

Benutzer- und Referenzhandbuch



Copyright © 1998–2009, Altova GmbH. All rights reserved. Use of this software is governed by and subject to an Altova software license agreement. XMLSpy, MapForce, StyleVision, SemanticWorks, SchemaAgent, UModel, DatabaseSpy, DiffDog, Authentic, AltovaXML, MissionKit, and ALTOVA as well as their logos are trademarks and/or registered trademarks 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 3rd party copyrighted software or material that is protected by copyright and subject to other terms and conditions as detailed on the Altova website at http://www.altova.com/logo_3rdparty.html



Altova Authentic 2009 Desktop Edition Benutzer- und Referenzhandbuch

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

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

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

Published: 2009

© 2009 Altova GmbH

Inhaltsverzeichnis

1	Willkommen bei Authentic Desktop	3
2	Benutzerhandbuch	6
2.1	Einführung	7
2.1.1	Die grafische Benutzeroberfläche	8
	– Hauptfenster	10
	– Projektfenster	12
	– Info-Fenster	14
	– Eingabehilfen	15
	– Fenster "Meldungen"	16
	– Menüleiste, Symbolleisten und Statusleiste	17
2.1.2	Die Applikationsumgebung	19
	– Einstellungen und Anpassung	20
	– Tutorials, Projekte, Beispiele	21
	– Authentic Desktop Features and Help, and Altova Products	22
2.2	Tutorial Authentic-Ansicht	23
2.2.1	Öffnen eines XML-Dokuments in der Authentic-Ansicht	25
2.2.2	Die Benutzeroberfläche der Authentic-Ansicht	27
2.2.3	Node-Operationen	30
2.2.4	Eingabe von Daten in der Authentic-Ansicht	33
2.2.5	Eingabe von Attributwerten	36
2.2.6	Hinzufügen von Entities	37
2.2.7	Drucken des Dokuments	38
2.3	Authentic-Ansicht	39
2.3.1	Übersicht über die Benutzeroberfläche	40
2.3.2	Authentic-Ansicht Symbolleistenschaltflächen	42
2.3.3	Authentic-Ansicht Hauptfenster	45
2.3.4	Authentic-Ansicht Eingabehilfen	48
2.3.5	Authentic-Ansicht Kontextmenüs	52
2.4	Bearbeitung in der Authentic-Ansicht	54
2.4.1	Grundlegendes zur Bearbeitung	55
2.4.2	Tabellen in der Authentic-Ansicht	58
	– SPS-Tabellen	59
	– XML-Tabellen	61
2.4.3	Bearbeiten einer DB	65

	– Navigieren in einer DB-Tabelle.....	66
	– DB-Abfragen.....	67
	– Ändern einer DB-Tabelle.....	72
2.4.4	Symbole zum Bearbeiten von XML-Tabellen	74
2.4.5	Arbeiten mit Datumsangaben	76
	– Datumswahl	77
	– Texteintrag	78
2.4.6	Definieren von Entities	79
2.4.7	Grafiken in der Authentic-Ansicht	81
2.4.8	Verwendung von Tasten in der Authentic-Ansicht	82
2.5	Browser-Ansicht	83
2.6	Globale Altova-Ressourcen	84
2.6.1	Definieren globaler Ressourcen	85
	– Dateien	87
	– Ordner	90
	– Datenbanken	91
	– Kopieren von Konfigurationen.....	93
2.6.2	Verwendung globaler Ressourcen	94
	– Zuweisen von Dateien und Ordnern.....	95
	– Wechseln von Konfigurationen.....	98
2.7	Authentic Desktop in Visual Studio	99
2.7.1	Installieren des Authentic Plugin	100
2.7.2	Unterschiede zur Standalone-Version	101
2.8	Authentic Desktop in Eclipse	102
2.8.1	Installation des Authentic Desktop Plugin für Eclipse	103
2.8.2	Einstiegspunkte für Authentic in Eclipse	107
2.9	Benutzerreferenz	111
2.9.1	Menü "Datei"	112
	– Neu	113
	– Öffnen	114
	– Neu laden	117
	– Kodierung	118
	– Schließen, Alle schließen.....	119
	– Speichern, Speichern unter, Alles speichern.....	120
	– Als Mail senden.....	124
	– Drucken	125
	– Druckereinrichtung.....	126
	– Zuletzt verwendete Dateien, Beenden.....	127
2.9.2	Menü "Bearbeiten"	128
	– Rückgängig, Wiederherstellen.....	129
	– Ausschneiden, Kopieren, Einfügen, Löschen.....	130
	– Alles markieren.....	131
	– Suchen, Weitersuchen.....	132

	– Ersetzen...	133
2.9.3	Menü "Projekt"	134
	– Neues Projekt	136
	– Projekt öffnen	137
	– Projekt neu laden	138
	– Projekt schließen	139
	– Projekt speichern	140
	– Versionskontrolle	141
	Installationsverfahren für Versionskontrollsysteme	144
	Aus Versionskontrolle öffnen	152
	Versionskontrolle aktivieren	153
	Aktuellste Version holen	154
	Abrufen	154
	Ordner abrufen	155
	Auschecken	156
	Einchecken	156
	Auschecken rückgängig	157
	Zu Versionskontrolle hinzufügen	158
	Von Versionskontrolle ausgliedern	159
	Aus Versionskontrolle freigeben	159
	Verlauf anzeigen	160
	Unterschiede anzeigen	161
	Eigenschaften anzeigen	163
	Status aktualisieren	163
	Versionskontrollmanager	163
	– Dateien zu Projekt hinzufügen	164
	– Globale Ressource zu Projekt hinzufügen	165
	– URL zu Projekt hinzufügen	166
	– Aktive Datei zu Projekt hinzufügen	167
	– Aktive und verwandte Dateien zu Projekt hinzufügen	168
	– Projektordner zu Projekt hinzufügen	169
	– Externes Verzeichnis zum Projekt hinzufügen	170
	– Externes Webverzeichnis zum Projekt hinzufügen	173
	– Projekteigenschaften	176
	– Zuletzt verwendete Projekte	178
2.9.4	Menü "XML"	179
	– Einfügen	180
	XInclude einfügen	180
	– Anhängen	183
	XInclude anhängen	183
	– Child hinzufügen	186
	Child XInclude anhängen	186
	– Wohlgeformtheit prüfen	189
	– Validieren	190
	– Eingabehilfen aktualisieren	192

2.9.5	Menü "XSL/XQuery"	193
	– XSL-Transformation	194
	– XSL:FO-Transformation	195
	– XSL-Parameter/XQuery-Variablen	197
2.9.6	Menü "Authentic"	201
	– Neues Dokument	202
	– Datenbankdaten bearbeiten	203
	– StyleVision Stylesheet zuweisen	204
	– Neue Zeile mit XML-Daten für Bearbeitung auswählen	205
	– XML Entities definieren	206
	– Markup-Symbole ausblenden, Kleine/Große/Gemischte Markup-Symbole einblenden	208
	– Zeile anhängen/einfügen/kopieren/löschen	209
	– Zeile nach oben/unten	210
2.9.7	Menü "Ansicht"	211
	– WSDL-Design-Ansicht	212
	– XBRL-Taxonomieansicht	213
	– Authentic-Ansicht	214
	– Browser-Ansicht	215
2.9.8	Menü "Browser"	216
	– Zurück	217
	– Vorwärts	218
	– Abbrechen	219
	– Aktualisieren	220
	– Schriftgröße	221
	– Neues Fenster	222
2.9.9	Menü "Extras"	223
	– Rechtschreibung	224
	– Rechtschreibung - Optionen	226
	– Skripting-Umgebung	230
	– Makros	231
	– Projekt	232
	– Globale Ressourcen	233
	– Aktive Konfiguration	234
	– Anpassen	235
	– Befehle	235
	– Symbolleisten	236
	– Tastatur	237
	– Menü	242
	– Makros	244
	– Plug-Ins	245
	– Optionen	246
	– Kontextmenü 'Anpassen'	247
	– Optionen	250
	– Datei	250

	Dateiarten.....	251
	Ansicht.....	252
	XBRL-Schriftarten.....	252
	Kodierung.....	253
	XSL.....	254
	Skripting-Umgebung.....	255
	Versionskontrolle.....	256
2.9.10	Menü "Fenster"	257
	– Überlappend.....	258
	– Horizontal anordnen.....	259
	– Vertikal anordnen.....	260
	– Projekt-Fenster.....	261
	– Info-Fenster.....	262
	– Eingabehilfen.....	263
	– Ausgabefenster.....	264
	– Projekt- und Eingabehilfen.....	265
	– Alle ein/aus	266
	– Liste aller offenen Fenster.....	267
2.9.11	Menü "Hilfe"	268
	– Inhaltsverzeichnis.....	269
	– Index	270
	– Suche	271
	– Tastaturzuordnungen.....	272
	– Aktivierung, Bestellformular, Registrieren, Updates.....	273
	– Support Center, Fragen und Antworten, Komponenten downloaden.....	275
	– Internet	276
	– Über	277

3 Referenz für Programmierer 280

3.1	Anmerkungen zur Release	281
3.2	Skripterstellung	283
3.2.1	Übersicht über die Verwendung	285
	– Skripting-Projekte und das Projektfenster.....	286
	– Bearbeitung im Hauptfenster.....	289
	– Formulare, Event Handler und Makros.....	291
	– Skripting -Einstellungen.....	293
3.2.2	Formulare	294
	– Editor-Ansichten.....	295
	– Formulardesign.....	298
	– Objekteigenschaften und Events.....	300
	– Objekt: TheView.....	303
	– Beispiel: Erstellen eines Formulars.....	304
	– Beispiel: Verwenden von Farben.....	309

3.2.3	Application Event Handler	310
3.2.4	Makros	313
	– Erstellen eines neuen Makros.....	314
	– Erstellen von Funktionen für das Makro.....	315
	– Ausführen eines Makros.....	317
	– Ändern eines Makros.....	319
3.2.5	Menüs	320
	– File	321
	– Edit	322
	– View	323
	– Project	324
	– Layout	325
	– Draw	328
	– Window	329
3.2.6	Anmerkungen zur Programmierung	330
3.3	Plugins	331
3.3.1	Registrierung von IDE Plug-Ins	332
3.3.2	ActiveX Controls	333
3.3.3	Konfigurations-XML	334
3.3.4	ATL-Beispieldateien	337
	– Schnittstellenbeschreibung (IDL).....	338
	– Klassendefinition.....	340
	– Implementierung.....	341
3.3.5	IXMLSpyPlugIn	343
	– OnCommand.....	344
	– OnUpdateCommand.....	345
	– OnEvent	346
	– GetUIModifications.....	348
	– GetDescription.....	349
3.4	Die XMLSpy API	350
3.4.1	Übersicht	351
	– Objektmodell.....	352
	– Einfacher Dokumentzugriff.....	354
	– Fehlerbehandlung.....	356
	– Events	358
	– Import und Export von Daten.....	362
	– Ändern der Dokumentstruktur mit Hilfe von XMLData.....	365
	– DOM und XMLData.....	369
	– Nicht mehr verwendete Zeilenoperationen in der Authentic-Ansicht.....	372
	– Nicht mehr verwendete Bearbeitungsoptionen in der Authentic-Ansicht.....	373
3.4.2	Beispiele für die Verwendung	374
	– JScript: Bubble-Sortierung dynamischer Tabellen.....	375

– VBScript: Verwendung von Events auf Objektebene.....	377
3.4.3 Schnittstellen	379
– Application	380
Events.....	381
<i>OnBeforeOpenDocument</i>	381
<i>OnBeforeOpenProject</i>	381
<i>OnDocumentOpened</i>	382
<i>OnProjectOpened</i>	382
ActiveDocument.....	383
AddMacroMenuItem.....	383
Application.....	383
ClearMacroMenu	384
CurrentProject.....	384
Dialogs.....	384
Documents.....	384
FindInFiles	385
GetDatabaseImportElementList.....	385
GetDatabaseSettings.....	386
GetDatabaseTables.....	386
GetExportSettings	387
GetTextImportElementList.....	387
GetTextImportExportSettings.....	388
ImportFromDatabase.....	388
ImportFromSchema.....	389
ImportFromText.....	390
ImportFromWord	390
NewProject.....	391
OpenProject.....	391
Parent	392
Quit	392
ReloadSettings.....	392
RunMacro.....	393
ScriptingEnvironment.....	393
ShowApplication.....	393
ShowFindInFiles	393
ShowForm.....	394
URLDelete	394
URLMakeDirectory.....	395
Visible.....	395
WarningNumber.....	395
WarningText	395
– AuthenticDataTransfer	397
dropEffect.....	397
getData.....	398
ownDrag.....	398
type	398

– AuthenticRange.....	399
AppendRow.....	400
Application.....	401
CanPerformAction.....	401
CanPerformActionWith.....	402
Clone.....	402
CollapsToBegin.....	402
CollapsToEnd.....	403
Copy.....	403
Cut	403
Delete.....	404
DeleteRow.....	404
DuplicateRow.....	404
ExpandTo.....	405
FirstTextPosition.....	405
FirstXMLData.....	406
FirstXMLDataOffset.....	407
GetElementAttributeNames.....	408
GetElementAttributeValue.....	408
GetElementHierarchy.....	409
GetEntityNames	409
Goto.....	410
GotoNext.....	410
GotoNextCursorPosition.....	411
GotoPrevious.....	411
GotoPreviousCursorPosition.....	412
HasElementAttribute.....	412
InsertEntity.....	412
InsertRow	413
IsCopyEnabled.....	413
IsCutEnabled.....	414
IsDeleteEnabled.....	414
IsEmpty	414
IsEqual.....	415
IsFirstRow.....	415
IsInDynamicTable.....	415
IsLastRow.....	415
IsPasteEnabled.....	416
IsTextStateApplied.....	416
LastTextPosition	416
LastXMLData.....	417
LastXMLDataOffset.....	418
MoveBegin.....	419
MoveEnd.....	419
MoveRowDown	420

MoveRowUp.....	420
Parent.....	420
Paste.....	421
PerformAction.....	421
Select.....	422
SelectNext.....	422
SelectPrevious.....	423
SetElementAttributeValue.....	424
SetFromRange.....	425
Text.....	425
– AuthenticView.....	426
Events.....	427
<i>OnBeforeCopy</i>	427
<i>OnBeforeCut</i>	427
<i>OnBeforeDelete</i>	428
<i>OnBeforeDrop</i>	428
<i>OnBeforePaste</i>	429
<i>OnDragOver</i>	430
<i>OnKeyboardEvent</i>	431
<i>OnMouseEvent</i>	432
<i>OnSelectionChanged</i>	433
Application.....	434
AsXMLString.....	434
DocumentBegin.....	435
DocumentEnd.....	435
Event.....	435
Goto.....	436
IsRedoEnabled.....	436
IsUndoEnabled.....	437
MarkupVisibility.....	437
Parent.....	437
Print.....	437
Redo.....	438
Selection.....	438
Undo.....	438
UpdateXMLInstanceEntities.....	439
WholeDocument.....	439
XMLDataRoot.....	440
– CodeGeneratorDlg.....	441
Application.....	441
CompatibilityMode.....	442
CPPSettings_DOMType.....	442
CPPSettings_GenerateVC6ProjectFile.....	442
CPPSettings_GenerateVSProjectFile.....	443
CPPSettings_LibraryType.....	443
CPPSettings_UseMFC.....	443
CSharpSettings_ProjectType.....	443

OutputPath.....	444
OutputPathDialogAction.....	444
OutputResultDialogAction.....	444
Parent.....	445
ProgrammingLanguage.....	445
PropertySheetDialogAction.....	445
TemplateFileName.....	445
– DatabaseConnection.....	447
ADOConnection.....	447
AsAttributes.....	448
CommentIncluded.....	448
CreateMissingTables.....	448
CreateNew.....	448
DatabaseKind.....	449
DatabaseSchema.....	449
ExcludeKeys.....	449
File.....	449
IncludeEmptyElements.....	450
NullReplacement.....	450
NumberDateTimeFormat.....	450
ODBCConnection.....	451
SQLSelect.....	451
TextFieldLen.....	451
– Dialogs.....	452
Application.....	452
CodeGeneratorDlg.....	452
FileSelectionDlg.....	453
Parent.....	453
SchemaDocumentationDlg.....	453
GenerateSampleXMLDlg.....	453
DTDSchemaGeneratorDlg.....	454
FindInFilesDlg.....	454
WSDLDocumentationDlg.....	454
– Document.....	456
Events.....	457
OnBeforeSaveDocument.....	457
OnBeforeCloseDocument.....	458
OnBeforeValidate.....	459
OnCloseDocument.....	459
OnViewActivation.....	460
Application.....	460
AssignDTD.....	460
AssignSchema.....	461
AssignXSL.....	461
AssignXSLFO.....	461
AsXMLString.....	462

AuthenticView.....	462
Close.....	463
ConvertDTDOOrSchema.....	463
ConvertDTDOOrSchemaEx.....	463
CreateChild.....	464
CreateSchemaDiagram.....	464
CurrentViewMode.....	465
DataRoot.....	465
DocEditView.....	465
Encoding.....	465
EndChanges.....	466
ExecuteXQuery.....	466
ExportToDatabase.....	467
ExportToText.....	467
FullName.....	468
GenerateDTDOOrSchema.....	469
GenerateDTDOOrSchemaEx.....	469
GenerateProgramCode.....	470
GenerateSampleXML.....	470
GenerateSchemaDocumentation.....	470
GenerateWSDLDocumentation.....	471
GetExportElementList.....	471
GetPathName (nicht mehr verwendet).....	472
GridView.....	472
IsModified.....	472
IsValid.....	472
IsWellFormed.....	474
Name.....	474
Parent.....	475
Path.....	475
RootElement.....	475
Save.....	475
SaveAs.....	476
Saved.....	476
SaveInString.....	476
SaveToURL.....	477
SetActiveDocument.....	477
Document.SetEncoding (nicht mehr verwendet).....	477
SetExternalIsValid.....	479
Document.SetPathName (nicht mehr verwendet).....	479
StartChanges.....	479
Suggestions.....	480
SwitchViewMode.....	480
TextView.....	480
Title.....	480

TransformXSL	481
TransformXSLEx	481
TransformXSLFO	481
UpdateViews	482
UpdateXMLData	482
– Documents	483
Count	483
Item	483
NewFile	484
NewFileFromText	484
OpenFile	484
OpenURL	485
OpenURLDialog	485
– DTDSchemaGeneratorDlg	487
Application	487
AttributeTypeDefinition	487
DTDSchemaFormat	488
FrequentElements	488
GlobalAttributes	488
MaxEnumLength	488
MergeAllEqualNamed	489
OnlyStringEnums	489
OutputPath	489
OutputPathDialogAction	489
Parent	490
ResolveEntities	490
TypeDetection	490
ValueList	490
– ElementList	491
Count	491
Item	491
RemoveElement	491
– ElementListItem	493
ElementKind	493
FieldCount	493
Name	493
RecordCount	493
– ExportSettings	495
CreateKeys	495
ElementList	495
EntitiesToText	496
ExportAllElements	496
ExportCompleteXML	496
FromAttributes	496
FromSingleSubElements	496

FromTextValues.....	497
IndependentPrimaryKey.....	497
Namespace.....	497
StartFromElement.....	497
SubLevelLimit.....	497
– FileSelectionDlg.....	498
Application.....	498
DialogAction.....	498
FullName.....	499
Parent.....	499
– FindInFilesDlg.....	500
AdvancedXMLSearch.....	500
Application.....	500
DoReplace.....	501
FileExtension.....	501
Find.....	501
IncludeSubfolders.....	501
MatchCase.....	502
MatchWholeWord.....	502
Parent.....	502
RegularExpression.....	502
Replace.....	502
ReplaceOnDisk.....	503
SearchInProjectFilesDoExternal.....	503
SearchLocation.....	503
ShowResult.....	503
StartFolder.....	503
XMLAttributeContents.....	504
XMLAttributeNames.....	504
XMLCDATA.....	504
XMLComments.....	504
XMLElementContents.....	505
XMLElementNames.....	505
XMLPI.....	505
XMLRest.....	505
– FindInFilesResult.....	506
Application.....	506
Count.....	506
Document.....	506
Item.....	507
Parent.....	507
Path.....	507
– FindInFilesResultMatch.....	508
Application.....	508
Length.....	508

Line.....	508
LineText.....	509
Parent.....	509
Position.....	509
Replaced.....	509
– FindInFilesResults.....	510
Application.....	510
Count.....	510
Item.....	510
Parent.....	510
– GenerateSampleXMLDlg.....	511
Application.....	511
Parent.....	511
NonMandatoryAttributes.....	512
NonMandatoryElements - obsolete.....	512
TakeFirstChoice - obsolete.....	512
RepeatCount.....	512
FillWithSampleData - obsolete.....	512
FillElementsWithSampleData.....	513
FillAttributesWithSampleData.....	513
ContentOfNillableElementsIsNonMandatory.....	513
TryToUseNonAbstractTypes.....	513
Optimization.....	513
SchemaOrDTDAssignment.....	514
LocalNameOfRootElement.....	514
NamespaceURIOfRootElement.....	514
OptionsDialogAction.....	514
– GridView.....	515
Events.....	515
OnBeforeDrag.....	515
OnBeforeDrop.....	515
OnBeforeStartEditing.....	516
OnEditingFinished.....	516
OnFocusChanged.....	517
CurrentFocus.....	517
Deselect.....	518
IsVisible.....	518
Select.....	518
SetFocus.....	518
– SchemaDocumentationDlg.....	519
AllDetails.....	520
Application.....	520
DiagramFormat.....	520
EmbedDiagrams.....	521
IncludeAll.....	521
IncludeAttributeGroups.....	521

IncludeComplexTypes.....	521
IncludeGlobalAttributes.....	522
IncludeGlobalElements.....	522
IncludeGroups.....	522
IncludeIndex.....	523
IncludeLocalAttributes.....	523
IncludeLocalElements.....	523
IncludeRedefines.....	524
IncludeReferencedSchemas.....	524
IncludeSimpleTypes.....	524
MultipleOutputFiles.....	524
OptionsDialogAction.....	525
OutputFile.....	525
OutputFileDialogAction.....	525
OutputFormat.....	526
Parent.....	526
ShowAnnotations.....	526
ShowAttributes.....	527
ShowChildren.....	527
ShowDiagram.....	527
ShowEnumerations.....	528
ShowIdentityConstraints.....	528
ShowNamespace.....	528
ShowPatterns.....	529
ShowProgressBar.....	529
ShowProperties.....	529
ShowResult.....	529
ShowSingleFacets.....	530
ShowSourceCode.....	530
ShowType.....	530
ShowUsedBy.....	531
– SpyProject.....	532
CloseProject.....	532
ProjectFile.....	532
RootItems.....	532
SaveProject.....	533
SaveProjectAs.....	533
– SpyProjectItem.....	534
ChildItems.....	534
FileExtensions.....	534
ItemType.....	534
Name.....	535
Open.....	535
ParentItem.....	535
Path.....	535

ValidateWith.....	535
XMLForXSLTransformation.....	536
XSLForXMLTransformation.....	536
XSLTransformationFileExtension.....	536
XSLTransformationFolder.....	536
– SpyProjectItems.....	537
AddFile.....	537
AddFolder.....	537
AddURL.....	537
Count.....	538
Item.....	538
RemoveItem.....	538
– TextImportExportSettings.....	539
CommentIncluded.....	539
DestinationFolder.....	539
EnclosingCharacter.....	539
Encoding.....	540
EncodingByteOrder.....	540
FieldDelimiter.....	540
FileExtension.....	540
HeaderRow.....	540
ImportFile.....	541
RemoveDelimiter.....	541
RemoveNewline.....	541
– TextView.....	542
Events.....	542
OnBeforeShowSuggestions.....	542
OnChar.....	543
Application.....	543
GetRangeText.....	544
GoToLineChar.....	544
Length.....	544
LineCount.....	544
LineFromPosition.....	544
LineLength.....	545
MoveCaret.....	545
Parent.....	545
PositionFromLine.....	545
ReplaceText.....	545
SelectionEnd.....	546
SelectionStart.....	546
SelectText.....	546
SelText.....	546
Text.....	546
– WSDLDocumentationDlg.....	547

AllDetails	548
Application.....	548
DiagramFormat	548
EmbedDiagrams.....	548
IncludeAll.....	549
IncludeBinding.....	549
IncludeImportedWSDLFiles	549
IncludeMessages	550
IncludeOverview	550
IncludePortType.....	550
IncludeService.....	550
IncludeTypes.....	551
MultipleOutputFiles	551
OptionsDialogAction.....	551
OutputFile	552
OutputFileDialogAction.....	552
OutputFormat	552
Parent	553
ShowBindingDiagram	553
ShowExtensibility.....	553
ShowMessageParts.....	553
ShowPort.....	554
ShowPortTypeDiagram.....	554
ShowPortTypeOperations	554
ShowProgressBar	555
ShowResult.....	555
ShowServiceDiagram	555
ShowSourceCode	556
ShowTypesDiagram	556
ShowUsedBy.....	556
– XMLData	557
AppendChild.....	558
CountChildren.....	558
CountChildrenKind.....	558
EraseAllChildren.....	559
EraseCurrentChild.....	559
GetChild.....	560
GetChildKind.....	560
GetCurrentChild.....	561
GetFirstChild.....	561
GetNextChild	561
HasChildrenKind.....	562
HasChildren.....	563
InsertChild.....	563
IsSameNode	564

