurück. Die Methoden haben keine Parameter.

IsEditUndoEnabled	Authentic.EditUndo	Macht eine Bearbeitungsoperation rückgängig
IsEditRedoEnabled	Authentic.EditRedo	Stellt eine Bearbeitungsoperation wieder her
IsEditCopyEnabled	Authentic.EditCopy	Kopiert den markierten Text in die Zwischenablage
IsEditCutEnabled	Authentic.EditCut	Schneidet den markierten Text aus und legt ihn in die Zwischenablage
IsEditPasteEnabled	Authentic.EditPaste	Fügt den Text aus der Zwischenablage an der Cursorposition ein
IsEditClearEnabled	Authentic.EditClear	Löscht den markierten Text aus dem XML-Dokument

5 Suchen und Ersetzen

Die [Authentic.FindDialog](#)-Methode öffnet ein Suchdialogfeld, in welches der Benutzer einen Suchbegriff eingeben kann. Mit Hilfe der [Authentic.FindNext](#)-Methode wird die nächste Instanz desselben Elements gesucht. Die `FindNext`-Methode kann mit Hilfe der Booleschen Eigenschaft `IsFindNextEnabled` getestet werden. Eine Variante der `FindDialog`-Methode ist die [Authentic.ReplaceDialog](#)-Methode. Mit dieser Operation wird ein bestimmtes Element gesucht und durch einen bestimmten Wert, den der Benutzer in das Ersetzungsdialogfeld eingegeben hat, ersetzt. Bei Aufruf der `FindNext`-Methode wird das nächste Element gesucht und ersetzt.

6 Zeilenoperationen

Sie können in der Authentic-Ansicht eine Struktur sich wiederholender Elemente als dynamische Tabelle anlegen, in der jede Zeile eine Instanz des sich wiederholenden Elements darstellt. Nachdem Sie die dynamische Tabelle erstellt haben, kann der Benutzer, der mit der Authentic-Ansicht arbeitet, die Zeilen und deren Daten manipulieren. Diese Zeilenoperationen können mit Hilfe eines Skripts ausgeführt werden.

Wenn die Zeilenoperationen von einem externen Skript ausgeführt werden sollen, sind zwei Schritte erforderlich:

- Zuerst wird überprüft, ob die Zeile, in der sich der Cursor befindet, eine Eigenschaft verwendet. Eine Eigenschaft wie z.B. `IsRowInsertEnabled` wird verwendet und gibt den Wert "True" oder "False" zurück.
- Wenn der zurückgegebene Wert "True" ist, kann die benötigte Zeilenmethode aufgerufen werden.

Im Folgenden sehen Sie eine Liste von Eigenschaften und Methoden, die Zeilenoperationen ausführen. Jede Eigenschaft gibt einen Booleschen Wert zurück und die Methoden haben keine Parameter.

IsRowInsertEnabled	Authentic.RowInsert	Zeileneinfügeoperation
IsRowAppendEnabled	Authentic.RowAppend	Operation zum Anhängen einer Zeile
IsRowDeleteEnabled	Authentic.RowDelete	Operation zum Löschen einer Zeile
IsRowMoveUpEnabled	Authentic.RowMoveUp	Verschiebt die XML-Daten um eine Zeile nach oben
IsRowMoveDownEnabled	Authentic.RowMoveDown	Verschiebt die XML-Daten um eine Zeile nach unten
IsRowDuplicateEnabled	Authentic.RowDuplicate	Dupliziert die aktuelle Zeile

7 Tastaturkürzel

Falls die Authentic 2005 Browser Edition sich im Eingabemodus befindet, stehen die folgenden Tastaturkürzel zur Verfügung:

Strg + P	Dokument drucken
Strg + Z	Rückgängig
Strg + Y	Wiederherstellen
Strg + X	Ausschneiden
Strg + C	Kopieren
Strg + V	Einfügen
Strg + A	Alle auswählen
Strg + F	Suchdialogfeld öffnen
Strg + H	Ersetzungsdialogfeld öffnen

8 Textstatus-Schaltflächen

Die Symbolleiste der Authentic-Ansicht der Authentic 2005 Browser Edition unterstützt Textstatussymbole. Dabei handelt es sich um Symbole, die Elemente mit vordefinierten Textformatierungseigenschaften einfügen. Um diese Funktion verwenden zu können, muss das Element, das als Textstatus-Symbol erstellt werden soll

- im Schema als globales Template deklariert sein und
- das SPS muss die erforderlichen Definitionen enthalten (siehe StyleVision-Dokumentation)

Das Plug-In benötigt den Pfad zum `.bmp`, das als Textstatus-Symbol in der Symbolleiste verwendet werden soll. Diese URL wird als der Wert des [Authentic.TextStateBmpURL](#)-Ereignisses verwendet.

9 Eingabehilfen

Die Authentic 2005 Browser Edition gestattet Ihnen, Eingabehilfen wie in XMLSpy 2005 bereitzustellen. Auf diese Art hat ein Benutzer, der in der Authentic-Ansicht arbeitet, jederzeit Zugriff auf die Elemente, Attribute und Entities, die an einer bestimmten Stelle im Dokument zulässig sind. Die Eingabehilfen sind standardmäßig deaktiviert und werden nicht angezeigt, wenn sie nicht explizit aktiviert werden. Sie können die Eingabehilfen über die folgenden PROPERTYBAG-Parameter im OBJECT-Tag aktivieren:

EntryHelpersEnabled	TRUE / FALSE
EntryHelperSize	Größe in Pixeln
EntryHelperAlignment	Nimmt die SPYAuthenticToolBarAlignment -Werte auf
EntryHelperWindows	Nimmt die SPYAuthenticEntryHelperWindows -Werte auf. Jede Kombination ist zulässig (bit-check)

Anstatt die Parameter im OBJECT-Tag zu verwenden, können Sie auch Eigenschaften und Methoden in der API verwenden:

Eigenschaften

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

Methode

- [Authentic.RedrawEntryHelpers](#)

10 Zusatzpakete

Funktionsbeschreibung

Die Authentic 2005 Browser Edition unterstützt Zusatzpakete, die Funktionalitäten des Programmmoduls erweitern. Derzeit bieten wir die folgenden Pakete zur Rechtschreibprüfung an. Weitere Pakete sind in Vorbereitung.

- Amerikanisches Englisch
- Englisch, komplett (einschließlich medizinisches und juristisches Englisch)
- Alle Sprachen (einschließlich medizinische und juristische Termini)

Installation der Pakete

Bei Aktualisierung eines Plug-In ist es nicht notwendig, ein bereits installiertes Paket neu zu installieren. Das bereits installierte Paket wird von der neuen Version verwendet. Das Plug-In auf dem Client-Rechner liest die XML-Einstellungsdatei (Settings File) auf dem Server und ruft die Informationen über die bereitgestellten Pakete (Versionsnummer, Beschreibung usw.) ab. Außerdem kann der Benutzer über das Paketverwaltungsdialogfeld die Verwendung der Pakete steuern.

Mechanismus

Pakete sind auf dem Server gespeichert und können vom Plug-In jederzeit geladen werden. Sobald ein Paket installiert ist, wird es bei jedem Start der Applikation aktiviert, bis es vom Benutzer entfernt wird. Der Server stellt die Pakete bereit, die bei Bedarf auf den Client heruntergeladen werden. Wenn der Benutzer das Paket nicht verwenden möchte, entfällt das weitere Download.

Aktivieren der Paketverwaltungsfunktion

Um die Paketverwaltungsfunktion zu aktivieren, verwenden Sie die `LoaderSettingsFileURL` als `PROPERTYBAG`-Parameter im `OBJECT`-Tag, der die URL der `LoaderSettingsFile` für die Paketverwaltung definiert.

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

Beispiel für eine LoaderSettings XML-Datei

```
<?xml version="1.0" encoding="UTF-8"?>
<loadersettings>
  <package mode="user_demand" id="SentrySpellChecker_EAM_only"
    category="spelling" version="1">
    <packageurl>
      PlugInLoader/SentrySpellChecker_EAM_only.pck
    </packageurl>
    <description>
      This package contains the Sentry Spell Checker
      engine with an American English lexicon.
    </description>
  </package>
  <package mode="user_demand" id="SentrySpellChecker_EALL"
    category="spelling" version="1">
    <packageurl>
      PlugInLoader/SentrySpellChecker_EALL.pck
    </packageurl>
    <description>
      This package contains the Sentry Spell Checker engine
      with all English lexicons including Legal and Medical.
    </description>
  </package>
  <package mode="user_demand" id="SentrySpellChecker_ALL"
    category="spelling" version="1">
    <packageurl>
      PlugInLoader/SentrySpellChecker_ALL.pck
    </packageurl>
  </package>
```

```
<description>
  This package contains the Sentry Spell Checker engine
  with all lexicons including all English Legal and
  English Medical.
</description>
</package>
</loadersettings>
```

Beachten Sie bitte Folgendes:

- Der Pfad zum Paket muss als Inhalt des `packageurl`-Child von `package` definiert sein. Dieser Pfad kann absolut oder relativ zur HTML-Datei angegeben werden.
- Wenn Sie ein Paket aus der XML-Datei entfernen (durch Entfernen eines `package`-Elements) wird das Paket auf dem Client nicht installiert.
- Das `mode`-Attribut des `package`-Elements dient dazu, den Grad der Kontrolle festzulegen, die der Benutzer über die Entscheidung hat, ob ein Paket installiert werden soll. Bei Verwendung des Werts "user" wird der Benutzer bei jedem Start des Plug-In gefragt, ob das Paket installiert werden soll. Bei Verwendung des Werts "user_demand" wird das Paket installiert, wenn der Benutzer dies explizit verlangt. Im Fall des Rechtschreibprüfpakets wird die Installation des Pakets durch Klicken auf die Symbolleistschaltfläche aktiviert. Bei Verwendung des Werts "force" wird das Paket installiert, ohne dass der Benutzer darüber benachrichtigt wird.
- Der Inhalt des `description` Child von `package` kann editiert werden.

11 Verwendung von XMLData

Mit XMLData haben Sie Zugriff auf die Elemente der aktuell angezeigten XML-Datei. Sie können damit alle erforderlichen Änderungen an den Elementen der XML-Schemastruktur vornehmen. Die Hauptfunktion von XMLData ist:

1. Zugriff auf die Namen und Werte jeder Art von Datenelementen (z.B. Elemente, Attribute)
2. Erstellung von neuen Datenelementen jeder Art
3. Einfügen und Anhängen neuer Elemente
4. Löschen vorhandener Child-Elemente

Wenn Sie mit der XMLData-Oberfläche bereits vertraut sind, da Sie sie in der XMLSpy API verwendet haben, beachten Sie bitte einige wichtige Dinge, wenn Sie neue Elemente erstellen und in die XML-Datei einfügen oder bestehende Elemente umbenennen. Lesen Sie dazu bitte die folgenden Abschnitte "Erstellen und Einfügen neuer XMLData-Objekte" und "Name und Wert von Elementen".

Struktur von XMLData

Bevor Sie die XMLData-Schnittstelle verwenden können, müssen Sie wissen, wie eine bestehende XML-Datei auf eine XMLData-Struktur gemappt wird. Beachten Sie vor allem, dass XMLData keinen eigenen Zweig von Objekten für Attribute hat.

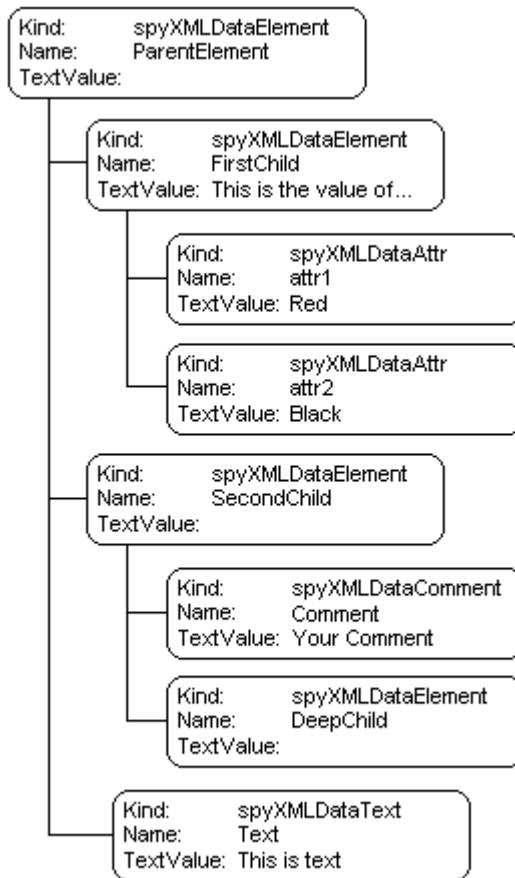
Die Attribute eines Elements sind gleichzeitig Children des Elements. Die [XMLData.Kind](#)-Eigenschaft gibt Ihnen die Möglichkeit, zwischen den verschiedenen Child-Typen eines Elements zu unterscheiden.

Beispiel:

Dieser XML-Code,

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

wird auf die folgende XMLData-Objektstruktur gemappt:



Das Parent-Element aller XML-Elemente in einer Datei ist die Eigenschaft [Authentic.XMLRoot](#). Über dieses XMLData-Objekt werden alle anderen XML-Elemente in der Struktur referenziert.

Name und Wert von Elementen

Mit Hilfe der Eigenschaften [XMLData.Name](#) und [XMLData.TextValue](#) können Sie den Namen und Wert aller XML-Elementtypen abrufen und ändern. Es kann vorkommen, dass mehrere Arten von XMLData-Objekten und leere Elemente keinen Textwert haben, der damit verknüpft ist.

Es ist nicht ratsam, den Namen eines bestehenden XML-Elements in der Authentic-Ansicht zu ändern. Der Name bestimmt, wie der Inhalt des Elements in der Authentic-Ansicht angezeigt wird. Nähere Informationen zu diesem Thema finden Sie in der Dokumentation zu StyleVision.

Erstellen und Einfügen neuer XMLData-Objekte

Um neue XML Language Entities einzufügen, gehen Sie folgendermaßen vor:

1. Erstellen Sie das neue XMLData-Objekt:
Erstellen Sie mit Hilfe der [Authentic.CreateChild](#)-Methode ein neues XMLData-Objekt. Definieren Sie den Namen und Wert, bevor Sie die neue XML-Entity einfügen (siehe Punkt 3).
2. Suchen Sie den richtigen Ordner für das neue XMLData-Objekt:
Um ein neues XMLData-Objekt einzufügen benötigen Sie zuerst eine Referenz auf das Parent-Datenelement. Wenn das neue Child-Datenelement das letzte Child-Datenelement des Parent-Datenelements werden soll, verwenden Sie zum Einfügen

des XMLData-Objekts die [XMLData.AppendChild](#)-Methode.

Falls das neue Child-Datenelement an einer anderen Stelle in der Sequenz der Child-Objekte angelegt werden soll, verwenden Sie [XMLData.GetFirstChild](#) und [XMLData.GetNextChild](#), um den Iterator zu dem Child zu verschieben, vor dem das neue Child eingefügt werden soll.

3. Fügen Sie das neue Child mit Hilfe von [XMLData.InsertChild](#) ein. Das neue Child-Datenelement wird unmittelbar vor dem aktuellen Child-Datenelement eingefügt. Es kann in der Authentic-Ansicht vorkommen, dass zusammen mit dem erstellten Element zusätzliche Child Nodes zur XML-Datei hinzugefügt werden. Dies hängt von den Node-Einstellungen ab, die Sie über StyleVision ändern können.

Im folgenden Beispiel wird ein drittes Child-Datenelement zwischen die Elemente <FirstChild> und <SecondChild> eingefügt:

```
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
```

Child-Elemente sollten in einer bestimmten Reihenfolge eingefügt werden. Fügen Sie nach Möglichkeit keine Attribute nach anderen Child-Elementen in die Sequenz ein. Dies bedeutet, dass Attribute keine übergeordneten Elemente eines anderen Typs haben dürfen und dass ein anderes Element kein nachfolgendes Element vom Typ Attribut haben darf.

In der Authentic-Ansicht ist eine spezielle Vorgangsweise erforderlich, um den Textwert eines XML-Elements anzuzeigen (Dies gilt nicht für Attribute). Jeder Text (oder Inhalt) muss Teil eines eigenen Child Node vom Typ Text sein. Sie können ein solches Element mit Hilfe von [Authentic.CreateChild](#) unter Verwendung des Parameterwerts 6 erstellen. Anstatt den Textwert des Elements direkt festzulegen, müssen Sie den Textwert des Child Node definieren.

Kopieren vorhandener XMLData-Objekte

Die Methoden XMLData.InsertChild und XMLData.AppendChild können nicht zum Einfügen bestehender XMLData-Objekte an einer anderen Stelle in derselben Datei verwendet werden. Diese Methoden funktionieren nur bei neuen XMLData-Objekten.

Anstatt InsertChild oder AppendChild zu verwenden, müssen Sie die Objekthierarchie manuell kopieren. Im Folgenden sehen Sie ein Beispiel für eine in JavaScript geschriebene Funktion, mit der XMLData rekursiv kopiert werden:

```
// this function returns a complete copy of the 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;
}
```

Löschen von XMLData-Objekten

XMLData bietet zwei Methoden zum Löschen von Child-Objekten: [XMLData.EraseAllChildren](#) und [XMLData.EraseCurrentChild](#).

Zum Löschen von XMLData-Objekten benötigen Sie Zugriff auf das Parent-Element der zu löschenden Elemente. Verwenden Sie [XMLData.GetFirstChild](#) und [XMLData.GetNextChild](#), um eine Referenz auf das XMLData-Parent-Objekt zu erhalten.

Beispiele zum Löschen von XML-Elementen finden Sie unter der Beschreibung der Methoden [XMLData.EraseAllChildren](#) und [XMLData.EraseCurrentChild](#).

12 DOM und XMLData

Die XMLData-Schnittstelle bietet im Vergleich zu DOM weniger Methoden, ist um einiges einfacher und gibt Ihnen vollen Zugriff auf die XML-Struktur des aktuellen Dokuments. Die XMLData-Schnittstelle ist ein minimalistischer Lösungsansatz zum Lesen und Ändern bestehender oder neu erstellter XML-Daten. Sie können aber auch eine DOM-Struktur verwenden, weil Sie von einer externen Quelle darauf zugreifen können oder einfach die MSXML DOM-Implementierung bevorzugen.

Die nachfolgend beschriebenen Funktionen **ProcessDOMNode()** und **ProcessXMLDataNode()** konvertieren alle Segmente einer XML-Struktur zwischen XMLData und DOM.

So verwenden Sie die **ProcessDOMNode()**-Funktion:

- Übergeben Sie das Root-Element des DOM-Segments, das Sie in **objNode** konvertieren möchten und
- Übergeben Sie das Plug-In-Objekt mit der CreateChild()-Methode in **objCreator**.

So verwenden Sie die **ProcessXMLDataNode()**-Funktion:

- Übergeben Sie das Root-Element des XMLData-Segments in **objXMLData** und
- Übergeben Sie das mit MSXML in **xmlDoc** erzeugte DOMDocument-Objekt.

```

////////////////////////////////////
// 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;
}
```


Kapitel 4

Objekte

Objekte

1 Authentic

Siehe auch

Methoden

[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](#)

Eigenschaften

[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](#)

Ereignisse

[SelectionChanged](#)

[ControllInitialized](#)

Beschreibung

Authentic-Klasse

1.1 Authentic.ApplyTextState

Siehe auch

Deklaration: `ApplyTextState(elementName als String)`

Beschreibung

Wendet den vom Parameter `elementName` definierten Textstatus an oder entfernt ihn. Ein typisches Beispiel für den Parameter `elementName` wäre "strong" oder "italic".

Ein XML-Dokument enthält Datensegmente, die Subelemente enthalten können. Nehmen wir als Beispiel den folgenden HTML-Code:

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

Aufgrund des HTML-Tags `` wird das Wort "fragment" fett angezeigt. Dies geschieht jedoch nur deshalb, weil der HTML-Parser weiß, dass der Tag `` "bold", also fett bedeutet. XML bietet viel mehr Flexibilität. Sie können jede beliebige Aktion für jeden beliebigen XML-Tag definieren. Wichtig ist, dass Sie mit XML einen Textstatus auf einen Text anwenden können, jedoch muss dieser Textstatus im Schema enthalten sein.