	Kind.....	564
	MayHaveChildren.....	564
	Name.....	564
	Parent.....	565
	TextValue.....	565
3.4.4	Schnittstellen (nicht mehr verwendet)	566
	– AuthenticEvent (obsolete).....	567
	altKey (nicht mehr verwendet).....	568
	altLeft (nicht mehr verwendet).....	569
	button (nicht mehr verwendet).....	570
	cancelBubble (nicht mehr verwendet).....	571
	clientX (nicht mehr verwendet).....	572
	clientY (nicht mehr verwendet).....	573
	ctrlKey (nicht mehr verwendet).....	574
	ctrlLeft (nicht mehr verwendet).....	575
	dataTransfer (nicht mehr verwendet).....	576
	fromElement (nicht mehr verwendet).....	577
	keyCode (nicht mehr verwendet).....	577
	propertyName (nicht mehr verwendet).....	578
	repeat (nicht mehr verwendet).....	578
	returnValue (nicht mehr verwendet).....	578
	shiftKey (nicht mehr verwendet).....	579
	shiftLeft (nicht mehr verwendet).....	580
	srcElement (nicht mehr verwendet).....	581
	type (nicht mehr verwendet).....	582
	– AuthenticSelection (nicht mehr verwendet).....	584
	End (nicht mehr verwendet).....	585
	EndTextPosition (nicht mehr verwendet).....	585
	Start (nicht mehr verwendet).....	586
	StartTextPosition (nicht mehr verwendet).....	586
	– OldAuthenticView (nicht mehr verwendet).....	588
	ApplyTextState (nicht mehr verwendet).....	590
	CurrentSelection (nicht mehr verwendet).....	591
	EditClear (nicht mehr verwendet).....	591
	EditCopy(nicht mehr verwendet).....	592
	EditCut (nicht mehr verwendet).....	592
	EditPaste (nicht mehr verwendet).....	593
	EditRedo (nicht mehr verwendet).....	593
	EditSelectAll (nicht mehr verwendet).....	594
	EditUndo (nicht mehr verwendet).....	594
	event (nicht mehr verwendet).....	595
	GetAllowedElements (nicht mehr verwendet).....	595
	GetNextVisible (nicht mehr verwendet).....	598
	GetPreviousVisible (nicht mehr verwendet).....	599
	IsEditClearEnabled (nicht mehr verwendet).....	600

IsEditCopyEnabled (nicht mehr verwendet).....	600
IsEditCutEnabled (nicht mehr verwendet).....	601
IsEditPasteEnabled (nicht mehr verwendet).....	601
IsEditRedoEnabled (nicht mehr verwendet).....	602
IsEditUndoEnabled (nicht mehr verwendet).....	602
IsRowAppendEnabled (nicht mehr verwendet).....	603
IsRowDeleteEnabled (nicht mehr verwendet).....	603
IsRowDuplicateEnabled (nicht mehr verwendet).....	604
IsRowInsertEnabled (nicht mehr verwendet).....	604
IsRowMoveDownEnabled (nicht mehr verwendet).....	606
IsRowMoveUpEnabled (nicht mehr verwendet).....	606
IsTextStateApplied (nicht mehr verwendet).....	607
IsTextStateEnabled (nicht mehr verwendet).....	607
LoadXML (nicht mehr verwendet).....	608
MarkUpView (nicht mehr verwendet).....	608
RowAppend (nicht mehr verwendet).....	610
RowDelete (nicht mehr verwendet).....	610
RowDuplicate (nicht mehr verwendet).....	611
RowInsert (nicht mehr verwendet).....	611
RowMoveDown (nicht mehr verwendet).....	612
RowMoveUp (nicht mehr verwendet).....	612
SaveXML (nicht mehr verwendet).....	613
SelectionMoveTabOrder (nicht mehr verwendet).....	614
SelectionSet (nicht mehr verwendet).....	615
XMLRoot (nicht mehr verwendet).....	616
3.4.5 Enumerationen	618
– SPYAttributeTypeDefinition.....	619
– SPYAuthenticActions.....	620
– SPYAuthenticDocumentPosition.....	621
– SPYAuthenticElementActions.....	622
– SPYAuthenticElementKind.....	623
– SPYAuthenticMarkupVisibility	624
– SPYDatabaseKind.....	625
– SPYDialogAction.....	626
– SPYDOMType.....	627
– SPYDTDSchemaFormat.....	628
– SPYEncodingByteOrder.....	629
– SPYExportNamespace.....	630
– SPYFindInFilesSearchLocation.....	631
– SPYFrequentElements.....	632
– SPYImageKind.....	633
– SPYKeyEvent.....	634
– SPYLibType.....	635
– SPYLoading.....	636
– SPYMouseEvent.....	637

– SPYNumberDateTimeFormat	638
– SPYProgrammingLanguage	639
– SPYProjectItemTypes	640
– SPYProjectType	641
– SPYSampleXMLGenerationOptimization	642
– SPYSampleXMLGenerationSchemaOrDTDAssignment	643
– SPYSchemaDefKind	644
– SPYSchemaDocumentationFormat	645
– SPYTextDelimiters	646
– SPYTextEnclosing	647
– SPYTypeDetection	648
– SPYURLTypes	649
– SPYViewModes	650
– SPYVirtualKeyMask	651
– SPYXMLDataKind	652
3.5 Verwendung der XMLSpy API mit Java	653
3.5.1 Sample source code	655
3.5.2 SpyApplication	657
3.5.3 SpyCodeGeneratorDlg	658
3.5.4 SpyDatabaseConnection	659
3.5.5 SpyDialogs	660
3.5.6 SpyDoc	661
3.5.7 SpyDocuments	663
3.5.8 SpyDTDSchemaGeneratorDlg	664
3.5.9 SpyElementList	665
3.5.10 SpyElementListItem	666
3.5.11 SpyExportSettings	667
3.5.12 SpyFileSelectionDlg	668
3.5.13 SpyFindInFilesDlg	669
3.5.14 SpyFindInFilesMatch	670
3.5.15 SpyFindInFilesResult	671
3.5.16 SpyFindInFilesResults	672
3.5.17 SpyGenerateSampleXMLDlg	673
3.5.18 SpyGridView	674
3.5.19 SpyProject	675
3.5.20 SpyProjectItem	676
3.5.21 SpyProjectItems	677
3.5.22 SpySchemaDocumentationDlg	678
3.5.23 SpyTextImportExportSettings	680
3.5.24 SpyTextView	681
3.5.25 SpyWSDLDocumentationDlg	682
3.5.26 SpyXMLData	684

3.5.27	Authentic	685
	– SpyAuthenticRange.....	685
	– SpyAuthenticView.....	687
	– SpyDocEditSelection.....	688
	– SpyDocEditView.....	689
3.5.28	Vordefinierte Konstanten	690
	– SPYAttributeTypeDefinition.....	691
	– SPYAuthenticActions.....	692
	– SPYAuthenticDocumentPosition.....	693
	– SPYAuthenticElementKind.....	694
	– SPYAuthenticMarkupVisibility.....	695
	– SPYDatabaseKind.....	696
	– SPYDialogAction.....	697
	– SPYDOMType.....	698
	– SPYDTDSchemaFormat.....	699
	– SPYEncodingByteOrder.....	700
	– SPYExportNamespace.....	701
	– SPYFindInFilesSearchLocation.....	702
	– SPYFrequentElements.....	703
	– SPYImageKind.....	704
	– SPYLibType.....	705
	– SPYLoading.....	706
	– SPYNumberDateTimeFormat.....	707
	– SPYProgrammingLanguage.....	708
	– SPYProjectItemTypes.....	709
	– SPYProjectType.....	710
	– SPYSampleXMLGenerationOptimization.....	711
	– SPYSampleXMLGenerationSchemaOrDTDAssignment.....	712
	– SPYSchemaDefKind.....	713
	– SPYSchemaDocumentationFormat.....	714
	– SPYTextDelimiters.....	715
	– SPYTextEnclosing.....	716
	– SPYTypeDetection.....	717
	– SPYURLTypes.....	718
	– SpyViewModes.....	719
	– SPYWhitespaceComparison.....	720
	– SPYXMLDataKind.....	721
3.6	AuthenticDesktop Integration	722
3.6.1	Integration auf Applikationsebene	723
	– Beispiel: HTML.....	724
	Instantiieren des Control	724
	Hinzufügen einer Schaltfläche zum Öffnen eines Standarddokuments.....	724
	Verbindung zu benutzerdefinierten Events herstellen.....	724
3.6.2	Integration auf Dokumentenebene	725

– Verwendung von AuthenticDesktopControl.....	726
– Verwendung von AuthenticDesktopControlDocument.....	727
– Verwendung von AuthenticDesktopControlPlaceHolder.....	728
– Abfragen von AuthenticDesktop Befehlen.....	729
– Beispiele	730
C#	730
<i>Einführung</i>	730
<i>Platzieren des AuthenticDesktopControl</i>	730
HTML.....	730
<i>Instantiieren des AuthenticDesktopControl</i>	731
<i>Erstellen eines Bearbeitungsfensters</i>	731
<i>Erstellen eines Projektfensters</i>	731
<i>Erstellen eines Placeholder für Eingabehilfenfenster</i>	732
3.6.3 Befehlstabelle für Authentic Desktop	733
– Menü "Datei".....	734
– Menü "Bearbeiten".....	735
– Menü "Projekt".....	736
– Menü "XML".....	737
– Menü "XSL/XQuery".....	738
– Menü "Authentic".....	739
– Menü "Ansicht".....	740
– Menü "Browser".....	741
– Menü "Extras".....	742
– Menü "Fenster".....	743
– Menü "Hilfe".....	744
– Diverse Menübefehle	745
3.6.4 Aufrufen von AuthenticDesktopAPI	752
3.6.5 Objektreferenz	753
– AuthenticDesktopCommand.....	754
Accelerator.....	754
ID	754
IsSeparator.....	754
Label.....	755
StatusText.....	755
SubCommands.....	755
ToolTip.....	755
– AuthenticDesktopCommands.....	756
Count.....	756
Item.....	756
– AuthenticDesktopControl.....	757
Eigenschaften	757
<i>Appearance</i>	757
<i>Application</i>	758
<i>BorderStyle</i>	758
<i>CommandsList</i>	758
<i>EnableUserPrompts</i>	758
<i>IntegrationLevel</i>	758

<i>MainMenu</i>	759
<i>Toolbars</i>	759
Methoden.....	759
<i>Exec</i>	759
<i>Open</i>	759
<i>QueryStatus</i>	760
Events.....	760
<i>OnCloseEditingWindow</i>	760
<i>OnContextChanged</i>	761
<i>OnDocumentOpened</i>	761
<i>OnFileChangedAlert</i>	761
<i>OnLicenseProblem</i>	761
<i>OnOpenedOrFocused</i>	761
<i>OnToolWindowUpdated</i>	762
<i>OnUpdateCmdUI</i>	762
<i>OnValidationWindowUpdated</i>	762
– AuthenticDesktopControlDocument.....	763
Eigenschaften.....	763
<i>Appearance</i>	763
<i>BorderStyle</i>	764
<i>Document</i>	764
<i>IsModified</i>	764
<i>Path</i>	764
<i>ReadOnly</i>	764
Methoden.....	765
<i>Exec</i>	765
<i>New</i>	765
<i>Open</i>	765
<i>QueryStatus</i>	765
<i>Reload</i>	766
<i>Save</i>	766
<i>SaveAs</i>	766
Events.....	766
<i>OnActivate</i>	767
<i>OnContextChanged</i>	767
<i>OnDocumentClosed</i>	767
<i>OnDocumentOpened</i>	767
<i>OnDocumentSaveAs</i>	767
<i>OnFileChangedAlert</i>	768
<i>OnModifiedFlagChanged</i>	768
<i>OnSetEditorTitle</i>	768
– AuthenticDesktopControlPlaceHolder.....	769
Eigenschaften.....	769
<i>Label</i>	769
<i>PlaceholderWindowID</i>	769
<i>Project</i>	770
Methoden.....	770
<i>OpenProject</i>	770
<i>CloseProject</i>	770
Events.....	770
<i>OnModifiedFlagChanged</i>	771
<i>OnSetLabel</i>	771

– Enumerationen.....	772
ICActiveXIntegrationLevel.....	772
AuthenticDesktopControlPlaceholderWindow.....	772

4 Anhänge 774

4.1	Technische Daten	775
4.1.1	OS- und Arbeitsspeichieranforderungen	776
4.1.2	Altova XML Parser	777
4.1.3	Altova XSLT- und XQuery-Prozessor	778
4.1.4	Unicode-Unterstützung	779
	– Windows 2000 und Windows XP	780
	– Rechts nach Links-Schreibrichtung.....	781
4.1.5	Internet-Verwendung	782
4.2	Lizenzinformationen	783
4.2.1	Electronic Software Distribution	784
4.2.2	Rechte am geistigen Eigentum	785
4.2.3	Endbenutzer-Lizenzvereinbarung	786

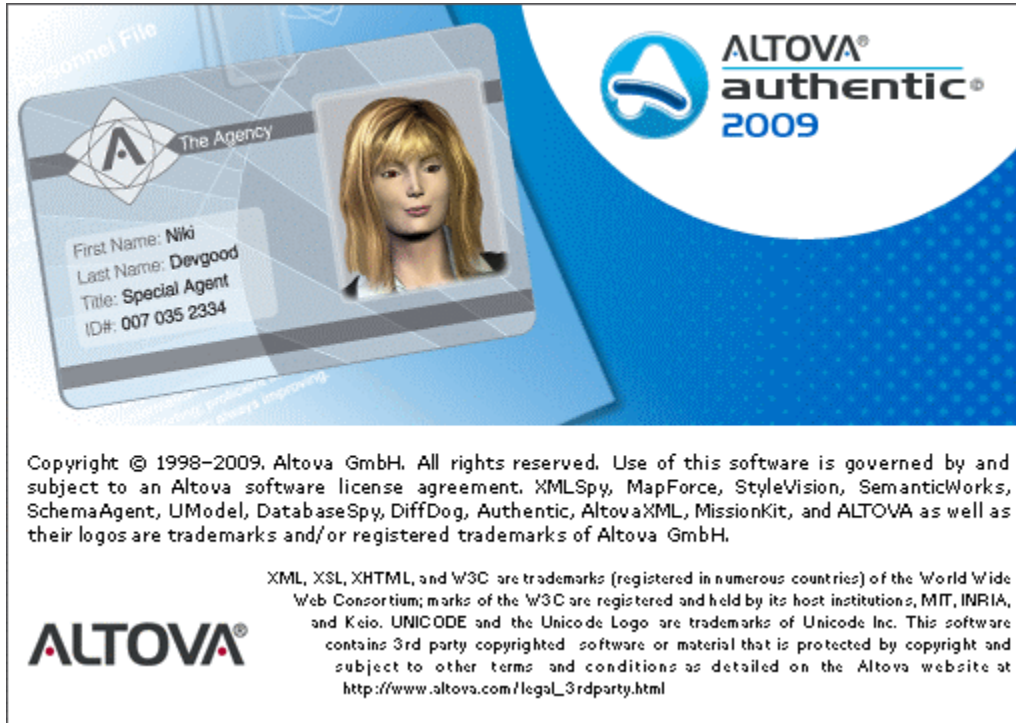
Index

Kapitel 1

Willkommen bei Authentic Desktop

Willkommen bei Authentic Desktop

Die **Altova® Authentic® 2009 Desktop Edition** ist ein innovativer visueller Lösungsansatz zur Erstellung von XML-Dokumenten, bei dem der Endbenutzer sich nicht mit den technischen Aspekten von XML befassen muss.

The splash screen features a blue background with a white circle on the right. On the left, there is a stylized ID card for 'The Agency' with a photo of a woman and the following text: 'First Name: Niki', 'Last Name: Devgood', 'Title: Special Agent', and 'ID#: 007 035 2334'. The Altova Authentic 2009 logo is in the top right. Below the ID card, there is a copyright notice and a disclaimer. The Altova logo is in the bottom left.

Copyright © 1998–2009, Altova GmbH. All rights reserved. Use of this software is governed by and subject to an Altova software license agreement. XMLSpy, MapForce, StyleVision, SemanticWorks, SchemaAgent, UModel, DatabaseSpy, DiffDog, Authentic, AltovaXML, MissionKit, and ALTOVA as well as their logos are trademarks and/or registered trademarks 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 3rd party copyrighted software or material that is protected by copyright and subject to other terms and conditions as detailed on the Altova website at http://www.altova.com/legal_3rdparty.html

Kapitel 2

Benutzerhandbuch

Benutzerhandbuch

Dieses Benutzerhandbuch enthält Tutorials und Erklärungen zu den verschiedenen Features der Authentic-Ansicht, um Ihnen eine Einführung in das Programm zu geben. Außerdem finden Sie darin einen umfassenden Referenzteil, in dem die Funktionen der Benutzeroberfläche beschrieben sind. Das Benutzerhandbuch ist in die folgenden Abschnitte gegliedert:

- eine [Einführung](#), die eine Übersicht über die Benutzeroberfläche und die Authentic Desktop-Umgebung enthält.
- ein [Tutorial](#), um Sie mit Authentic Desktop vertraut zu machen.
- eine Beschreibung der Benutzeroberfläche der [Authentic-Ansicht](#), einer WYSIWYG-Ansicht des XML-Dokuments. Diese Ansicht gestattet Benutzern, XML-Dokumente zu erstellen und zu editieren, als würde es sich um einfache Textdokumente oder interaktive Formulare handeln. Der XML Markup-Code wird dem Benutzer nicht angezeigt, sodass er sich auf den Inhalt des Dokuments konzentrieren kann. Die Authentic-Ansicht ist die Hauptansicht von Authentic Desktop.
- eine [Benutzerreferenz](#) mit Beschreibungen aller Fenster und Menübefehle, die in Authentic Desktop zur Verfügung stehen.

1 Einführung

Diese Einführung enthält eine Beschreibung :

- [der Benutzeroberfläche der Applikation](#) und
- [der Applikationsumgebung](#).

Der [Abschnitt "Benutzeroberfläche"](#) beginnt mit einer Übersicht über die Benutzeroberfläche. Anschließend werden die einzelnen Fenster der Benutzeroberfläche näher beschrieben. Außerdem erfahren Sie, wie Sie die Größe der Fenster anpassen und die Fenster verschieben können und mit den Fenstern und der Benutzeroberfläche arbeiten.

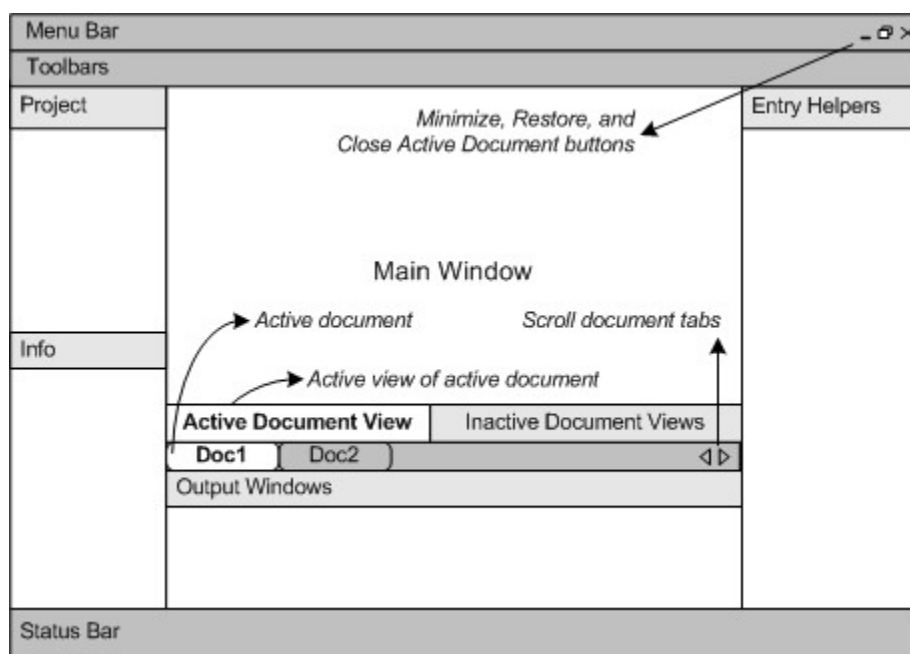
Im [Abschnitt "Applikationsumgebung"](#) werden die verschiedenen Einstellungen erläutert, mit denen festgelegt wird, wie die Dateien angezeigt und bearbeitet werden können. Außerdem wird darin erklärt, wie und wo Sie Ihre Applikation anpassen können. In diesem Abschnitt erfahren Sie, wo wichtige Beispiel- und Tutorial-Dateien auf Ihrem Rechner installiert wurden. Weiter hinten in diesem Abschnitt finden Sie einen Link zur [Altova Website](#), wo Sie eine Feature Matrix Ihrer Applikation finden, Ihr Benutzerhandbuch in verschiedenen Formaten abrufen können und Informationen über die verschiedenen Support-Optionen für Sie finden. Des Weiteren finden Sie dort Informationen über andere Produkte von Altova.

1.1 Die grafische Benutzeroberfläche

Die grafische Benutzeroberfläche besteht aus einem Hauptfenster und mehreren Seitenleisten (siehe *Abbildung unten*). Standardmäßig sind die Seitenleisten rund um das Hauptfenster angeordnet und werden in die folgenden Gruppen eingeteilt:

- Projektfenster
- Info-Fenster
- Eingabehilfen: Elemente, Attribute, Entities, usw. (je nachdem, welche Dokumentart gerade aktiv ist)
- Ausgabefenster: Meldungen

Das Hauptfenster und die Seitenleisten werden in den Unterabschnitten dieses Abschnitts beschrieben.



Außerdem enthält die Benutzeroberfläche eine Menüleiste, eine Statusleiste sowie Symbolleisten, die alle in einem Unterabschnitt dieses Abschnitts beschrieben sind.

Ein- und Ausblenden der Seitenleisten

Die Seitenleistengruppen (Projektfenster, Info-Fenster, Eingabehilfen, Ausgabefenster) können über die entsprechenden Befehle im Menü **Fenster** ein- und ausgeblendet werden. Eine Seitenleiste, die gerade angezeigt wird, kann auch durch Rechtsklick auf die Titelleiste der Seitenleiste (oder die Titelleiste der mit Registern versehenen Seitenleistengruppe) und Auswahl des Befehls **Ausblenden** ausgeblendet werden.

Seitenleisten freischwebend anzeigen und andocken

Eine einzelne Seitenleiste kann entweder frei schwebend oder andockt angezeigt werden. Wenn ein frei schwebendes Fenster andockt wird, wird es an der Stelle andockt, an der es zuletzt andockt war. Ein Fenster kann auch als Register in einem anderen Fenster andockt werden.

Ein Fenster kann auf die folgenden Arten an- oder abgedockt werden:

- Klicken Sie mit der rechten Maustaste auf die Titelleiste eines Fensters und wählen Sie

- den entsprechenden Befehl aus (**Abgedockt** oder **Angedockt**).
- Doppelklicken Sie auf die Titelleiste des Fensters. Wenn das Fenster angedockt war, wird es nun frei schwebend angezeigt. Wenn es frei schwebend angezeigt wurde, wird es nun dort angedockt, wo es zuletzt angedockt war.
 - Ziehen Sie das Fenster (an der Titelleiste als Ziehpunkt) aus seiner angedockten Position bis es frei schwebend angezeigt wird. Ziehen Sie ein frei schwebendes Fenster (an seiner Titelleiste) an die Stelle, an der es angedockt werden soll. Es erscheinen zwei Gruppen blauer Pfeile. Mit den äußeren vier Pfeilen können Sie das Fenster relativ zum Applikationsfenster andocken (am oberen, rechten, unteren oder linken Rand der Benutzeroberfläche). Mit den inneren Pfeilen können Sie das Fenster relativ zu dem Fenster andocken, über dem sich der Cursor gerade befindet. Wenn Sie ein Fenster über den Mittelpunkt der inneren Pfeile ziehen (oder auf die Titelleiste eines Fensters), wird es als Registerkarte in dem Fenster angezeigt, in das es gezogen wurde.

Um ein Fenster, das als Registerkarte angezeigt wird, frei schwebend anzuzeigen, doppelklicken Sie auf sein Register. Um ein als Register angezeigtes Fenster aus einer Gruppe derartiger Fenster herauszuziehen, ziehen Sie es am Register heraus.