So bedeutet z.B. im Beispiel mit den Dateien `OrgChart.xml` `OrgChart.sps`, `OrgChart.xsd` der Tag `` dasselbe wie "bold". Um diesen Textstatus anzuwenden, wird die Methode

ApplyTextState() aufgerufen. Wie auch bei den die Zeilen- und Bearbeitungsoperationen muss getestet werden, ob der Textstatus angewendet werden kann.

Siehe auch [IsTextStateEnabled](#) und [IsTextStateApplied](#).

1.2 Authentic.attachCallback

Siehe auch

Deklaration: `attachCallback(bstrName als String, varCallback als Variant)`

Beschreibung

In der Authentic-Ansicht stehen Ereignisse zur Verfügung, die mittels benutzerdefinierter Callback-Funktionen abgearbeitet werden können. Alle Ereignishandler akzeptieren keine Parameter und alle Werte, die zurückgegeben werden, werden ignoriert. Um Informationen abzurufen, wenn ein bestimmtes Ereignis abgearbeitet wird, müssen Sie die entsprechenden Eigenschaften des `event`-Objekts aufrufen.

Liste der derzeit verfügbaren Ereignisse:

```
ondragover
ondrop
onkeydown
onkeyup
onkeypress
onmousemove
onmouseup
onmousedown
oneditpaste
oneditcut
oneditcopy
```

Seit Version 3.0.0.0:

```
ondoceditcommand
```

Seit Version: 5.3.0.0:

```
onbuttondoubleclick
```

JavaScript-Beispiel:

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

1.3 AuthenticView

Siehe auch

Deklaration: [AuthenticView](#) als [AuthenticView](#) (schreibgeschützt)

Beschreibung

Gibt ein Objekt zurück, das Zugriff auf Authentic-Ansicht-spezifische Eigenschaften und Methoden verschafft.

Die [AuthenticView](#)-Funktion überlappt sich mit den bestehenden ansichtsspezifischen Funktionen. Zukünftige Versionen der Authentic-Ansicht werden alle ansichtsspezifischen Funktionen enthalten. Das AuthenticView-Objekt ist die empfohlene Schnittstelle für alle zukünftigen Implementierungen.

Beispiele

Nähere Informationen zur Verwendung des AuthenticView-Objekts finden Sie unter "[Sortieren dynamischer Tabellen mit bubble-sort](#)".

1.4 Authentic.AutoHideUnusedCommandGroups

Deklaration: `AutoHideUnusedCommandGroups` als `Boolean`

Beschreibung

True, wenn nicht verwendete Symbolleisten-Befehlsgruppen automatisch ausgeblendet werden (z.B. XML-Tabellenbefehle, wenn die aktuelle SPS-Datei XML-Tabellen nicht unterstützt)

Standardwert: true

1.5 Authentic.BaseURL

Deklaration: BaseURL als String

Beschreibung

Diese Eigenschaft definiert eine alternative URL, die zur Auflösung von relativen Pfadangaben verwendet wird.

Wenn keine URL definiert wurde, wird der Ordner der aktuellen XML-Datei verwendet.

1.6 Authentic.ClearSelection

Deklaration: [ClearSelection\(\)](#)

Beschreibung

Mit dieser Methode wird die aktuelle Auswahl aufgehoben. Danach schlägt jeder Versuch, die Auswahl mit Hilfe der CurrentSelection-Eigenschaft zu holen, fehl, bis ein Node ausgewählt wird oder ein erfolgreicher Aufruf von [SelectionSet\(\)](#) verarbeitet wurde.

Um einen Node löschen zu können, der die aktuelle Auswahl enthält, muss ein andere Node ausgewählt oder die Auswahl mittels ClearSelection() aufgehoben werden.

1.7 Authentic.ClearUndoRedo

Deklaration: [ClearUndoRedo\(\)](#)

Beschreibung

Mit dieser Methode wird der gesamte Rückgängig/Wiederherstellen-Puffer gelöscht.

1.8 Authentic.ControlInitialized

Siehe auch

Deklaration: [ControlInitialized](#)

Beschreibung

Dieses Ereignis wird bei der Erstellung und Initialisierung des Controls ausgelöst.

Siehe auch [Connection Point-Ereignisse](#).

1.9 Authentic.CreateChild

Siehe auch

Deklaration: `CreateChild`(*nKind* als [SPYXMLDataKind](#)) als [XMLData](#)

Rückgabewert

Neuer XML Node

Beschreibung

Die CreateChild-Methode dient zur Erstellung neuer Nodes, die Sie über die [XMLData](#)-Schnittstelle in die XML-Struktur des aktuellen Dokuments einfügen können.

Siehe auch [XMLData.AppendChild](#) und [XMLData.InsertChild](#)

1.10 Authentic.CurrentSelection

Siehe auch

Deklaration: [CurrentSelection](#) als [AuthenticSelection](#)

Beschreibung

Die Eigenschaft verschafft Zugriff auf die aktuelle Auswahl in der Authentic-Ansicht.

Mit dem unten gezeigten Beispielcode wird der komplette Text der aktuellen Auswahl abgerufen:

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;
}
```

1.11 Authentic.DesignDataLoadObject

Siehe auch

Deklaration: [DesignDataLoadObject](#) als [AuthenticLoadObject](#)

Beschreibung

Das DesignDataLoadObject enthält eine Referenz auf das SPS-Dokument. Mit Hilfe des SPS-Dokuments, das normalerweise in StyleVision erstellt wird, wird die WYSIWYG-Editierumgebung erstellt.

Ein Beispiel dazu finden Sie unter [SchemaLoadObject](#).

1.12 Authentic.EditClear

Siehe auch

Deklaration: [EditClear](#)

Beschreibung

Hebt die aktuelle Auswahl auf.

1.13 Authentic.EditCopy

Siehe auch

Deklaration: [EditCopy](#)

Beschreibung

Kopiert die aktuelle Auswahl in die Zwischenablage.

1.14 Authentic.EditCut

Siehe auch

Deklaration: [EditCut](#)

Beschreibung

Schneidet die aktuelle Auswahl aus dem Dokument aus und kopiert sie in die Zwischenablage.

1.15 Authentic.EditPaste

Siehe auch

Deklaration: [EditPaste](#)

Beschreibung

Fügt den Inhalt aus der Zwischenablage in das Dokument ein.

1.16 Authentic.EditRedo

Siehe auch

Deklaration: [EditRedo](#)

Beschreibung

Stellt die zuletzt rückgängig gemachte Aktion wieder her.

1.17 Authentic.EditSelectAll

Siehe auch

Deklaration: [EditSelectAll](#)

Beschreibung

Mit dieser Methode wird das komplette Dokument ausgewählt.

1.18 Authentic.EditUndo

Siehe auch

Deklaration: [EditUndo](#)

Beschreibung

Die letzte Aktion wird rückgängig gemacht.

1.19 Authentic.EnableModifications

Deklaration: [EnableModifications](#) als [Boolean](#)

Beschreibung

"True", wenn die durch die Authentic 2005 Browser Edition am XML-Inhalt vorgenommenen Änderungen aktiviert sind. Siehe auch [Modified](#)-Eigenschaft.

Standardwert: TRUE

1.20 Authentic.EntryHelperAlignment

Deklaration: [EntryHelperAlignment](#) als [SPYAuthenticToolbarAlignment](#)

Beschreibung

Diese Eigenschaft dient zum Festlegen der Position der Eingabehilfen. Der Standardwert ist 3. Damit werden die Eingabehilfen an der rechten Seite platziert.

1.21 Authentic.EntryHelpersEnabled

Deklaration: `EntryHelpersEnabled` als `Boolean`

Beschreibung

"True", wenn die Eingabehilfen der Authentic 2005 Browser Edition aktiviert sind.
Diese Eigenschaft kann verwendet werden, um die Eingabehilfen zu aktivieren bzw. zu deaktivieren.

Standardwert: FALSE

1.22 Authentic.EntryHelperSize

Deklaration: `EntryHelperSize` als `Integer`

Beschreibung

Mit Hilfe dieser Eigenschaft können Sie die anfangs verwendete Größe der Eingabehilfen in Pixeln angeben. Setzen Sie die Eigenschaft auf -1, wenn dieser Wert ignoriert werden soll.

Standardwert: -1

1.23 Authentic.EntryHelperWindows

Deklaration: [EntryHelperWindows](#) als [SPYAuthenticEntryHelperWindows](#)

Beschreibung

Mit Hilfe dieser Eigenschaft können Sie definieren, welche der Eingabehilfen angezeigt werden sollen. Die Werte können kombiniert werden, um mehrere Eingabehilfenfenster anzuzeigen. Der Standardwert ist 7. Bei Auswahl dieses Werts werden alle drei Eingabehilfen eingeblendet.

1.24 Authentic.event

Siehe auch

Deklaration: `event` als [AuthenticEvent](#)

Beschreibung

Die Ereigniseigenschaft enthält ein Ereignis-Datenobjekt mit Informationen über das aktuelle Ereignis.

1.25 Authentic.FindDialog

Siehe auch

Deklaration: [FindDialog](#)

Beschreibung

Zeigt den FindDialog an.

Siehe auch [Suchen und Ersetzen](#).

1.26 Authentic.FindNext

Siehe auch

Deklaration: [FindNext](#)

Beschreibung

Diese Methode führt eine "Weitersuchen"-Operation aus.

Siehe auch [Suchen und Ersetzen](#).

1.27 Authentic.GetAllAttributes

Siehe auch

Deklaration: `GetAllAttributes(pForElement als XMLData, pElements als Variant)`

Beschreibung

GetAllAttributes() gibt die für das angegebene Element zulässigen Attribute als String-Array zurück.

JavaScript-Beispiel:

```
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-Beispiel:

```
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
End Sub
```

```
    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
```

1.28 Authentic.GetAllowedElements

Siehe auch

Deklaration: `GetAllowedElements(nAction als SPYAuthenticElementActions, pStartElement als XMLData, pEndElement als XMLData, pElements als Variant)`

Beschreibung

`GetAllowedElements()` gibt die für die verschiedenen von `nAction` definierten Aktionen zulässigen Elemente zurück.

JavaScript-Beispiel:

```
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-Beispiel:

```
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
```

1.29 Authentic.GetFileVersion

Siehe auch

Deklaration: `GetFileVersion`(*strVersion* als `String`)

Beschreibung

Die Methode gibt die Version der Komponente als String im Format 5.0.0.0 zurück.

1.30 Authentic.GetNextVisible

Siehe auch

Deklaration: `GetNextVisible(pElement als XMLData) als XMLData`

Beschreibung

Die Methode liefert das nächste sichtbare XML-Element im Dokument.

1.31 Authentic.GetPreviousVisible

Siehe auch

Deklaration: `GetPreviousVisible(pElement als XMLData) als XMLData`

Beschreibung

Die Methode liefert das vorhergehende XML-Element im Dokument.

1.32 Authentic.IsEditClearEnabled

Siehe auch

Deklaration: [IsEditClearEnabled](#) als [Boolean](#)

Beschreibung

True, wenn [EditClear](#) möglich ist.

Siehe auch [Bearbeitungsoperationen](#).

1.33 Authentic.IsEditCopyEnabled

Siehe auch

Deklaration: [IsEditCopyEnabled](#) als [Boolean](#)

Beschreibung

True, wenn Kopieren in die Zwischenablage möglich ist.

Siehe auch [EditCopy](#) und [Bearbeitungsoperationen](#).

1.34 Authentic.IsEditCutEnabled

Siehe auch

Deklaration: [IsEditCutEnabled](#) als [Boolean](#)

Beschreibung

True, wenn [EditCut](#) derzeit möglich ist.

Siehe auch [Bearbeitungsoperationen](#).

1.35 Authentic.IsEditPasteEnabled

Siehe auch

Deklaration: [IsEditPasteEnabled](#) als [Boolean](#)

Beschreibung

True, wenn [EditPaste](#) möglich ist.

Siehe auch [Bearbeitungsoperationen](#).

1.36 Authentic.IsEditRedoEnabled

Siehe auch

Deklaration: [IsEditRedoEnabled](#) als [Boolean](#)

Beschreibung

True, wenn [EditRedo](#) derzeit möglich ist.

Siehe auch [Bearbeitungsoperationen](#).

1.37 Authentic.IsEditUndoEnabled

Siehe auch

Deklaration: [IsEditUndoEnabled](#) als [Boolean](#)

Beschreibung

True, wenn [EditUndo](#) möglich ist.

Siehe auch [Bearbeitungsoperationen](#).

1.38 Authentic.IsFindNextEnabled

Siehe auch

Deklaration: [IsFindNextEnabled](#) als [Boolean](#)

Beschreibung

True, wenn FindNext derzeit möglich ist. False, wenn der Suchbegriff nicht mehr vorkommt.

Siehe auch [Suchen und Ersetzen](#) und [FindDialog](#).

1.39 Authentic.IsRowAppendEnabled

Siehe auch

Deklaration: [IsRowAppendEnabled](#) als [Boolean](#)

Beschreibung

True, wenn [RowAppend](#) möglich ist.

Siehe auch [Zeilenoperationen](#).

1.40 Authentic.IsRowDeleteEnabled

Siehe auch

Deklaration: [IsRowDeleteEnabled](#) als [Boolean](#)

Beschreibung

True, wenn [RowDelete](#) möglich ist.

Siehe auch [Zeilenoperationen](#).

1.41 Authentic.IsRowDuplicateEnabled

Siehe auch

Deklaration: [IsRowDuplicateEnabled](#) als [Boolean](#)

Beschreibung

True, wenn [RowDuplicate](#) möglich ist.

Siehe auch [Zeilenoperationen](#).

1.42 Authentic.IsRowInsertEnabled

Siehe auch

Deklaration: [IsRowInsertEnabled](#) als [Boolean](#)

Beschreibung

True, wenn [RowInsert](#) möglich ist.

Siehe auch [Zeilenoperationen](#).

1.43 Authentic.IsRowMoveDownEnabled

Siehe auch

Deklaration: [IsRowMoveDownEnabled](#) als [Boolean](#)

Beschreibung

True, wenn [RowMoveDown](#) möglich ist.

Siehe auch [Zeilenoperationen](#).

1.44 Authentic.IsRowMoveUpEnabled

Siehe auch

Deklaration: [IsRowMoveUpEnabled](#) als [Boolean](#)

Beschreibung

True, wenn [RowMoveUp](#) möglich ist.

Siehe auch [Zeilenoperationen](#).

1.45 Authentic.IsTextStateApplied

Siehe auch

Deklaration: `IsTextStateApplied(elementName als String) als Boolean`

Beschreibung

Überprüft, ob der Textstatus bei der aktuellen Selektion/Position bereits angewendet wurde. Typische Beispiele für den Parameter `elementName` sind "strong" und "italic".

1.46 Authentic.IsTextStateEnabled

Siehe auch

Deklaration: `IsTextStateEnabled(elementName als String) als Boolean`

Beschreibung

Überprüft, ob ein Textstatus für die aktuelle Selektion/Position angewendet werden kann. Typische Beispiele für den Parameter `elementName` sind "strong" und "italic".

1.47 Authentic.LoadXML

Siehe auch

Deklaration: `LoadXML(xmlString als String)`

Beschreibung

Lädt den angegebenen String als das aktuelle XML-Dokument in die Authentic-Ansicht. Der neue Inhalt wird sofort angezeigt.

1.48 Authentic.MarkUpView

Siehe auch

Deklaration: `MarkUpView`(*kind* als [SPYAuthenticMarkupVisibility](#))

Beschreibung

Standardmäßig wird das Dokument wie ein HTML-Dokument angezeigt. Manchmal ist es jedoch hilfreich, wenn die Editier-Tags angezeigt werden. Mit Hilfe dieser Methode können Sie verschiedene Arten von Markup-Tags anzeigen.

1.49 Authentic.Modified

Deklaration: Modified als Boolean

Beschreibung

True, wenn XML-Inhalt geändert wurde.

Diese Eigenschaft ist schreibgeschützt.

1.50 Authentic.Print

Siehe auch

Deklaration: [Print](#)

Beschreibung

Druckt das aktuell editierte Dokument.

1.51 **Authentic.PrintPreview**

Siehe auch

Deklaration: [PrintPreview](#)

Beschreibung

Zeigt eine Druckvorschau des gerade editierten Dokuments an.

1.52 Authentic.RedrawEntryHelpers

Deklaration: [RedrawEntryHelpers\(\)](#)

Beschreibung

RedrawEntryHelpers holt die Werte aus den Eigenschaften [EntryHelpersEnabled](#), [EntryHelperAlignment](#), [EntryHelperSize](#) und [EntryHelperWindows](#) und zeichnet die Eingabehilfenfenster neu.

1.53 Authentic.ReloadToolbars

Deklaration: [ReloadToolbars\(\)](#)

Beschreibung

ReloadToolbars liest die [ToolbarRows](#) Sammlung und zeichnet die Symbolleisten und Ansichten neu.

1.54 Authentic.ReplaceDialog

Siehe auch

Deklaration: [ReplaceDialog](#)

Beschreibung

Zeigt den ReplaceDialog an.

Siehe auch [Suchen und Ersetzen](#).

1.55 Authentic.Reset

Siehe auch

Deklaration: [Reset](#)

Beschreibung

Wurde durch andere Funktionen ersetzt und es wird angeraten diese nicht mehr zu verwenden.

1.56 Authentic.RowAppend

Siehe auch

Deklaration: [RowAppend](#)

Beschreibung

Hängt eine Zeile an der aktuellen Position an.

Siehe auch [Zeilenoperationen](#).

1.57 Authentic.RowDelete

Siehe auch

Deklaration: [RowDelete](#)

Beschreibung

Löscht die aktuell ausgewählte(n) Zeile(n).

Siehe auch [Zeilenoperationen](#).

1.58 Authentic.RowDuplicate

Siehe auch

Deklaration: [RowDuplicate](#)

Beschreibung

Diese Methode dupliziert die aktuell ausgewählten Zeilen.

Siehe auch [Zeilenoperationen](#).

1.59 Authentic.RowInsert

Siehe auch

Deklaration: [RowInsert](#)

Beschreibung

Fügt eine neue Zeile unmittelbar oberhalb der aktuellen Auswahl ein.

Siehe auch [Zeilenoperationen](#).

1.60 Authentic.RowMoveDown

Siehe auch

Deklaration: [RowMoveDown](#)

Beschreibung

Verschiebt die aktuelle Zeile um eine Position nach unten.

Siehe auch [Zeilenoperationen](#).

1.61 Authentic.RowMoveUp

Siehe auch

Deklaration: [RowMoveUp](#)

Beschreibung

Verschiebt die aktuelle Zeile um eine Position nach oben.

Siehe auch [Zeilenoperationen](#).

1.62 Authentic.Save

Siehe auch

Deklaration: [Save](#)

Beschreibung

Speichert das Dokument unter der in der Eigenschaft [XMLDataSaveUrl](#) definierten URL. Bei Untrusted Versionen können Sie auch einen vollständigen lokalen Pfad verwenden.

Das Plug-In sendet einen HTTP PUT Request an den Server, um die aktuell angezeigte XML-Datei zu speichern.

1.63 Authentic.SaveButtonAutoEnable

Deklaration: `SaveButtonAutoEnable` als `Boolean`

Beschreibung

Ist diese Eigenschaft auf TRUE gesetzt, wird der Ein/Aus-Status der Schaltfläche "Speichern" in der Symbolleiste des Control entsprechend dem Modified Flag des Dokuments gesetzt.

Der Standardwert ist FALSE.

1.64 Authentic.SavePOST

Siehe auch

Deklaration: [SavePOST](#)

Beschreibung

Speichert das Dokument unter der von der Eigenschaft [XMLDataSaveUrl](#) definierten URL. Bei Untrusted Versionen können Sie auch einen vollständigen lokalen Pfad verwenden. Das Plug-In sendet einen HTTP POST Request an den Server, um die aktuell angezeigte XML-Datei zu speichern.

Überprüfen, ob eine Datei gespeichert wurde oder nicht

Wenn das Authentic Plug-In eine HTTP-Antwort ≥ 300 erhält, geht es davon aus, dass die Datei **nicht** gespeichert wurde und zeigt daraufhin (standardmäßig) eine Meldung an, die den HTTP-Fehler enthält. Anschließend wird eine zweite (deaktivierbare) Meldung angezeigt, in der der Benutzer darauf hingewiesen wird, dass die Datei nicht gespeichert wurde. Es obliegt dem Applikationsentwickler, sicherzustellen, dass die korrekte HTTP-Antwort zurückgegeben wird, je nachdem ob das Dokument gespeichert werden konnte oder nicht. Der entsprechende PHP-Code könnte etwa folgendermaßen aussehen:

```
<?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" );
?>
```

1.65 Authentic.SaveXML

Siehe auch

Deklaration: [SaveXML](#) als [String](#)

Rückgabewert

XML-Struktur als String

Beschreibung

Speichert die aktuellen XML-Daten in einem String, der an den Caller zurückgegeben wird.

1.66 Authentic.SchemaLoadObject

Siehe auch

Deklaration: [SchemaLoadObject](#) als [AuthenticLoadObject](#)

Beschreibung

Das SchemaLoadObject enthält eine Referenz zum XML-Schema-Dokument für die aktuelle XML-Datei. Das Schema-Dokument wird normalerweise mit XMLSpy 2005 generiert.

Beispiel

```
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
```

Der obige Code definiert alle URL-Eigenschaften der zu ladenden Objekte und ruft [StartEditing](#) zum Laden und Anzeigen der Dateien auf. Bei Untrusted Versionen können Sie auch einen vollständigen lokalen Pfad verwenden. Der aktuelle Inhalt und Status des Plug-In werden zurückgesetzt.

1.67 Authentic.SelectionChanged

Siehe auch

Deklaration: [SelectionChanged](#) als [VT_0019](#)

Beschreibung

Dieses Ereignis wird ausgelöst, wenn der Benutzer die aktuelle Auswahl ändert.

Siehe auch [Connection Point-Ereignisse](#).

1.68 Authentic.SelectionMoveTabOrder

Siehe auch

Deklaration: `SelectionMoveTabOrder`(*bForward* als `Boolean`, *bTag* als `Boolean`)

Beschreibung

`SelectionMoveTabOrder()` verschiebt die aktuelle Auswahl nach vorne oder nach hinten.

Wenn `bTag` "false" ist und die letzte Zelle einer Tabelle ausgewählt ist, wird eine neue Zeile hinzugefügt.

1.69 Authentic.SelectionSet

Siehe auch

Deklaration: SelectionSet(*pStartElement* als XMLData, *nStartPos* als long, *pEndElement* als XMLData, *nEndPos* als long) als Boolean

Beschreibung

Verwenden Sie SelectionSet(), um eine neue Auswahl in der Authentic-Ansicht zu definieren. Sie können pEndElement auf Null (nichts) setzen, wenn die Auswahl nicht mehr als ein (pStartElement) XML-Element umfassen soll.

1.70 Authentic.SetUnmodified

Deklaration: [SetUnmodified\(\)](#)

Beschreibung

Nach Aufruf dieser Methode wird der aktuelle Zustand der Rückgängig/Wiederholen-Puffers als ungeänderter Zustand des zugrunde liegenden XML-Dokuments verwendet und der [Modified](#) Flag wird auf FALSE gesetzt.

1.71 Authentic.StartEditing

Siehe auch

Deklaration: [StartEditing](#) als [Boolean](#)

Rückgabewert

True, wenn alle Dateien erfolgreich geladen wurden und angezeigt werden.

Beschreibung

Startet die Editierung des aktuellen Dokuments. Es ist wichtig, dass zuerst die Eigenschaften der Lade-Objekte [SchemaLoadObject](#), [DesignDataLoadObject](#) und [XMLDataLoadObject](#) festgelegt werden.

1.72 Authentic.StartSpellChecking

Deklaration: [StartSpellChecking\(\)](#)

Beschreibung

Dieser Befehl öffnet das Dialogfeld für die Rechtschreibprüfung, wenn ein Paket mit dem Rechtschreibprüf-Prozessor und den erforderlichen Wörterbüchern verfügbar und aktiviert ist.

1.73 Authentic.TextStateBmpURL

Deklaration: `TextStateBmpURL` als `String`

Beschreibung

Die URL, von der die Bitmaps für die Textstatus-Symbole abgerufen werden sollen.
Wenn keine URL definiert ist, werden keine Textstatus-Symbole angezeigt.

Beispiele

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

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

1.74 Authentic.TextStateToolBarLine

Deklaration: `TextStateToolBarLine` als `long`

Beschreibung

Die Symbolleiste (Zeilennummer) in der die Textstatus-Symbole angezeigt werden sollen. Textstatus-Symbole können an bestehende Symbolleisten angehängt oder in neue Symbolleisten platziert werden.
Standardwert: 1

1.75 Authentic.ToolbarRows

Deklaration: `ToolbarRows` als [AuthenticToolbarRows](#)

Beschreibung

Enthält eine Sammlung aller Symbolleisten, die angezeigt werden sollen. Informationen zum Löschen, Hinzufügen oder Ändern von Symbolleisten finden Sie unter [AuthenticToolbarRows](#).

1.76 Authentic.ToolbarsEnabled

Deklaration: ToolbarsEnabled als Boolean

Beschreibung

True, wenn Authentic 2005 Browser Edition-Symbolleisten aktiviert sind.
Diese Eigenschaft kann zum Aktivieren oder Deaktivieren aller Symbolleisten verwendet werden.
Standardwert: true

1.77 Authentic.ToolbarToolTipsEnabled

Deklaration: `ToolbarToolTipsEnabled` als `Boolean`

Beschreibung

True, wenn die Tooltips für die Authentic 2005 Browser Edition-Symbolleisten aktiviert sind.
Standardwert: true

1.78 Authentic.UICommands

Deklaration: `UICommands` als [AuthenticCommands](#)

Beschreibung

Enthält eine Sammlung aller verfügbaren Symbolleistenbefehle (mit Beschreibung).
Für diese Befehle können in den verschiedenen Symbolleisten Schaltflächen platziert werden.
Schreibgeschützt.

Beispiel

Holt alle verfügbaren Befehle, Befehlsgruppen, Beschreibungen und zeigt Sie in einem Nachrichtenfeld an.

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

1.79 Authentic.ValidateDocument

Siehe auch

Deklaration: `ValidateDocument(showResults als Boolean) als Boolean`

Rückgabewert

Ergebnis der Validierung

Beschreibung

Überprüft die aktuellen XML-Daten auf Ihre Gültigkeit hinsichtlich der XML-Schemadaten. Wenn der Parameter `showResults` `FALSE` ist, werden die Validierungsfehler nicht angezeigt, ansonsten werden sie angezeigt.

1.80 Authentic.validationBadData

Siehe auch

Deklaration: `validationBadData` als [XMLData](#)

Beschreibung

Diese Eigenschaft kann zusätzliche Informationen über den letzten Validierungsfehler liefern. Sie wird nach Aufrufen von `ValidateDocument()` gesetzt und ist entweder Null oder sie enthält eine Referenz auf das fehlerverursachende XML-Element.

1.81 Authentic.validationMessage

Siehe auch

Deklaration: `validationMessage` als `String`

Beschreibung

Wenn die Validierung fehlgeschlagen ist (nach Aufruf von `ValidateDocument`), speichert diese Eigenschaft einen `String` mit der Fehlermeldung.

1.82 Authentic.XMLDataLoadObject

Siehe auch

Deklaration: [XMLDataLoadObject](#) als [AuthenticLoadObject](#)

Beschreibung

Das XMLDataLoadObject enthält eine Referenz auf das XML-Dokument, das gerade bearbeitet wird. Das XML-Dokument wird normalerweise mit Hilfe von XMLSpy 2005 definiert und anhand einer Datenbank oder eines anderen Geschäftsprozesses generiert.

Beispiel dazu finden Sie unter [SchemaLoadObject](#).

1.83 Authentic.XMLDataSaveUrl

Siehe auch

Deklaration: [XMLDataSaveUrl](#) als [String](#)

Beschreibung

Wenn die XML-Daten geändert wurden, können diese über eine URL wieder auf einem Server gespeichert werden. Mit dieser Eigenschaft wird der Pfad definiert, unter dem die XML-Daten gespeichert werden. Bei Untrusted Versionen können Sie auch den vollständigen lokalen Pfad verwenden.

Siehe auch die Methoden [Authentic.Save](#) und [Authentic.SavePOST](#).

1.84 Authentic.XMLRoot

Siehe auch

Deklaration: [XMLRoot](#) als [XMLData](#)

Beschreibung

XMLRoot ist das Parent-Element der aktuell angezeigten XML-Struktur. Über die [XMLData](#) Schnittstelle haben Sie vollen Zugriff auf den kompletten Dateiinhalt.

Weitere Informationen dazu siehe auch [Using XMLData](#).

1.85 Authentic.XMLTable

Deklaration: [XMLTable](#) als [AuthenticXMLTableCommands](#)

Beschreibung

Holt einen Satz aller XML-Tabellenbefehle.
Schreibgeschützt.

2 AuthenticCommand

Methoden

Eigenschaften

[CommandID](#)

[Group](#)

[ShortDescription](#)

[Name](#)

2.1 AuthenticCommand.CommandID

Deklaration: `CommandID` als [SPYAuthenticCommand](#)

Beschreibung

Enthält die CommandId des Befehls
Mögliche Werte: siehe AuthenticToolBarButton
Schreibgeschützt

Beispiel

Siehe Beispiel unter [Authentic.UICommands](#)

2.2 AuthenticCommand.Group

Deklaration: [Group](#) als [SPYAuthenticCommandGroup](#)

Beschreibung

Die CommandGroup, zu der der Befehl gehört.
Schreibgeschützt

Beispiel

Siehe Beispiel unter [Authentic.UICommands](#)

2.3 AuthenticCommand.ShortDescription

Deklaration: [ShortDescription](#) als [String](#)

Beschreibung

Kurzbeschreibung des Befehls (z.B. der Tooltipp-Text)
Schreibgeschützt

Beispiel

Siehe Beispiel unter [Authentic.UICommands](#)

2.4 AuthenticCommand.Name

Deklaration: Name als String

Beschreibung

Für zukünftige Verwendungszwecke

3 AuthenticCommands

Methoden

[Item](#)

Eigenschaften

[Count](#)

3.1 AuthenticCommands.Count

Deklaration: Count als long

Beschreibung

Anzahl der verfügbaren Benutzerschnittstellen-Befehle.
Schreibgeschützt

3.2 AuthenticCommands.Item

Deklaration: `Item` (`nPosition` als `long`) als [AuthenticCommand](#)

Beschreibung

Liefert den Befehl in Position `nPosition`. `nPosition` beginnt mit 1.

4 AuthenticDataTransfer

Siehe auch

Methoden

[getData](#)

Eigenschaften

[dropEffect](#)

[ownDrag](#)

[type](#)

Beschreibung

AuthenticDataTransfer-Schnittstelle.

4.1 AuthenticDataTransfer.dropEffect

Siehe auch

Deklaration: `dropEffect` als `long`

Beschreibung

Die Eigenschaft speichert das Drop-Verhalten aus dem Standard-Ereignishandler. Sie können das Drop-Verhalten definieren, wenn Sie diesen Wert ändern und [AuthenticEvent.cancelBubble](#) auf TRUE setzen.

4.2 AuthenticDataTransfer.getData

Siehe auch

Deklaration: [getData](#) als [Variant](#)

Beschreibung

getData holt die aktuellen Daten, die mit diesem dataTransfer-Objekt verknüpft sind. Siehe auch [AuthenticDataTransfer.type](#).

4.3 AuthenticDataTransfer.ownDrag

Siehe auch

Deklaration: `ownDrag` als `Boolean`

Beschreibung

Die Eigenschaft ist TRUE, wenn die Quelle des Drag-Vorgangs von innerhalb der Authentic-Ansicht kommt.

4.4 AuthenticDataTransfer.type

Siehe auch

Deklaration: `type` als `String`

Beschreibung

Enthält den Datentyp, den Sie mit der [AuthenticDataTransfer.getData](#) Methode holen.

Derzeit werden die folgenden Datentypen unterstützt:

OWN	Daten aus dem Plug-In selbst
TEXT	Nur Text
UNICODETEXT	Nur Text als UNICODE
IUNKNOWN	Unbekannte Referenz auf IDataObject-Instanz

5 AuthenticEvent

Siehe auch

Eigenschaften

[altKey](#)

[altLeft](#)

[ctrlKey](#)

[ctrlLeft](#)

[shiftKey](#)

[shiftLeft](#)

[keyCode](#)

[repeat](#)

[button](#)

[clientX](#)

[clientY](#)

[dataTransfer](#)

[srcElement](#)

[fromElement](#)

[propertyName](#)

[cancelBubble](#)

[returnValue](#)

[type](#)

Beschreibung

AuthenticEvent-Schnittstelle.

5.1 AuthenticEvent.altKey

Siehe auch

Deklaration: altKey als Boolean

Beschreibung

True, wenn Sie die rechte ALT-Taste drücken.

5.2 AuthenticEvent.altLeft

Siehe auch

Deklaration: altLeft als Boolean

Beschreibung

True, wenn die linke ALT-Taste gedrückt wird.

5.3 AuthenticEvent.button

Siehe auch

Deklaration: `button` als `long`

Beschreibung

Gibt an, welche Maustaste gedrückt wird:

- 0 Keine Taste
- 1 Linke Taste
- 2 Rechte Taste
- 3 Sowohl die linke als auch die rechte Taste
- 4 Mittlere Taste
- 5 Linke und mittlere Taste
- 6 Rechte und mittlere Taste
- 7 Alle drei Tasten

Die Ereignisse `onbuttondown` und `onbuttonup` setzen den Tastenwert auf verschiedene Arten. Das `onbuttonup`-Ereignis setzt immer nur den Wert für die Taste, die gerade losgelassen wurde und das Ereignis ausgelöst hat, unabhängig davon, welche anderen Tasten in diesem Moment ebenfalls gedrückt sind.

5.4 AuthenticEvent.cancelBubble

Siehe auch

Deklaration: `cancelBubble` als `Boolean`

Beschreibung

Setzt `cancelBubble` auf TRUE, wenn der Standard-Ereignishandler nicht aufgerufen werden soll.

5.5 AuthenticEvent.clientX

Siehe auch

Deklaration: `clientX` als `long`

Beschreibung

X-Wert der aktuellen Mausposition in den Client-Koordinaten.

5.6 AuthenticEvent.clientY

Siehe auch

Deklaration: `clientY` als `long`

Beschreibung

Y-Wert der aktuellen Mausposition in den Client-Koordinaten.

5.7 AuthenticEvent.ctrlKey

Siehe auch

Deklaration: ctrlKey als Boolean

Beschreibung

True, wenn die rechte Strg-Taste gedrückt wird.

5.8 AuthenticEvent.ctrlLeft

Siehe auch

Deklaration: ctrlLeft als Boolean

Beschreibung

True, wenn die linke Strg-Taste gedrückt wird.

5.9 AuthenticEvent.dataTransfer

Siehe auch

Deklaration: [dataTransfer](#) als [Variant](#)

Beschreibung

Eigenschaft dataTransfer

5.10 AuthenticEvent.fromElement

Siehe auch

Deklaration: [fromElement](#) als [Variant](#)

Beschreibung

Derzeit wird diese Eigenschaft von keinem Ereignis gesetzt.

5.11 AuthenticEvent.keyCode

Siehe auch

Deklaration: `keyCode` als `long`

Beschreibung

Tastaturcode der aktuell gedrückten Taste.

Diese Eigenschaft ist mit Lese- und Schreibzugriff.

5.12 AuthenticEvent.propertyName

Siehe auch

Deklaration: `propertyName` als `String`

Beschreibung

Derzeit wird diese Eigenschaft von keinem Ereignis gesetzt.

5.13 AuthenticEvent.repeat

Siehe auch

Deklaration: repeat als Boolean

Beschreibung

True, wenn sich das onkeydown-Ereignis wiederholt.

5.14 AuthenticEvent.returnValue

Siehe auch

Deklaration: returnValue als Variant

Beschreibung

Verwenden Sie returnValue, um einen Rückgabewert für Ihren Ereignishandler zu setzen.

5.15 AuthenticEvent.shiftKey

Siehe auch

Deklaration: `shiftKey` als `Boolean`

Beschreibung

True, wenn die rechte Umschalttaste gedrückt wird.

5.16 AuthenticEvent.shiftLeft

Siehe auch

Deklaration: `shiftLeft` als `Boolean`

Beschreibung

True, wenn die linke Umschalttaste gedrückt wird.

5.17 AuthenticEvent.srcElement

Siehe auch

Deklaration: [srcElement](#) als [Variant](#)

Beschreibung

Element, das das aktuelle Ereignis auslöst.
Dies ist normalerweise ein [XMLData](#)-Objekt.

Seit Version 3.0.0.0:

Die Eigenschaft kann auch eine Referenz auf ein [AuthenticCommand](#)-Objekt enthalten, wenn Sie von einem ondoceditcommand-Ereignis aus gesetzt wurde.

5.18 AuthenticEvent.type

Siehe auch

Deklaration: type als String

Beschreibung

Derzeit wird diese Eigenschaft von keinem Ereignis gesetzt.

6 AuthenticLoadObject

Siehe auch

Eigenschaften

[String](#)

[URL](#)

Beschreibung

Das XMLSpyXMLLoadSave-Objekt dient dazu, die Quelle für die Dateien festzulegen, die geladen werden müssen. Sie können den Inhalt entweder direkt über die String-Eigenschaft festlegen oder als externen Pfad über die URL-Eigenschaft.

Nähere Informationen zur Verwendung der Eigenschaften finden Sie auch unter [Authentic.SchemaLoadObject](#), [Authentic.DesignDataLoadObject](#) und [Authentic.XMLDataLoadObject](#).

6.1 AuthenticLoadObject.String

Siehe auch

Deklaration: `String` als `String`

Beschreibung

Mit Hilfe dieser Eigenschaft können Sie die XML-Struktur von einem String aus setzen. Wenn Sie diese Eigenschaft verwenden möchten, muss die URL-Eigenschaft des Objekts leer sein.

6.2 AuthenticLoadObject.URL

Siehe auch

Deklaration: [URL](#) als [String](#)

Beschreibung

Die Eigenschaft sollte eine gültige URL für den Lade- oder Speichervorgang enthalten. Derzeit werden die HTTP-Protokolle http, https, ftp und gopher unterstützt.

7 AuthenticRange

Siehe auch

In der ersten Tabelle werden die Eigenschaften und Methoden von AuthenticRange aufgelistet, mit Hilfe derer Sie durch das Dokument navigieren und bestimmte Abschnitte auswählen können.

Eigenschaften

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

Methoden

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

In der folgenden Tabelle, werden die Methoden zum Ändern des Inhalts aufgelistet. Viele davon können durch Rechtsklick mit der Maus aufgerufen werden.

Eigenschaften

[Text](#)

Bearbeitungsoperationen

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

Operationen für dynamische Tabellen

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

Die folgenden Methoden stellen die Funktionalitäten der Authentic-Eingabehilfenfenster für Range (Bereichs)-Objekte bereit.

Operationen der Eingabehilfenfenster

Elemente	Attribute	Entities
CanPerformActionWith	GetElementAttributeValue	GetEntityNames
CanPerformAction	GetElementAttributeNames	InsertEntity
PerformAction	GetElementHierarchy	
	HasElementAttribute	
	SetElementAttributeValue	

Beschreibung

Bei AuthenticRange-Objekten der Automation-Schnittstelle handelt es sich um die Selektion. Sie können damit den Cursor auf jede Cursorposition in der Authentic-Ansicht setzen oder einen Abschnitt des Dokuments auswählen. Die für AuthenticRange-Objekte verfügbaren Operationen haben dann auf diese Auswahl dieselbe Wirkung wie die entsprechenden Operationen der Benutzeroberfläche auf die aktuelle Auswahl in der Authentic-Ansicht. Der Hauptunterschied ist, dass Sie eine beliebige Anzahl von AuthenticRange-Objekten gleichzeitig verwenden können, während es bei der Benutzeroberfläche nur genau eine Selektion gibt.

Um zunächst einmal ein Bereichsobjekt zu erhalten, verwenden Sie [AuthenticView.Selection](#), um einen Bereich zu erhalten, der der aktuellen Selektion auf der Benutzeroberfläche entspricht. Alternativ dazu können einige einfache Bereiche auch über die schreibgeschützten Eigenschaften [AuthenticView.DocumentBegin](#), [AuthenticView.DocumentEnd](#) und [AuthenticView.WholeDocument](#) aufgerufen werden. Die flexibelste Methode ist [AuthenticView.Goto](#), mit Hilfe derer Sie in nur einem Aufruf zu einem bestimmten Bereich im Dokument navigieren können. Für komplexere Auswahlvorgänge können Sie die obigen Methoden kombinieren. Die verschiedenen Navigationsmethoden zu Bereichsobjekten sind in der ersten Tabelle auf dieser Seite aufgelistet.

Ein andere Methode, einen Dokumentenabschnitt auszuwählen, ist die Verwendung der Positionseigenschaften des Bereichsobjekts. Es stehen zwei Systeme zur Positionierung zur Verfügung, die beliebig kombiniert werden können:

- **Absolute** Textcursor-Positionen, beginnend mit Position 0 am Dokumentanfang, können für den Anfang und das Ende eines Bereichs gesetzt und abgerufen werden. Nähere Informationen dazu finden Sie unter [FirstTextPosition](#) und [LastTextPosition](#). Zur Verwendung dieser Methode sind komplexe interne Berechnungen erforderlich, daher sollte sie mit Vorsicht verwendet werden.
- Das **XMLData**-Element und eine Textposition innerhalb dieses Elements können für den Anfang und das Ende eines Bereichs gesetzt und abgerufen werden. Nähere Informationen dazu finden Sie unter [FirstXMLData](#), [FirstXMLDataOffset](#), [LastXMLData](#) und [LastXMLDataOffset](#). Diese Methode ist äußerst effizient, erfordert aber Kenntnisse über die zugrunde liegende Dokumentenstruktur. Sie können damit XMLData-Objekte suchen und Operationen daran durchführen, die sonst über die Benutzeroberfläche nicht zur Verfügung stehen.

Änderungen am Dokumenteninhalte können durch verschiedene Methoden vorgenommen werden:

- mit Hilfe der [Text](#)-Eigenschaft können Sie den durch das Bereichsobjekt ausgewählten Dokumententext abrufen. Ist sie gesetzt, wird der ausgewählte Dokumententext durch den neuen Text ersetzt.
- die Standard-Dokumentbearbeitungsfunktionen [Cut](#), [Copy](#), [Paste](#) und [Delete](#).
- Tabellenoperationen für Tabellen, die dynamisch wachsen können.

- Methoden, die die Funktionalitäten der Authentic-Eingabehilfenfenster abbilden.
- Zugriff auf die **XMLData**-Objekte des zugrunde liegenden Dokuments, damit diese direkt geändert werden können.

7.1 AuthenticRange.AppendRow

Siehe auch

Methode: `AppendRow ()` als `Boolean`

Beschreibung

Wenn sich der Anfang des Bereichs innerhalb einer dynamischen Tabelle befindet, wird mit der Methode eine neue Zeile am Ende der ausgewählten Tabelle eingefügt. Die Auswahl des Bereichs wird geändert, sodass sie auf den Beginn der neuen Zeile zeigt. Die Funktion gibt den Wert `true` zurück, wenn das Anhängen erfolgreich war. Andernfalls wird `false` zurückgegeben.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

Beispiele