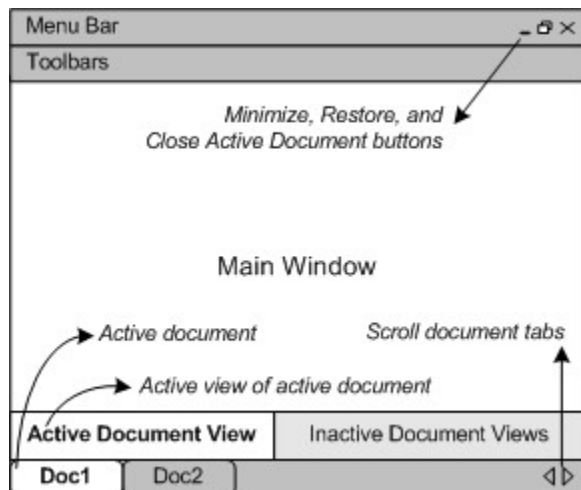
Automatisches Ausblenden der Seitenleisten

Mit der Funktion zum automatischen Ausblenden können Sie angedockte Seitenleisten als Schaltflächen am Rand des Applikationsfensters minimieren, damit Sie mehr Platz für das Hauptfenster und andere Seitenleisten haben. Wenn Sie über eine minimierten Seitenleiste scrollen, wird diese ausgerollt.

Um Seitenleisten automatisch auszublenden und wiederherzustellen, klicken Sie auf das Pin-Symbol in der Titelleiste des Seitenleistenfensters (oder rechtsklicken Sie auf die Titelleiste und wählen Sie den Befehl **Automatisch ausblenden**).

1.1.1 Hauptfenster

Das Hauptfenster (*Abbildung unten*) ist der Bereich, in dem alle Dokumente angezeigt und bearbeitet werden.



Dateien im Hauptfenster

- Sie können eine beliebige Anzahl an Dokumenten gleichzeitig geöffnet haben und bearbeiten.
- Jedes offene Dokument hat sein eigenes Fenster und ein Register mit seinem Namen am unteren Rand des Hauptfensters. Um ein Dokument zum aktiven zu machen, klicken Sie auf sein Register.
- Wenn mehrere Dateien geöffnet sind, werden manchmal auf Grund von Platzmangel in der Dokumentregisterleiste nicht alle Register angezeigt. Um Dokumentregister zu sehen, (i) verwenden Sie entweder die Scroll-Schaltflächen rechts unten in der Dokumentregisterleiste oder (ii) wählen Sie das gewünschte Dokument aus der Liste im unteren Bereich des Menüs [Fenster](#) aus.
- Wenn das aktive Dokument maximiert ist, befinden sich die Schaltflächen zum **Minimieren**, **Wiederherstellen** und **Schließen** auf der rechten Seite der Menüleiste. Wenn ein Dokument überlappend, nebeneinander oder minimiert angezeigt wird, befinden sich die Schaltflächen zum **Maximieren**, **Wiederherstellen** und **Schließen** in der Titelleiste des Dokumentfensters.
- Wenn Sie eine Datei maximieren, werden auch alle anderen offenen Dateien maximiert.
- Um offene Dateien überlappend oder nebeneinander anzuzeigen, verwenden Sie die Befehle im [Menü Fenster](#).
- Sie können offene Dateien auch durch Verwendung von **Strg+Tab** oder **Strg+F6** in der Reihenfolge aktivieren, in der sie geöffnet wurden.
- Durch Rechtsklick auf ein Dokumentregister wird ein Kontextmenü mit einer Auswahl an **Datei**-Befehlen wie z.B. **Drucken** und **Schließen** angezeigt.

Ansichten im Hauptfenster

Das aktive Dokument kann auf verschiedene Arten angezeigt und bearbeitet werden. Die verfügbaren Ansichten werden in einer Leiste oberhalb der Dokumentregister angezeigt (*siehe Abbildung oben*), wobei die aktive Ansicht markiert erscheint. Um eine Ansicht auszuwählen, klicken Sie auf die entsprechende Ansichtsschaltfläche oder rufen Sie im Menü [Ansicht](#) den entsprechenden Befehl auf.

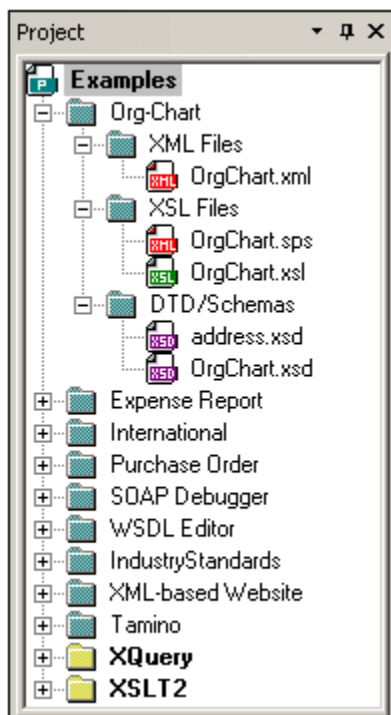
Verfügbar sind entweder Editier- oder Browser-Ansichten:

- [Authentic-Ansicht](#): Zum Bearbeiten von XML-Dokumenten auf Basis von StyleVision Power Stylesheets
- [Browser-Ansicht](#): Eine integrierte Browser-Ansicht, die sowohl CSS- als auch XSL-Stylesheets unterstützt.

Hinweis: Sie können die Standardansicht für bestimmte Dateierweiterungen im Dialogfeld [Extras | Optionen](#) anpassen: Gehen Sie dazu auf dem Register "Dateiarten" zum Fenster "Standardansicht".

1.1.2 Projektfenster

Ein Projekt ist eine Sammlung von Dateien, die miteinander auf eine von Ihnen festgelegte Art in Zusammenhang stehen. In der Abbildung unten enthält z.B. ein Projekt namens `Examples` die Dateien für verschiedene Beispiele in Beispielordnern, von denen jeder weitere Unterordner enthalten kann. Im Projekt `Examples` ist der Beispielordner `OrgChart` in Unterordner für XML-, XSL- und Schema-Dateien gegliedert.



Projekte dienen also als Sammelbehälter für gemeinsam verwendete Dateien, damit diese schneller aufgerufen werden können. Des Weiteren können Sie Schemas und XSLT-Dateien für einzelne Ordner definieren, um die Batch-Verarbeitung für Dateien in einem Ordner zu ermöglichen.

Projektooperationen

Die Befehle für Ordneroperationen stehen im Menü **Projekt** zur Verfügung. Einige Befehle stehen in den Kontextmenüs (Aufruf mittels Rechtsklick) des Projekts und seinen Ordnern zur Verfügung.

- Im Projektfenster ist immer nur ein Projekt geöffnet. Wenn ein neues Projekt erstellt oder ein bestehendes geöffnet wird, wird das Projekt, das gerade im Projektfenster geöffnet ist, durch das neue ersetzt.
- Nachdem an einem Projekt Änderungen vorgenommen wurden, muss das Projekt (durch Auswahl des Befehls **Projekt | Projekt speichern**) gespeichert werden.
- Das Projekt hat eine Baumstruktur bestehend aus Ordnern, Dateien und anderen Ressourcen. Diese können in jeder Ebene und bis zu einer beliebigen Tiefe hinzugefügt werden.
- Projektordner sind "*semantische*" Ordner, die eine logische Dateigruppierung darstellen und **nicht** der hierarchischen Organisation von Dateien auf Ihrer Festplatte entsprechen müssen.
- Ordner können physischen Verzeichnissen in Ihrem Dateisystem entsprechen und eine direkte Beziehung zu diesen haben. Solche Ordner werden *externe Ordner* genannt

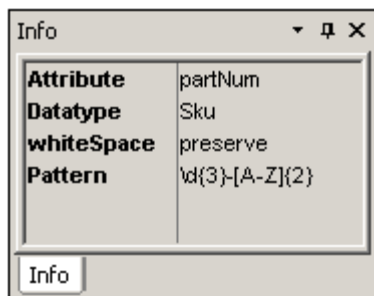
und im Projektfenster durch ein gelbes Ordnersymbol gekennzeichnet (im Gegensatz zu normalen Projektordnern, die grün angezeigt werden). Externe Projektordner müssen explizit mit dem Befehl **Aktualisieren** synchronisiert werden.

- Ein Ordner kann eine beliebige Mischung von Dateiartern enthalten. Alternativ dazu können Sie (im Dialogfeld "Eigenschaften" für diesen Ordner) für jeden Ordner Dateierweiterungen definieren, damit bestimmte Dateiartern jederzeit über einen Ordner aufgerufen werden können. Wenn eine Datei zum übergeordneten Ordner hinzugefügt wird, wird sie automatisch zu dem Unterordner hinzugefügt, für den diese Dateierweiterung definiert wurde.
- Im Projektordner kann ein Ordner in einen anderen Ordner oder an eine andere Stelle im selben Ordner gezogen werden; eine Datei kann in einen anderen Ordner gezogen werden, nicht aber innerhalb desselben Ordners (in dem die Dateien alphabetisch geordnet sind) verschoben werden. Außerdem können Dateien und Ordner aus dem Windows Datei-Explorer in das Projektfenster gezogen werden.
- Jeder Ordner hat bestimmte Eigenschaften, die im Dialogfeld "Eigenschaften" dieses Ordners definiert werden. Dazu gehören die Dateierweiterungen für diesen Ordner, das Schema, mit dem die XML-Dateien validiert werden, die XSLT-Datei, mit der die XML-Dateien transformiert werden sollen usw.
- Die Batch-Verarbeitung von Dateien in einem Ordner erfolgt durch Rechtsklick auf den Ordner und Auswahl des entsprechenden Befehls aus dem Kontextmenü.

Anmerkung: Das Projektfenster kann über das Menü **Fenster** ein- und ausgeblendet werden.

1.1.3 Info-Fenster

Das Fenster "Info" (*Abbildung unten*) enthält detaillierte Informationen über das Element oder Attribut, in dem sich der Cursor gerade befindet.



Anmerkung: Sie können das Info-Fenster über das Menü **Fenster** ein- oder ausblenden.

1.1.4 Eingabehilfen

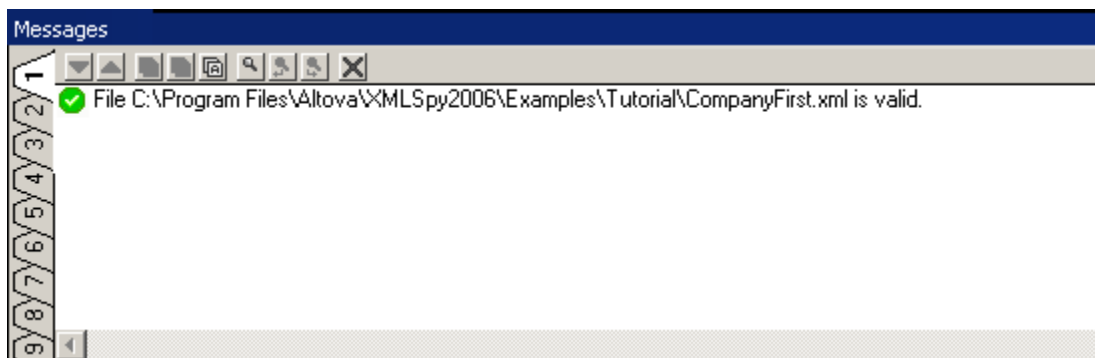
Eingabehilfen sind intelligente Editierfunktionen, mit Hilfe derer Sie schnell gültige XML-Dokumente erstellen können. Wenn Sie ein Dokument bearbeiten, sehen Sie in den Eingabehilfen je nach Cursorposition Bearbeitungsoptionen zur Struktur. Die Eingabehilfen beziehen die erforderlichen Informationen aus der zugrunde liegenden DTD, dem XML-Schema und/oder dem StyleVision Power Stylesheet. Wenn Sie z.B. ein XML-Datendokument bearbeiten, werden in den entsprechenden Eingabehilfenfenstern die Elemente, Attribute und Entities angezeigt (sowie dazugehörige Informationen), die an der aktuellen Cursorposition eingefügt werden können.

Hinweis:

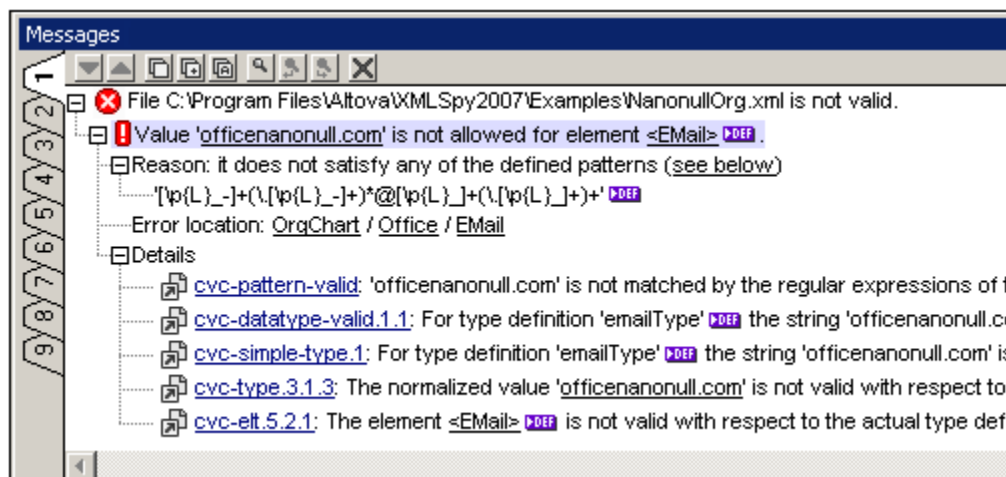
- Über die Menüoption **Fenster | Eingabehilfen** können Sie die Eingabehilfen ein- oder ausblenden.

1.1.5 Fenster "Meldungen"

Im Fenster "Meldungen" werden Meldungen über in Authentic Desktop ausgeführte Aktionen sowie Fehlermeldungen und andere Ausgabemeldungen angezeigt. Wenn z.B. ein XML-, XML-Schema, DTD- oder XQuery-Dokument validiert wird gültig ist, wird im Validierungsfenster eine entsprechende Meldung (*Abbildung unten*) angezeigt.



Andernfalls wird eine Meldung, die den Fehler beschreibt, angezeigt (*Abbildung unten*). Beachten Sie, dass es Links (schwarzen Text) zu Nodes und Node-Inhalt im XML-Dokument gibt sowie Links zum entsprechenden Abschnitt in der Spezifikation im Internet, in dem die betreffende Regel beschrieben wird (blauer Text). Wenn Sie auf eines der violetten Dev-Symbole klicken, wird die entsprechende Schemadefinition in der Schema-Ansicht geöffnet.



Das Validierungs-Ausgabefenster ist in allen Ansichten aktiv. Wenn Sie jedoch auf einen Link zum XML-Dokumentinhalt klicken, wird der entsprechende Node in der Text-Ansicht markiert. Beachten Sie jedoch: Wenn ein XML-Schema in der Schema-Ansicht validiert wird, ändert sich die Ansicht bei Klicken auf das Dev-Symbol nicht.

Anmerkung: Der Befehl **Validieren** (im Menü XML) wird normalerweise auf das aktive Dokument angewendet. Sie können den Befehl jedoch auch auf eine Datei oder eine Gruppe von Dateien im aktiven Projekt anwenden. Wählen Sie die gewünschte Datei bzw. den gewünschten Ordner im Projekt-Fenster (durch Anklicken) aus und klicken Sie auf **XML | Validieren** oder **F8**. Ungültige Dateien in einem Projekt werden geöffnet und im Hauptfenster aktiviert. Falls die Datei ungültig ist, wird eine entsprechende Fehlermeldung angezeigt.

1.1.6 Menüleiste, Symbolleisten und Statusleiste

Menüleiste

Die Menüleiste ([siehe Abbildung](#)) enthält die verschiedenen Menüs der Applikation. Es gelten die folgenden Konventionen:

- Wenn Befehle in einem Menü in einer Ansicht oder an einer bestimmten Stelle im Dokument **nicht** zur Verfügung stehen, sind sie ausgegraut und deaktiviert.
- Bei Auswahl einiger Menübefehle erscheint ein Untermenü mit einer Liste zusätzlicher Optionen. Solche Menübefehle weisen rechts neben dem Befehlsnamen einen nach rechts weisenden Pfeil auf.
- Bei einigen Menübefehlen wird ein Dialogfeld aufgerufen, in dem Sie weitere Informationen eingeben müssen, damit der Befehl ausgeführt werden kann. Diese Befehle sind durch drei Auslassungspunkte (...) nach dem Namen des Befehls gekennzeichnet.
- Um einen Menübefehl aufzurufen, klicken Sie auf den Menünamen und anschließend auf den Befehl. Enthält ein Menüeintrag ein Untermenü, wird dieses geöffnet, wenn Sie die Maus über den Menüeintrag positionieren. Klicken Sie auf den gewünschten Untermenüeintrag.
- Sie können ein Menü auch durch Drücken der entsprechenden Tastenkombination öffnen. Die Tastenkombination für die einzelnen Menüs lautet **Alt+TASTE**, wobei TASTE für den unterstrichenen Buchstaben im Menünamen steht. Die Tastenkombination für das Menü **Datei** ist z.B. **Alt+a**.
- Die Auswahl eines Menübefehls (also eines Befehls in einem Menü) erfolgt durch das nacheinander-Auswählen (i) des Menüs durch die entsprechende Tastenkombination (siehe voriger Punkt) und (ii) anschließendes Drücken der Tastenkombination für den jeweiligen Befehl (**Alt+TASTE**, wobei TASTE dem unterstrichenen Buchstaben im Befehlsnamen entspricht). Um z.B. eine neue Datei zu erstellen, (**Datei | Neu**), drücken Sie **Alt+a** und dann **Alt+N**.
- Einige Menübefehle können **direkt** durch Drücken eines speziellen **Tastenkürzels** oder einer Tastenkombination (**Strg+TASTE**) aufgerufen werden. Bei Befehlen, denen ein Tastenkürzel zugewiesen wurde, wird rechts vom Befehl das Tastenkürzel oder die Tastenkombination angezeigt. So können Sie z.B. eine neue Datei mit der Tastenkombination **Strg+N** erstellen und eine XML-Datei durch Drücken von **F8** validieren. Im Dialogfeld "Anpassen" (**Extras | Anpassen**) können Sie auf dem Register "Tastatur" Ihre [eigenen Tastenkürzel erstellen](#).

Symbolleisten

Die Symbolleisten ([siehe Abbildung](#)) enthalten Schaltflächen zum Aufrufen von Menübefehlen. Wenn Sie den Mauszeiger über die Schaltfläche halten, wird der Name des Befehls angezeigt. Um den Befehl auszuführen, klicken Sie auf die Schaltfläche.

Die Symbolleisten-Schaltflächen sind in Gruppen angeordnet. Im Dialogfeld [Extras | Anpassen | Symbolleisten](#) können Sie einstellen, welche Symbolleistengruppen angezeigt werden sollen. Diese Einstellungen gelten für die aktuelle Ansicht. Um Einstellungen für eine andere Ansicht vorzunehmen, wechseln Sie zu dieser Ansicht und nehmen Sie anschließend unter [Extras | Anpassen | Symbolleisten](#) Ihre Einstellungen vor. Sie können Symbolleistengruppen auf der GUI auch an eine andere Stelle ziehen, indem Sie darauf klicken und mit dem Ziehpunkt (oder an der Titelleiste) an die gewünschte Stelle verschieben. Wenn Sie auf den Ziehpunkt doppelklicken, wird die Symbolleiste abgedockt und frei schwebend angezeigt; wenn Sie auf die Titelleiste doppelklicken, wird die Symbolleiste an der Stelle angedockt, an der sie sich vorher befunden hat.

Statusleiste

Die Statusleiste befindet sich am unteren Rand des Applikationsfensters ([siehe Abbildung](#)). In ihr werden die folgenden Informationen angezeigt: (i) Statusinformationen über das Laden von

Dateien und (ii) Informationen über Menübefehle und Befehlsschaltflächen in den Symbolleisten, wenn Sie den Cursor darüber platzieren.

1.2 Die Applikationsumgebung

In diesem Abschnitt beschreiben wir verschiedene Aspekte der Applikation, die für den Anfang wichtig sind. Gehen Sie diesen Abschnitt durch, um sich mit Authentic Desktop vertraut zu machen. Dieser Abschnitt enthält wichtige Informationen über Einstellungen und Anpassungen und verschafft Ihnen einen allgemeinen Überblick darüber.

Dieser Abschnitt ist folgendermaßen gegliedert:

- [Einstellungen und Anpassung](#): Hier wird beschrieben wie und wo wichtige Einstellungen und anpassbare Optionen definiert werden können.
- [Tutorials, Projekte, Beispiele](#): Hier finden Sie die Pfade der verschiedenen im Applikationspaket enthaltenen Dateien, die nicht Teil des Programms selbst sind.
- [Produktfeatures, Dokumentation und Altova-Produkte](#): Hier finden Sie Links zur [Altova Website](#), wo Sie Informationen über Produktfeatures, ein Hilfe in zusätzlichen Formaten und andere Altova-Produkte finden.

1.2.1 Einstellungen und Anpassung

Sie können Authentic Desktop mittels verschiedener Einstellungen und Optionen konfigurieren. In diesem Abschnitt erhalten Sie einen kurzen Überblick über diese Optionen. Dieser Abschnitt ist in die folgenden Teile gegliedert.

- Einstellungen
- Anpassung

Einstellungen

Auf den verschiedenen Registern im Dialogfeld "Optionen" wird eine Reihe wichtiger Authentic Desktop-Einstellungen definiert. Gehen Sie die verschiedenen Optionen selbst durch, um zu sehen, welche Optionen zur Verfügung stehen.

Anpassung

Sie können verschiedene Aspekte von Authentic Desktop anpassen, wie unter anderem auch das Aussehen der Benutzeroberfläche. Die entsprechenden Optionen dazu finden Sie im Dialogfeld "Anpassen" (Aufruf über den Menübefehl [Extras | Anpassen](#)).

Die verschiedenen Konfigurationsoptionen sind im Abschnitt [Benutzerreferenz](#) beschrieben.

1.2.2 Tutorials, Projekte, Beispiele

Das Authentic Desktop Installationspaket enthält Tutorials, Projekte und Beispieldateien.

Pfade von Tutorials, Projekten und Beispieldateien

Die Authentic Desktop-Tutorials, -Projekte und -Beispieldateien sind in folgendem Ordner installiert:

```
C:\Documents and Settings\<username>\My Documents\  
Altova\Authentic2009\AuthenticExamples\
```

Der Ordner `My Documents\Altova\Authentic2009` wird für jeden Benutzer, der auf dem PC einen Account hat, im entsprechenden Ordner dieses Benutzers unter `<username>` installiert. Bei diesem Installationssystem stehen jedem Benutzer seine eigenen `AuthenticExamples`-Ordner in einem eigenen Arbeitsbereich zur Verfügung.

Anmerkung zum Authentic Desktop-Hauptordner

Bei Installation von Authentic Desktop auf einem Rechner, wird ein Hauptordner `Altova\Authentic2009` unter folgendem Pfad angelegt:

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

Wenn ein Benutzer Authentic Desktop auf diesem Rechner zum ersten Mal startet, erstellt Authentic Desktop eine Kopie dieses Hauptordners im Ordner `<username>\My Documents\`. Es ist daher wichtig, dass Sie beim Arbeiten mit Tutorial- oder Beispieldateien nicht im Hauptordner arbeiten, da die bearbeiteten Dateien sonst in den Benutzerordner eines Benutzers kopiert werden, der Authentic Desktop zu einem späteren Zeitpunkt zum ersten Mal startet.

Pfade der Tutorial-, Projekt- und Beispieldateien

Alle Tutorial-, Projekt- und Beispieldateien befinden sich im Ordner .

1.2.3 Authentic Desktop Features and Help, and Altova Products

Auf der Altova-Website, www.altova.com, finden Sie eine Unmenge an Informationen zu Authentic Desktop und Ressourcen. Darunter die folgenden:

Liste der Authentic Desktop-Features

Die Altova-Website bietet [eine Liste der Authentic Desktop Features](#).

Authentic Desktop Hilfe

Diese Dokumentation ist die von Altova zur Verfügung gestellte Hilfe zu Authentic Desktop. Sie steht in Form des in Authentic Desktop integrierten Hilfesystems zur Verfügung, die mit dem Befehl **Hilfe** oder durch Drücken von **F1** aufgerufen werden kann. Des Weiteren stehen die Benutzerhandbücher für alle Altova-Produkte auch in den folgenden Formaten zur Verfügung:

- [Online-HTML-Handbücher](#): Werden über die Support-Seite der Altova-Website aufgerufen
- [Druckbare PDF-Dokumente](#): können von der Altova-Website heruntergeladen und lokal ausgedruckt werden
- [Druckausgabe](#): können über einen Link auf der Altova-Website bestellt werden

Support-Optionen

Wenn Sie weitere Informationen darüber benötigen, welche Informationen im Benutzerhandbuch (also in dieser Dokumentation) zur Verfügung stehen, oder wenn Sie eine Frage zu Altova-Produkten haben, besuchen Sie unser [Support Center](#) auf der Altova-Website. Hier finden Sie folgende Informationen:

- Links zu unseren [FAQ-Seiten](#)
- [Diskussionsforen](#) zu Altova-Produkten und allgemeinen XML-Themen
- [Online Support-Formulare](#), über die Sie Anfragen an den Support senden können, falls Sie über ein Support- und Wartungspaket verfügen. Unser Support-Team wird sich um Ihre Anfrage kümmern.

Altova-Produkte

Eine Liste aller Altova-Produkte finden Sie auf der [Altova Website](#).

2 Tutorial Authentic-Ansicht

In der Authentic-Ansicht können Sie XML-Dokumente über eine grafische WYSIWYG-Oberfläche (*Abbildung unten*) wie in einer Textverarbeitungsanwendung, z.B. Microsoft Word, bearbeiten. Sie müssen dabei nur die Daten eingeben und sich nicht um die Formatierung des Dokuments kümmern, da die Formatierung bereits im Stylesheet definiert ist, das festlegt, wie das XML-Dokument in der Authentic-Ansicht dargestellt wird. Das Stylesheet (StyleVision Power Stylesheet, in diesem Tutorial als SPS bezeichnet) wird im Produkt Altova StyleVision erstellt:

Nanonull, Inc.

Location:

Street:	119 Oakstreet, Suite 4876	Phone:	+1 (321) 555 5155 0
City:	Vereno	Fax:	+1 (321) 555 5155 4
State & Zip:	DC 29213	E-mail:	office@nanonull.com

Vereno Office Summary: 4 departments, 15 employees.

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

Ein XML-Dokument kann in der Authentic in zweierlei Hinsicht bearbeitet werden: (i) Sie können die Struktur des Dokuments bearbeiten (z.B. Dokumentteile wie z.B. Absätze oder Überschriften hinzufügen oder löschen), und (ii) Sie können Daten eingeben (den Inhalt der Dokumentteile).

Dieses Tutorial enthält eine schrittweise Anleitung zu Folgendem:

- Öffnen einer bestehenden XML-Datei in der Authentic-Ansicht. Damit ein Dokument in der Authentic -Ansicht bearbeitet werden kann, mit es mit einer SPS-Datei verknüpft sein.
- Aussehen der Benutzeroberfläche der Authentic-Ansicht und eine allgemeine Beschreibung der wichtigsten Bearbeitungsfunktionen.
- Bearbeiten der Dokumentstruktur durch Einfügen und Löschen von Nodes.
- Eingabe von Daten in das XML-Dokument.
- Eingabe von Attributwerten über die Eingabehilfe "Attribute" und die Entity-Werte
- Drucken des Dokuments

Denken Sie daran, dass dieses Tutorial als Einführung gedacht ist und daher nicht auf komplizierte Aspekte eingeht. Zusätzliches Referenzmaterial und nähere Informationen zu

einzelnen Funktionen finden Sie in den Abschnitten zur [Oberfläche der Authentic-Ansicht](#).

Voraussetzungen für das Tutorial

Sie finden alle für das Tutorial benötigten **Dateien** im Ordner `C:\Documents and Settings\<username>\My Documents\Altova\<ProductName>\Examples` des Altova-Applikationsordners. Es sind die Dateien:

- `NanonullOrg.xml` (das XML-Dokument, das Sie öffnen)
- `NanonullOrg.sps` (das StyleVision Power Stylesheet, mit dem das XML-Dokument verknüpft ist)
- `NanonullOrg.xsd` (das XML-Schema, auf dem das XML-Dokument und das StyleVision Power Stylesheet basieren und mit dem diese verknüpft sind)
- `nanonull.gif` und `Altova_right_300.gif` (zwei Bilddateien, die in dem Tutorial verwendet werden)

Bitte beachten Sie: An einigen Stellen des Tutorials werden Sie gebeten, sich den XML-Text des XML-Dokuments anzusehen (also nicht die Authentic-Ansicht des Dokuments). Wenn die von Ihnen verwendete Altova Produktversion keine Textansicht enthält (dies ist bei der kostenlosen Authentic Desktop und der Authentic Browser der Fall), verwenden Sie einen einfachen **Texteditor**, wie Wordpad oder Notepad, um den Text des XML-Dokuments anzuzeigen.

Vorsicht: Verwenden Sie für das Tutorial am besten eine Kopie der Datei `NanonullOrg.xml`, damit Sie das Original im gegebenen Fall jederzeit wieder aufrufen können.

2.1 Öffnen eines XML-Dokuments in der Authentic-Ansicht

Sie können in der Authentic-Ansicht ein vorhandenes XML-Dokument bearbeiten oder ein neues XML-Dokument erstellen. In diesem Tutorial werden Sie ein vorhandenes XML-Dokument in der Authentic-Ansicht öffnen (wie in diesem Abschnitt beschrieben) und erfahren, wie Sie es bearbeiten können (Beschreibung siehe nächste Abschnitte). Außerdem erfahren Sie in diesem Abschnitt, wie ein neues XML-Dokument erstellt werden kann, um in der Authentic-Ansicht bearbeitet zu werden.

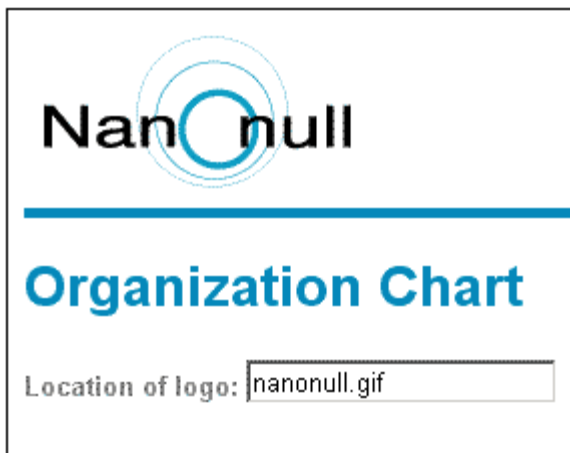
Öffnen eines vorhandenen XML-Dokuments

Die Datei, die Sie öffnen werden, heißt `NanonullOrg.xml`. Sie befindet sich im Ordner `Examples` der Altova-Applikation. Es gibt zwei Möglichkeiten, um `NanonullOrg.xml` zu öffnen:

- Klicken Sie in Ihrem Altova-Produkt auf **Datei | Öffnen**, suchen Sie anschließend die Datei `NanonullOrg.xml` im Dialogfeld, das nun angezeigt wird und klicken Sie auf "Öffnen".
- Suchen Sie die Datei im Windows Explorer, rechtsklicken Sie darauf und wählen Sie Ihr Altova-Produkt als die Applikation aus, mit der die Datei geöffnet werden soll.

Die Datei `NanonullOrg.xml` wird direkt in der Authentic-Ansicht (*Abbildung unten*) geöffnet.

•



Beachten Sie: Wie ein XML-Dokument in der Authentic-Ansicht angezeigt wird, wird im SPS definiert. Ohne ein SPS kann das Dokument nicht in der Authentic-Ansicht dargestellt werden.

Erstellen eines neuen XML-Dokuments auf Basis eines SPS

Sie können auch ein neues XML-Dokument erstellen, das auf einem SPS basiert. Dabei gibt es zwei Möglichkeiten: über den Menübefehl **Datei | Neu** und über den Menübefehl **Authentic | Neues Dokument erstellen**. In beiden Fällen wird ein SPS ausgewählt.

Über Datei | Neu

1. Wählen Sie **Datei | Neu** und wählen Sie im Dialogfeld "Neues Dokument anlegen" XML als den Dateityp der neu zu erstellenden Datei aus.
2. Klicken Sie auf **STYLEVISION Stylesheet auswählen...** und suchen Sie das gewünschte StyleVision Power Stylesheet.

Über Authentic | Neues Dokument erstellen

1. Wählen Sie **Authentic | Neues Dokument**.
2. Wählen Sie im Dialogfeld "Neues Dokument erstellen" das gewünschte SPS aus.

Wenn dem StyleVision Power Stylesheet eine XML-Vorlagendatei zugewiesen wurde, werden die Daten in der XML-Vorlagendatei als Ausgangsdaten der in der Authentic-Ansicht erstellten Vorlage verwendet.

2.2 Die Benutzeroberfläche der Authentic-Ansicht

Die Benutzeroberfläche der Authentic-Ansicht besteht aus einem Hauptfenster, in dem Sie die Dokumentdaten eingeben und bearbeiten, und drei Eingabehilfen. Die Bearbeitung eines Dokuments geht ganz einfach. Wenn Sie die Markup-Tags des Dokuments sehen möchten, blenden Sie die Markup-Tags ein. Geben Sie anschließend den Inhalt Ihres Dokuments ein. Um die Dokumentstruktur zu ändern, verwenden Sie entweder das Kontextmenü oder die Eingabehilfe "Elemente".

Anzeige der XML-Node Tags (Dokument-Markup-Tags)

Ein XML-Dokument ist im Grunde eine Hierarchie von Nodes. Z.B.:

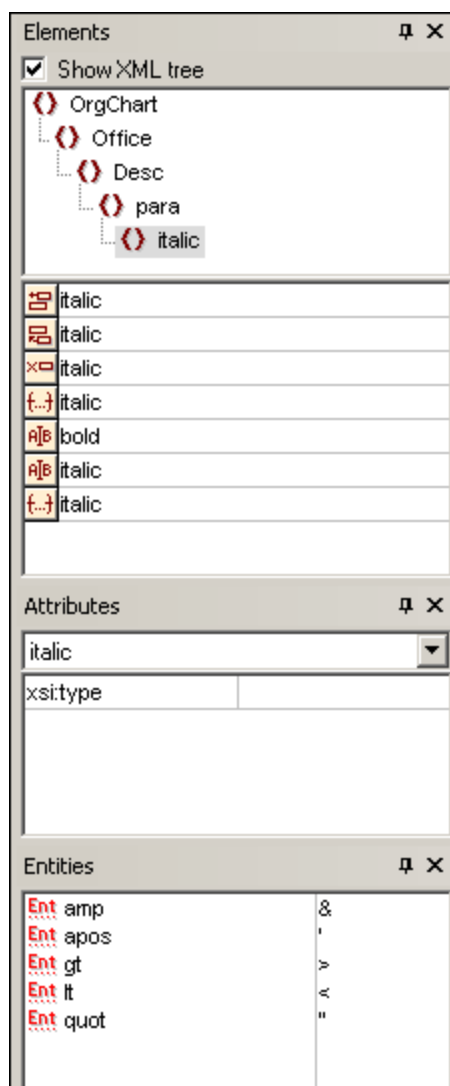
```
<DocumentRoot>
  <Person id="ABC001">
    <Name>Alpha Beta</Name>
    <Address>Some Address</Address>
    <Tel>1234567</Tel>
  </Person>
</DocumentRoot>
```

Standardmäßig werden die Node-Tags in der Authentic-Ansicht nicht angezeigt. Sie können sie durch Auswahl des Menübefehls **Authentic | Große Markup-Symbole einblenden** (oder der Symbolleisten-Schaltfläche ) einblenden. Große Markup-Tags enthalten den Namen des jeweiligen Node. Alternativ dazu können Sie kleine Markup-Symbole (keine Node-Namen in den Tags) und gemischte Markup-Tags (eine Mischung aus großen, kleinen und keinen Markup-Tags, welche durch den Designer des Stylesheets definiert ist; die Standardeinstellung für gemischte Markup-Symbole ist kein Markup) auswählen.

Sie können den Text des XML-Dokuments in der Text-Ansicht Ihres Altova-Produkts oder in einem Text Editor ansehen.

Eingabehilfen

Die Benutzeroberfläche bietet drei Eingabehilfen (*Abbildung unten*), die standardmäßig am rechten Rand des Applikationsfensters angeordnet sind. Es sind dies die Eingabehilfen "Element", "Attribute" und "Entities".



Element-Eingabehilfe: In der Element-Eingabehilfe werden Elemente angezeigt, die an der aktuellen Cursorposition oder der im Hauptfenster ausgewählten Stelle eingefügt und entfernt werden können. Beachten Sie, dass die Eingabehilfe kontextsensitiv ist; ihr Inhalt ändert sich, je nachdem, wo sich der Cursor gerade befindet, bzw. welcher Text ausgewählt ist. Der Inhalt der Eingabehilfe kann auch auf eine andere Art geändert werden: Wenn in der XML-Struktur ein anderer Node ausgewählt wird, werden in der Eingabehilfe die für diesen Node relevanten Elemente angezeigt. Sie können die Element-Eingabehilfe erweitern, um die XML-Struktur anzuzeigen, indem Sie das Kontrollkästchen "XML-Baumstruktur anzeigen" am oberen Rand der Eingabehilfe aktivieren (*siehe Abbildung oben*). In der XML-Baumstruktur wird die Hierarchie der Nodes vom obersten Element-Node bis hinunter zu dem im Hauptfenster ausgewählten Node angezeigt.

Attribute-Eingabehilfe: In der Attribute-Eingabehilfe werden die Attribute des im Hauptfenster ausgewählten Elements sowie deren Werte angezeigt. Attributwerte können in der Attribute-Eingabehilfe eingegeben oder bearbeitet werden. In der Auswahlliste der Attribute-Eingabehilfe können Element-Nodes von der obersten Elementebene bis hinunter zum ausgewählten Element ausgewählt werden. Bei Auswahl eines Elements aus dieser Dropdown-Liste werden die Attribute dieses Elements in der Eingabehilfe angezeigt, wo sie anschließend bearbeitet werden können.

Entities-Eingabehilfe: Die Entities-Eingabehilfe ist nicht kontextsensitiv. Hier werden alle im Dokument deklarierten Entities angezeigt. Wenn Sie auf eine Entity doppelklicken, wird sie an der Cursorposition eingefügt. Eine Beschreibung, wie man Entites für ein Dokument hinzufügt, finden Sie im Abschnitt [Benutzeroberfläche der Authentic-Ansicht](#).

Kontextmenü

Wenn Sie mit der rechten Maustaste auf eine Stelle in der Authentic-Ansicht doppelklicken, erscheint ein Kontextmenü zum entsprechenden Node. Im Kontextmenü finden Sie Befehle, die Ihnen Folgendes ermöglichen:

- Einfügen von Nodes an dieser Stelle oder vor oder nach dem ausgewählten Node. Das Untermenü enthält eine Liste der Nodes, die an der jeweiligen Stelle eingefügt werden können.
- Entfernen des ausgewählten Node (falls dies im Schema zulässig ist) oder aller entfernbaren übergeordneter Elemente. Die Nodes, die (laut Schema) entfernt werden dürfen, werden im Untermenü aufgelistet.
- Einfügen von Entities und CDATA-Abschnitten. Die für das Dokument deklarierten Entities werden in einem Untermenü aufgelistet. CDATA-Abschnitte können nur innerhalb von Text eingefügt werden.
- Ausschneiden, Kopieren, Einfügen und Löschen von Dokumentinhalt.

Anmerkung: Nähere Informationen über die Benutzeroberfläche finden Sie unter [Benutzeroberfläche der Authentic-Ansicht](#).

2.3 Node-Operationen

Es gibt in der Authentic-Ansicht von XML-Dokumenten zwei grundlegende Node-Typen: **Element-Nodes** und **Attribut-Nodes**. Diese Nodes sind alle mit Tags versehen, die Sie [einblenden können](#). Es gibt auch andere Nodes im Dokument, wie z.B. Text-Nodes (die mit keinen Markup-Tags versehen sind) und CDATA-Abschnitt-Nodes (welche mit Markup versehen sind, um sie vom umgebenden Text abzugrenzen).

Die in diesem Abschnitt beschriebenen Node-Operationen beziehen sich nur auf Element-Nodes und Attribut-Nodes. Beim Ausprobieren der hier beschriebenen Operationen sollten Sie am besten die [großen Markup-Symbole einblenden](#)

Anmerkung:

Beachten Sie bitte, dass Sie **nur Elemente derselben oder einer übergeordneten Ebene** vor oder nach dem aktuellen Element einfügen oder anhängen können. Elemente derselben Ebene sind **gleichrangige** Elemente. Bei **gleichrangigen** Elementen eines Absatzelements handelt es sich entweder um andere Absatzelemente oder um Listen, eine Tabelle, eine Grafik usw. handeln. Gleichrangige Elemente können vor oder nach dem Element vorkommen. Elemente übergeordneter Ebenen sind **Vorfahren**-Elemente und Geschwisterelemente von Vorfahren. Bei Vorfahren eines Absatzelements könnte es sich um einen Abschnitt, ein Kapitel, einen Artikel usw. handeln. Ein Absatz in einem gültigen XML-Dokument hat bereits Vorfahren. Wenn Sie daher in der Authentic-Ansicht ein Element einer übergeordneten Ebene erstellen, wird dieses als Geschwisterelement des jeweiligen Vorfahrenelements erstellt. Wenn z.B. ein Abschnittselement an einen Absatz angehängt wird, wird es als Geschwisterelement des Abschnitts erstellt, der das aktuelle Absatzelement enthält - und zwar als der letzte Abschnitt auf dieser Ebene.

Durchführen von Node-Operationen

Node-Operationen können entweder durch Auswahl eines Befehls im [Kontextmenü](#) oder durch Klicken auf den Eintrag für die Node-Operation in der [Element-Eingabehilfe](#) durchgeführt werden. In einige Fällen kann ein Element oder Attribut hinzugefügt werden, indem Sie in der Authentic-Ansicht des Dokuments auf den [Link Node hinzufügen](#) klicken. In speziellen Fällen, in denen Elemente als Absätze oder Listeneinträge definiert wurden, können Sie durch Drücken der [Eingabetaste](#) ein neues gleichrangiges Element derselben Art erstellen, wenn sich der Cursor in einem derartigen Element befindet. In diesem Abschnitt erfahren Sie außerdem, wie Nodes über die Funktionen [Element anwenden](#), [Node entfernen](#) und [Element löschen](#) erstellt und gelöscht werden können.

Einfügen von Elementen

Elemente können an den folgenden Stellen eingefügt werden:

- an der Cursorposition innerhalb eines Element-Node. Die an dieser Stelle verfügbaren Elemente werden im Untermenü des Kontextmenüs **Einfügen** aufgelistet. In der Element-Eingabehilfe werden Elemente, die an dieser Stelle eingefügt werden können, durch ein  Symbol markiert. Platzieren Sie im Dokument `NanonullOrg.xml` den Cursor in das Element `para` und erstellen Sie über das Kontextmenü und die Element-Eingabehilfe die Elemente `bold` und `italic`.
- vor oder nach dem ausgewählten Element oder einem seiner Vorfahren, wenn dies laut Schema gestattet ist. Wählen Sie im ersten Untermenü, das angezeigt wird, das Element (ausgewähltes Element oder einen Vorfahren dieses Elements) aus, vor/nach dem das Element eingefügt werden soll. Wählen Sie nun im zweiten Untermenü, das nun angezeigt wird, das gewünschte Element aus, das Sie einfügen möchten. In der Element-Eingabehilfe werden Elemente, die vor oder nach dem ausgewählten Element eingefügt werden können, durch die Symbole  bzw.  gekennzeichnet. Beachten

Sie, dass Sie in der Element-Eingabehilfe Elemente nur vor/nach dem ausgewählten Element einfügen können, nicht aber vor/nach dem Vorfahrenelement. Probieren Sie diesen Befehl aus, indem Sie den Cursor zuerst in das Element `para` und anschließend in die Tabelle mit der Liste der Angestellten platzieren.

Link "Node hinzufügen"

Wenn ein Element oder Attribut im Dokument-Design nicht aber im XML-Dokument selbst vorhanden ist, wird an der Stelle im Dokument, an der dieser Node definiert ist, ein Node `hinzufügen` Link angezeigt. Um diesen Link in der Zeile mit dem Text *Location of logo* zu sehen, wählen Sie den Node `@href` im Element `CompanyLogo` aus und löschen Sie ihn (durch Drücken der **Entf**-Taste). Der Link `@href hinzufügen` wird im Element `CompanyLogo`, das Sie bearbeitet haben, angezeigt (*Abbildung unten*). Wenn Sie auf den Link klicken, wird der Node `@href` zum XML-Dokument hinzugefügt. Das Textfeld innerhalb des Tags `@href` wird angezeigt, weil im Design festgelegt ist, dass der Node `@href` auf diese Art hinzuzufügen ist. Sie müssen nun noch den Wert (oder Inhalt) des Node `@href` eingeben. Geben Sie den Text `nanonull.gif` ein.



Wenn das Content Model eines Elements nicht eindeutig ist, wenn z.B. festgelegt ist, dass eine Sequenz von Child-Elementen in jeder beliebigen Reihenfolge angezeigt werden kann, so wird der Link `... hinzufügen` angezeigt. Beachten Sie, dass kein Node-Name definiert ist. Wenn Sie auf den Link klicken, erscheint eine Liste von Elementen, die an dieser Stelle eingefügt werden dürfen.

Anmerkung: Der Link "Node hinzufügen" erscheint direkt in der Dokumentvorlage; es gibt keinen entsprechenden Eintrag dazu im Kontextmenü oder in der Element-Eingabehilfe.

Erstellen neuer Elemente mit der Eingabetaste

In Fällen, in denen ein Element (vom Stylesheet Designer) als Absatz oder Listeneintrag formatiert werden muss, wird bei Drücken der Eingabetaste ein neuer Node dieser Art nach dem aktuellen Node eingefügt, wenn sich der Cursor innerhalb eines solchen Node befindet. Sie können diese Funktion im Dokument `NanonullOrg.xml` ausprobieren. Gehen Sie zum Ende des Node `para` (genau vor den End-Tag) und drücken Sie die **Eingabetaste**.

Anwenden von Elementen

In Elementen mit gemischtem Inhalt (denjenigen, die sowohl Text als auch Child-Elemente enthalten) können Sie Textinhalt markieren und darauf eines der zulässigen Child-Elemente anwenden. Der ausgewählte Text wird zum Inhalt des angewendeten Elements. Um Elemente anzuwenden, wählen Sie im Kontextmenü den Befehl "Anwenden" und wählen Sie anschließend eines der vorgeschlagenen Elemente aus. (Wenn auf den ausgewählten Text keine Elemente angewendet werden können, wird der Befehl "Anwenden" im Kontextmenü

nicht angezeigt.) In der Element-Eingabehilfe werden Elemente, die für eine Auswahl angewendet werden können, durch das Symbol  gekennzeichnet. Wählen Sie im Dokument `NanonullOrg.xml` Text im Element gemischten Inhalts `para` aus und probieren Sie aus, wie sich die Elemente `bold` und `italic` bei Anwendung auswirken.

Der Stylesheet Designer hätte auch eine Symbolleiste-Schaltfläche für die Anwendung eines Elements erstellen können. Im Dokument `NanonullOrg.xml` können die Elemente `bold` und `italic` durch Klicken auf die Symbole "bold" bzw. "italic" in der Authentic-Symbolleiste der Applikation angewendet werden.

Entfernen von Nodes

Ein Node kann entfernt werden, wenn das Dokument dadurch nicht ungültig wird. Bei Entfernung eines Node werden der Node und sein gesamter Inhalt gelöscht. Sie können einen Node mit dem Kontextmenübefehl **entfernen** entfernen. Wenn der Befehl "Entfernen" markiert wird, erscheint ein Untermenü, das alle Nodes enthält, die entfernt werden dürfen, beginnend beim ausgewählten Node bis zum obersten Node des Dokuments. Um einen zu löschenden Node auszuwählen, platzieren Sie den Cursor in den Node oder markieren Sie den Node (oder einen Teil davon). In der Element-Eingabehilfe werden Nodes, die entfernt werden dürfen, durch das Symbol  gekennzeichnet. Sie können einen solchen Node auch entfernen, indem Sie ihn auswählen und die **Entf**-Taste drücken. Probieren Sie im Dokument `NanonullOrg.xml` die oben beschriebenen Methoden aus, indem Sie einige Nodes entfernen. Sie können Ihre Änderungen mit **Strg+Z** rückgängig machen.

Löschen von Elementen

Element-Nodes, bei denen es sich um Child-Elemente mit gemischtem Inhalt (sowohl Text- als auch Element-Children) handelt, können gelöscht werden. Sie können das gesamte Element löschen, wenn Sie den Node auswählen oder den Cursor als Einfügepunkt in den Node setzen. Sie können den Element-Markup-Code eines Textfragments innerhalb des Elements löschen, indem Sie das Textfragment markieren. Klicken Sie nach dem Markieren des Fragments im Kontextmenü auf den Befehl **Löschen** und wählen Sie anschließend das zu löschende Element aus. In der Element-Eingabehilfe werden Elemente, die aus einer bestimmten Auswahl gelöscht werden können, durch das Symbol  (Einfügepunktauswahl) und  (Bereichsauswahl) gekennzeichnet. Probieren Sie das Löschen von Elementen im Dokument `NanonullOrg.xml` an den Child-Elementen `bold` und `italic` des Elements `para` (welches gemischten Inhalt hat) aus.