```
'-----  
'                                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
```

7.2 AuthenticRange.Application

Siehe auch

Eigenschaft: [Application](#) als [Authentic](#) (read-only)

Beschreibung

Liefert das Applikationsobjekt.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.3 AuthenticRange.CanPerformAction

Siehe auch

Methode: [CanPerformAction](#) (*eAction* als [SPYAuthenticActions](#), *strElementName* als [String](#)) als [Boolean](#)

Beschreibung

CanPerformAction und die dazugehörigen Methoden gestatten den Zugriff auf die Eingabehilfenfunktionen von Authentic. Mit Hilfe dieser Funktion können Sie den Inhalt eines Dokuments einfach und auf konsistente Weise bearbeiten, ohne wissen zu müssen, wo genau die Änderung vorgenommen wird. Der Anfang des Bereichsobjekts dient dazu, die nächste gültige Position zu finden, an der die angegebene Aktion durchgeführt werden kann. Wird diese Position gefunden, gibt die Methode *True* zurück, andernfalls *False*.

HINWEIS: Um alle gültigen Elementnamen für eine bestimmte Aktion zu finden, verwenden Sie [CanPerformActionWith](#).

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.
- 2007 Es wurde eine ungültige Aktion definiert.

Beispiele

Siehe [PerformAction](#).

7.4 AuthenticRange.CanPerformActionWith

Siehe auch

Methode: [PerformActionWith](#) (*eAction* als [SPYAuthenticActions](#), *out_arrElementNames* als [Variant](#))

Beschreibung

PerformActionWith und die dazugehörigen Methoden ermöglichen den Zugriff auf die Eingabehilfenfunktionen von Authentic. Mit Hilfe dieser Funktion können Sie den Inhalt eines Dokuments einfach und auf konsistente Weise bearbeiten, ohne wissen zu müssen, wo genau die Änderung vorgenommen wird.

Diese Methode gibt einen Array der Elementnamen zurück, an denen die angegebene Aktion ausgeführt werden kann.

HINWEIS: Verwenden Sie [CanPerformActionWith](#), um die Aktion anzuwenden.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.
- 2007 Es wurde eine ungültige Aktion definiert.

Beispiele

Siehe [PerformAction](#).

7.5 AuthenticRange.Clone

Siehe auch

Methode: [Clone](#) () als [AuthenticRange](#)

Beschreibung

Gibt eine Kopie des Bereichsobjekts zurück.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.6 AuthenticRange.CollapsToBegin

Siehe auch

Methode: [CollapsToBegin](#) () als [AuthenticRange](#)

Beschreibung

Setzt das Ende des Bereichsobjekts and seinen Beginn. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.7 AuthenticRange.CollapsToEnd

Siehe auch

Methode: [CollapsToEnd](#) () als [AuthenticRange](#)

Beschreibung

Setzt den Beginn des Bereichsobjekts an seinen Anfang. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.8 AuthenticRange.Copy

Siehe auch

Methode: [Copy](#) () als [Boolean](#)

Beschreibung

Gibt *False* zurück, wenn der Bereich keine Dokumentabschnitte enthält, die kopiert werden können.

Gibt *True* zurück, wenn Text sowie - im Falle von vollständig selektierten XML-Elementen - die Elemente selbst in den Kopier-/Einfügebepuffer kopiert wurden.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.9 AuthenticRange.Cut

Siehe auch

Methode: `Cut ()` als `Boolean`

Beschreibung

Gibt *False* zurück, wenn der Bereich Dokumentabschnitte enthält, die nicht gelöscht werden dürfen.

Gibt den Wert *True* zurück, wenn Text sowie - im Falle von vollständig selektierten XML-Elementen - die Elemente selbst aus dem Dokument gelöscht wurden und im Kopier-/Einfügebepuffer gespeichert wurden.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.10 AuthenticRange.Delete

Siehe auch

Methode: `Delete ()` als `Boolean`

Beschreibung

Gibt *False* zurück, wenn der Bereich Dokumentabschnitte enthält, die nicht gelöscht werden dürfen.

Gibt den Wert *True* zurück, wenn Text sowie - im Falle von vollständig selektierten XML-Elementen - die Elemente selbst aus dem Dokument gelöscht wurden.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.11 AuthenticRange.DeleteRow

Siehe auch

Methode: `DeleteRow ()` als `Boolean`

Beschreibung

Wenn sich der Anfang des Bereichs innerhalb einer dynamischen Tabelle befindet, löscht diese Methode die ausgewählte Zeile. Die Auswahl des Bereichs wird geändert und zeigt auf das nächste Element nach der gelöschten Zeile. Die Funktion gibt *true* zurück, wenn der Löschvorgang erfolgreich war, andernfalls wird *false* zurückgegeben.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

Beispiele

```
'-----  
'                               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
```

7.12 AuthenticRange.DuplicateRow

Siehe auch

Methode: `DuplicateRow` () als `Boolean`

Beschreibung

Wenn sich der Anfang des Bereichs innerhalb einer dynamischen Tabelle befindet, fügt diese Methode eine Kopie der aktuellen Zeile hinter der ausgewählten Zeile ein. Die Funktion gibt `true` zurück, wenn der Kopiervorgang erfolgreich war, andernfalls wird `false` zurückgegeben.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

Beispiele

```
'-----  
'                               VBScript  
' duplicate row in current dynamically growable table  
'-----  
Dim objRange  
Set objRange = objPlugin.ActiveDocument.AuthenticView.Selection  
  
' check if we can insert soemthing  
If objRange.IsInDynamicTable Then  
    objRange.DuplicateRow  
    ' objRange points to begining of new row  
    objRange.Select  
End If
```

7.13 AuthenticRange.ExpandTo

Siehe auch

Methode: [ExpandTo](#) (*eKind* als [SPYAuthenticElementKind](#)) als [AuthenticRange](#)

Beschreibung

Wählt das gesamte Element vom Typ *eKind* aus, das an der ersten Cursorposition des Bereichs beginnt bzw. diese enthält. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2003 Die Erweiterung des Bereichs würde über das Ende des Dokuments hinausgehen.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.14 AuthenticRange.FirstTextPosition

Siehe auch

Eigenschaft: [FirstTextPosition](#) als [Long](#)

Beschreibung

Setzt oder holt den am weitesten links befindlichen Textpositionsindex des Bereichsobjekts. Dieser Index ist immer kleiner oder gleich der [LastTextPosition](#). Die Indizierung beginnt am Anfang des Dokuments mit 0 und erhöht sich schrittweise mit jeder neuen Position, an der sich der Textcursor befinden kann. Eine Erhöhung der Textposition um 1 hat dieselbe Wirkung wie die Rechtspfeil-Taste. Eine Verringerung der Textposition um den Wert 1 hat dieselbe Wirkung wie die Linkspfeil-Taste.

Wenn Sie *FirstTextPosition* auf einen Wert setzen, der größer ist als die aktuelle [LastTextPosition](#), wird [LastTextPosition](#) auf den Wert der neuen *FirstTextPosition* gesetzt.

HINWEIS: Verwenden Sie Textcursorpositionen mit Vorsicht, da dies im Vergleich zur Cursorpositionierung auf Basis von XMLData ein aufwändiger Vorgang ist.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.
- 2006 Es wurde eine Textposition außerhalb des Dokuments angegeben

Beispiele

```

'-----
'                               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 "Oops!"
End If

```

7.15 AuthenticRange.FirstXMLData

Siehe auch

Eigenschaft: [FirstXMLData](#) als [XMLData](#)

Beschreibung

Setzt oder holt das erste XMLData-Element im zugrunde liegenden Dokument, das zum Teil oder zur Gänze vom Bereich ausgewählt ist. Der genaue Beginn der Auswahl wird durch das [FirstXMLDataOffset](#) Attribut definiert.

Wenn FirstXMLData auf ein neues Datenobjekt gesetzt wird, wird [FirstXMLDataOffset](#) auf die erste Cursorposition innerhalb dieses Elements gesetzt. Es dürfen nur XMLData-Objekte verwendet werden, für die eine Cursorposition möglich ist. Wenn Sie *FirstXMLData* setzen / wählt [FirstXMLDataOffset](#) eine Position aus, die größer ist als der aktuelle Wert von [LastXMLData](#) / [LastXMLDataOffset](#). Das zweite wird auf die neue Anfangsposition verschoben.

HINWEIS: Sie können mit Hilfe der Eigenschaften [FirstXMLData](#) und [LastXMLData](#) direkt auf das zugrunde liegende XML-Dokument zugreifen und dieses manipulieren, wenn die mit dem [AuthenticRange](#)-Objekt verfügbaren Methoden nicht ausreichen.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.
- 2008 Interner Fehler
- 2009 Das XMLData-Objekt kann nicht aufgerufen werden.

Beispiele

```
'-----  
'                               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
```

7.16 AuthenticRange.FirstXMLDataOffset

Siehe auch

Eigenschaft: [FirstXMLDataOffset](#) als [Long](#)

Beschreibung

Setzt oder holt den Cursorpositions-Offset im [FirstXMLData](#)-Element für den Anfang des Bereichs. Die Offset-Positionen basieren auf den von der [Text](#)-Eigenschaft zurückgegebenen Zeichen und beginnen mit dem Wert 0. Verwenden Sie beim Setzen eines neuen Offset den Wert -1, um den Offset an die letzte mögliche Position im Element zu setzen. Seien Sie besonders in den folgenden Fällen vorsichtig:

- Die Textform von Einträgen in Auswahllisten, Kontrollkästchen und ähnlichen Steuerelementen unterscheidet sich unter Umständen von dem, was auf dem Bildschirm angezeigt wird. Der Daten-Offset basiert zwar auf diesem Text, doch gibt es nur zwei gültige Offset-Positionen, eine am Anfang und eine am Ende des Eintrags. Wenn Sie versuchen, den Offset in die Mitte des Eintrags zu positionieren, wird der Offset an das Ende des Eintrags gesetzt.
- Die Textform von XML Entities kann sich in ihrer Darstellung von der auf dem Bildschirm dargestellten Länge unterscheiden. Der Offset basiert auf der Textform der Entities.

Wenn [FirstXMLData](#) / [FirstXMLDataOffset](#) eine Position hinter dem aktuellen [LastXMLData](#) / [LastXMLDataOffset](#) auswählt, wird letzteres an die neue Anfangsposition verschoben.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Es wurde ein ungültiger Offset-Wert angegeben.
Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

Beispiele

```

'-----
'                               VBScript
' Select the complete text of an XMLData element
' using XMLData based selection and ExpandTo
'-----
Dim objAuthenticView
Set objAuthenticView = objPlugin.AuthenticView

' first we use the XMLData based range properties
' to select all text of the first XMLData element
' in the current selection
Dim objRange
Set objRange = objAuthenticView.Selection
objRange.FirstXMLDataOffset = 0 ' start at beginning of element text
objRange.LastXMLData = objRange.FirstXMLData ' select only one element
objRange.LastXMLDataOffset = -1 ' select till its end

' the same can be achieved with the ExpandTo method
Dim objRange2
Set objRange2 = objAuthenticView.Selection.ExpandTo(spyAuthenticTag)

' were we successful?
If objRange.IsEqual(objRange2) Then
    objRange.Select()
Else

```

```
        MsgBox "Oops"  
End If
```

7.17 AuthenticRange.GetElementAttributeNames

Siehe auch

Methode: [GetElementAttributeNames](#) (*strElementName* als [String](#), *out_arrAttributeNames* als [Variant](#))

Beschreibung

Liefert die Namen aller Attribute für das einschließende Element mit dem angegebene Namen. Verwenden Sie die Element- / Attribut-Paare, um den Attributwert mit Hilfe der Methoden [GetElementAttributeValue](#) und [SetElementAttributeValue](#) zu setzen oder zu holen.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Es wurde ein ungültiger Elementname angegeben.
Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

Beispiele

Siehe [SetElementAttributeValue](#).

7.18 AuthenticRange.GetElementAttributeValue

Siehe auch

Methode: `GetElementAttributeValue` (*strElementName* als `String`, *strAttributeName* als `String`) als `String`

Beschreibung

Liefert für das mit *strElementName* definierte Element den Wert des in *strAttributeName* definierten Attributs. Wenn das Attribut unterstützt wird, ihm jedoch kein Wert zugewiesen wurde, wird ein leerer String zurückgegeben. Um herauszufinden, welche Attribute von einem Element unterstützt werden, verwenden Sie [GetElementAttributeNames](#) oder [HasElementAttribute](#).

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Es wurde ein ungültiger Elementname angegeben.
Es wurde ein ungültiger Attributname angegeben.
Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

Beispiele

Siehe [SetElementAttributeValue](#).

7.19 AuthenticRange.GetElementHierarchy

Siehe auch

Methode: [GetElementHierarchy](#) (*out_arrElementNames* als *Variant*)

Beschreibung

Liefert die Namen aller XML-Elemente, die Parent-Elemente der aktuellen Auswahl sind. Innere Elemente werden vor den diese einschließenden Elementen aufgelistet. Wenn sich die aktuelle Auswahl nicht innerhalb eines einzelnen XMLData-Elements befindet, wird eine leere Liste zurückgegeben.

Zusammen mit dem Bereichsobjekt identifizieren die Namen der Elementhierarchie XMLData-Elemente eindeutig. Die Attribute dieser Elemente können direkt durch [GetElementAttributeNames](#) und die dazugehörigen Methoden aufgerufen werden.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

Beispiele

Siehe [SetElementAttributeValue](#).

7.20 AuthenticRange.GetEntityNames

Siehe auch

Methode: [GetEntityNames](#) (*out_arrEntityNames* als *Variant*)

Beschreibung

Liefert die Namen aller definierten Entities. Die Liste der abgerufenen Entities ist unabhängig von der aktuellen Auswahl oder der aktuellen Position. Verwenden Sie einen dieser Namen mit der [InsertEntity](#)-Funktion.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

Beispiele

Siehe [InsertEntity](#).

7.21 AuthenticRange.Goto

Siehe auch

Methode: `Goto` (*eKind* als `SPYAuthenticElementKind`, *nCount* als `Long`, *eFrom* als `SPYAuthenticDocumentPosition`) als [AuthenticRange](#)

Beschreibung

Setzt den Bereich an den Anfang des *nCount*-Elements vom Typ *eKind*. Die Anfangsposition wird durch den Parameter *eFrom* definiert.

Verwenden Sie zum Navigieren zum Ende des Dokuments positive Positionswerte für *nCount*. Verwenden Sie negative Werte, um zum Anfang des Dokuments zu navigieren. Die Methode gibt das geänderte Range-Objekt zurück.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2003 Das Ziel befindet sich hinter dem Ende des Dokuments.
- 2004 Das Ziel befindet sich vor dem Anfang des Dokuments.
- 2005 Es wurde ein ungültiger Elementtyp definiert.
Es wurde eine ungültige Anfangsposition definiert.
Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.22 AuthenticRange.GotoNext

Siehe auch

Methode: [GotoNext](#) (*eKind* als [SPYAuthenticElementKind](#)) als [AuthenticRange](#)

Beschreibung

Setzt den Bereich an den Anfang des nächsten Elements vom Typ *eKind*. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2003 Das Ziel befindet sich hinter dem Ende des Dokuments.
- 2005 Es wurde ein ungültiger Elementtyp definiert.
Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

Beispiele

```
-----  
'          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
```

7.23 AuthenticRange.GotoNextCursorPosition

Siehe auch

Methode: [GotoNextCursorPosition](#) () als [AuthenticRange](#)

Beschreibung

Setzt den Bereich auf die nächste Cursorposition hinter dem Ende der aktuellen Endposition.
Gibt das geänderte Objekt zurück.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2003 Das Ziel befindet sich hinter dem Ende des Dokuments.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.24 AuthenticRange.GotoPrevious

Siehe auch

Methode: [GotoPrevious](#) (*eKind* als [SPYAuthenticElementKind](#)) als [AuthenticRange](#)

Beschreibung

Setzt den Bereich an den Anfang des Elements vom Typ *eKind*, welches sich vor dem Anfang des aktuellen Bereichs befindet. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2004 Das Ziel befindet sich vor dem Anfang des Dokuments.
- 2005 Es wurde ein ungültiger Elementtyp definiert.
Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

Beispiele

```
'-----  
'                               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
```

7.25 AuthenticRange.GotoPreviousCursorPosition

Siehe auch

Methode: [GotoPreviousCursorPosition](#) () als [AuthenticRange](#)

Beschreibung

Setzt den Bereich der Cursorposition unmittelbar vor die aktuelle Position. Gibt das geänderte Objekt zurück.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2004 Das Ziel befindet sich vor dem Anfang des Dokuments.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.26 AuthenticRange.HasElementAttribute

Siehe auch

Methode: `HasElementAttribute` (*strElementName* als `String`, *strAttributeName* als `String`) als `Boolean`

Beschreibung

Testet, ob das einschließende Element mit dem Namen *strElementName* das in *strAttributeName* definierte Attribut unterstützt.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Es wurde ein ungültiger Elementname definiert.
Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.27 AuthenticRange.InsertEntity

Siehe auch

Methode: `InsertEntity` (*strEntityName* als `String`)

Beschreibung

Ersetzt die Bereichsauswahl durch die angegebene Entity. Die Entity muss einer der Entity-Namen sein, die von [GetEntityNames](#) zurückgegeben werden.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Es wurde ein ungültiger Name für den Eintrag angegeben.

Beispiele