Tabellen und Tabellenstruktur

Es gibt zwei Arten von Tabellen in der Authentic-Ansicht:

- *SPS-Tabellen (statische und dynamische)*. Die allgemeine Struktur von SPS-Tabellen wird durch den Stylesheet Designer festgelegt. Innerhalb dieser grundlegenden Struktur sind die einzigen zulässigen Änderungen an der Struktur inhaltsbezogene. So können Sie z. B. neue Zeilen zu einer dynamischen SPS-Tabelle hinzufügen.
- *XML-Tabellen*, in welchen Sie entscheiden, dass der Inhalt eines bestimmten Node (z.B. eines Node für personenspezifische Daten) als Tabelle dargestellt werden soll. Wenn der Stylesheet Designer die Erstellung dieses Node als XML-Tabelle ermöglicht hat, können Sie die Struktur der Tabelle festlegen sowie ihren Inhalt bearbeiten. XML-Tabellen werden im Abschnitt [Verwendung von Tabellen in der Authentic-Ansicht](#) näher beschrieben.

2.4 Eingabe von Daten in der Authentic-Ansicht

Daten werden im XML-Dokument direkt im Hauptfenster der Authentic-Ansicht eingegeben. Zusätzlich dazu können Daten für Attribute (also der Attributwert) über [die Attribute-Eingabehilfe eingegeben werden](#). Daten werden entweder (i) direkt als Text oder (ii) durch Auswahl einer Option in einem Dateneingabeelement, eingegeben, das anschließend auf einen vordefinierten Texteintrag gemappt wird.

Hinzufügen von Textinhalt

Sie können Elementinhalt und Attributwerte im Hauptfenster der Authentic-Ansicht direkt als Text eingeben. Platzieren Sie dazu den Cursor an die gewünschte Stelle und geben Sie den Text ein. Sie können Text auch aus der Zwischenablage in das Dokument kopieren. Inhalt kann auch mit den normalen Inhaltsbearbeitungsmechanismen wie **Entfernen** und **Feststelltaste**, durch Markieren des Texts und Eingabe des Ersetzungstexts oder durch Löschen des markierten Texts bearbeitet werden.

Um z.B. den Namen der Firma im Feld `Name` des Elements `Office` zu ändern, platzieren Sie den Cursor hinter Nanonull und geben Sie "USA" ein, um den Namen von "Nanonull, Inc." in "Nanonull USA, Inc." zu ändern.



Wenn Text editierbar ist, können Sie den Cursor hineinsetzen und den Text markieren, andernfalls ist das nicht der Fall. Versuchen Sie, einen der **Feldnamen** (nicht den Wert im Feld) "Street", "City", "State/Zip" im Adressblock zu ändern. Sie können den Cursor nicht in diesen Text platzieren, da dies kein XML-Inhalt ist, sondern aus dem StyleVision Power Stylesheet stammt.

Einfügen von Sonderzeichen und Entities

Bei der Eingabe von Daten werden folgende Inhaltsarten speziell behandelt:

- *Sonderzeichen, die als XML-Markup-Symbole verwendet werden* (kaufmännisches Und, Apostroph, Größer als, Kleiner als und Anführungszeichen). Diese Zeichen stehen als [vordefinierte Entities](#) zur Verfügung und können durch Doppelklick auf die entsprechende Entity in der Entities-Eingabehilfe im Dokument eingefügt werden. Wenn diese Zeichen häufig vorkommen (z.B. in Programmcode Listen), können Sie innerhalb von CDATA-Abschnitten eingegeben werden. Klicken Sie zum Einfügen eines CDATA-Abschnitts mit der rechten Maustaste an die gewünschte Stelle und wählen Sie im Kontextmenü den Befehl **CDATA-Abschnitt einfügen**. Der XML-Prozessor ignoriert alle Markup-Zeichen innerhalb von CDATA-Abschnitten. Dies bedeutet auch, dass Sonderzeichen innerhalb von CDATA-Abschnitten als das Zeichen selbst und nicht die Entity-Referenz eingegeben werden müssen.
- *Sonderzeichen, die nicht über die Tastatur eingegeben werden können*, sollten durch Kopieren des Zeichens aus der Zeichentabelle Ihres Systems an die gewünschte Stelle im Dokument eingegeben werden.
- *Ein häufig verwendeter Textstring* kann als [Entity definiert werden](#), die in der Entities-Eingabehilfe angezeigt wird. Die [Entity wird an der gewünschten Stelle eingefügt](#). Platzieren Sie dazu den Cursor an die gewünschte Stelle und doppelklicken Sie in der Entities-Eingabehilfe auf die Entity. Diese Funktion eignet sich dazu, um Dokumente auf aktuellem Stand zu halten, da der Wert des Textstrings nur einmal gespeichert ist. Wenn der Wert geändert werden muss, müssen Sie nur die

Entity-Definition ändern.

Anmerkung: Wenn Markup-Tags in der Authentic -Ansicht ausgeblendet werden, werden leere Elemente leicht übersehen. Um sicherzustellen, dass Sie kein leeres Element übersehen haben [aktivieren Sie ein der Optionen große Markup-Symbole \(oder kleine Markup-Symbole\) anzeigen](#).

Probieren Sie die Verwendung der einzelnen oben beschriebenen Arten von Text aus.

Hinzufügen von Inhalt über ein Dateneingabe-Steuerelement

Beim Editieren von Inhalt wird dieser direkt durch Eingabe des Texts als Elementinhalt eingegeben. Es gibt aber noch eine andere Methode, um **Elementinhalt** (oder Attributwerte) in der Authentic-Ansicht einzugeben: mit Hilfe von Dateneingabe-Steuerelemente.

Nachfolgend sehen Sie eine Liste von Dateneingabe-Steuerelementen in der Authentic-Ansicht sowie eine Erklärung dazu, wie Daten bei jedem dieser Elemente in die XML-Datei eingegeben werden.

Dateneingabe-Steuerelement	Daten in der XML-Datei
Eingabefeld (Textfeld)	Text wird vom Benutzer eingegeben
Eingabefeld mit mehreren Zeilen	Text wird vom Benutzer eingegeben
Dropdown-Liste	Auswahl des Benutzers wird auf einen Wert gemappt
Kontrollkästchen	Auswahl des Benutzers wird auf einen Wert gemappt
Optionsfeld	Auswahl des Benutzers wird auf einen Wert gemappt
Schaltfläche	Auswahl des Benutzers wird auf einen Wert gemappt

In der statischen Tabelle mit den Adressfeldern (siehe unten) stehen zwei Dateneingabe-Steuerelemente zur Verfügung: Ein Eingabefeld für das Feld "Zip" (PLZ) und eine Dropdown-Liste für das Feld "State" (Bundesstaat). Die Werte, die Sie in die Textfelder eingeben, werden direkt als XML-Inhalt der entsprechenden Elemente eingegeben. Bei anderen Dateneingabe-Steuerelementen wird Ihre Auswahl auf einen Wert gemappt.

The screenshot shows a web form with the following fields and values:

- Street:** 119 Oakstreet, Suite 4876
- City:** Vereno
- State & Zip:** A dropdown menu is open showing options DC, DE, FL, and GA. The Zip field next to it contains the value 29213.

Below the form, there is a blue link labeled "Vereno Office Summary" and a line of text that reads "artments, 15 employees."

Hier sehen Sie den XML-Text für die oben gezeigte Authentic-Ansicht:

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

Beachten Sie, dass die Dropdown-Listen-Auswahl "DC" auf einen Wert "DC" gemappt ist. Der Wert im Feld "zip" wird direkt als Inhalt des Elements `ipo:zip` eingegeben.

2.5 Eingabe von Attributwerten

Ein Attribut ist eine Eigenschaft eines Elements. Ein Element kann beliebig viele Attribute haben. Attribute haben Werte. Manchmal müssen XML-Daten als Attributwert eingegeben werden. In der Authentic-Ansicht werden Attributwerte auf zwei Arten eingegeben:

- Als Inhalt im Hauptfenster, wenn das Attribut so definiert wurde, dass es die Eingabe des Werts auf diese Art zulässt.
- In der Attribut-Eingabehilfe

Attributwerte im Hauptfenster

Attributwerte können als Fließtext oder als Text in einem Eingabefeld eingegeben werden oder als vom Benutzer aus einer Auswahl gewählter Eintrag, der auf einen XML-Wert gemappt wird. Attributwerte werden auf dieselbe Art eingegeben, wie Element-Inhalt: siehe [Eingabe von Daten in der Authentic-Ansicht](#). In solchen Fällen erfolgt die Unterscheidung zwischen Element-Inhalt und Attributwerten durch das StyleVision Power Stylesheet und die Daten werden entsprechend behandelt.

Attributwerte in der Attribut-Eingabehilfe

Wenn Sie einen Attributwert eingeben oder ändern wollen, können Sie dies in der Attribut-Eingabehilfe tun. Zuerst wird der Attribut-Node in der Authentic-Ansicht ausgewählt, dann wird der Wert des Attributs in der Attribut-Eingabehilfe eingegeben oder bearbeitet. Der Pfad des in `NanonullOrg.xml` verwendeten Logos wird als der Wert des `href` Attributs des Elements `CompanyLogo` gespeichert. Um ein anderes Logo zu verwenden, gehen Sie vor wie folgt:

1. Wählen Sie das Element `CompanyLogo` aus, indem Sie auf den Tag `CompanyLogo` klicken. Die Attribute des Elements `CompanyLogo` werden in der Attribut-Eingabehilfe angezeigt.
2. Ändern Sie in der Attribut-Eingabehilfe den Wert des Attributs `href` von `nanonull.gif` in `Altova_right_300.gif` (ein Bild im Ordner `Examples`).

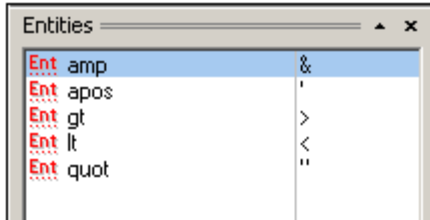


Daraufhin wird das Nanonull-Logo durch das Altova-Logo ersetzt.

Anmerkung: Entities können nicht in der Attribut-Eingabehilfe eingegeben werden.

2.6 Hinzufügen von Entities

Bei einer Entity in der Authentic-Ansicht handelt es sich normalerweise (nicht notwendigerweise) um XML-Daten wie z.B. ein einzelnes Zeichen, einen Textstring oder auch ein Fragment eines XML-Dokuments. Auch eine Binärdatei wie z.B. eine Bilddatei kann eine Entity sein. Alle verfügbaren Entities werden in der Entities-Eingabehilfe (*Abbildung unten*) angezeigt. Um eine Entity einzufügen, platzieren Sie den Cursor an die gewünschte Stelle im Dokument und doppelklicken Sie anschließend in der Entities-Eingabehilfe auf die Entity. Beachten Sie, dass Sie Entities nicht in der Attribut-Eingabehilfe eingeben können.



Das kaufmännische Und (&) hat in XML eine spezielle Bedeutung (wie auch das Apostroph, das "Größer als"- und das "Kleiner als"-Zeichen und das doppelte Anführungszeichen). Um diese Zeichen einzufügen, werden Entities verwendet, damit sie nicht mit Zeichen, die eine XML-Bedeutung haben, verwechselt werden. Diese Zeichen sind in der Authentic-Ansicht als Entities verfügbar.

Ändern Sie in `NanonullOrg.xml` den Jobtitel von Joe Martin (in Marketing) in Marketing Manager Europe & Asia. Gehen Sie dazu folgendermaßen vor:

1. Platzieren Sie den Cursor dorthin, wo das kaufmännische Und eingefügt werden soll.
2. Doppelklicken Sie auf die Entity, die als "amp" aufgelistet ist. Daraufhin wird ein kaufmännisches Und eingefügt (*Abbildung unten*)

Marketing (2)		
First	Last	Title
Joe	Martin	Marketing Manager Europe &

Anmerkung:

Die Entities-Eingabehilfe ist nicht kontextsensitiv. Es werden immer alle verfügbaren Entities angezeigt, egal wo sich der Cursor gerade befindet. Dies bedeutet nicht, dass eine Entity überall im Dokument eingefügt werden kann. Wenn Sie sich nicht sicher sind, validieren Sie das Dokument nach dem Einfügen der Entity mit: **XML | Validieren (F8)**.

Definieren Ihrer eigenen Entities

Als Dokument-Bearbeiter können Sie auch Ihre eigenen Entities definieren. Eine Beschreibung dazu finden Sie im Abschnitt [Definieren von Entities in der Authentic-Ansicht](#).

2.7 Drucken des Dokuments

Bei Ausdruck eines XML-Dokuments von der Authentic-Ansicht aus bleibt die in der Authentic-Ansicht sichtbare Formatierung erhalten.

So drucken Sie die Datei `NanonullOrg.xml`:

1. Wechseln Sie in den Modus "Markup-Symbole ausblenden", wenn Sie gerade einen anderen Modus verwenden, da sonst auch die Markup-Symbole gedruckt werden.
2. Wählen Sie **Datei | Druckvorschau**, um eine Vorschau aller Seiten anzuzeigen. In der Abbildung unten sehen Sie einen Teil einer um 50 % verkleinerten Seite in der Druckansicht.

Page 1 of 6

ALTOVA

www.altova.com

Organization Chart

Location of logo:

Nanonull, Inc.

Location:

Street: 119 Oakstreet, Suite 4876 City: Vereno State & Zip: DC 29213	Phone: +1 (321) 555 5155 0 Fax: +1 (321) 555 5155 4 E-mail: office@nanonull.com

[Vereno Office Summary:](#) 4 departments, 15 employees.

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

Beachten Sie, dass die Formatierung auf der Seite dieselbe ist, wie in der Authentic-Ansicht.

3. Um die Datei zu drucken, wählen Sie **Datei | Drucken**.

Beachten Sie, dass Sie auch eine Version des Dokuments drucken können, in der (kleine) Markup-Symbole angezeigt werden. Wechseln Sie dazu in den Modus "Kleine Markup-Symbole einblenden" oder "Große Markup-Symbole einblenden" und wählen Sie den Befehl "Drucken". In beiden Modi werden im Ausdruck kleine Markup-Symbole angezeigt.

3 Authentic-Ansicht

Die Authentic-Ansicht steht zur Verfügung, wenn Sie auf das Register "Authentic" des aktiven Dokuments klicken. Wenn dem XML-Dokument kein SPS zugewiesen wurde, werden Sie aufgefordert, dies zu tun.

Dieser Abschnitt enthält:

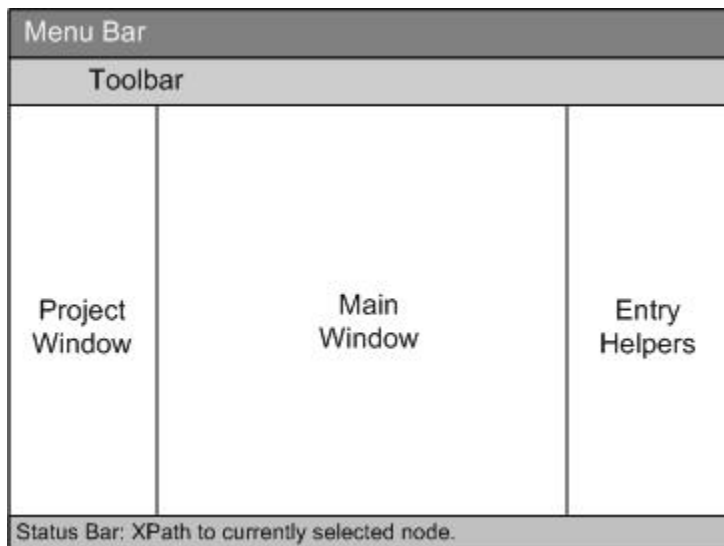
- eine Übersicht über die Benutzeroberfläche
- eine Beschreibung der Symbolleistensymbole der Authentic-Ansicht
- eine Beschreibung der Ansichtsmodi im Authentic-Ansicht-Hauptfenster
- eine Beschreibung der Eingabehilfen und deren Verwendung
- eine Beschreibung der verschiedenen in der Authentic-Ansicht verfügbaren Kontextmenüs
- eine Erläuterung zur Verwendung der verschiedenen Funktionen der Authentic-Ansicht

Weitere Informationen zur Authentic-Ansicht finden Sie:

- im Tutorial zur Authentic-Ansicht, in dem beschrieben wird, wie Sie mit der Authentic-Ansicht arbeiten. Sie finden das Tutorial in der Dokumentation zu Altova XMLSpy und zur Altova Authentic Desktop (siehe Abschnitt Tutorials) sowie [online](#).
- Eine ausführliche Beschreibung der Menübefehle der Authentic-Ansicht finden Sie im Abschnitt "Benutzerreferenz" Ihrer Produktdokumentation.

3.1 Übersicht über die Benutzeroberfläche

Die Authentic-Ansicht besteht aus einer Menüleiste am oberen Rand des Fensters und drei Bereichen: dem Projekt-Fenster, dem Hauptfenster und den Eingabehilfen-Fenstern (siehe Abbildung unten).



Menüleiste

Die Menüs der Menüleiste werden in der Produktdokumentation im Abschnitt "Benutzerreferenz" näher beschrieben.

Symbolleiste

Eine Beschreibung der Schaltflächen der Symbolleiste finden Sie im Abschnitt [Symbolleistenschaltflächen der Authentic-Ansicht](#).

Projekt-Fenster

XML-, XSL-, HTML-Schema und Entity-Dateien können in einem Projekt zusammen gruppiert werden. Verwenden Sie zum Erstellen und Bearbeiten der Liste der Projektdateien die Befehle im Menü "Projekt" (Beschreibung siehe Abschnitt "Benutzerreferenz" der Produktdokumentation). Die Liste der Projektdateien wird im Projekt-Fenster angezeigt. Um eine Datei im Projekt-Fenster aufzurufen, doppelklicken Sie darauf.

Hauptfenster

Das Hauptfenster ist der Bereich, in dem das XML-Dokument angezeigt und bearbeitet wird. Eine Beschreibung dazu finden Sie im Abschnitt [Authentic-Ansicht Hauptfenster](#).

Eingabehilfen

Dieser Bereich enthält drei Eingabehilfen-Fenster: Elemente, Attribute und Entities. Welche Einträge in diesen Fenstern (Element- und Attribute-Eingabehilfen) zu sehen sind, ist vom Kontext abhängig, d.h. davon, an welcher Stelle im Dokument sich der Cursor gerade befindet. Um ein Element oder eine Entity in das Dokument einzugeben, doppelklicken Sie in der Eingabehilfe darauf. Der Wert eines Attributs wird in der Attribute-Eingabehilfe im Feld "Wert" des entsprechenden Attributs eingegeben. Nähere Informationen dazu finden Sie im Abschnitt [Authentic-Ansicht Eingabehilfen](#).

Statusleiste

In der Statusleiste wird der XPath zum aktuell ausgewählten Node angezeigt.

Kontextmenüs

Dies sind Menüs, die angezeigt werden, wenn Sie im Hauptfenster auf die rechte Maustaste klicken. Es stehen Ihnen kontextsensitive Editierbefehle zur Verfügung, mit Hilfe derer Sie die Struktur und den Inhalt des ausgewählten Nodes bearbeiten können. Diese Befehle gestatten das Einfügen, Anhängen oder Löschen eines Nodes, das Hinzufügen von Entities und das Ausschneiden und Einfügen von Inhalt.

3.2 Authentic-Ansicht Symbolleistenschaltflächen

Die Schaltflächen der Symbolleiste der Authentic-Ansicht sind Befehls-Shortcuts. Einige Symbole kennen Sie sicher bereits von anderen Windows-Anwendungen oder anderen Altova-Produkten her, andere sind vielleicht neu für Sie. In diesem Abschnitt werden die für die Authentic-Ansicht spezifischen Befehle beschrieben.

In der nachfolgenden Beschreibung sind verwandte Befehle gemeinsam gruppiert.

Wechseln in die Authentic-Ansicht

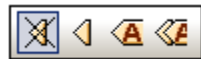


Wenn das XML-Dokument mit einem StyleVision Power Stylesheet verknüpft ist, wechseln Sie durch Auswahl des Befehls **Ansicht | Authentic** aus einer anderen Ansicht in die Authentic-Ansicht.

Wenn das Dokument **nicht** mit einem StyleVision Power Stylesheet verknüpft ist, wird ein Dialogfeld angezeigt, in dem Sie aufgefordert werden, das Dokument mit einem StyleVision Power Stylesheet zu verknüpfen. Falls Sie eine Meldung erhalten, dass keine temporäre Datei (Temp-Datei) erstellt werden konnte, wenn Sie in die Authentic-Ansicht zu wechseln versuchen, wenden Sie sich an Ihren Systemadministrator. Der Systemadministrator muss die Standard-Sicherheits-ID für "non-power User" ändern, damit diesen das Anlegen von Ordnern und Dateien gestattet ist.

Ein- und Ausblenden von XML Markup

In der Authentic-Ansicht können die Tags für alle, einige oder keines der XML-Elemente oder Attribute eingeblendet werden - entweder mit Namen (große Markup-Symbole) oder ohne Namen (kleine Markup-Symbole). Die vier Markup-Symbole werden in der Symbolleiste angezeigt, die entsprechenden Befehle finden Sie im Menü **Authentic**.



Markup-Symbole ausblenden. Alle XML-Tags mit Ausnahme der reduzierten Tags werden ausgeblendet. Wenn Sie im Modus "Markup-Symbole ausblenden" auf einen Tag, der reduziert wurde, doppelklicken (wodurch er normalerweise erweitert wird), wird der Inhalt des Nodes angezeigt und die Tags ausgeblendet.



Kleine Markup-Symbole einblenden. Die XML-Element/Attribut-Tags werden ohne Namen angezeigt.



Große Markup-Symbole einblenden. Die XML-Elemente/Attribute werden mit ihren Namen angezeigt.



Mixed Markup-Symbole einblenden. Im StyleVision Power Stylesheet können Sie für jedes XML-Element oder Attribut einzeln festlegen, wie es im "Mixed Markup"-Modus angezeigt werden soll (mit großen oder kleinen Markup-Symbolen oder gar nicht). Im Mixed Markup-Modus der Authentic-Ansicht ist die Anzeige der Markup-Symbole also benutzerdefiniert. Beachten Sie allerdings, dass diese Einstellungen von der Person vorgenommen werden, die das StyleVision Power Stylesheet erstellt hat.

Bearbeiten von dynamischen Tabellenstrukturen

Zeilen in einer **dynamischen SPS-Tabelle** sind Wiederholungen einer Datenstruktur. Jede Zeile steht für eine Instanz eines einzelnen Elements, daher hat jede Zeile dieselbe

XML-Substruktur wie die darauf folgende.

Die Befehle zum Bearbeiten dynamischer Tabellen dienen zum Bearbeiten der Zeilen einer dynamischen SPS-Tabelle, d.h. Sie können die Anzahl und Reihenfolge der Element-Instanzen ändern. Sie können jedoch nicht die Spalten einer dynamischen SPS-Tabelle bearbeiten, da dadurch die Substruktur der einzelnen Elementinstanzen geändert würde.

Die Symbole zum Bearbeiten dynamischer Tabellen werden in der Symbolleiste angezeigt und können auch über das Menü **Authentic** aufgerufen werden.



Zeile anhängen



Zeile einfügen



Zeile kopieren (d.h. der Inhalt einer Zelle wird dupliziert)



Zeile nach oben



Zeile nach unten



Zeile löschen

Bitte beachten Sie: Diese Befehle sollten nur bei **dynamischen SPS-Tabellen** und nicht innerhalb von statischen SPS-Tabellen verwendet werden. Die verschiedenen Arten der in der Authentic-Ansicht verwendeten Tabellen werden weiter hinten im Abschnitt [Tabellen in der Authentic-Ansicht](#) näher beschrieben.

Erstellen und Bearbeiten von XML-Tabellen

Sie können Ihre eigenen Tabellen einfügen, wenn Sie Daten in Tabellenform darstellen möchten. Solche Tabellen werden als XML-Tabellen eingefügt. Sie können die Struktur einer XML-Tabelle bearbeiten und die Tabelle formatieren. Die Symbole dazu stehen in der Symbolleiste zur Verfügung (siehe Abbildung unten). Im Abschnitt [XML-Tabellenbearbeitungsbefehle](#) finden Sie nähere Informationen zu diesen Befehlen.



Die Befehle zu diesen Symbolen **stehen nicht als Menübefehle** zur Verfügung. Beachten Sie außerdem, dass die entsprechende Funktion erst im StyleVision Power Stylesheet aktiviert und konfiguriert werden muss, damit Sie XML-Tabellen verwenden können.

Eine ausführliche Beschreibung der in der Authentic-Ansicht verwendeten Tabellen bzw. wie diese erstellt und bearbeitet werden, finden Sie weiter hinten in der Dokumentation im Abschnitt [Tabellen in der Authentic-Ansicht](#).

Symbole zur Textformatierung

Text wird in der Authentic-Ansicht formatiert, indem man darauf ein XML-Element oder -Attribut anwendet, das die gewünschte Formatierung aufweist. Wurde eine solche Formatierung definiert, kann der Autor des StyleVision Power Stylesheet in der Authentic-Ansicht Symbole zur

Verfügung stellen, mit denen die Formatierung angewendet wird.

Um eine Textformatierung mit Hilfe eines Textformatierungssymbols anzuwenden, markieren Sie den gewünschten Text und klicken Sie auf das entsprechende Symbol.

Symbole für die Navigation in DB-Zeilen



Die Pfeilsymbole von links nach rechts sind: Gehe zum ersten Datensatz in der DB; Gehe zum vorhergehenden Datensatz; Dialogfeld "Gehe zu Datensatz #"; Gehe zum nächsten Datensatz; und Gehe zum letzten Datensatz.



Mit diesem Symbol wird das Dialogfeld "Datenbankabfrage bearbeiten" aufgerufen. Hier können Sie eine Abfrage eingeben. In der Authentic-Ansicht werden die abgefragten Datensätze angezeigt.

3.3 Authentic-Ansicht Hauptfenster

In der Authentic-Ansicht gibt es vier Ansichtsmodi: Große Markup-Symbole; Kleine Markup-Symbole; Mixed Markup-Symbole; und Alle Markup-Symbole ausblenden. Damit können Sie die Markup-Informationen für Dokumente mit unterschiedlicher Genauigkeit anzeigen.

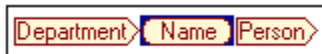
Um zwischen diesen Ansichtsmodi zu wechseln, verwenden Sie die Befehle im Menü "Authentic" oder die Symbole der Symbolleiste (siehe vorhergehender Abschnitt, [Authentic-Ansicht Symbolleistenschaltflächen](#)).

Große Markup-Symbole

In diesem Modus werden die Start- und End-Tags von Elementen und Attributen mit dem Namen des Elements/Attributs im Tag angezeigt:

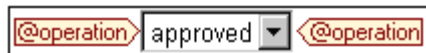


Das Element `Name` in der Abbildung oben ist **erweitert**, d.h. sein Start- und End-Tag sowie der Inhalt der Elements werden angezeigt. Durch Doppelklicken auf den Start- oder End-Tag eines Elements/Attributs können Sie dieses **reduzieren**.



Um das reduzierte Element/Attribut zu erweitern, doppelklicken Sie auf den reduzierten Tag.

Im Modus für große Markup-Symbole werden Attribute durch das Symbol @ im Start- und End-Tag gekennzeichnet:



Kleine Markup-Symbole

In diesem Modus werden die Start- und End-Tags von Elementen/Attributen ohne Namen angezeigt:

▶Nanonull, Inc.◀

Location: ▶US◀

Street: ▶▶119 Oakstreet, Suite 4876◀◀	Phone: ▶+1 (321) 555 5155 0◀
City: ▶▶Vereno◀◀	Fax: ▶+1 (321) 555 5155 4◀
State & Zip: ▶▶DC◀◀ 29213	E-mail: ▶office@nanonull.com◀

▶▶Vereno◀◀ Office Summary: 4 departments, 15 employees.▶▶

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

◀◀

Um ein Element/Attribut zu reduzieren bzw. zu erweitern, doppelklicken Sie auf den entsprechenden Tag.

Mixed Markup-Symbole

Im Mixed Markup-Modus wird eine benutzerdefinierte Markup-Ansicht angezeigt. Der Autor des StyleVision Power Stylesheet kann für einzelne Elemente/Attribute im Dokument große, kleine oder gar keine Markup-Symbole einstellen. Der Benutzer der Authentic-Ansicht sieht diese benutzerdefinierte Ansicht im "Mixed Markup"-Ansichtsmodus.

Alle Markup-Symbole ausblenden

In diesem Modus werden alle XML-Markup-Symbole ausgeblendet. Da die Formatierung in der Authentic-Ansicht die Formatierung des gedruckten Dokuments ist, handelt es sich bei diesem Modus um eine WYSIWYG-Ansicht des Dokuments.

Anzeige von Inhalt

In der Authentic-Ansicht wird Inhalt auf zwei Arten angezeigt:

- Reiner Text. Sie geben den Text ein und dieser Text wird der Inhalt des Elements bzw. der Wert des Attributs.

Department Name
Enter the department
 Name Person

- Dateneingabe-Hilfen. Dabei handelt es sich entweder um ein Eingabefeld (Textfeld), ein Eingabefeld mit mehreren Zeilen, eine Dropdown-Liste, ein Kontrollkästchen oder ein Optionsfeld. Text, den Sie in Eingabefelder und Eingabefelder mit mehreren Zeilen eingeben, wird der XML-Inhalt des Elements bzw. der Wert des Attributs.

documentid documentid

Bei den anderen Dateneingabe-Hilfen wird durch Ihre Auswahl der entsprechende

XML-Wert erzeugt, der im StyleVision Power Stylesheet definiert wurde. Wenn also im Beispiel unten "approved" ausgewählt wird, wird dieser Eintrag auf einen XML-Wert "1" oder auf "approved" oder einen beliebigen anderen Eintrag gemappt, während "not approved" dem Wert "0" oder "not approved" oder einem beliebigen anderen Wert zugeordnet wird.



Optionale Nodes

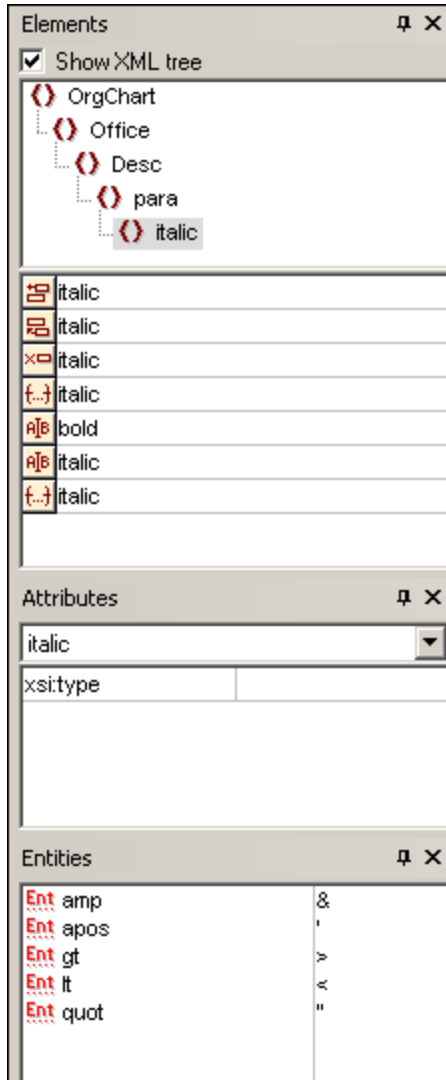
Wenn ein Element oder Attribut gemäß dem referenzierten Schema **optional** ist, wird eine Eingabeaufforderung vom Typ "add [*Element/Attribut*]" angezeigt.



Wenn Sie auf die Eingabeaufforderung klicken, wird das Element hinzugefügt und der Cursor so gesetzt, dass Daten eingegeben werden können. Wenn es mehrere optionale Nodes gibt, wird die Aufforderung "add..." angezeigt. Wenn Sie darauf klicken, wird ein Menü mit den optionalen Nodes angezeigt.

3.4 Authentic-Ansicht Eingabehilfen

Die Authentic-Ansicht verfügt über drei Eingabehilfen: für Elemente, Attribute und Entities. Sie werden als Fenster am rechten Rand der Authentic-Ansicht angezeigt.



Die Element- und Attribut-Eingabehilfen sind kontextsensitiv, d.h. was in der Eingabehilfe angezeigt wird, hängt davon ab, an welcher Stelle im Dokument der Cursor sich befindet. Die in der Entities-Eingabehilfe angezeigten Entities sind nicht kontextsensitiv; es werden - unabhängig von der Cursorposition - alle im Dokument zulässigen Entities angezeigt.

Im Folgenden werden die einzelnen Eingabehilfen beschrieben.

Element-Eingabehilfe

Die Element-Eingabehilfe besteht aus zwei Teilen:

- dem oberen Teil mit einer XML-Baumstruktur, der über das Kontrollkästchen **Show xml tree** ein- und ausgeblendet werden kann. In der XML-Baumstruktur sehen Sie die übergeordneten Nodes bis hinauf zum Root-Element des aktuellen Elements des Dokuments. Wenn Sie auf ein Element in der XML-Baumstruktur klicken, werden die dazugehörigen Elemente (wie in nächsten Punkt dieser Liste beschrieben) im unteren Teil der Element-Eingabehilfe angezeigt

- Im unteren Teil der Element-Eingabehilfe werden die Elemente aufgelistet, die im, vor und nach dem ausgewählten Element/Textbereich in der Authentic-Ansicht eingefügt werden können, entfernt werden können oder darauf (d.h. durch Ersetzung) angewendet werden können. Was Sie mit einem in der Eingabehilfe aufgelisteten Element tun können, wird in der Eingabehilfe durch das Symbol auf der linken Seite des Elementnamens angezeigt. Im Folgenden finden Sie eine Liste der in der Element-Eingabehilfe vorkommenden Symbole mit einer Erklärung dazu.

Um einen Node aus der Eingabehilfe zu verwenden, klicken Sie auf sein Symbol.



Hinter Element einfügen

Das Element in der Eingabehilfe wird hinter dem ausgewählten Element eingefügt. Beachten Sie, dass es auf der richtigen hierarchischen Ebene angehängt wird. Wenn sich Ihr Cursor z.B. innerhalb eines `//sect1/para`-Elements befindet und Sie ein `sect1`-Element anhängen, wird das neue `sect1`-Element nicht als Geschwisterelement von `//sect1/para` angehängt, sondern als `sect1`-Element, das ein Parent dieses `para`-Elements ist.



Vor Element einfügen

Das Element in der Eingabehilfe wird vor dem ausgewählten Element eingefügt. Beachten Sie, dass das Element genau wie beim vorhergehenden Befehl auf der richtigen hierarchischen Ebene eingefügt wird.



Element entfernen

Löscht das Element und dessen Inhalt.



Element einfügen

Sie können ein Element aus der Eingabehilfe auch **innerhalb** eines Elements einfügen. Wenn Sie den Cursor in ein Element hineinsetzen, können die zulässigen Child-Elemente dieses Elements eingefügt werden. Beachten Sie, dass zulässige Child-Elemente sowohl Teil eines "elements-only Content Model" als auch Teil eines "Mixed Content Model" (Text plus Child-Elemente) sein können.

Ein zulässiges Child-Element kann entweder dann eingefügt werden, wenn ein Textbereich markiert wurde oder wenn der Cursor als Einfügepunkt in den Text platziert wurde.

- Wenn ein Textbereich ausgewählt und ein Element eingefügt wurde, wird der Textbereich der Inhalt des eingefügten Elements.
- Bei Auswahl einer Cursorposition wird das Element an dieser Stelle eingefügt

Nachdem das Element eingefügt wurde, kann es durch Klicken auf eines der beiden Symbole zum Löschen eines Elements (in der Element-Eingabehilfe) gelöscht werden. Welches der beiden Symbole angezeigt wird, ist abhängig davon, ob Sie einen Textbereich auswählen oder ob Sie den Cursor als Einfügepunkt in den Text setzen (siehe unten).



Element anwenden

Wenn Sie in Ihrem Dokument ein Element auswählen (indem Sie in der "Große Markup-Symbol anzeigen"-Ansicht entweder auf den Start- oder den End-Tag eines Elements klicken) und wenn dieses Element durch ein anderes Element ersetzt werden kann (z.B. kann in einem Element gemischten Inhalts wie `para`, ein `italic` Element durch das `bold` Element ersetzt werden), bedeutet dieses Symbol, dass das Element in der Eingabehilfe auf das ausgewählte (Original-)

Element angewendet werden kann. Der Befehl **Element anwenden** kann auch auf einen Textbereich innerhalb eines Elements gemischten Inhalts angewendet werden; der Textbereich wird als Inhalt des angewendeten Elements erstellt.

- Wenn das angewendete Element ein **Child-Element mit demselben Namen** wie das Child-Element des Originalelements hat und das Original-Element eine Instanz dieses Child-Elements enthält, dann wird das Child-Element des Originals im neuen Element-Inhalt beibehalten.
- Wenn das angewendete Element **kein Child-Element mit demselben Namen** wie das instantiierte Child-Element des Originalelements enthält, dann wird das instantiierte Child-Element des Originalelements als Geschwisterelement von etwaigen Child-Elementen oder Elementen des neuen Elements angehängt.
- Wenn das angewendete Element **ein Child-Element hat, für das es im Content Model des Originalelements kein Äquivalent gibt**, wird dieses Child-Element nicht direkt erstellt, doch bietet Ihnen die Authentic-Ansicht die Möglichkeit, dieses einzufügen.

Wenn Sie anstatt eines Elements einen Textbereich auswählen, wird bei Anwendung eines Elements auf diesen Bereich an dieser Stelle das angewendete Element erstellt, wobei der markierte Text dessen Inhalt wird. Wenn der Cursor sich an einem Einfügepunkt befindet, ist die Anwendung eines Elements nicht zulässig.



Element löschen (wenn ein Bereich ausgewählt ist)

Dieses Symbol erscheint, wenn Text innerhalb eines Elements gemischten Inhalts ausgewählt wird. Wenn Sie auf dieses Symbol klicken, wird das Element, das den ausgewählten Bereich einschließt, gelöscht.



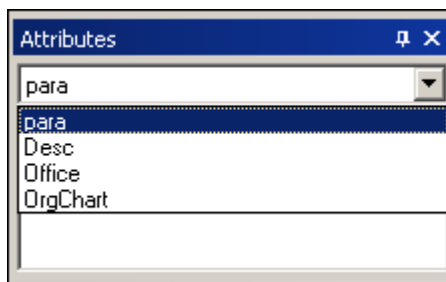
Element löschen (wenn ein Einfügepunkt ausgewählt ist)

Dieses Symbol erscheint, wenn der Cursor in ein Element platziert wird, das ein Child eines Elements gemischten Inhalts ist. Wenn Sie auf das Symbol klicken, wird dieses zeileninterne Element gelöscht.

Attribut-Eingabehilfe

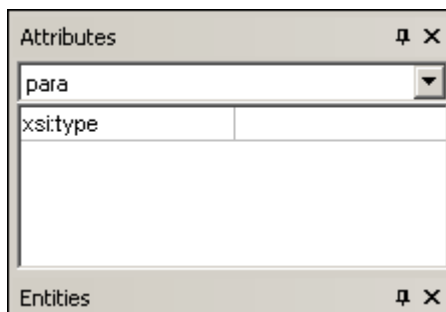
Die Attribut-Eingabehilfe besteht aus einem Listenfeld mit einer Dropdown-Liste von Attributen. Das ausgewählte Element (Sie können auf den Start- oder End-Tag klicken oder den Cursor irgendwo im Element-Inhalt platzieren) wird in der Dropdown-Liste angezeigt.

Die in den nachfolgenden Abbildungen gezeigte Attribut-Eingabehilfe hat ein `para`-Element in der Dropdown-Liste. Wenn Sie auf den Abwärtspfeil in der Liste klicken, erscheint eine Liste aller **übergeordneten Elemente** des `para`-Elements hinauf bis zum Root-Element, in diesem Fall OrgChart.



Unterhalb der Dropdown-Liste wird eine Liste gültiger Attribute für dieses Element - in diesem

Fall für das `para`-Element angezeigt. Wenn ein Attribut für ein bestimmtes Element obligatorisch ist, wird es fett angezeigt. (Im Beispiel unten gibt es keine obligatorischen Attribute)

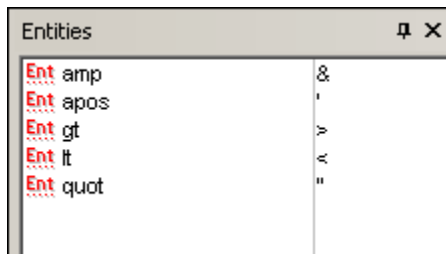


Um den Wert für ein Attribut einzugeben, klicken Sie in das Wertefeld des Attributs und geben Sie den Wert ein. Daraufhin wird das Attribut mit dem entsprechenden Wert im XML-Dokument erstellt.

Im Falle des Attributs `xsi:nil`, welches in der Eingabehilfe "Attribut" erscheint, wenn ein nillable Element ausgewählt wurde, kann der Wert des Attributs `xsi:nil` nur durch Auswahl eines der zulässigen Werte (`true` oder `false`) aus der Dropdown-Liste der Werte des Attributs eingegeben werden.

Entities-Eingabehilfe

Über die Entities-Eingabehilfe können Sie eine Entity in Ihr Dokument einfügen. Entities dienen zum Einfügen von Sonderzeichen oder Textfragmenten, die in einem Dokument öfter vorkommen (z.B. der Name einer Firma). Um eine Entity einzufügen, setzen Sie den Cursor an die gewünschte Stelle im Text und doppelklicken Sie in der Entities-Eingabehilfe auf die Entity.



Anmerkung:

Eine interne Entity ist eine Entity, deren Wert in der DTD definiert wurde. Eine externe Entity ist eine Entity, deren Wert in einer externen Quelle z.B. einer anderen XML-Datei enthalten ist. Sowohl interne als auch externe Entities werden in der Entities-Eingabehilfe aufgelistet. Beim Einfügen einer Entity - egal ob intern oder extern - wird die Entity und nicht ihr Wert in den XML-Text eingefügt. Wenn es sich um eine interne Entity handelt, wird in der Authentic-Ansicht **der Wert der Entity** angezeigt. Dies bedeutet, dass eine XML-Datei, die eine externe Entity ist, in der Authentic-Ansicht als Entity angezeigt wird; die Entity wird in der Authentic-Ansicht nicht durch ihren Inhalt ersetzt.

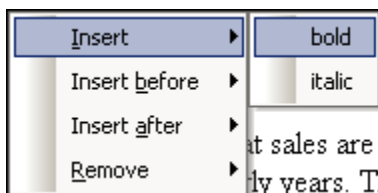
Sie können in der Authentic-Ansicht auch Ihre **eigenen Entities definieren**: Nähere Informationen dazu finden Sie im Abschnitt "Bearbeitung in der Authentic -Ansicht" unter [Definieren von Entities](#).

3.5 Authentic-Ansicht Kontextmenüs

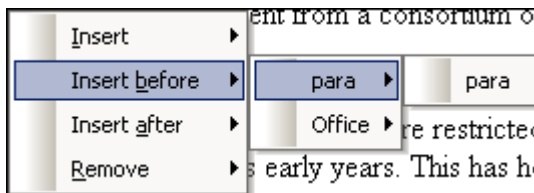
Wenn Sie mit der rechten Maustaste auf einen ausgewählten Dokumenteninhalte oder Node klicken, wird ein Kontextmenü mit Befehlen, die für die Auswahl oder Cursorposition relevant sind, angezeigt.

Einfügen von Elementen

In der Abbildung unten sehen Sie das Kontextmenü mit dem Untermenü **Einfügen**, das eine Liste aller Elemente enthält, die an der aktuellen Cursorposition eingefügt werden können. Im Untermenü **Einfügen vor** werden alle Elemente aufgelistet, die vor dem aktuellen Element eingefügt werden können. Im Untermenü **Einfügen nach** werden alle Elemente aufgelistet, die nach dem aktuellen Element eingefügt werden können. In der Abbildung unten ist das aktuelle Element das `para`-Element. Die Elemente *italic* und **bold** können innerhalb des aktuellen `para`-Elements eingefügt werden.



Die Elemente `para` und `Office` können vor dem `para`-Element eingefügt werden.



Die meisten der Befehle aus dem Kontextmenü werden unter [Authentic-Ansicht Eingabehilfen](#) beschrieben.

Entfernen eines Node

Wenn Sie den Mauszeiger über den Befehl **Entfernen** positionieren, erscheint eine Menülste bestehend aus dem ausgewählten Node und allen seinen übergeordneten Nodes bis hinauf zum Dokument-Element, die entfernt werden können (ohne dass das Dokument ungültig wird). Klicken Sie auf das Element, das Sie entfernen möchten. Auf diese Art lässt sich ein Element oder ein beliebiges übergeordnetes Element einfach entfernen. Beachten Sie, dass bei Entfernung eines übergeordneten Elements auch alle untergeordneten Elementen einschließlich des ausgewählten Elements entfernt werden.

Einfügen einer Entity

Wenn Sie den Cursor über den Befehl "Entity einfügen" positionieren, erscheint ein Untermenü mit einer Liste aller deklarierten Entities. Wenn Sie auf eine Entity klicken, wird sie am ausgewählten Node eingefügt. Eine Beschreibung, wie man Entities für das Dokument definiert, finden Sie unter [Definieren von Entities](#).

Einfügen eines CDATA-Abschnitts

Dieser Befehl ist aktiviert, wenn Sie den Cursor innerhalb von Text setzen. Wenn Sie auf den Befehl klicken, wird am Cursor-Punkt ein CDATA-Abschnitt eingefügt. Der CDATA-Abschnitt steht innerhalb eines Start- und End-Tags; um diese Tags anzuzeigen, aktivieren Sie die großen oder kleinen Markup-Symbole. Innerhalb von CDATA-Abschnitten werden XML-Markup-Symbole und Parsing ignoriert. XML-Markup-Zeichen (Zeichen wie das kaufmännische Und, Apostroph, größer als, kleiner als und Anführungszeichen) werden nicht

als Markup behandelt sondern als Literalzeichen. Daher eignen sich CDATA-Abschnitte für Text wie z.B. Auflistungen von Programmcode, welche XML-Markup-Zeichen enthalten.

Entfernen

Mit dem Befehl "Entfernen" können Sie den ausgewählten Node und seinen Inhalt entfernen. Ein Node wird dann als ausgewählt betrachtet, wenn Sie den Cursor in den Node platzieren oder auf den Start- oder End-Tag des Node klicken.

Löschen

Mit dem Befehl "Löschen" wird der Element-Markup-Code rund um die Auswahl gelöscht. Wenn der gesamte Node ausgewählt ist, wird der Element-Markup-Code für den gesamten Node gelöscht. Wenn ein Textabschnitt ausgewählt ist, wird der Element-Markup-Code nur rund um diesen Textabschnitt gelöscht.

4 Bearbeitung in der Authentic-Ansicht

In diesem Abschnitt werden wichtige Features der Authentic-Ansicht näher beschrieben. Bei den Features, die hier behandelt werden, handelt es sich entweder um häufig gebrauchte Funktionen oder solche, deren Funktionsweise und Konzepte näher erklärt werden müssen.

In diesem Abschnitt werden folgende Themen behandelt:

- Es gibt in der Authentic-Ansicht drei verschiedene Tabellenarten. Im Abschnitt [Tabellen in der Authentic-Ansicht](#) werden diese drei Tabellenarten (statisches SPS, dynamisches SPS und XML) näher erläutert und deren Verwendung wird erklärt. Die Beschreibung beginnt mit einem allgemeinen Überblick. Anschließend wird näher auf die Einzelheiten eingegangen.
- Die Datumsauswahl ist ein grafischer Kalender, der Daten im korrekten XML-Format eingibt, wenn Sie auf ein Datum klicken. Siehe [Datumsauswahl](#).
- Eine Entity ist ein Kürzel für ein Sonderzeichen oder einen Textstring. Sie können Ihre eigenen Entities definieren, um Sonderzeichen oder Textstrings durch Eingabe der Entity einfügen zu können. Nähere Informationen dazu finden Sie unter [Definieren von Entities](#).
- Die [Grafikformate](#), die in der Authentic-Ansicht angezeigt werden können.

4.1 Grundlegendes zur Bearbeitung

Bei der Bearbeitung in der Authentic-Ansicht wird ein XML-Dokument bearbeitet. In der Authentic-Ansicht kann der Markup-Code, der die Struktur des XML-Dokuments definiert, jedoch ausgeblendet werden, sodass nur mehr der Inhalt des Dokuments angezeigt wird (*erste Abbildung unten*). Sie müssen sich daher nicht mit dem technischen Hintergrund von XML auseinandersetzen und können das Dokument wie ein normales Textdokument bearbeiten. Bei Bedarf können Sie den Markup-Code während der Bearbeitung jederzeit einblenden (*zweite Abbildung unten*).

Vereno Office Summary: 4 departments, 16 employees.

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

Ein editierbares Dokument in der Authentic-Ansicht ohne XML-Markup-Tags

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

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

para para

Ein editierbares Dokument in der Authentic-Ansicht mit XML-Markup-Tags

Bearbeitung von Text

Ein Dokument in der Authentic-Ansicht besteht im Wesentlichen aus Text und Bildern. Um den Text im Dokument zu bearbeiten, platzieren Sie den Cursor an die Stelle im Text, an der Sie den Text einfügen möchten und geben Sie den Text ein. Sie können Text mit den gängigen Tastenkombinationen (wie z.B. der Entf-Taste) kopieren, verschieben und löschen und Drag-and-Drop-Mechanismen verwenden. Eine Ausnahme ist die **Eingabetaste**. Da das Dokument in der Authentic-Ansicht vorformatiert ist, werden und können zwischen Einträgen keine Zeilen oder Leerzeichen eingefügt werden. Die **Eingabetaste** in der Authentic dient daher dazu, eine

weitere Instanz des gerade bearbeiteten Elements einzufügen und sollte ausschließlich zu diesem Zweck verwendet werden.