```
'-----  
'                               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
```

7.28 AuthenticRange.InsertRow

Siehe auch

Methode: `InsertRow ()` als `Boolean`

Beschreibung

Wenn sich der Anfang des Bereichs innerhalb einer dynamischen Tabelle befindet, wird mit der Methode eine neue Zeile vor der aktuellen eingefügt. Die Auswahl des Bereichs wird geändert, sodass sie auf den Beginn der neu eingefügten Zeile zeigt. Die Funktion gibt den Wert *true* zurück, wenn das Einfügen erfolgreich war. Andernfalls wird *false* zurückgegeben.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

Beispiele

```
'-----  
'                               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
```

7.29 AuthenticRange.IsEmpty

Siehe auch

Methode: `IsEmpty ()` als `Boolean`

Beschreibung

Testet, ob die erste und die letzte Position des Bereichs identisch sind.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.30 AuthenticRange.IsEqual

Siehe auch

Methode: [IsEqual](#) (*objCmpRange* als [AuthenticRange](#)) als [Boolean](#)

Beschreibung

Testet, ob der Beginn und das Ende der beiden Bereiche identisch sind.

Fehler

- 2001 Eines der beiden verglichenen Bereichsobjekte ist ungültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.31 AuthenticRange.IsInDynamicTable

Siehe auch

Methode: `IsInDynamicTable ()` als `Boolean`

Beschreibung

Testet, ob sich der Anfang eines Bereichs innerhalb einer Tabelle befindet, die andere Zeilenoperationen unterstützt.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.32 AuthenticRange.LastTextPosition

Siehe auch

Eigenschaft: [LastTextPosition](#) als [Long](#)

Beschreibung

Setzt oder holt den am weitesten rechts befindlichen Textpositionsindex des Bereichsobjekts. Dieser Index ist immer größer oder gleich der [FirstTextPosition](#). Die Indizierung beginnt am Anfang des Dokuments mit 0 und erhöht sich schrittweise mit jeder neuen Position, an der sich der Textcursor befinden kann. Eine Erhöhung der Textposition um 1 hat dieselbe Wirkung wie die Rechtspfeil-Taste. Eine Verringerung der Textposition um den Wert 1 hat dieselbe Wirkung wie die Linkspfeil-Taste.

Wenn Sie *LastTextPosition* auf einen Wert setzen, der kleiner ist als die aktuelle [FirstTextPosition](#), wird [FirstTextPosition](#) auf den Wert der neuen *LastTextPosition* gesetzt.

HINWEIS: Verwenden Sie Textcursorpositionen mit Vorsicht, da dies im Vergleich zur Cursorpositionierung auf Basis von XMLData ein aufwändiger Vorgang ist.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.
- 2006 Es wurde eine Textposition außerhalb des Dokuments angegeben

Beispiele

```

-----
'                               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

```

7.33 AuthenticRange.LastXMLData

Siehe auch

Eigenschaft: [LastXMLData](#) als [XMLData](#)

Beschreibung

Setzt oder holt das letzte XMLData-Element im zugrunde liegenden Dokument, das zum Teil oder zur Gänze vom Bereich ausgewählt ist. Das genaue Ende der Auswahl wird durch das [LastXMLDataOffset](#) Attribut definiert.

Wenn *LastXMLData* auf ein neues Datenobjekt gesetzt wird, wird [LastXMLDataOffset](#) auf die letzte Cursorposition innerhalb dieses Elements gesetzt. Es dürfen nur XMLData-Objekte verwendet werden, für die es eine Cursorposition gibt. Wenn Sie *LastXMLData* / [LastXMLDataOffset](#) setzen, wählen Sie eine Position aus, die kleiner ist als der aktuelle Wert von [FirstXMLData](#) / [FirstXMLDataOffset](#). Letzteres wird auf die neue Endposition verschoben.

HINWEIS: Sie können mit Hilfe der Eigenschaften [FirstXMLData](#) und [LastXMLData](#) direkt auf das zugrunde liegende XML-Dokument zugreifen und dieses manipulieren, wenn die mit dem [AuthenticRange](#)-Objekt verfügbaren Methoden nicht ausreichen.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.
- 2008 Interner Fehler
- 2009 Das XMLData-Objekt kann nicht aufgerufen werden.

7.34 AuthenticRange.LastXMLDataOffset

Siehe auch

Eigenschaft: [LastXMLDataOffset](#) als [Long](#)

Beschreibung

Setzt oder holt die Cursorposition im [LastXMLData](#)-Element für das Ende des Bereichs.

Die Offset-Positionen basieren auf den von der [Text](#)-Eigenschaft zurückgegebenen Zeichen und beginnen mit dem Wert 0. Verwenden Sie beim Setzen eines neuen Offset den Wert -1, um den Offset an die letzte mögliche Position im Element zu setzen. Seien Sie besonders in den folgenden Fällen vorsichtig:

- Die Textform von Einträgen in Auswahllisten, Kontrollkästchen und ähnlichen Steuerelementen unterscheidet sich unter Umständen von dem, was auf dem Bildschirm angezeigt wird. Der Daten-Offset basiert zwar auf diesem Text, doch gibt es nur zwei gültige Offset-Positionen, eine am Anfang und eine am Ende des Eintrags. Wenn Sie versuchen, den Offset in die Mitte des Eintrags zu positionieren, wird der Offset an das Ende des Eintrags gesetzt.
- Die Textform von XML Entities kann sich in ihrer Darstellung von der auf dem Bildschirm dargestellten Länge unterscheiden. Der Offset basiert auf der Textform der Entities z.B. `&`.

Wenn [LastXMLData](#) / [LastXMLDataOffset](#) eine Position vor dem aktuellen [FirstXMLData](#) / [FirstXMLDataOffset](#) auswählt, wird letzteres an die neue Endposition verschoben.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Es wurde ein ungültiger Offset-Wert angegeben.
Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

Beispiele

```

' -----
'                               VBScript
' Select the complete text of an XMLData element
' using XMLData based selection and ExpandTo
' -----
Dim objAuthenticView
Set objAuthenticView = objPlugin.AuthenticView

' first we use the XMLData based range properties
' to select all text of the first XMLData element
' in the current selection
Dim objRange
Set objRange = objAuthenticView.Selection
objRange.FirstXMLDataOffset = 0 ' start at beginning of element text
objRange.LastXMLData = objRange.FirstXMLData ' select only one element
objRange.LastXMLDataOffset = -1 ' select till its end

' the same can be achieved with the ExpandTo method
Dim objRange2
Set objRange2 = objAuthenticView.Selection.ExpandTo(spyAuthenticTag)

' were we successful?
If objRange.IsEqual(objRange2) Then
    objRange.Select()

```

```
Else  
    MsgBox "Oops"  
End If
```

7.35 AuthenticRange.MoveBegin

Siehe auch

Methode: [MoveBegin](#) (*eKind* als [SPYAuthenticElementKind](#), *nCount* als [Long](#)) als [AuthenticRange](#)

Beschreibung

Verschiebt den Anfang des Bereichs an den Anfang des *nCount* Elements vom Typ *eKind*. Die Zählung beginnt am aktuellen Beginn des Bereichsobjekts.

Mit Hilfe von positiven Zahlen für *nCount* können Sie die Position zum Ende des Dokuments hin verschieben, mit Hilfe von negativen Zahlen können Sie die Position zum Beginn des Dokuments hin verschieben. Das Ende des Bereichs wird nicht verschoben, es sei denn, der neue Anfang des Bereichs wäre größer als der Wert. In diesem Fall wird das Ende auf den neuen Anfang verschoben. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2003 Das Ziel befindet sich hinter dem Ende des Dokuments.
- 2004 Das Ziel befindet sich vor dem Anfang des Dokuments.
- 2005 Es wurde ein ungültiger Elementtyp definiert.
Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.36 AuthenticRange.MoveEnd

Siehe auch

Methode: [MoveEnd](#) (*eKind* als [SPYAuthenticElementKind](#), *nCount* als [Long](#)) als [AuthenticRange](#)

Beschreibung

Verschiebt das Ende des Bereichs an den Anfang des *nCount* Elements vom Typ *eKind*. Die Zählung beginnt am aktuellen Beginn des Bereichsobjekts.

Mit Hilfe von positiven Zahlen für *nCount* können Sie die Position zum Ende des Dokuments hin verschieben, mit Hilfe von negativen Zahlen können Sie die Position zum Beginn des Dokuments hin verschieben. Das Ende des Bereichs wird nicht verschoben, es sei denn, der neue Anfang des Bereichs wäre größer als der Wert. In diesem Fall wird das Ende an den neuen Anfang verschoben. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2003 Das Ziel befindet sich hinter dem Ende des Dokuments.
- 2004 Das Ziel befindet sich vor dem Anfang des Dokuments.
- 2005 Es wurde ein ungültiger Elementtyp definiert.
Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.37 AuthenticRange.MoveRowDown

Siehe auch

Methode: `MoveRowDown` () als `Boolean`

Beschreibung

Wenn sich der Anfang eines Bereichs innerhalb einer dynamischen Tabelle befindet und eine Zeile ausgewählt wird, die nicht die letzte Zeile in dieser Tabelle ist, vertauscht diese Methode diese Zeile mit der Zeile unmittelbar danach. Die Auswahl des Bereichs verschiebt sich mit der Zeile, ändert sich aber sonst nicht. Die Funktion gibt den Wert `true` zurück, wenn das Verschieben erfolgreich war, ansonsten wird der Wert `false` zurückgegeben.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.38 AuthenticRange.MoveRowUp

Siehe auch

Methode: [MoveRowUp](#) () als [Boolean](#)

Beschreibung

Wenn sich der Anfang eines Bereichs innerhalb einer dynamischen Tabelle befindet und eine Zeile ausgewählt wird, die nicht die erste Zeile in dieser Tabelle ist, vertauscht diese Methode diese Zeile mit der Zeile unmittelbar oberhalb davon. Die Auswahl des Bereichs verschiebt sich mit der Zeile, ändert sich aber sonst nicht. Die Funktion gibt den Wert *true* zurück, wenn das Verschieben erfolgreich war, ansonsten wird der Wert *false* zurückgegeben.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

Beispiele

Siehe [JScript - Bubble Sort Dynamic Tables](#).

7.39 AuthenticRange.Parent

Siehe auch

Eigenschaft: [Parent](#) als [AuthenticView](#) (read-only)

Beschreibung

Ruft die Ansicht auf, die dieses Bereichsobjekt enthält.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.40 AuthenticRange.Paste

Siehe auch

Methode: `Paste ()` als `Boolean`

Beschreibung

Gibt *False* zurück, wenn der Kopier-/Einfügebuffer leer ist oder die aktuelle Auswahl nicht durch den Inhalt dieses Puffers ersetzt werden kann.

Andernfalls wird die aktuelle Auswahl gelöscht, der Inhalt dieses Puffers eingefügt und der Wert *true* zurückgegeben.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

7.41 AuthenticRange.PerformAction

Siehe auch

Methode: `PerformAction` (*eAction* als `SPYAuthenticActions`, *strElementName* als `String`) als `Boolean`

Beschreibung

`PerformAction` und die dazugehörigen Methoden verleihen Zugriff auf die Eingabehilfenfunktionen von Authentic. Mit Hilfe dieser Funktion können Sie Dokumentinhalt schnell und auf konsistente Weise ändern, ohne wissen zu müssen, wo genau die Änderung vorgenommen wird. Der Anfang des Bereichsobjekts dient dazu, die nächste gültige Position zu finden, an der die angegebene Aktion durchgeführt werden kann. Wenn es keine solche Position gibt, gibt die Methode den Wert `false` zurück. Andernfalls wird das Dokument geändert und der Bereich zeigt auf den Anfang der Änderung.

HINWEIS: Um Elementnamen zu finden, die als der zweite Parameter übergeben werden können, verwenden Sie [CanPerformActionWith](#).

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.
- 2007 Es wurde eine ungültige Aktion definiert.

Beispiele

```

'-----
'           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

```

7.42 AuthenticRange.Select

Siehe auch

Methode: [Select \(\)](#)

Beschreibung

Macht diesen Bereich zur aktuellen Auswahl der Ansicht. Dasselbe Ergebnis erzielen Sie bei Verwendung von: '*objRange.Parent.Selection = objRange*'

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.

Beispiele

```
'-----  
'                               VBScript  
'-----  
  
' set current selection to end of document  
objPlugin.objAuthenticView.DocumentEnd.Select()
```

7.43 AuthenticRange.SelectNext

Siehe auch

Methode: [SelectNext](#) (*eKind* als [SPYAuthenticElementKind](#)) als [AuthenticRange](#)

Beschreibung

Wählt das Element vom Typ *eKind* nach dem aktuellen Bereichsende aus. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2003 Das Ziel befindet sich hinter dem Ende des Dokuments.
- 2005 Es wurde ein ungültiger Elementtyp definiert.
Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

Beispiele

```
-----  
'                               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
```

7.44 AuthenticRange.SelectPrevious

Siehe auch

Methode: [GotoPrevious](#) (*eKind* als [SPYAuthenticElementKind](#)) als [AuthenticRange](#)

Beschreibung

Wählt das Element vom Typ *eKind* vor dem Beginn es aktuelle Bereichs aus. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2004 Das Ziel befindet vor dem Beginn des Dokuments.
- 2005 Es wurde ein ungültiger Elementtyp definiert.
Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

Beispiele

```
-----  
'           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
```

7.45 AuthenticRange.SetElementAttributeValue

Siehe auch

Methode: `SetElementAttributeValue` (*strElementName* als `String`, *strAttributeName* als `String`, *strAttributeValue* als `String`)

Beschreibung

Liefert den Wert des in *strAttributeName* Attributs für das mit *strElementName* bezeichnete Element. Wenn das Attribut unterstützt wird, aber keinen zugewiesenen Wert hat, wird ein leerer String zurückgegeben. Um die Namen der von einem Element unterstützten Attribute zu ermitteln, verwenden Sie [GetElementAttributeNames](#) oder [HasElementAttribute](#).

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Es wurde ein ungültiger Elementname angegeben.
Es wurde ein ungültiger Attributname angegeben.
Es wurde ein ungültiger Attributwert angegeben.

Beispiele

```

'-----
'           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

```

7.46 AuthenticRange.SetFromRange

Siehe auch

Methode: [SetFromRange](#) (*objSrcRange* als [AuthenticRange](#))

Beschreibung

Setzt das Bereichsobjekt auf dieselbe Anfangs- und Endposition wie *objSrcRange*.

Fehler

- 2001 Eines der beiden Bereichsobjekte ist ungültig.
- 2005 Es wurde kein Objekt als Quellobjekt definiert.

7.47 AuthenticRange.Text

Siehe auch

Eigenschaft: [Text](#) als [String](#)

Beschreibung

Setzt oder holt den vom Bereichsobjekt ausgewählten Textinhalt.

Die Anzahl der abgerufenen Zeichen muss nicht unbedingt übereinstimmen, da es zwischen dem Anfang und dem Ende des ausgewählten Bereichs Textcursor-Positionen gibt. Die meisten Dokumentelemente unterstützen eine Bereichsendeposition des Cursors, die nicht mit der Bereichsanfangsposition des folgenden Elements übereinstimmt. Dropdown-Listen verfügen nur über eine Cursorposition, können aber Strings beliebiger Länge auswählen. Bei Optionsfeldern und Kontrollkästchen enthält der Text eigenschaftswert den String des entsprechenden XML-Elements.

Wenn mit dem Bereich mehr als ein Element ausgewählt wird, besteht der Text aus einer Verkettung der Einzeltexte. XML Entities werden erweitert, so dass statt '&' '&' erscheint.

Wenn der Text auf den leeren String gesetzt wird, wird kein XML-Element gelöscht. Verwenden Sie stattdessen [Cut](#), [Delete](#) oder [PerformAction](#).

Fehler

- 2001 Das Authentic Range-Objekt oder das dazugehörige View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

8 AuthenticSelection

Siehe auch

Eigenschaften

[Start](#)

[StartTextPosition](#)

[End](#)

[EndTextPosition](#)

8.1 AuthenticSelection.End

Siehe auch

Deklaration: [End](#) als [XMLData](#)

Beschreibung

XML-Element, an dem die aktuelle Auswahl endet.

8.2 AuthenticSelection.EndTextPosition

Siehe auch

Deklaration: [EndTextPosition](#) als [long](#)

Beschreibung

Position in [Authentic.End](#), an der die Auswahl endet.

8.3 AuthenticSelection.Start

Siehe auch

Deklaration: [Start](#) als [XMLData](#)

Beschreibung

XML-Element, an dem die aktuelle Auswahl beginnt.

8.4 AuthenticSelection.StartTextPosition

Siehe auch

Deklaration: [StartTextPosition](#) als [long](#)

Beschreibung

Position in [Authentic.Start](#), an der die Auswahl beginnt.

9 AuthenticToolBarButton

Methoden

Eigenschaften

CommandID

9.1 AuthenticToolBarButton.CommandID

Deklaration: `CommandID` als [SPYAuthenticCommand](#)

Beschreibung

Die CommandId der Symbolleisten-Schaltfläche.

10 AuthenticToolBarButtons

Methoden

[Item](#)

[NewButton](#)

[NewCustomButton](#)

[NewSeparator](#)

[Remove](#)

Eigenschaften

[Count](#)

10.1 AuthenticToolBarButtons.Count

Deklaration: Count als long

Beschreibung

Holt die tatsächliche Anzahl der Schaltflächen der Symbolleiste.
Schreibgeschützt.

10.2 AuthenticToolBarButtons.Item

Deklaration: `Item`(n als `long`) als [AuthenticToolBarButton](#)

Beschreibung

Holt die nte Schaltfläche der aktuellen Symbolleiste. n beginnt mit 1.

Beispiel

Siehe Beispiel unter [AuthenticToolBarButtons.NewButton](#)

10.3 AuthenticToolBarButtons.NewButton

Deklaration: `NewButton`(`nPosition` als `long`, `nCommandId` als `SPYAuthenticCommand`)

Beschreibung

Fügt eine neue Schaltfläche für den Befehl "nCommandId" an der Position "nPosition" der Symbolleiste ein. nPosition beginnt mit 1.

Beispiel

Fügt eine neue Symbolleiste hinzu, ordnet sie am unteren Rand an und fügt neue Symbolleisten-Schaltflächen hinzu.

```
objPlugIn.ToolbarRows.NewRow(3)           // add a new Toolbar (Row
                                           3)
set ToolbarRow = objPlugIn.ToolbarRows.Item(3)
set Buttons = ToolbarRow.Buttons

ToolbarRow.Alignment = 2                 // align Toolbar to bottom
Buttons.NewButton    1, 2                 // add Print Button
Buttons.NewButton    1, 3                 // add PrintPreview Button
Buttons.NewSeparator 2                    // add Separator
Buttons.NewButton    1, 4                 // add Validate Button
```

Bei Aufruf von StartEditing oder ReloadToolbars werden die geänderten Symbolleisteneinstellungen verwendet.

10.4 AuthenticToolBarButtons.NewCustomButton

Deklaration: `NewCustomButton`(`nPosition` als `long`, `strName` als `String`, `strTooltip` als `String`, `strBitmapURL` als `String`)

Beschreibung

Fügt in der Symbolleiste an der Position `nPosition` eine neue benutzerdefinierte Schaltfläche mit dem Namen `strName` ein. `nPosition` beginnt bei 1. `strTooltip` wird als Text für den Tooltip verwendet.

`strBitmapURL` ist der Pfad zur Bitmap-Datei für die neue Schaltfläche. Dieser Pfad ist relativ zum Pfad, der mit der `TextStateBmpURL`-Eigenschaft definiert wurde.

Beispiel

Angenommen, Sie haben eine benutzerdefinierte Schaltfläche mit dem Namen "MyFunction" in Ihre Symbolleisten eingefügt. Im folgenden `OnCommand()` sehen Sie, was geschehen muss, wenn der Benutzer auf Ihre Schaltfläche geklickt hat und wie Sie den ein/aus-Status dafür festlegen.

Verwenden Sie bitte die `attachCallBack`-Methode, um das "ondoceditcommand"-Ereignis mit `OnCommand()` zu verknüpfen.