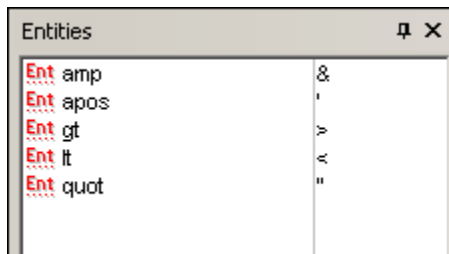
Formatierung von Text

Eines der grundlegenden Prinzipien von XML-Dokumentsystemen ist, dass Inhalt und Darstellung separat gehalten werden. Das XML-Dokument enthält den Inhalt, während das Stylesheet die Darstellung (Formatierung) enthält. In der Authentic-Ansicht wird das XML-Dokument mit Hilfe eines Stylesheet dargestellt, d.h. die gesamte Formatierung, die Sie in der Authentic-Ansicht sehen, wird vom Stylesheet erzeugt. Wenn Sie Text fett angezeigt sehen, wurde die Fettformatierung vom Stylesheet vorgegeben. Wenn Sie eine Liste oder Tabelle sehen, wurde diese Darstellung vom Stylesheet vorgegeben. Das XML-Dokument, das Sie in der Authentic-Ansicht bearbeiten, enthält nur den Inhalt und keinerlei Formatierungen. Die Formatierung ist im Stylesheet definiert. Für Sie als Benutzer der Authentic-Ansicht bedeutet dies, dass Sie den von Ihnen bearbeiteten Text nicht formatieren müssen oder können. Sie bearbeiten nur den Inhalt. Die Formatierung, die automatisch auf den Inhalt angewendet wird, ist mit dem semantischen und/oder strukturellen Wert der Daten verknüpft, die Sie bearbeiten. So erhält beispielsweise eine E-Mail-Adresse (die als semantische Einheit betrachtet werden könnte) automatisch eine bestimmte Formatierung, weil es sich um eine E-Mail-Adresse handelt. Ebenso muss eine Überschrift (sowohl eine strukturelle als auch eine semantische Einheit) an einer bestimmten Stelle im Dokument vorkommen und wird automatisch so formatiert, wie dies der Designer des Stylesheet für Überschriften definiert hat. Sie können weder die Formatierung der E-Mail-Adresse noch die der Überschrift ändern. Sie ändern nur den Inhalt der E-Mail-Adresse oder der Überschrift.

In manchen Fällen muss Inhalt speziell formatiert werden; z.B. ein Textstring der fett gedruckt werden soll. In allen diesen Fällen muss die Darstellung mit einem Strukturelement des Dokuments verknüpft sein. So wird z.B. ein Textstring, der fett dargestellt werden soll, strukturell vom restlichen Inhalt durch Markup-Code getrennt, dem der Stylesheet Designer Fettformatierung zuweisen wird. Wenn Sie als Benutzer der Authentic-Ansicht einen solchen Textstring verwenden möchten, müssen Sie diesen in den entsprechenden Element Markup-Code einschließen. Nähere Informationen darüber finden Sie in dieser Dokumentation im Abschnitt [Element-Eingabehilfe](#) unter dem Befehl "Element einfügen".

Einfügen von Entities

Einige Zeichen sind in XML-Dokumenten für Markup-Tags reserviert und können in normalem Text nicht verwendet werden. Dies sind die Zeichen Kaufmännisches Und (&), Apostroph ('), kleiner als (<), größer als (>) und doppeltes Anführungszeichen ("). Wenn Sie diese Zeichen in Ihren Daten verwenden möchten, müssen Sie sie als Entity-Referenzen über die [Eingabehilfe "Entities"](#) (Abbildung unten) einfügen.



XML bietet auch die Möglichkeit benutzerdefinierte Entities zu erstellen. Dabei kann es sich um Folgendes handeln: (i) Sonderzeichen, die auf Ihrer Tastatur nicht zur Verfügung stehen, (ii) Textstrings, die Sie in Ihrem Dokumentinhalt wiederverwenden möchten, (iii) XML-Datenfragmente oder (iv) andere Ressourcen, wie z.B. Bilder. Sie können in der Applikation Authentic-Ansicht [Ihre eigenen Entities definieren](#). Nachdem Sie diese definiert haben, werden die Entities in der [Eingabehilfe "Entities"](#) angezeigt und können dann in das Dokument eingefügt

werden.

Einfügen von CDATA-Abschnitten

CDATA-Abschnitte sind Textabschnitte in einem XML-Dokument, die der XML-Parser nicht als XML-Daten verarbeitet. Mit Hilfe dieser Abschnitte können Sie große Textabschnitte auslassen, wenn die Ersetzung von Sonderzeichen durch Entity-Referenzen nicht wünschenswert ist; ein Beispiel wäre Programmcode oder ein XML-Fragment, das mit seinen Markup-Tags wiedergegeben werden soll. CDATA-Abschnitte können innerhalb von Elementinhalt vorkommen und werden am Beginn und Ende von `<![CDATA[ bzw. ] ]>` eingeschlossen. Folglich sollte der Textstring `] ]>` nicht in einem CDATA-Abschnitt vorkommen, da damit das Ende des Abschnitts gekennzeichnet würde. In diesem Fall kann das Zeichen "größer als" durch ihre Entity-Referenz (`>`) dargestellt werden. Um einen CDATA-Abschnitt in ein Element einzufügen, platzieren Sie den Cursor an der gewünschten Stelle, rechtsklicken Sie und wählen Sie im Kontextmenü den Befehl **CDATA-Abschnitt einfügen**. Um die CDATA-Abschnittstags in der Authentic-Ansicht anzuzeigen, [wechseln Sie in die Markup-Anzeige](#). Alternativ dazu können Sie den Text markieren, der in einen DATA-Abschnitt eingeschlossen werden soll, und den Befehl **CDATA-Abschnitt einfügen** wählen.

Bearbeiten von Links und Navigieren zum Ziel eines Links

Ein Hyperlink besteht aus zwei Teilen: dem Link-Text und dem Ziel des Links. Sie können den Link-Text bearbeiten, indem Sie auf den Text klicken und ihn ändern. Sie können jedoch das Ziel des Links nicht bearbeiten. (Das Ziel des Links wird vom Designer des Stylesheet definiert (entweder durch Eingabe einer statischen Zieladresse oder durch Abrufen der Zieladresse aus Daten im XML-Dokument). In der Authentic-Ansicht können Sie zum Ziel des Links navigieren, indem Sie die Strg-Taste drücken und auf den Link-Text klicken. (Beachten Sie: Wenn Sie nur auf den Link klicken, können Sie den Link-Text nur bearbeiten.)

4.2 Tabellen in der Authentic-Ansicht

Die drei Tabellentypen fallen in zwei Kategorien: SPS-Tabellen (statische und dynamische) und XML-Tabellen.

SPS-Tabellen: Es gibt zwei Arten von SPS-Tabellen: statische und dynamische. SPS-Tabellen werden vom Autor des StyleVision Power Stylesheet, mit dem Ihr XML-Dokument verknüpft ist, erstellt. Sie können selbst keine SPS-Tabelle in das XML-Dokument einfügen, doch können Sie Daten in die Felder der SPS-Tabelle eingeben und Zeilen in dynamischen SPS-Tabellen hinzufügen und löschen. Im nachfolgenden Abschnitt zu [SPS-Tabellen](#) werden die Funktionen dieser Tabellen näher beschrieben.

XML-Tabellen werden vom Benutzer der Authentic-Ansicht eingefügt. Auf diese Art können Sie auf Wunsch an jeder Stelle im Dokument, an der dies zulässig ist, Tabellen einfügen. Die Editierfeatures von [XML-Tabellen](#) und die [Bearbeitungssymbole für XML-Tabellen](#) werden im Folgenden beschrieben.

4.2.1 SPS-Tabellen

In der Authentic-Ansicht werden zwei Arten von SPS-Tabellen verwendet: statische Tabellen und dynamische Tabellen.

Statische Tabellen sind von der Struktur und der Art des Inhalts der Zellen unveränderbar. Als Benutzer der Authentic-Ansicht können Sie Daten in die Tabellenzellen eingeben, die Struktur der Tabellen (d.h. Hinzufügen von Zeilen oder Spalten usw.) oder den Inhaltstyp einer Zelle jedoch nicht verändern. Die Eingabe von Daten erfolgt entweder durch Eingabe von Text oder durch Auswahl von Optionen in Form von Kontrollkästchen, Optionsfeldern oder Dropdown-Listen. Nachdem Sie Daten eingegeben haben, können Sie diese bearbeiten.

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

Bitte beachten Sie: Die Symbole oder Befehle zum Bearbeiten dynamischer Tabellen **dürfen nicht** zum Bearbeiten statischer Tabellen verwendet werden.

Dynamische Tabellen enthalten Zeilen, die eine sich wiederholende Datenstruktur darstellen, d.h. jede Zeile weist dieselbe Datenstruktur auf (Dies ist bei statischen Tabellen nicht der Fall). Sie können daher Zeilen bearbeiten, also Zeilen anhängen, einfügen, nach oben oder nach unten verschieben und löschen. Diese Befehle stehen im Menü **Authentic** und als Symbole in der Symbolleiste (siehe Abb. unten) zur Verfügung.



Um diese Befehle zu verwenden, platzieren Sie den Cursor an eine beliebige Stelle in der gewünschten Zeile und wählen Sie den entsprechenden Befehl aus.

Administration

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

Zum Navigieren innerhalb der Zellen einer Tabelle verwenden Sie die Pfeiltasten der Tastatur (nach oben, nach unten, nach links, nach rechts). Um von einer Zelle zur nächsten zu springen, drücken Sie die Tabulatortaste. Wenn Sie in der letzten Zelle einer Zeile die Tabulatortaste

drücken, wird eine neue Zeile erstellt.

4.2.2 XML-Tabellen

XML-Tabellen können von Ihnen, dem Benutzer der Authentic-Ansicht verwendet werden. Sie können damit Tabellen überall im XML-Dokument, wo dies zulässig ist, einfügen. Diese Funktion eignet sich zum Einfügen von Tabellendaten in Ihr Dokument. Diese Tabellen werden als Tabellen gedruckt, wenn Sie den Druckauftrag direkt von der Authentic-Ansicht aus geben. Wenn Sie Ausgabedokumente auch mit Hilfe von XSLT Stylesheets erzeugen, besprechen Sie die gewünschte Ausgabe mit dem Autor des StyleVision Power Stylesheet.

Beachten Sie bitte, dass XML-Tabellen nur an bestimmten Stellen, an denen dies zulässig ist, eingefügt werden können. Dies ist im Schema (DTD oder XML Schema) definiert. Wenn Sie Tabellen an anderen Stellen einfügen möchten, muss dies mit dem für das StyleVision Power Stylesheet Verantwortlichen besprochen werden.

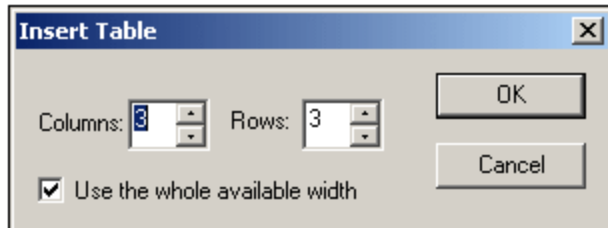
Arbeiten mit XML-Tabellen

Die Arbeit mit XML-Tabellen erfordert drei Schritte: Einfügen der Tabelle, Formatierung der Tabelle und Eingabe von Daten. Die Befehle für die Arbeit mit XML-Tabellen stehen in Form von Symbolen in der Symbolleiste zur Verfügung (siehe [Symbole zur Bearbeitung von XML-Tabellen](#)).

Einfügen von Tabellen

So fügen Sie eine XML-Tabelle ein:

1. Setzen Sie Ihren Cursor an die Stelle, an der die Tabelle eingefügt werden soll und klicken Sie auf das Symbol . (Beachten Sie dass XML-Tabellen nur an bestimmten im Schema definierten Stellen eingefügt werden können). Das Dialogfeld "Tabelle einfügen" (siehe unten) wird nun angezeigt.



2. Wählen Sie die Anzahl der Spalten und Zeilen aus und geben Sie an, ob die Tabelle sich über die gesamte Breite erstrecken soll. Daraufhin wird anhand der Angaben oben die folgende Tabelle erstellt.

Sie können später Spalten hinzufügen und löschen, Zeilen erstellen und Spalten zusammenfügen. Erstellen Sie daher zuerst die grundlegende Struktur.

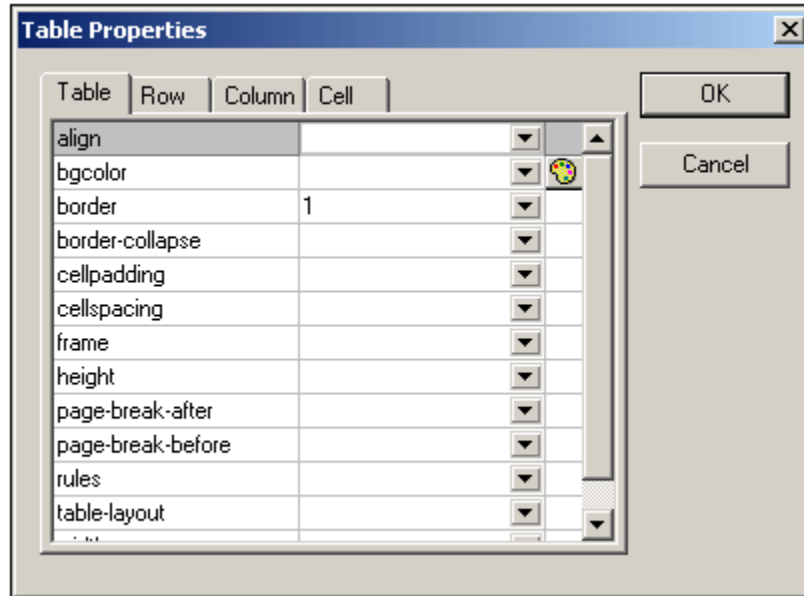
Bitte beachten Sie: Alle Änderungen an der Tabellenstruktur müssen mit Hilfe der Befehle des Tabellenmenüs vorgenommen werden. Sie können nicht durch Änderung von Attributwerten in der Attribut-Eingabehilfe gemacht werden.

Formatieren von Tabellen und Eingabe von Daten

So formatieren Sie Ihre Tabelle:

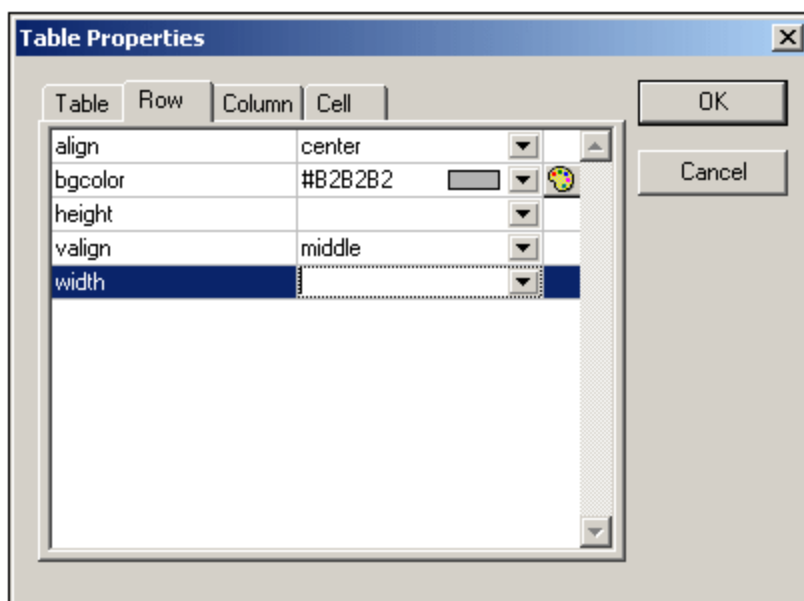
1. Platzieren Sie Ihren Cursor zum Formatieren der Tabelle an eine beliebige Stelle in der

Tabelle und klicken Sie auf das Symbol  (Tabelleneigenschaften). Daraufhin erscheint das Dialogfeld "Tabelleneigenschaften" (siehe unten), in dem Sie die Formatierung für die Tabelle, oder für einzelne Zeilen, Spalten oder Zellen definieren können.



2. Setzen Sie die Eigenschaften "cellspacing" und "cellpadding" auf "0". Ihre Tabelle wird nun folgendermaßen aussehen:

3. Setzen Sie nun den Cursor in die erste Zeile, um diese zu formatieren und klicken Sie auf das Symbol  (Tabelleneigenschaften). Klicken Sie auf das Register "Row".



Da es sich bei der ersten Zeile um die Kopfzeile handelt, definieren Sie eine Hintergrundfarbe, um diese Zeile von den anderen Zeilen zu unterscheiden. Beachten Sie dass die Zeileneigenschaften in der Abbildung oben definiert wurden. Geben Sie anschließend den Text für die Spaltenüberschriften ein. Ihre Tabelle sieht nun folgendermaßen aus:

Name	Telephone	Email

4. Beachten Sie, dass die Ausrichtung wie angegeben mittig (centered) ist. Angenommen, Sie möchten nun die Spalte "Telephone" in die Spalten "Office" und "Home" unterteilen. Um dies zu tun, müssen Sie Zellen teilen. Setzen Sie die Cursor in die Zelle "Telephone" und klicken Sie auf das Symbol  (Vertikal teilen). Ihre Tabelle sieht nun aus wie folgt:

Name	Telephone	Email

5. Setzen Sie nun den Cursor in die Zelle unterhalb der Zelle mit dem Eintrag "Telephone" und klicken Sie auf das Symbol  (Horizontal teilen). Geben Sie dann die Spaltenüberschriften "Office" und "Home" ein. Ihre Tabelle sieht nun folgendermaßen aus:

Name	Telephone	Email
	Office	Home

Nun müssen Sie die einzelnen Zellen in der Spalte "Telephone" vertikal teilen.

Mit Hilfe der Symbole zum Bearbeiten von Tabellen können Sie des Weiteren Spalten und Zeilen hinzufügen und löschen und Zellinhalte vertikal ausrichten. Die Symbole zum Bearbeiten von XML-Tabellen sind in der Benutzerreferenz im Abschnitt "XML-Tabellensymbole" näher beschrieben.

Navigieren innerhalb von Tabellenzellen

Um innerhalb der Zellen der XML-Tabelle navigieren zu können, verwenden Sie die Pfeiltasten der Tastatur.

Eingabe von Daten in eine Zelle

Um Daten in eine Zelle einzugeben, setzen Sie den Cursor in eine Zelle und geben Sie die Daten ein.

Formatieren von Text

Text in einer XML-Tabelle muss wie anderer Text im XML-Dokument mit Hilfe von XML-Elementen oder Attributen formatiert werden. Um ein Element hinzuzufügen, markieren Sie den Text und doppelklicken Sie auf das gewünschte Element in der Element-Eingabehilfe. Um einen Attributwert zu definieren, setzen Sie den Cursor in das Textfragment und geben Sie den erforderlichen Attributwert in der Attribut-Eingabehilfe ein. Nachdem Sie den Kopfzeilentext fett formatiert haben, sieht Ihre Tabelle folgendermaßen aus:

Name	Telephone		Email
	Office	Home	

Der Text in der Abbildung oben wurde durch Anklicken und Doppelklick auf das Element `strong` formatiert, für das eine globale Vorlage vorhanden ist, in der als Schriftbreite "Fett" definiert ist. Die Textformatierung wird sofort sichtbar.

Bitte beachten Sie: Damit die Textformatierung in der Authentic-Ansicht angezeigt wird, muss in StyleVision für das entsprechende Element eine globale Vorlage mit der erforderlichen Textformatierung erstellt worden sein.

4.3 Bearbeiten einer DB

In der Authentic-Ansicht können Sie Datenbank (DB)-Tabellen bearbeiten und die Daten in einer DB speichern. Dieser Abschnitt enthält eine ausführliche Beschreibung der Funktionen, die Ihnen zum Bearbeiten einer DB-Tabelle zur Verfügung stehen. Beachten Sie bitte die folgenden Punkte:

- Die Anzahl der Datensätze in einer DB-Tabelle, die in der Authentic-Ansicht angezeigt werden können, wurde unter Umständen vom Autor des StyleVision Power Stylesheet absichtlich eingeschränkt, um das Design kompakter zu gestalten. In diesem Fall wird anfangs nur diese beschränkte Anzahl an Datensätzen in der Authentic-Ansicht geladen. Sie können die restlichen Datensätze der DB-Tabelle mit Hilfe der Symbole zur Navigation in DB-Tabellenzeilen (siehe [Navigieren in einer DB-Tabelle](#)) laden und anzeigen.
- Sie können eine [Datenbankabfrage](#) durchführen, um bestimmte Datensätze anzuzeigen.
- Sie können Datensätze hinzufügen, ändern und löschen und Ihre Änderungen in der DB speichern. Siehe [Ändern einer DB-Tabelle](#).

So öffnen Sie ein DB-basiertes StyleVision Power Stylesheet in der Authentic Ansicht:

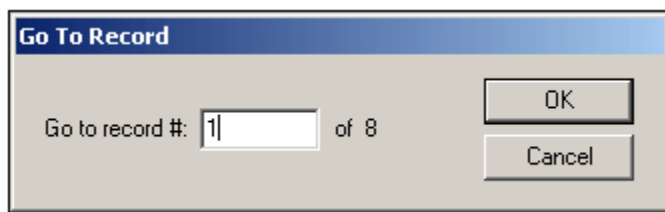
- Klicken Sie auf **Authentic | Datenbank bearbeiten**, und suchen Sie nach dem gewünschten StyleVision Power Stylesheet.

4.3.1 Navigieren in einer DB-Tabelle

Die Befehle zum Navigieren in den Zeilen einer DB-Tabelle stehen in Form von Schaltflächen im Authentic-Ansichtsdokument zur Verfügung. Normalerweise gibt es zu jeder DB-Tabelle einen Navigationsbereich mit vier oder fünf Schaltflächen.



Die Pfeilsymbole von links nach rechts sind: Gehe zum ersten Datensatz in der DB; Gehe zum vorhergehenden Datensatz; Dialogfeld "Gehe zu Datensatz #" öffnen (*siehe Abbildung unten*); Gehe zum nächsten Datensatz; und Gehe zum letzten Datensatz.



Um durch eine DB-Tabelle zu navigieren, klicken Sie auf die gewünschte Schaltfläche.

XML-Datenbanken

Im Fall von XML-Datenbanken wie z.B. IBM DB2 enthält eine Zelle (oder Zeile) ein einziges XML-Dokument. Daher wird immer nur eine einzige Zeile auf einmal in die Authentic-Ansicht geladen. Um ein XML-Dokument zu laden, das sich in einer anderen Zeile befindet, verwenden Sie die Menübefehl [Authentic | Neue Zeile mit XML-Daten für Bearbeitung auswählen](#).

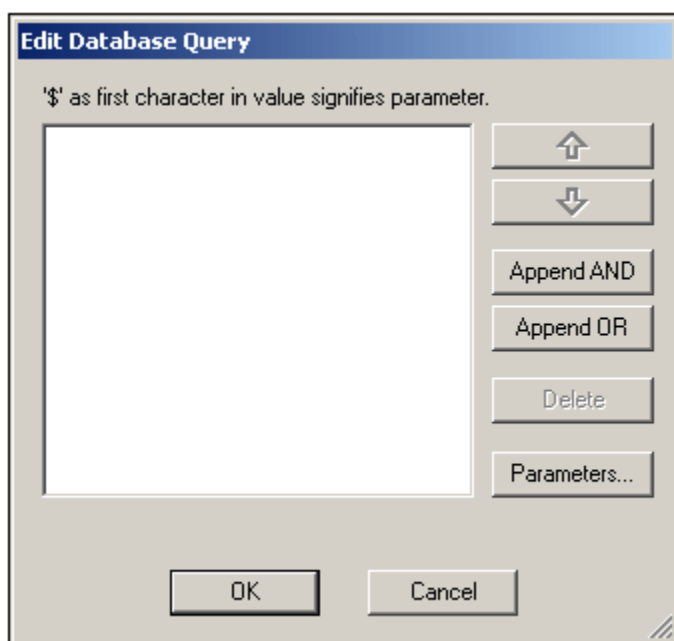
4.3.2 DB-Abfragen

Mit Hilfe einer DB-Abfrage können Sie die Datensätze einer in der Authentic-Ansicht angezeigten Tabelle abfragen. Eine Abfrage wird für eine einzelne Tabelle erstellt. Es kann für jede Tabelle nur eine Abfrage erstellt werden. Sie können während der Bearbeitung eines Dokuments jederzeit eine Abfrage erstellen. Wenn das in der Authentic-Ansicht angezeigte Dokument zu dem Zeitpunkt, zu dem Sie die Abfrage absenden, nicht gespeicherte Änderungen enthält, werden Sie gefragt, ob **alle** Änderungen im Dokument gespeichert oder verworfen werden sollen. Beachten Sie, dass auch an anderen Tabellen vorgenommene Änderungen gespeichert/verworfen werden. Nach Absenden der Abfrage wird die Tabelle mit den aktuellen Abfragekriterien erneut geladen.