```
function OnCommand()
{
    // 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;
        }
    }
}
```

10.5 AuthenticToolBarButtons.NewSeparator

Deklaration: `NewSeparator(nPosition` als `long`)

Beschreibung

Fügt ein Trennzeichen an der Position `nPosition` der Symbolleiste ein. `nPosition` beginnt mit 1.

Beispiel

Siehe Beispiel unter [AuthenticToolBarButtons.NewButton](#)

10.6 AuthenticToolBarButtons.Remove

Deklaration: Remove(nPosition als long)

Beschreibung

Entfernt die Schaltfläche oder das Trennzeichen an Position nPosition der Symbolleiste. nPosition beginnt bei 1.

11 AuthenticToolBarRow

Methoden

[Alignment](#)

Eigenschaften

[Buttons](#)

11.1 AuthenticToolBarRowAlignment

Deklaration: Alignment(nAlign als [SPYAuthenticToolBarAlignment](#))

Beschreibung

Liefert oder setzt die Anordnung der Symbolleiste im Plug-In.

Beispiel

Anordnen aller Symbolleisten am unteren Rand:

```
for each ToolbarRow in objPlugin.ToolbarRows
  ToolbarRow.Alignment = 2
next
```

11.2 AuthenticToolBarRowButtons

Deklaration: Buttons als [AuthenticToolBarButtons](#)

Beschreibung

Liefert alle Schaltflächen der Symbolleiste

Beispiel

Siehe Beispiel unter [AuthenticToolBarButtons.NewButton](#)

12 AuthenticToolBarRows

Methoden

[Item](#)

[RemoveRow](#)

[NewRow](#)

Eigenschaften

[Count](#)

12.1 AuthenticToolbarRows.Count

Deklaration: Count als long

Beschreibung

Liefert die aktuelle Anzahl der definierten Symbolleisten.
Schreibgeschützt.

12.2 AuthenticToolBarRows.Item

Deklaration: `Item(nPosition als long)` als [AuthenticToolBarRow](#)

Beschreibung

Liefert die Symbolleiste in Position nPosition. nPosition beginnt mit "1".
Schreibgeschützt.

Beispiel

Siehe Beispiel unter [AuthenticToolBarButtons.NewButton](#)

12.3 AuthenticToolBarRows.RemoveRow

Deklaration: RemoveRow(nPosition als long)

Beschreibung

Entfernt die Symbolleiste an Position nPosition von der Benutzeroberfläche. nPosition beginnt mit 1.

12.4 AuthenticToolBarRows.NewRow

Deklaration: `NewRow(nPosition als long)`

Beschreibung

Fügt eine neue Symbolleistenzeile ein. nPosition beginnt bei 1.

Beispiel

Siehe Beispiel unter [AuthenticToolBarButtons.NewButton](#)

13 AuthenticView

Siehe auch

Eigenschaften	Methoden	Ereignisse
Application	Goto	
DocumentBegin	Print	
DocumentEnd	Redo	
MarkupVisibility	Undo	
Parent	UpdateXMLInstanceEntities	
Selection		
WholeDocument		

Beschreibung

AuthenticView und sein Child-Objekt [AuthenticRange](#) stellen die Schnittstelle für die Authentic-Ansicht bereit, über die Sie Dokument-Inhalte einfach und auf konsistente Art bearbeiten können. Die Funktionsüberschneidung mit bereits bestehenden Methoden und Eigenschaften im Authentic-Objekt ist beabsichtigt. Die Funktionen des Authentic-Objekts zur Bearbeitung von Inhalt werden dadurch nicht mehr benötigt. Es wird empfohlen, unbedingt die neue AuthenticView-Schnittstelle zu verwenden.

AuthenticView verschafft Ihnen einfachen Zugriff auf bestimmte Funktionen wie z.B. Drucken, mehrfaches Rückgängigmachen und die aktuelle Cursor-Auswahl oder -Position.

AuthenticView verwendet Objekte vom Typ [AuthenticRange](#), um das einfache Navigieren im Dokument und die problemlose Auswahl logischer Textelemente zu ermöglichen. Verwenden Sie für einfache Auswahlvorgänge die Eigenschaften [DocumentBegin](#), [DocumentEnd](#) oder [WholeDocument](#) und für komplexere Auswahlvorgänge die [Goto](#)-Methode. Informationen zum Navigieren relativ zu einem vorgegebenen Dokumentenbereich finden Sie unter den Methoden und Eigenschaften des [AuthenticRange](#)-Objekts.

13.1 AuthenticView.Application

Siehe auch

Eigenschaft: [Application](#) als [Authentic](#) (Schreibgeschützt)

Beschreibung

Ruft das Authentic 2005 Desktop Edition Applikationsobjekt auf.

Fehler

- 2000 Das Authentic View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

13.2 AuthenticView.DocumentBegin

Siehe auch

Eigenschaft: [DocumentBegin](#) als [AuthenticRange](#) (Schreibgeschützt)

Beschreibung

Ruft ein Bereichsobjekt ab, das auf den Beginn des Dokuments zeigt.

Fehler

- 2000 Das Authentic View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

13.3 AuthenticView.DocumentEnd

Siehe auch

Eigenschaft: [DocumentEnd](#) als [AuthenticRange](#) (Schreibgeschützt)

Beschreibung

Ruft ein Bereichsobjekt ab, das auf das Ende des Dokuments zeigt.

Fehler

- 2000 Das Authentic View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

13.4 AuthenticView.Goto

Siehe auch

Methode: `Goto` (*eKind* als `SPYAuthenticElementKind`, *nCount* als `Long`, *eFrom* als `SPYAuthenticDocumentPosition`) als [AuthenticRange](#)

Beschreibung

Ruft ein Bereichsobjekt auf, das auf den Anfang des *nCount*-Elements vom Typ *eKind* zeigt. Die Anfangsposition wird durch den Parameter *eFrom* definiert. Verwenden Sie positive Werte für *nCount*, um zum Ende des Dokuments zu navigieren. Verwenden Sie negative Werte, um zum Anfang des Dokuments zu navigieren.

Fehler

- 2000 Das Authentic View-Objekt ist nicht mehr gültig.
- 2003 Das Ziel befindet sich hinter dem Ende des Dokuments.
- 2004 Das Ziel befindet vor den Anfang des Dokuments.
- 2005 Es wurde ein ungültiger Elementtyp angegeben.
Die Anfangsposition ist keine Position aus `spyAuthenticDocumentBegin` oder `spyAuthenticDocumentEnd`.
Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

Beispiele

```
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
```

13.5 AuthenticView.MarkupVisibility

Siehe auch

Eigenschaft: [MarkupVisibility](#) als [SPYAuthenticMarkupVisibility](#)

Beschreibung

Setzt oder liefert den aktuellen Sichtbarkeitsstatus von Markup.

Fehler

- 2000 Das Authentic View-Objekt ist nicht mehr gültig.
- 2005 Es wurde ein ungültiger Enumerationswert angegeben.
Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

13.6 AuthenticView.Parent

Siehe auch

Eigenschaft: [Parent](#) als [Authentic](#) (schreibgeschützt)

Beschreibung

Liefert eine Referenz auf die Applikation.

Fehler

- 2000 Das Authentic View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

13.7 AuthenticView.Print

Siehe auch

Methode: `Print` (`bWithPreview` als `Boolean`, `bPromptUser` als `Boolean`)

Beschreibung

Druckt das in dieser Ansicht angezeigte Dokument. Wenn `bWithPreview` auf `True` gesetzt ist, wird das Dialogfeld "Druckansicht" angezeigt. Wenn `bPromptUser` auf `True` gesetzt ist, wird das Dialogfeld "Drucken" angezeigt. Wenn beide Parameter auf `False` gesetzt sind, wird das Dokument ohne weiteren Eingriff durch den Benutzer gedruckt.

Fehler

2000 Das Authentic View-Objekt ist nicht mehr gültig.

13.8 AuthenticView.Redo

Siehe auch

Methode: [Redo](#) () als [Boolean](#)

Beschreibung

Stellt die letzte mit dem Befehl "Rückgängig" rückgängig gemachte Aktion wieder her.

Fehler

- 2000 Das Authentic View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

13.9 AuthenticView.Selection

Siehe auch

Eigenschaft: [Selection](#) als [AuthenticRange](#)

Beschreibung

Setzt oder holt die aktuelle Textauswahl auf der Benutzeroberfläche.

Fehler

- 2000 Das Authentic View-Objekt ist nicht mehr gültig.
- 2002 Es ist keine Selektion aktiv.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

Beispiele

```
'-----  
'----- 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
```

13.10 AuthenticView.Undo

Siehe auch

Methode: `Undo ()` als `Boolean`

Beschreibung

Macht die letzte in dieser Ansicht vorgenommene Änderung am Dokument rückgängig.

Fehler

- 2000 Das Authentic View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

13.11 AuthenticView.UpdateXMLInstanceEntities

Siehe auch

Methode: [UpdateXMLInstanceEntities \(\)](#)

Beschreibung

Aktualisiert die interne Darstellung der deklarierten Entities und befüllt die Eingabehilfen erneut. Zusätzlich dazu wird der Validator neu geladen, damit die XML-Datei korrekt validiert werden kann. Beachten Sie, dass dadurch eventuell auch Schema-Dateien neu geladen werden.

Fehler

Die Methode gibt nie einen Fehler zurück.

Beispiel

```
// -----  
//           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();  
}
```

13.12 AuthenticView.WholeDocument

Siehe auch

Eigenschaft: [WholeDocument](#) als [AuthenticRange](#) (Schreibgeschützt)

Beschreibung

Ruft ein Bereichsobjekt ab, das das gesamte Dokument auswählt.

Fehler

- 2000 Das Authentic View-Objekt ist nicht mehr gültig.
- 2005 Für den zurückgegebenen Parameter wurde eine ungültige Adresse angegeben.

14 AuthenticXMLTableCommands

Methoden

[Insert](#)

[Delete](#)

[AppendRow](#)

[InsertRow](#)

[DeleteRow](#)

[AppendColumn](#)

[InsertColumn](#)

[DeleteColumn](#)

[JoinLeft](#)

[JoinRight](#)

[JoinUp](#)

[JoinDown](#)

[SplitHorizontal](#)

[SplitVertical](#)

[AlignVerticalTop](#)

[AlignVerticalCenter](#)

[AlignVerticalBottom](#)

[AlignHorizontalLeft](#)

[AlignHorizontalRight](#)

[AlignHorizontalCenter](#)

[AlignHorizontalJustify](#)

[EditProperties](#)

Eigenschaften

[MayInsert](#)

[MayDelete](#)

[MayAppendRow](#)

[MayInsertRow](#)

[MayDeleteRow](#)

[MayAppendCol](#)

[MayInsertCol](#)

[MayDeleteCol](#)

[MayJoinLeft](#)

[MayJoinRight](#)

[MayJoinUp](#)

[MayJoinDown](#)

[MaySplitHorizontal](#)

[MaySplitVertical](#)

[MayAlignVertical](#)

[MayAlignHorizontal](#)

[MayEditProperties](#)

14.1 AuthenticXMLTableCommands.AlignHorizontalCenter

Deklaration: [AlignHorizontalCenter\(\)](#)

Beschreibung

Setzt die horizontale Ausrichtung auf "zentriert" oder setzt das Attribut zurück, wenn es zuvor auf "zentriert" gesetzt war.

14.2 AuthenticXMLTableCommands.AlignHorizontalJustify

Deklaration: [AlignHorizontalJustify\(\)](#)

Beschreibung

Setzt die horizontale Ausrichtung auf "Blocksatz" oder setzt das Attribut zurück, wenn es zuvor auf "Blocksatz" gesetzt war.

14.3 AuthenticXMLTableCommands.AlignHorizontalLeft

Deklaration: [AlignHorizontalLeft\(\)](#)

Beschreibung

Setzt die horizontale Ausrichtung auf "links" oder setzt das Attribut zurück, wenn es zuvor auf "links" gesetzt war.

14.4 AuthenticXMLTableCommands.AlignHorizontalRight

Deklaration: [AlignHorizontalRight\(\)](#)

Beschreibung

Setzt die horizontale Ausrichtung auf "rechts" oder setzt das Attribut zurück, wenn es zuvor auf "rechts" gesetzt war.

14.5 AuthenticXMLTableCommands.AlignVerticalBottom

Deklaration: [AlignVerticalBottom\(\)](#)

Beschreibung

Setzt die vertikale Ausrichtung auf "unten" oder setzt das Attribut zurück, wenn es zuvor auf "unten" gesetzt war.

14.6 AuthenticXMLTableCommands.AlignVerticalCenter

Deklaration: [AlignVerticalCenter\(\)](#)

Beschreibung

Setzt die vertikale Ausrichtung auf "Mitte" oder setzt das Attribut zurück, wenn es zuvor auf "Mitte" gesetzt war.

14.7 AuthenticXMLTableCommands.AlignVerticalTop

Deklaration: [AlignVerticalTop\(\)](#)

Beschreibung

Setzt die vertikale Ausrichtung auf "oben" oder setzt das Attribut zurück, wenn es zuvor auf "oben" gesetzt war.

14.8 AuthenticXMLTableCommands.AppendCol

Deklaration: [AppendCol\(\)](#)

Beschreibung

Hängt eine Spalte an die aktuelle Tabelle an.

14.9 AuthenticXMLTableCommands.AppendRow

Deklaration: [AppendRow\(\)](#)

Beschreibung

Hängt eine Zeile an die aktuelle Tabelle an.

14.10 AuthenticXMLTableCommands.Delete

Deklaration: [Delete\(\)](#)

Beschreibung

Wenn der Benutzer den Vorgang bestätigt hat, löscht die Methode die aktuell ausgewählte Tabelle.

14.11 AuthenticXMLTableCommands.DeleteCol

Deklaration: [DeleteCol\(\)](#)

Beschreibung

Die Methode löscht die aktuell ausgewählte Spalte.
Wenn es nur mehr eine Spalte gibt und der Benutzer den Vorgang bestätigt, wird die gesamte Tabelle gelöscht.

14.12 AuthenticXMLTableCommands.DeleteRow

Deklaration: [DeleteRow\(\)](#)

Beschreibung

Die Methode löscht die aktuell ausgewählte Zeile.

Wenn es nur mehr eine Zeile gibt und der Benutzer den Vorgang bestätigt, wird die gesamte Tabelle gelöscht.

14.13 AuthenticXMLTableCommands.EditProperties

Deklaration: [EditProperties\(\)](#)

Beschreibung

Zeigt das Dialogfeld "Eigenschaften" für die ausgewählte Tabelle/Zelle an.

14.14 AuthenticXMLTableCommands.Insert

Deklaration: [Insert\(\)](#)

Beschreibung

Ruft ein Dialogfeld auf und fügt eine neue Tabelle in die bestehende XML-Datei ein.

14.15 AuthenticXMLTableCommands.InsertCol

Deklaration: [InsertCol\(\)](#)

Beschreibung

Fügt eine neue Spalte vor der aktuelle ausgewählten Spalte ein.

14.16 AuthenticXMLTableCommands.InsertRow

Deklaration: [InsertRow\(\)](#)

Beschreibung

Fügt eine Zeile vor der aktuell ausgewählten Zeile ein.

14.17 AuthenticXMLTableCommands.JoinDown

Deklaration: [JoinDown\(\)](#)

Beschreibung

Verbindet die aktuelle Zelle mit der Zelle unmittelbar darunter.

14.18 AuthenticXMLTableCommands.JoinLeft

Deklaration: [JoinLeft\(\)](#)

Beschreibung

Verbindet die aktuelle Zelle mit der Zelle unmittelbar links davon.

14.19 AuthenticXMLTableCommands.JoinRight

Deklaration: [JoinRight\(\)](#)

Beschreibung

Verbindet die aktuelle Zelle mit der Zelle unmittelbar rechts davon.

14.20 AuthenticXMLTableCommands.JoinUp

Deklaration: [JoinUp\(\)](#)

Beschreibung

Verbindet die aktuelle Zelle mit der Zelle unmittelbar darüber.

14.21 AuthenticXMLTableCommands.MayAlignHorizontal

Deklaration: `MayAlignHorizontal` als `Boolean`

Beschreibung

True, wenn die Attribute für die horizontale Ausrichtung für die aktuelle Zelle gesetzt werden können.

14.22 AuthenticXMLTableCommands.MayAlignVertical

Deklaration: `MayAlignVertical` als `Boolean`

Beschreibung

True, wenn die Attribute für die vertikale Ausrichtung für die aktuelle Zelle gesetzt werden können.

14.23 AuthenticXMLTableCommands.MayAppendCol

Deklaration: `MayAppendCol` als `Boolean`

Beschreibung

True, wenn eine Spalte angehängt werden kann.

14.24 AuthenticXMLTableCommands.MayAppendRow

Deklaration: `MayAppendRow` als `Boolean`

Beschreibung

True, wenn eine Zeile angehängt werden kann.

14.25 AuthenticXMLTableCommands.MayDelete

Deklaration: `MayDelete` als `Boolean`

Beschreibung

Die Eigenschaft ist "true", wenn eine Tabelle ausgewählt ist.

14.26 AuthenticXMLTableCommands.MayDeleteCol

Deklaration: `MayDeleteCol` als `Boolean`

Beschreibung

True, wenn die aktuelle Spalte gelöscht werden kann.

14.27 AuthenticXMLTableCommands.MayDeleteRow

Deklaration: `MayDeleteRow` als `Boolean`

Beschreibung

True, wenn die aktuelle Zeile gelöscht werden kann.

14.28 AuthenticXMLTableCommands.MayEditProperties

Deklaration: `MayEditProperties` als `Boolean`

Beschreibung

Die Eigenschaft ist "true", wenn das Dialogfeld "Eigenschaften" für die ausgewählte Tabelle/Zelle zur Verfügung steht.

14.29 AuthenticXMLTableCommands.MayInsert

Deklaration: `MayInsert` als `Boolean`

Beschreibung

Die Eigenschaft ist "true", wenn an der aktuellen Auswahl eine neue Tabelle eingefügt werden kann.

14.30 AuthenticXMLTableCommands.MayInsertCol

Deklaration: `MayInsertCol` als `Boolean`

Beschreibung

True, wenn eine Spalte eingefügt werden kann.

14.31 AuthenticXMLTableCommands.MayInsertRow

Deklaration: `MayInsertRow` als `Boolean`

Beschreibung

True, wenn eine Zeile eingefügt werden kann.

14.32 AuthenticXMLTableCommands.MayJoinDown

Deklaration: `MayJoinDown` als `Boolean`

Beschreibung

Die Eigenschaft ist "true", wenn die JoinDown-Operation für die aktuell ausgewählte Zelle möglich ist.

14.33 AuthenticXMLTableCommands.MayJoinLeft

Deklaration: `MayJoinLeft` als `Boolean`

Beschreibung

Die Eigenschaft ist true, wenn die JoinLeft-Operation für die aktuell ausgewählte Zelle möglich ist.

14.34 AuthenticXMLTableCommands.MayJoinRight

Deklaration: `MayJoinRight` als `Boolean`

Beschreibung

Die Eigenschaft ist true, wenn die JoinRight-Operation für die aktuell ausgewählte Zelle möglich ist.

14.35 AuthenticXMLTableCommands.MayJoinUp

Deklaration: `MayJoinUp` als `Boolean`

Beschreibung

Die Eigenschaft ist "true", wenn die JoinUp-Operation für die aktuell ausgewählte Zelle möglich ist.

14.36 AuthenticXMLTableCommands.MaySplitHorizontal

Deklaration: `MaySplitHorizontal` als `Boolean`

Beschreibung

True, wenn die aktuelle Zelle horizontal geteilt werden kann.

14.37 AuthenticXMLTableCommands.MaySplitVertical

Deklaration: `MaySplitVertical` als `Boolean`

Beschreibung

True, wenn die aktuelle Zelle vertikal geteilt werden kann.

14.38 AuthenticXMLTableCommands.SplitHorizontal

Deklaration: [SplitHorizontal\(\)](#)

Beschreibung

Die Methode teilt die aktuelle Zelle horizontal.

14.39 AuthenticXMLTableCommands.SplitVertical

Deklaration: [SplitVertical\(\)](#)

Beschreibung

Die Methode teilt die aktuelle Zelle vertikal.

15 XMLData

Siehe auch

Methoden

[InsertChild](#)

[AppendChild](#)

[EraseAllChildren](#)

[EraseCurrentChild](#)

[GetCurrentChild](#)

[GetFirstChild](#)

[GetNextChild](#)

[IsSameNode](#)

Eigenschaften

[Name](#)

[TextValue](#)

[HasChildren](#)

[MayHaveChildren](#)

[Kind](#)

[Parent](#)

Beschreibung

Mit Hilfe der XMLData-Schnittstelle können Sie den Inhalt der gerade angezeigten XML-Datei manipulieren. Die Schnittstelle ist ein vereinfachtes Pendant zur COM-Schnittstelle, die im Plug-In und in XMLSpy 2005 selbst implementiert ist.

Verwenden Sie zum Erstellen eines neuen XMLData-Objekts die CreateChild()-Methode der Plug-In-Schnittstelle.

15.1 XMLData.AppendChild

Siehe auch

Deklaration: `AppendChild(pNewData als XMLData)`

Beschreibung

AppendChild hängt pNewData als letztes Child an das XMLData-Objekt an. Siehe auch "[Using XMLData](#)".

Beispiel

```
Dim objCurrentParent
Dim objNewChild

Set objNewChild = objPlugIn.CreateChild(spyXMLDataElement)
Set objCurrentParent = objPlugIn.XMLRoot

objCurrentParent.AppendChild objNewChild

Set objNewChild = Nothing
```

15.2 XMLData.EraseAllChildren

Siehe auch

Deklaration: [EraseAllChildren](#)

Beschreibung

EraseAllChildren löscht alle verknüpften Children des XMLData-Objekts.

Beispiel

In diesem Beispiel werden alle Elemente des aktiven Dokuments gelöscht.

```
Dim objCurrentParent

Set objCurrentParent = objPlugIn.XMLRoot
objCurrentParent.EraseAllChildren
```

15.3 XMLData.EraseCurrentChild

Siehe auch

Deklaration: [EraseCurrentChild](#)

Beschreibung

EraseCurrentChild löscht das aktuelle XMLData Child-Objekt. Bevor Sie EraseCurrentChild aufrufen, müssen Sie mit [XMLData.GetFirstChild](#) einen internen Iterator initialisieren.

Beispiel

In diesem JavaScript-Beispiel werden alle Elemente mit dem Namen "EraseMe" gelöscht. Sie sehen im Code, dass innerhalb eines Schleifendurchlaufs EraseCurrentChild und GetNextChild aufgerufen werden können, sodass die Child-Elemente der Reihe nach abgearbeitet werden können.

```
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;
      }
    }
  }
}
```

15.4 XMLData.GetCurrentChild

Siehe auch

Deklaration: [GetCurrentChild](#) als [XMLData](#)

Rückgabewert

Gibt ein XML-Element als XMLData-Objekt zurück.

Beschreibung

[GetCurrentChild](#) holt das aktuelle Child. Bevor Sie [GetCurrentChild](#) aufrufen, müssen Sie mit [XMLData.GetFirstChild](#) einen internen Iterator initialisieren.

15.5 XMLData.GetFirstChild

Siehe auch

Deklaration: `GetFirstChild`(*nKind* als [SPYXMLDataKind](#)) als [XMLData](#)

Rückgabewert

Gibt ein XML-Element als XMLData-Objekt zurück.

Beschreibung

`GetFirstChild` initialisiert einen neuen Iterator und gibt das erste Child zurück. Setzen Sie *nKind* = -1, um einen Iterator für alle Arten von Children zu holen.

Beispiel

Siehe Beispiel unter [XMLData.GetNextChild](#).

15.6 XMLData.GetNextChild

Siehe auch

Deklaration: [GetNextChild](#) als [XMLData](#)

Rückgabewert

Gibt ein XML-Element als XMLData-Objekt zurück.

Beschreibung

GetNextChild geht zum nächsten Child dieses Elements. Bevor Sie GetNextChild aufrufen, müssen Sie mit [XMLData.GetFirstChild](#) einen internen Iterator initialisieren.

Überprüft, wann das letzte Child des Elements erreicht wurde (siehe Beispiel unten).

Beispiel

```
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)
```

15.7 XMLData.HasChildren

Siehe auch

Deklaration: [HasChildren](#) als [Boolean](#)

Beschreibung

Die Eigenschaft ist "true", wenn das Objekt ein Parent eines anderen XMLData-Objekts ist.

Diese Eigenschaft ist schreibgeschützt.

15.8 XMLData.InsertChild

Siehe auch

Deklaration: `InsertChild`(*pNewData* als [XMLData](#))

Beschreibung

`InsertChild` fügt das neue Child vor dem aktuellen Child ein (Siehe auch [XMLData.GetFirstChild](#), [XMLData.GetNextChild](#) zu Setzen des aktuellen Child).

15.9 XMLData.IsSameNode

Siehe auch

Deklaration: `IsSameNode`(*pNodeToCompare* als [XMLData](#)) als `Boolean`

Beschreibung

Gibt "true" zurück, wenn `pNodeToCompare` denselben Node referenziert wie das Objekt selbst.

15.10 XMLData.Kind

Siehe auch

Deklaration: Kind als [SPYXMLDataKind](#)

Beschreibung

Typ dieses XMLData-Objekts.

Diese Eigenschaft ist schreibgeschützt.

15.11 XMLData.MayHaveChildren

Siehe auch

Deklaration: [MayHaveChildren](#) als [Boolean](#)

Beschreibung

Gibt an, ob es zulässig ist, Children zu diesem XMLData-Objekt hinzuzufügen.

Diese Eigenschaft ist schreibgeschützt.

15.12 XMLData.Name

Siehe auch

Deklaration: `Name` als `String`

Beschreibung

Wird verwendet, um den Namen des XMLData-Objekts zu holen und zu ändern.

15.13 XMLData.Parent

Siehe auch

Deklaration: [Parent](#) als [XMLData](#)

Rückgabewert

Parent als XMLData-Objekt.

Kein Wert (oder NULL), wenn es kein Parent-Element gibt.

Beschreibung

Parent von diesem Element.

Diese Eigenschaft ist schreibgeschützt.

15.14 XMLData.TextValue

Siehe auch

Deklaration: [TextValue](#) als [String](#)

Beschreibung

Wird verwendet, um den Textwert von diesem XMLData-Objekt zu ändern und zu holen. Siehe auch "[Using XMLData](#)".

Kapitel 5

Enumerationen

Enumerationen

1 SPYXMLDataKind

Beschreibung

Die verschiedenen Typen von XMLData-Elementen, die für XML-Dokumente zur Verfügung stehen

Mögliche Werte:

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

2 SPYAuthenticElementActions

Beschreibung

Aktionen, die mit [GetAllowedElements](#) verwendet werden können.

Mögliche Werte:

k_ActionInsertAt	= 0
k_ActionApply	= 1
k_ActionClearSurr	= 2
k_ActionAppend	= 3
k_ActionInsertBefore	= 4
k_ActionRemove	= 5

3 SPYAuthenticCommand

Beschreibung

Enumeration aller verfügbaren Befehle.

Mögliche Werte:

```
// 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_CommandRowMoveDow  = 19
    n
    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

// seit TypeLib 1.2
k_CommandCheckSpelling      = 43
k_CommandAbout              = 44
k_CommandPackageManagement  = 45
```

4 SPYAuthenticCommandGroup

Beschreibung

Gruppen, zu denen ein Befehl gehören kann.

Mögliche Werte:

k_CommandGroupMain	= 0
k_CommandGroupEdit	= 1
k_CommandGroupMarkup	= 2
k_CommandGroupRow	= 3
k_CommandGroupXMLTables	= 4

5 SPYAuthenticToolbarAllignment

Beschreibung

Werte zur Definition der Symbolleistenposition.

Mögliche Werte:

k_ToolbarAlignTop	= 0
k_ToolbarAlignLeft	= 1
k_ToolbarAlignBottom	= 2
k_ToolbarAlignRight	= 3

6 SPYAuthenticEntryHelperWindows

Beschreibung

Identifizierung der verfügbaren Eingabehilfenfenster

Mögliche Werte:

k_Elements = 1

k_Attributes = 2

k_Entities = 4

7 SPYAuthenticElementKind

Beschreibung

Enumeration der verschiedenen zur Navigation und Auswahl innerhalb der Objekte [AuthenticRange](#) und [AuthenticView](#) verwendeten Elementarten.

Mögliche Werte:

spyAuthenticChar	= 0
spyAuthenticWord	= 1
spyAuthenticLine	= 3
spyAuthenticParagraph	= 4
spyAuthenticTag	= 6
spyAuthenticDocument	= 8
spyAuthenticTable	= 9
spyAuthenticTableRow	= 10
spyAuthenticTableColumn	= 11

8 SPYAuthenticActions

Beschreibung

Aktionen, die an [AuthenticRange](#)-Objekten ausgeführt werden können.

Mögliche Werte:

spyAuthenticInsertAt	= 0
spyAuthenticApply	= 1
spyAuthenticClearSurr	= 2
spyAuthenticAppend	= 3
spyAuthenticInsertBefore	= 4
spyAuthenticRemove	= 5

9 SPYAuthenticDocumentPosition

Beschreibung

Relative und absolute Position zum Navigieren mit [AuthenticRange](#)-Objekten.

Mögliche Werte:

spyAuthenticDocumentBegin	= 0
spyAuthenticDocumentEnd	= 1
spyAuthenticRangeBegin	= 2
spyAuthenticRangeEnd	= 3

10 SPYAuthenticMarkupVisibility

Beschreibung

Enumerationswerte zum Aus- oder Einblenden des Markups.

Mögliche Werte:

spyAuthenticMarkupHidden	= 0
spyAuthenticMarkupSmall	= 1
spyAuthenticMarkupLarge	= 2
spyAuthenticMarkupMixed	= 3

Kapitel 6

Altova Developer License Agreement

Altova Developer License Agreement

THIS IS A LEGAL DOCUMENT -- RETAIN FOR YOUR RECORDS

ALTOVA® END USER LICENSE AGREEMENT FOR AUTHENTIC® DESKTOP EDITION AND AUTHENTIC® BROWSER EDITION SOFTWARE

Licensor:

Altova GmbH
Rudolfsplatz 13a/9
A-1010 Wien
Austria

Important - Read Carefully. Notice to User:

This Altova End User License Agreement for Authentic® ("Authentic EULA") governs your right to use (i) the Authentic Desktop Edition software ("Authentic Desktop Edition") and/or (ii) the Authentic Browser Edition software ("Authentic Browser Edition") (each or collectively hereinafter referenced, as "Authentic Software"). This Authentic EULA is a legal document between you and Altova GmbH ("Altova"). It is important that you read this document before using the Altova-provided software and any accompanying documentation, including, without limitation, printed materials, 'online' files, or electronic documentation ("Documentation"). By clicking the "I accept" and "Next" buttons below, or by installing, or otherwise using the Authentic Software, you agree to be bound by the terms of this Authentic EULA as well as the Altova Privacy Policy ("Privacy Policy") including, without limitation, the warranty disclaimers, limitation of liability, data use and termination provisions below. You agree that this agreement is enforceable like any written agreement negotiated and signed by you. If you do not agree, you are not licensed to use the Authentic Software, and you must destroy any downloaded copies of the Authentic Software in your possession or control. Please go to our Web site at <http://www.altova.com/authenticeula> to download and print a copy of this Authentic EULA for your files and <http://www.altova.com/privacy> to review the privacy policy.

1. AUTHENTIC SOFTWARE LICENSE

- (a) **License Grant.** Upon your acceptance of this Authentic EULA, Altova grants you a personal non-exclusive, non-transferable (except as provided below), limited license to **install and use** a copy of the Authentic Software on your compatible computer for the purpose of viewing, distributing, sharing, and editing of XML files solely in connection with STYLEVISION® Power Stylesheets hereinafter defined, as further provided herein.
- (b) **Backup and Archival Copies.** You may make one backup and one archival copy of the Authentic Software, provided your backup and archival copies are not installed or used on any computer and further provided that all such copies shall bear the original and unmodified copyright, patent and other intellectual property markings that appear on or in the Authentic Software.
- (c) **Key Codes.** Upon acceptance of this Authentic EULA with respect to Authentic Desktop Edition, you will receive a key code that will enable you to activate or operate the Authentic Software. You may not relicense, reproduce or distribute any key code except with the express written permission of Altova.
- (d) **Title.** This Authentic EULA gives you a limited license to use and distribute to the limited extent permitted in Section 3, as applicable, the Authentic Software. Altova and its suppliers retain all right, title and interest, including all copyright and intellectual property rights,

in and to, the Authentic Software and all copies thereof. All rights not specifically granted in this Authentic EULA are reserved by Altova. As between you and Altova, documents, files, stylesheets, generated program code and schemas that are authored or created by you via your utilization of the Authentic Software, in accordance with its Documentation and the terms of this Authentic EULA, are your property.

(e) **Reverse Engineering.** You may not reverse engineer, decompile, disassemble or otherwise attempt to discover the source code, underlying ideas, underlying user interface techniques or algorithms of the Authentic Software by any means whatsoever, directly or indirectly, or disclose any of the foregoing, except to the extent you may be expressly permitted to decompile under applicable law in the European Union, if it is essential to do so in order to achieve operability of the Authentic Software with another software program, and you have first requested Altova to provide the information necessary to achieve such operability and Altova has not made such information available. Altova has the right to impose reasonable conditions and to request a reasonable fee before providing such information. Any information supplied by Altova or obtained by you, as permitted hereunder, may only be used by you for the purpose described herein and may not be disclosed to any third party or used to create any software which is substantially similar to the expression of the Authentic Software. Requests for information from users in the European Union with respect to the above should be directed to the Altova Customer Support Department.

(f) **Additional Restrictions.** You may not loan, rent, lease, sublicense, distribute or otherwise transfer all or any portion of the Authentic Software to third parties except to the limited extent expressly set forth in Section 3. You may not copy, distribute, or make derivative works of the Authentic Software except as expressly set forth above, and any copies that you are permitted to make pursuant to this Authentic EULA must contain the same copyright, patent and other intellectual property markings that appear on or in the Authentic Software. You may not modify, adapt or translate the Authentic Software. You may not, directly or indirectly, encumber or suffer to exist any lien or security interest on the Authentic Software; knowingly take any action that would cause the Authentic Software to be placed in the public domain; or use the Authentic Software in any computer environment not specified in this Authentic EULA. You will comply with applicable law and Altova's instructions regarding the use of the Authentic Software. You agree to notify your employees and agents who may have access to the Authentic Software of the restrictions contained in this Authentic EULA and to ensure their compliance with these restrictions. You may not alter or modify the Authentic Software or create a new installer for the Authentic Software. The Authentic Software is licensed and distributed by Altova for viewing, distributing, sharing, and editing of XML files solely in connection with STYLEVISION Power Stylesheets, defined as .sps files that are template files developed by Altova or its customers using Altova's STYLEVISION product. You are not authorized to integrate or use the Authentic Software with any (a) STYLEVISION Power Stylesheet(s) not developed in accordance with the Altova Software License Agreement available at <http://www.altova.com/eula> (b) other software or enhancement that uses Inter Application Communication (IAC) to programmatically interface with Authentic Software for the purpose of (i) enabling additional functionality normally not available in Authentic Software, (ii) providing functionality that competes with other Altova products

2. INTELLECTUAL PROPERTY RIGHTS

Acknowledgement of Altova's Rights. You acknowledge that the Authentic Software and any copies that you are authorized by Altova to make are the intellectual property of and are owned by Altova and its suppliers. The structure, organization and code of the Authentic Software are the valuable trade secrets and confidential information of Altova and its suppliers. The Authentic Software is protected by copyright, including without limitation by United States Copyright Law, international treaty provisions and applicable laws in the country in which it is being used. You acknowledge that Altova retains the ownership of all patents, copyrights, trade secrets, trademarks and other intellectual property rights pertaining to the software, and that Altova's ownership rights extend to any images, photographs, animations, videos, audio, music, text and "applets" incorporated into Authentic Software and all accompanying printed materials. You will take no actions which adversely affect Altova's intellectual property rights in the Authentic

Software. Trademarks shall be used in accordance with accepted trademark practice, including identification of trademark owners' names. Trademarks may only be used to identify printed output produced by the software, and such use of any trademark does not give you any right of ownership in that trademark. XMLSPY, AUTHENTIC, STYLEVISION, MAPFORCE, MARKUP YOUR MIND, AXAD, NANONULL, and ALTOVA are trademarks of Altova GmbH (registered in numerous countries). Unicode and the Unicode Logo are trademarks of Unicode, Inc. Windows, Windows 95, Windows 98, Windows NT, Windows 2000 and Windows XP are trademarks of Microsoft. W3C, CSS, DOM, MathML, RDF, XHTML, XML and XSL are trademarks (registered in numerous countries) of the World Wide Web Consortium (W3C); marks of the W3C are registered and held by its host institutions, MIT, INRIA and Keio. Except as expressly stated above, this Authentic EULA does not grant you any intellectual property rights in the Authentic Software. Notifications of claimed copyright infringement should be sent to Altova's copyright agent as further provided on the Altova Web Site

3. AUTHENTIC DESKTOP EDITION DISTRIBUTION

(a) **License Grant.** Upon your acceptance of this Authentic EULA as part of your use of the Authentic Desktop Edition, and subject to your ongoing compliance with its terms and conditions, Altova hereby grants Authentic Desktop Edition users a non-exclusive, non-transferable, limited worldwide license to reproduce the Setup Program for Authentic Desktop Edition and distribute the Setup Program for Authentic Desktop Edition in executable form to end users in the manner hereinafter provided.

(b) **Internal Use.** You may install one copy of such Setup Program for Authentic Desktop Edition on a computer file server within your internal network for the sole and exclusive purpose of installing Authentic Desktop Edition (to an unlimited number of client computers on your internal network). No other server or network use of the Authentic Desktop Edition is permitted, including but not limited to using the Authentic Desktop Edition (i) either directly or through commands, data or instructions from or to another computer or (ii) for internal network, internet or web hosting services.

(c) **External Use.** You may distribute the Setup Program for Authentic Desktop Edition to any third party electronically or via download from the website or on physical media such as CD-ROMS or diskettes as part of or in conjunction with products that you have developed.

(d) **Distribution Restrictions.** In addition to the restrictions and obligations provided in other sections of this Authentic EULA, your license to distribute the Setup Program for Authentic Desktop Edition is further subject to all of the following restrictions (i) Authentic Desktop Edition shall only be licensed and not sold, (ii) you may not make Authentic Desktop Edition available as a stand alone product and if distributed as part of a product bundle you may charge for the product bundle provided that you license such product bundle at the same or lower fee at which you license any reasonably equivalent product bundle which does not include Authentic Desktop Edition, (iii) You must use the Setup Program provides by Altova **AS IS** and may not impair, alter or remove Altova's Authentic EULA, (which will appear in the installation process and which an end user must accept in order to be able to install or operate Authentic Software) or any other files, (iv) Other versions of Authentic Software available cannot be distributed under this Agreement; and (v) You may not combine the Authentic Software with your product in such a way that your product modifies or generates STYLEVISION Power Stylesheets.

4. WARRANTY DISCLAIMER

THE AUTHENTIC SOFTWARE IS PROVIDED TO YOU FREE OF CHARGE, AND ON AN "AS-IS" BASIS. ALTOVA PROVIDES **NO** WARRANTIES FOR THE AUTHENTIC SOFTWARE. TO THE MAXIMUM EXTENT PERMITTED BY LAW, ALTOVA AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES AND REPRESENTATIONS, WHETHER EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE; MERCHANTABILITY; SATISFACTORY QUALITY, INFORMATIONAL CONTENT, OR ACCURACY, QUIET ENJOYMENT, TITLE, AND NON- INFRINGEMENT. ALTOVA DOES NOT WARRANT THAT THE AUTHENTIC SOFTWARE IS ERROR-FREE OR WILL

OPERATE WITHOUT INTERRUPTION. IF APPLICABLE LAW REQUIRES ANY WARRANTIES WITH RESPECT TO THE AUTHENTIC SOFTWARE, ALL SUCH WARRANTIES ARE LIMITED IN DURATION TO 30 DAYS FROM THE DATE OF INSTALLATION OR USE. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS AND YOU MAY ALSO HAVE OTHER LEGAL RIGHTS THAT VARY FROM STATE TO STATE. you agree that you are solely responsible for the accuracy and adequacy of the software for your intended use and you will indemnify and HOLD harmless ALTOVA from any 3rd party suit to the extent based upon the accuracy and adequacy of the software in your use. Without limitation, The Software is not intended for use in hazardous environments REQUIRING fail-safe controls INCLUDING without LIMITATION THE operation of nuclear facilities, aircraft navigation, communication systems, AIR traffic CONTROL, life SUPPORT, or weapons systems, where the failure of the Software could lead to death, personal INJURY, or severe physical or environmental damage.

5. **LIMITATION OF LIABILITY**

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL ALTOVA OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE AUTHENTIC SOFTWARE, THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT SERVICES, OR ANY PROVISION OF THIS AUTHENTIC END USER LICENSE AGREEMENT, EVEN IF ALTOVA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. WHERE LEGALLY, LIABILITY CANNOT BE EXCLUDED, BUT MAY BE LIMITED, ALTOVA'S LIABILITY AND THAT OF ITS SUPPLIERS SHALL BE LIMITED TO THE SUM OF FIFTY DOLLARS (USD \$50) IN TOTAL. BECAUSE SOME STATES AND JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY, THE ABOVE LIMITATION MAY NOT APPLY TO YOU. IN SUCH STATES AND JURISDICTIONS, ALTOVA'S LIABILITY AND THAT OF ITS SUPPLIERS SHALL BE LIMITED TO THE GREATEST EXTENT PERMITTED BY LAW. THE FOREGOING LIMITATIONS ON LIABILITY ARE INTENDED TO APPLY TO THE WARRANTIES AND DISCLAIMERS ABOVE AND ALL OTHER ASPECTS OF THIS AUTHENTIC END USER LICENSE AGREEMENT.

6. **SUPPORT**

Altova provides no technical support for the version of the Authentic software product edition that you have licensed unless you have elected to purchase a "Support & Maintenance Package ("SMP"). The Support Period, hereinafter defined, covered by such SMP shall be delineated at such time as you elect to purchase a SMP. Your rights with respect to support as well as your upgrade eligibility depend on your decision to purchase a SMP and the level of SMP that you have purchased. All Technical support is provided via a Web-based support form only, and there is no guaranteed response time. If you have purchased SMP, then solely for the duration of its delineated Support Period, **you are eligible to receive the version of the Authentic Software edition** that you have licensed and all maintenance releases and updates for that edition that are released during your Support Period. For the duration of your SMP's Support Period, you will also be eligible to receive upgrades to the comparable edition of the next version of the Software that succeeds the Software edition that you have licensed for applicable upgrades released during your Support Period. Software that is introduced as separate product is not included in SMP. Maintenance releases, updates and upgrades may or may not include additional features. In addition, Altova will provide Priority Technical Support to you for the duration of the Support Period. Priority Technical Support is provided via a Web-based support form only, and Altova will make commercially reasonable efforts to respond via e-mail to all requests within forty-eight (48) hours during Altova's business hours (MO-FR, 8am UTC –

10pm UTC, Austrian and US holidays excluded) and to make reasonable efforts to provide work-arounds to errors reported in the Software.

During the Support Period you may also report any Authentic Software problem or error to Altova. In its sole discretion, Altova may elect to correct or provide a usable work-around solution in an upcoming maintenance release or update, which is made available at certain times at Altova's sole discretion. If Altova, in its discretion, requests written verification of an error or malfunction discovered by you or requests supporting example files that exhibit the Authentic Software problem, you shall promptly provide such verification or files, by email, telecopy, or overnight mail, setting forth in reasonable detail the respects in which the Authentic Software fails to perform. You shall use reasonable efforts to cooperate in diagnosis or study of errors. Altova may include error corrections in maintenance releases, updates, or new major releases of the Authentic Software. Altova is not obligated to fix errors. Whether or not you have purchased the Support & Maintenance Package, technical support only covers issues or questions resulting directly out of the operation of the Authentic Software and Altova will not provide you with generic consultation, assistance, or advice under any circumstances. The provision of technical support in no way constitutes a warranty for Authentic Software or the use or performance thereof

Updating Authentic Software may require the updating of software not covered by this Authentic EULA before installation. Updates of the operating system and application software not specifically covered by this Authentic EULA are your responsibility and will not be provided by Altova under this Authentic EULA. Altova's support obligations under this Section 6 are contingent upon your proper use of the Authentic Software and your compliance with the terms and conditions of this Authentic EULA at all times. Altova shall be under no obligation to provide the above technical support if, in Altova's opinion, the Authentic Software has failed due to the following conditions: (i) damage caused by the relocation of the software to another location or CPU; (ii) alterations, modifications or attempts to change the Authentic Software without Altova's written approval; (iii) causes external to the Authentic Software, such as natural disasters, the failure or fluctuation of electrical power, or computer equipment failure; (iv) your failure to maintain the Authentic Software at Altova's specified release level; or (v) use of the Authentic Software with other software without Altova's prior written approval. It will be your sole responsibility to: (i) comply with all Altova-specified operating and troubleshooting procedures and then notify Altova immediately of Authentic Software malfunction and provide Altova with complete information thereof; (ii) provide for the security of your confidential information; (iii) establish and maintain backup systems and procedures necessary to reconstruct lost or altered files, data or programs.

- 7. AUTHENTIC SOFTWARE ACTIVATION, UPDATES, METERING AND DATA USE**
- Altova has a built-in license metering module that helps you to avoid any unintentional violation of this Authentic EULA and Altova may use your internal network for such license metering. The Authentic Software may use your internal network and internet connection for the purpose of transmitting license-related data at the time of installation, registration or update to an Altova-operated license server and validating the authenticity of the license-related data in order to protect Altova against unlicensed or illegal use of the Authentic Software and to improve customer support. Activation is based on the exchange of license related data between your computer and the Altova license server. You agree that Altova may use these measures and you agree to follow any applicable requirements. In the event that Altova provides LiveUpdate notification service(s) to you, Altova may use your internal network and internet connection for the purpose of transmitting license-related data to an Altova-operated LiveUpdate server to validate your license at appropriate intervals and determine if there is any update available for you. The terms and conditions of the Privacy Policy are set out in full at <http://www.altova.com/privacy> and are incorporated by reference into this Authentic EULA. By your acceptance of the terms of this Authentic EULA or use of the Authentic Software, you authorize the collection, use and disclosure of information collected by Altova for the purposes provided for in this Authentic EULA and/or the Privacy Policy as revised from time to time. European users understand and consent to the processing of personal information in the United States for the purposes described herein. Altova has the right in its sole discretion to amend this

provision of the Authentic EULA and/or Privacy Policy at any time. You are encouraged to review the terms of the Privacy Policy as posted on the Altova Web site from time to time.

8. EXPORT RULES AND GOVERNMENT RESTRICTED RIGHTS

The Authentic Software was developed entirely at private expense and is commercial computer software provided with RESTRICTED RIGHTS. Use, duplication or disclosure by the U.S. Government or a U.S. Government contractor or subcontractor is subject to the restrictions set forth in this Agreement and as provided in FAR 12.211 and 12.212 (48 C.F.R. §12.211 and 12.212) or DFARS 227. 7202 (48 C.F.R. §227-7202) as applicable. Consistent with the above as applicable, Commercial Computer Software and Commercial Computer Documentation is licensed to U.S. government end users only as commercial items and only with those rights as are granted to all other end users under the terms and conditions set forth in this Authentic EULA. Manufacturer is Altova GmbH, Rudolfsplatz, 13a/9, A-1010 Vienna, Austria/EU. You may not use or otherwise export or reexport the Authentic Software or Documentation except as authorized by United States law and the laws of the jurisdiction in which the Authentic Software was obtained. In particular, but without limitation, the Authentic Software or Documentation may not be exported or reexported (i) into (or to a national or resident of) any U.S. embargoed country or (ii) to anyone on the U.S. Treasury Department's list of Specially Designated Nationals or the U.S. Department of Commerce's Table of Denial Orders. By using the Authentic Software, you represent and warrant that you are not located in, under control of, or a national or resident of any such country or on any such list.

9. TERM AND TERMINATION

Without prejudice to any other rights or remedies of Altova, this Authentic EULA may be terminated (a) by you giving Altova written notice of termination; or (b) by Altova, at its option, giving you written notice of termination or (c) Altova giving you written notice of termination if you fail to comply with the terms and conditions of the Authentic EULA. Upon any termination of this Authentic EULA, you must cease all use of Authentic Software, licensed hereunder, destroy all copies then in your possession or control and take such other actions as Altova may reasonably request to ensure that no copies of the Authentic Software remain in your possession or control. The terms and conditions set forth in Sections 1 (d), (e), (f), 2, 4, 5, 8, and 10 survive termination of this agreement as applicable.

10. GENERAL PROVISIONS

If you are located in the European Union and are using the Authentic Software in the European Union and not in the United States, then this Authentic EULA will be governed by and construed in accordance with the laws of the Republic of Austria (excluding its conflict of laws principles and the U.N. Convention on Contracts for the International Sale of Goods) and you expressly agree that exclusive jurisdiction for any claim or dispute with Altova, or relating in any way to your use of the Authentic Software resides in the Handelsgericht Wien (Commercial Court, Vienna) and you further agree and expressly consent to the exercise of personal jurisdiction in the Handelsgericht Wien (Commercial Court, Vienna) in connection with any such dispute or claim.

If you are located in the United States or are using the Authentic Software in the United States then this Authentic EULA will be governed by and construed in accordance with the law of the Commonwealth of Massachusetts, USA (excluding its conflict of laws principles and the U.N. Convention on Contracts for the International Sale of Goods) and you expressly agree that exclusive jurisdiction for any claim or dispute with Altova or relating in any way to your use of the Authentic Software resides in the federal or state courts of Massachusetts and you further agree and expressly consent to the exercise of personal jurisdiction in the federal or state courts of Massachusetts in connection with any such dispute or claim.

If you are located outside of the European Union or the United States and are not using the Authentic Software in the United States, then this Authentic EULA will be governed by and construed in accordance with the laws of the Republic of Austria (excluding its conflict of laws principles and the U.N. Convention on Contracts for the International Sale of Goods) and you expressly agree that exclusive jurisdiction for any claim or dispute with Altova, or relating in any

way to your use of the Authentic Software resides in the Handelsgericht Wien (Commercial Court, Vienna) and you further agree and expressly consent to the exercise of personal jurisdiction in the Handelsgericht Wien (Commercial Court, Vienna) in connection with any such dispute or claim.

This Authentic EULA will not be governed by the conflict of law rules of any jurisdiction or the United Nations Convention on Contracts for the International Sale of Goods, the application of which is expressly excluded. This Authentic EULA contains the entire agreement and understanding of the parties with respect to the subject matter hereof, and supersedes all prior written and oral understandings of the parties with respect to the subject matter hereof. Any notice or other communication given under this Authentic EULA shall be in writing and shall have been properly given by either of us to the other if sent by certified or registered mail, return receipt requested, or by overnight courier to the address shown on Altova's Web site for Altova and the address shown in Altova's records for you, or such other address as the parties may designate by notice given in the manner set forth above. This Authentic EULA will bind and inure to the benefit of the parties and our respective heirs, personal and legal representatives, affiliates, successors and permitted assigns. The failure of either of us at any time to require performance of any provision hereof shall in no manner affect such party's right at a later time to enforce the same or any other term of this Authentic EULA. This Authentic EULA may be amended only by a document in writing signed by both of us. In the event of a breach or threatened breach of this Authentic EULA by either party, the other shall have all applicable equitable as well as legal remedies. Each party is duly authorized and empowered to enter into and perform this Authentic EULA. If, for any reason, any provision of this Authentic EULA is held invalid or otherwise unenforceable, such invalidity or unenforceability shall not affect the remainder of this Authentic EULA, and this Authentic EULA shall continue in full force and effect to the fullest extent allowed by law. The parties knowingly and expressly consent to the foregoing terms and conditions.

Last updated 2004-09-22

Index

A

ANSI, 9

versions of Authentic Browser Plug-in, 9

AuthenticRange-Objekt, 24

Authentic, 45, 46, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 69, 70, 71, 72, 74, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 97, 98, 101, 102, 103, 104, 105, 106, 107, 108, 109, 111, 112, 113, 114, 115, 116, 118, 126, 127, 128, 129, 130, 131

attachCallBack, 46
 ControlInitialized, 53
 CreateChild, 54
 CurrentSelection, 55
 DesignDataLoadObject, 56
 EditClear, 57
 EditCopy, 58
 EditCut, 59
 EditPaste, 60
 EditRedo, 61
 EditSelectAll, 62
 EditUndo, 63
 event, 69
 FindDialog, 70
 FindNext, 71
 GetAllAttributes, 72
 GetAllowedElements, 74
 GetFileVersion, 76
 GetNextVisible, 77
 GetPreviousVisible, 78
 IsEditClearEnabled, 79
 IsEditCopyEnabled, 80
 IsEditCutEnabled, 81
 IsEditPasteEnabled, 82
 IsEditRedoEnabled, 83
 IsEditUndoEnabled, 84
 IsFindNextEnabled, 85
 IsRowAppendEnabled, 86
 IsRowDeleteEnabled, 87
 IsRowDuplicateEnabled, 88
 IsRowInsertEnabled, 89
 IsRowMoveDownEnabled, 90
 IsRowMoveUpEnabled, 91

IsTextStateApplied, 92
 IsTextStateEnabled, 93
 LoadXML, 94
 MarkupView, 95
 Print, 97
 PrintPreview, 98
 ReplaceDialog, 101
 Reset, 102
 RowAppend, 103
 RowDelete, 104
 RowInsert, 106
 RowMoveDown, 107
 RowMoveUp, 108
 Save, 109
 SavePOST, 111
 SaveXML, 112
 SchemaLoadObject, 113
 SelectionChanged, 114
 SelectionMoveTabOrder, 115
 SelectionSet, 116
 StartEditing, 118
 TextState anwenden, 45
 ValidateDocument, 126
 validationBadData, 127
 validationMessage, 128
 XMLDataLoadObject, 129
 XMLDataSaveUrl, 130
 XMLRoot, 131

Authentic Browser, 42

herunterladen, 4
 specifying version to use, 9
 verfügbare Versionen, 4

Authentic RowDuplicate, 105

Authentic View, 9

dimensions of in Plug-in, 9

AuthenticDataTransfer, 142, 143, 144, 145

dropEffect, 142
 getData, 143
 ownDrag, 144
 type, 145

AuthenticEvent, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164

altKey, 147
 altLeft, 148
 button, 149
 cancelBubble, 150
 clientX, 151
 clientY, 152

AuthenticEvent, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164

ctrlKey, 153
 ctrlLeft, 154
 dataTransfer, 155
 fromElement, 156
 keyCode, 157
 propertyName, 158
 repeat, 159
 returnValue, 160
 shiftKey, 161
 shiftLeft, 162
 srcElement, 163
 type, 164

Authentic-Objekt, 24**AuthenticRange, 168, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219**

AppendRow, 171
 Application, 172
 CanPerformAction, 173
 CanPerformActionWith, 174
 Close, 175
 CollapsToBegin, 176
 CollapsToEnd, 177
 Copy, 178
 Cut, 179
 Delete, 180
 DeleteRow, 181
 DuplicateRow, 182
 ExpandTo, 183
 FirstTextPosition, 184
 FirstXMLData, 185
 FirstXMLDataOffset, 186
 GetElementAttributeNames, 188
 GetElementAttributeValue, 189
 GetElementHierarchy, 190
 GetEntityNames, 191
 Goto, 192
 GotoNext, 193
 GotoNextCursorPosition, 194
 GotoPrevious, 195
 GotoPreviousCursorPosition, 196
 HasElementAttribute, 197
 InsertEntity, 198
 InsertRow, 199
 IsEmpty, 200

IsEqual, 201
 IsInDynamicTable, 202
 LastTextPosition, 203
 LastXMLData, 204
 LastXMLDataOffset, 205
 MoveBegin, 207
 MoveEnd, 208
 MoveRowDown, 210
 MoveRowUp, 209
 Parent, 211
 Paste, 212
 PerformAction, 213
 Select, 214
 SelectNext, 215
 SelectPrevious, 216
 SetElementAttributeValue, 217
 SetFromRange, 218
 Text, 219

AuthenticView, 48, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254

Application, 243
 DocumentBegin, 244
 DocumentEnd, 245
 Goto, 246
 MarkupVisibility, 247
 Parent, 248
 Print, 249
 Redo, 250
 Selection, 251
 Undo, 252
 WholeDocument, 254

AuthenticView-Objekt, 24**B****Bearbeitungsoperationen, 25****Browser, 1**

Document Editor plug-in, 1

C**CAB-Dateien, 4****Classid, 9**

listing of for various versions, 9

CLSID, 9

listing of for various versions, 9

Connection Point-Ereignisse, 21**controlinitialized, 21, 53****D****DB-based SPS, 18**

HTML page to reference, 18

DB-basiertes SPS, 5

Voraussetzungen für die Authentic 2005 Browser Edition, 5

Document Editor, 1

browser plug-in, 1

Dokumentinhalt, 24

aufrufen und ändern, 24

DOM, 37

und XMLData, 37

Dynamische Tabellen, 27**E****Eingabehilfen, 30****Einstellungen, 5**

für die Verwendung der Authentic 2005 Browser Edition in Verbindung mit DB-basierten SPS, 5

Ereignishandler, 21, 23**Ersetzen, 26****Event handling, 12**

in Authentic Browser Plug-in, 12

H**HTML page, 18**

for DB-based SPS, 18

sample for Authentic Browser Plug-in, 13, 15

HTML-Seite, 8

für das Authentic Browser Plug-in erstellen, 8

I**ID, 9**

of Object element, 9

Inhalt, 24

aufrufen und ändern, 24

L**Licensing, 9**

obsolete parameters and objects, 9

N**Non-Unicode, 9**

versions of Authentic Browser Plug-in, 9

O**Object element, 9**

description of, 9

P**Pakete, 31****Parameters, 9**

for Object element, 9

Plug-in, 1

Document Editor Browser, 1

R**Rechtschreibprüfung, 31**

S

Script element, 12

description of, 12

Scripting languages, 12

selectionchanged, 21

Subroutines, 12

in Authentic Browser Plug-in, 12

Suchen, 26

Suchen und ersetzen, 26

Symbolleisten-Schaltflächen, 23

Verhalten ändern, 23

T

Tastaturkürzel, 28

Textstatus-Schaltflächen, 29

Trusted, 9

versions of Authentic Browser Plug-in, 9

U

Unicode, 9

versions of Authentic Browser Plug-in, 9

Untrusted, 9

versions of Authentic Browser Plug-in, 9

V

Version, 9

specifying for use, 9

X

XMLData, 33, 304

IsSameNode, 304

und DOM, 37

XMLSpyLib, 141, 146, 295

AuthenticDataTransfer, 141

AuthenticEvent, 146

XMLSpyXMLData, 295

XMLSPYPLUGINLib, 165

Authentic, 42

XMLSpyXMLLoadSave, 165

XMLSpyXMLData, 296, 297, 298, 299, 300, 301, 302, 303, 305, 306, 307, 308, 309

AppendChild, 296

EraseAllChildren, 297

EraseCurrentChild, 298

GetCurrentChild, 299

GetFirstChild, 300

GetNextChild, 301

HasChildren, 302

InsertChild, 303

Kind, 305

MayHaveChildren, 306

Name, 307

Parent, 308

TextValue, 309

XMLSpyXMLLoadSave, 166, 167

String, 166

URL, 167

Z

Zeilenoperationen, 27