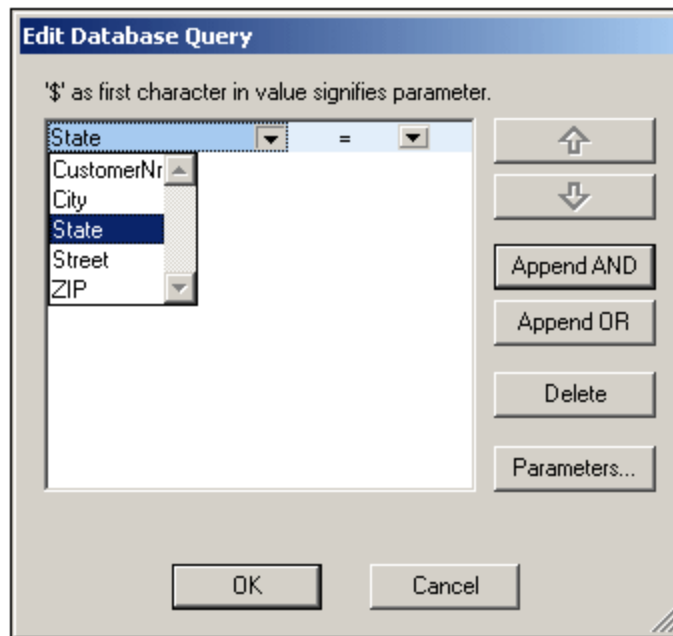
Bitte beachten Sie: Falls eine Meldung angezeigt wird, dass zu viele Tabellen geöffnet sind, können Sie die Anzahl der offenen Tabellen mit Hilfe einer Abfrage durch Filtern reduzieren.

So erstellen Sie eine Abfrage und senden diese ab:

1. Klicken Sie auf die Schaltfläche "Abfrage"  für die entsprechende Tabelle, um das Dialogfeld "Datenbankabfrage bearbeiten" aufzurufen (*siehe Abbildung unten*). Diese Schaltfläche erscheint normalerweise am oberen Rand einer DB-Tabelle oder unterhalb davon. Wenn keine Abfrage-Schaltfläche vorhanden ist, hat der Autor des StyleVision Power Stylesheet die DB-Abfragefunktion für diese Tabelle nicht aktiviert.



2. Klicken Sie auf die Schaltfläche **AND anhängen** oder **OR anhängen**. Daraufhin wird ein leeres Abfragekriterium (siehe unten) angehängt.



4. Geben Sie den Ausdruck für das Kriterium ein. Ein Ausdruck besteht aus: (i) einem Feldnamen (der in der dazugehörigen Dropdown-Liste zur Verfügung steht); (ii) einem Operator (der in der zugehörigen Dropdown-Liste zur Verfügung steht); und (iii) einem Wert (direkt einzugeben). Nähere Informationen zum Erstellen von Ausdrücken finden Sie im Abschnitt [Ausdrücke in Kriterien](#) weiter unten.
5. Wenn Sie ein weiteres Kriterium hinzufügen möchten, klicken Sie - je nachdem, welchen logischen Operator (AND oder OR) Sie verwenden wollen, um Ihre Kriterien zu verknüpfen - auf die Schaltfläche **AND anhängen** bzw. **OR anhängen**. Fügen Sie nun das neue Kriterium hinzu. Nähere Informationen zu logischen Operatoren finden Sie im Abschnitt [Neuordnen von Kriterien in DB-Abfragen](#).

Ausdrücke in Kriterien

Ausdrücke in DB-Abfragekriterien bestehen aus einem Feldnamen, einem Operator und einem Wert. Als **Feldnamen** stehen die Child-Elemente der obersten Ebene der Datentabelle zur Verfügung; die Namen dieser Felder werden in der Dropdown-Liste aufgelistet (*siehe Abbildung oben*). Unterhalb sehen Sie eine Liste der verfügbaren **Operatoren**:

=	Ist gleich
<>	Ist nicht gleich
<	Kleiner als
<=	Kleiner oder gleich
>	Größer als
>=	Größer oder gleich
LIKE	Phonetisch ähnlich
NOT LIKE	Phonetisch nicht ähnlich
IS NULL	Ist leer
NOT NULL	Ist nicht leer

Wenn IS NULL oder NOT NULL ausgewählt ist, ist das Wertefeld deaktiviert. **Werte** müssen ohne Anführungszeichen (oder andere Begrenzungszeichen) eingegeben werden. Außerdem müssen Werte dieselbe Formatierung wie die des entsprechenden DB-Felds haben, sonst ist das Ergebnis des Ausdrucks FALSE. Wenn z.B. ein Kriterium für ein Feld des date-Datentyps in einer MS Access DB einen Ausdruck `StartDate=25/05/2004` enthält, ist das Ergebnis der Abfrage FALSE, da der date-Datentyp in einer MS Access DB das Format `JJJJ-MM-TT` hat.

Verwendung von Parametern mit DB-Abfragen

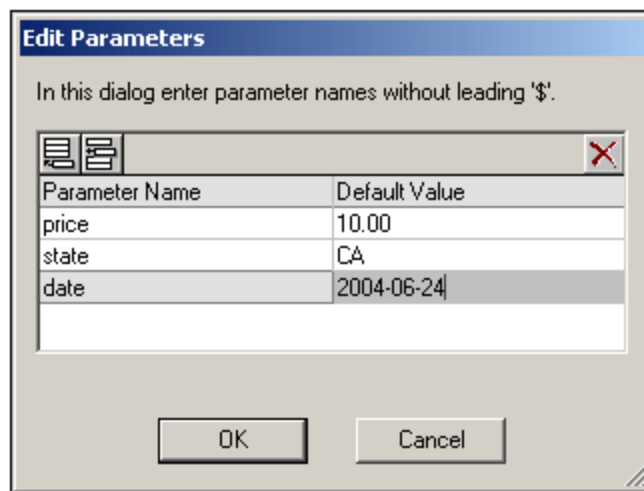
Sie können beim Erstellen von Abfragen den Namen eines **Parameters** als Wert eines Ausdrucks eingeben. Parameter sind Variablen, die anstelle von Literal-Werten in Abfragen verwendet werden können. Zuerst müssen Sie den Parameter und seinen Wert deklarieren und ihn anschließend in einem Ausdruck verwenden. Der Wert des Parameters wird als der Wert dieses Ausdrucks verwendet. Bei den Parametern, die Sie im Dialogfeld "Parameter bearbeiten" hinzufügen, handelt es sich um solche, die bereits für das Stylesheet deklariert wurden. In diesem Fall setzt der neue Wert den Wert im Stylesheet außer Kraft.

Parameter sind nützlich, wenn Sie einen einzelnen Wert in mehreren Ausdrücken verwenden wollen.

Deklarieren von Parametern über das Dialogfeld "Datenbankabfrage bearbeiten"

So deklarieren Sie Parameter:

1. Klicken Sie im Dialogfeld **Datenbankabfrage bearbeiten** auf die Schaltfläche **Parameter...**, um das unten gezeigte Dialogfeld **Parameter bearbeiten** aufzurufen.



2. Klicken Sie auf Anhängen  oder Einfügen .
2. Geben Sie den Namen und Wert des Parameters in die entsprechenden Felder ein.

Bitte beachten Sie: Das Dialogfeld "Parameter bearbeiten" enthält **alle** Parameter, die für das Stylesheet definiert wurden. Es ist zwar ein Fehler, im StyleVision Power Stylesheet einen nicht deklarierten Parameter zu verwenden, doch ist es kein Fehler, einen Parameter zu deklarieren und diesen nicht zu verwenden.

Verwendung von Parametern in Abfragen

Um den Namen eines Parameters als Wert eines Ausdrucks zu verwenden:

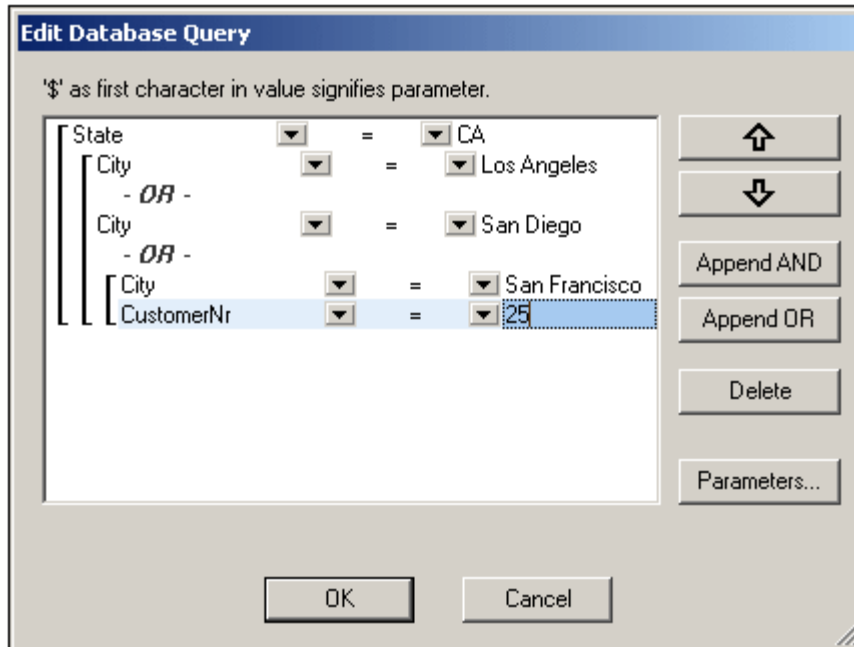
- Geben Sie das \$-Zeichen in das Werteingabefeld ein, gefolgt vom Namen des Parameters (ohne Leerzeichen).

Bitte beachten Sie: Wenn der Parameter bereits deklariert wurde, wird der Eintrag grün angezeigt. Wenn der Parameter noch nicht deklariert wurde, erscheint der Eintrag rot und der Parameter muss deklariert werden.

Neuordnen von Kriterien in DB-Abfragen

- Die logische Struktur der DB-Abfrage und die Beziehung zwischen zwei Kriterien oder

Kriteriengruppen wird grafisch angezeigt. Jede Ebene der logischen Struktur wird durch eine eckige Klammer angezeigt. Zwei Kriterien oder Kriteriengruppen nebeneinander bedeuten, dass der AND-Operator verwendet wurde, während bei Trennung von zwei Kriterien durch das Wort OR der OR-Operator verwendet wurde. Die Kriterien dienen auch dazu, dem Benutzer einen klaren Überblick über die logische Struktur der DB-Abfrage zu verschaffen.



Die in der obigen Abbildung gezeigte DB-Abfrage würde in Textform folgendermaßen aussehen:

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

Sie können die DB-Abfrage umstrukturieren, indem Sie ein Kriterium oder eine Gruppe von Kriterien im Vergleich zu den anderen Kriterien in der DB-Abfrage nach oben oder unten verschieben. Um ein Kriterium oder eine Gruppe von Kriterien zu verschieben, gehen Sie vor wie folgt:

1. Wählen Sie das Kriterium durch Anklicken aus oder wählen Sie eine ganze Ebene aus, indem Sie auf die Klammer für diese Ebene klicken.
2. Klicken Sie im Dialogfeld auf den Nach oben- oder Nach unten-Pfeil.

Beachten Sie die folgenden Punkte:

- Wenn sich das in Richtung der Verschiebung benachbarte Kriterium auf derselben Ebene befindet, werden die beiden Kriterien vertauscht.
- Eine Gruppe von Kriterien (d.h. ein Kriterium mit einer Klammer) ändert seine Position innerhalb derselben Ebene; die Ebene ändert sich nicht.
- Ein einzelnes Kriterium ändert seine Position innerhalb derselben Ebene. Wenn sich das benachbarte Kriterium weiter außen/innen (d.h. nicht auf derselben Ebene) befindet, wird das ausgewählte Kriterium **immer um eine Ebene** nach außen/innen verschoben.

Um ein Kriterium in einer DB-Abfrage zu löschen, wählen Sie es aus und klicken Sie auf **Löschen**.

Ändern einer DB-Abfrage

So ändern Sie eine DB-Abfrage:

1. Klicken Sie auf die Schaltfläche "Abfrage" , um das Dialogfeld "Datenbankabfrage bearbeiten" aufzurufen. Sie können nun die Ausdrücke in jedem der aufgelisteten Kriterien ändern, neue Kriterien hinzufügen oder Kriterien aus der DB-Abfrage löschen.
2. Klicken Sie auf OK. Die Daten aus der DB werden automatisch in StyleVision neu geladen, um die Änderungen an der DB-Abfrage anzuzeigen.

4.3.3 Ändern einer DB-Tabelle

Hinzufügen eines Datensatzes

So fügen Sie einen Datensatz hinzu:

1. Setzen Sie den Cursor in die gewünschte Zeile der DB-Tabelle und klicken Sie auf das Symbol  (um eine Zeile anzuhängen) oder  (um eine Zeile einzufügen). Daraufhin wird ein neuer Datensatz in der temporären XML-Datei erstellt.
2. Klicken Sie auf den Befehl **Datei | Speichern...**, um den neuen Datensatz zur Datenbank hinzuzufügen. In der Authentic-Ansicht wird in der Datenbank-Tabelle eine neue Zeile für den neuen Datensatz angehängt. Der `AltovaRowStatus` für diesen Datensatz wird auf `A` (für Added = Hinzugefügt) gesetzt.

Wenn Sie Daten für den neuen Datensatz eingeben, werden diese fett und unterstrichen angezeigt. Auf diese Art können Sie neu hinzugefügte Datensätze von vorhandenen unterscheiden - wenn vorhandene Datensätze nicht mit diesen Texteingenschaften formatiert wurden. Datentypfehler werden rot gekennzeichnet.

Der neue Datensatz wird zur DB hinzugefügt, wenn Sie **Datei | Speichern...** auswählen. Nachdem ein neuer Datensatz zur Datenbank hinzugefügt wurde, wird das `AltovaRowStatus`-Feld initialisiert (gekennzeichnet durch `---`) und der Datensatz wird in der Authentic-Ansicht als regulärer Datensatz angezeigt.

Ändern eines Datensatzes

Um einen Datensatz zu ändern, setzen Sie den Cursor an die gewünschte Stelle in der DB-Tabelle und nehmen Sie die gewünschten Änderungen vor. Wenn die Anzahl der Datensätze beschränkt ist, müssen Sie eventuell (mit Hilfe der oben beschriebenen Navigationssymbole) zum gewünschten Datensatz navigieren.

Wenn Sie einen Datensatz bearbeiten, werden die Einträge in allen Feldern des Datensatzes unterstrichen und der `AltovaRowStatus` aller primären Instanzen dieses Datensatzes wird auf `U` (für Updated = Aktualisiert) gesetzt. Für alle sekundären Instanzen dieses Datensatzes wird der `AltovaRowStatus` auf `u` (klein geschrieben) gesetzt. Die primären und sekundären Instanzen eines Datensatzes werden durch die Struktur der DB definiert - und ebenfalls von dem davon generierten XML-Schema. Wenn z.B. eine Adresstabelle in einer Kundentabelle enthalten ist, kann die Adresstabelle im Design-Dokument in zwei Instantiierungstypen vorkommen: als Adresstabelle selbst und innerhalb von Instantiierungen der Kundentabelle. Derjenige der beiden Typen, der geändert wird, ist der primär geänderte Typ. Andere Typen - es kann mehr als einen anderen Typ geben - sind sekundäre Typen. Datentypfehler werden rot markiert.

Sie speichern die Änderungen in der DB durch Klicken auf **Datei | Speichern...**. Nachdem ein geänderter Datensatz in der DB gespeichert wurde, wird das `AltovaRowStatus` Feld initialisiert (gekennzeichnet durch `---`) und der Datensatz wird in der Authentic-Ansicht als regulärer Datensatz angezeigt.

Bitte beachten Sie:

- Selbst wenn nur ein einziges Feld eines Datensatzes in der Authentic-Ansicht geändert wird, wird der gesamte Datensatz beim Speichern der Daten in der Datenbank aktualisiert.
- Der Datumswert `0001-01-01` ist bei einigen Datenbanken als `NULL`-Wert definiert und könnte daher eine Fehlermeldung verursachen.

Löschen eines Datensatzes

So löschen Sie einen Datensatz:

1. Setzen Sie den Cursor in die Zeile für den entsprechenden Datensatz und klicken Sie auf das Symbol . Der zu löschende Datensatz erscheint nun durchgestrichen. Der `AltovaRowStatus` erhält folgenden Wert: Primäre Instanzen des Datensatzes werden auf `D` gesetzt; sekundäre Instanzen auf `d`; indirekt gelöschte Datensätze werden mit `x` gekennzeichnet. Indirekt gelöschte Datensätze sind Felder im gelöschten Datensatz, die in einer separaten Tabelle enthalten sind. So könnte z.B. in einer Kundentabelle eine Adresstabelle enthalten sein. Wenn ein Kundendatensatz gelöscht würde, würde indirekt auch der entsprechende Adressdatensatz gelöscht. Wenn in der Kundentabelle ein Adressdatensatz gelöscht würde, würde der Adressdatensatz in der Kundentabelle primär gelöscht, jedoch würde derselbe Datensatz in einer unabhängigen Adresstabelle sekundär gelöscht, wenn diese instantiiert würde.
2. Sie speichern die Änderungen in der DB durch Klicken auf **Datei | Speichern....**

Bitte beachten Sie: Wenn Sie Daten in der DB speichern, wird der Rückgängig-Befehl zurückgesetzt. Sie können Operationen, die vor dem Speichern durchgeführt wurden, nicht mehr rückgängig machen.

4.4 Symbole zum Bearbeiten von XML-Tabellen

Die Befehle zum Bearbeiten von XML-Tabellen stehen in Form von Symbolen in der Symbolleiste zur Verfügung. Beachten Sie, dass es zu diesen Symbolen keine Menübefehlentsprechungen gibt.

Ausführliche Informationen zur Verwendung von XML-Tabellen finden Sie unter [XML-Tabellen](#).

Tabelle einfügen



Der Befehl "Tabelle einfügen" fügt eine **CALS / HTML-Tabelle** an der aktuellen Cursorposition ein.

Tabelle löschen



Der Befehl "Tabelle löschen" löscht die derzeit aktive Tabelle.

Zeile anhängen



Der Befehl "Zeile anhängen" hängt eine Zeile an das Ende der derzeit aktiven Tabelle an.

Spalte anhängen



Der Befehl "Spalte anhängen" hängt eine Spalte an das Ende der derzeit aktiven Tabelle an.

Zeile einfügen



Der Befehl "Zeile einfügen" fügt eine Zeile oberhalb der aktuellen Cursorposition in der derzeit aktiven Tabelle ein.

Spalte einfügen



Der Befehl "Spalte einfügen" fügt eine Spalte links von der derzeit aktiven Cursorposition in der derzeit aktiven Tabelle ein.

Links zusammenfügen



Der Befehl "Links zusammenfügen" fügt die aktive Zelle (an der Cursorposition) mit der Zelle links davon zusammen. Die Tags von beiden Zellen bleiben in der neuen Zelle erhalten, die Spaltenüberschriften bleiben unverändert.

Rechts zusammenfügen



Der Befehl "Rechts zusammenfügen" fügt die aktive Zelle (an der Cursorposition) mit der Zelle rechts davon zusammen. Die Tags von beiden Zellen bleiben in der neuen Zelle erhalten, die Spaltenüberschriften bleiben unverändert.

Unten zusammenfügen



Der Befehl "Unten zusammenfügen" fügt die aktive Zelle (an der Cursorposition) mit der Zelle unterhalb davon zusammen. Die Tags von beiden Zellen bleiben in der neuen Zelle erhalten, die Spaltenüberschriften bleiben unverändert.

Oben zusammenfügen



Der Befehl "Oben zusammenfügen" fügt die aktive Zelle (an der Cursorposition) mit der Zelle oberhalb davon zusammen. Die Tags von beiden Zellen bleiben in der neuen Zelle erhalten, die Spaltenüberschriften bleiben unverändert.

Horizontal teilen



Der Befehl "Horizontal teilen" erstellt eine neue Zelle rechts von der derzeit aktiven Zelle. Die Größe der beiden Zellen entspricht nun der Größe der Originalzelle

Vertikal teilen



Der Befehl "Vertikal teilen" erstellt eine neue Zelle unterhalb von der derzeit aktiven Zelle.

Bündig oben



Dieser Befehl richtet den Inhalt der Zellen bündig am oberen Zellenrand aus.

Vertikal zentrieren



Dieser Befehl zentriert den Inhalt der Zelle.

Bündig unten

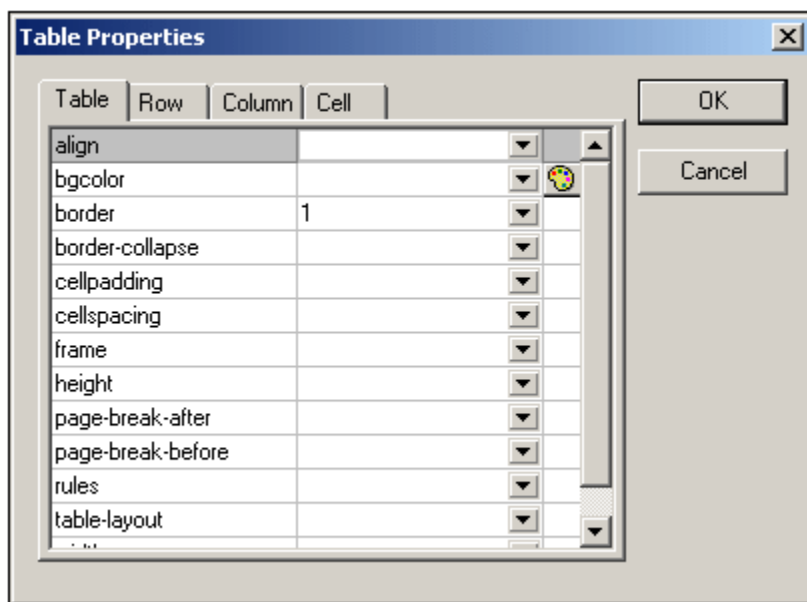


Dieser Befehl richtet den Inhalt der Zellen bündig am unteren Zellenrand aus.

Tabelleneigenschaften



Der Befehl "Tabelleneigenschaften" öffnet das Dialogfeld "Tabelleneigenschaften". Dieses Symbol wird nur bei HTML-Tabellen aktiv, bei CALS-Tabellen ist es deaktiviert.



4.5 Arbeiten mit Datumsangaben

Es gibt in der Authentic-Ansicht zwei Methoden, um Datumsangaben zu bearbeiten:

- Die Eingabe oder Änderung des Datums erfolgt über die [Datumswahl](#).
- Das Datum wird durch [Eingabe des Werts](#) eingegeben bzw. geändert.

Welche Methode der Benutzer der Authentic-Ansicht verwendet, ist im SPS definiert. Beide Methoden sind in den beiden Unterkapiteln dieses Abschnitts beschrieben.

Anmerkung zu Datumsformaten

Zur Speicherung des Datums im XML-Dokument stehen mehrere Datums-Datentypen zur Verfügung. Bei jedem dieser Datums-Datentypen muss das Datum in einem bestimmten lexikalischen Format gespeichert werden, damit das XML-Dokument gültig ist. So muss das Datum z.B. beim Datentyp `xs:date` das lexikalische Format `JJJJ-MM-TT` aufweisen. Wenn das Datum in einem `xs:date` Node anders als in diesem Format eingegeben wurde, ist das XML-Dokument ungültig.

Um sicherzustellen, dass das Datum im richtigen Format eingegeben wird, kann der Designer des SPS die grafische Datumswahl in das Design integrieren. Dadurch wird sichergestellt, dass das mit der Datumswahl ausgewählte Datum im richtigen lexikalischen Format eingegeben wird. Falls keine Datumswahl vorhanden ist, sollte in der Authentic-Ansicht sichergestellt werden, dass das Datum im korrekten Format eingegeben wird. Bei der Validierung des XML-Dokuments erhalten Sie nützliche Tipps über das erforderliche lexikalische Format.

4.5.1 Datumswahl

Die Datumswahl ist ein grafischer Kalender zur Eingabe eines Datums in einem Standardformat in ein XML-Dokument. Es ist für die Verarbeitung des Dokuments wichtig, dass ein Standardformat verwendet wird. Das Datumswahlsymbol erscheint neben dem Datumsfeld, das dadurch geändert wird (siehe Abbildung unten).

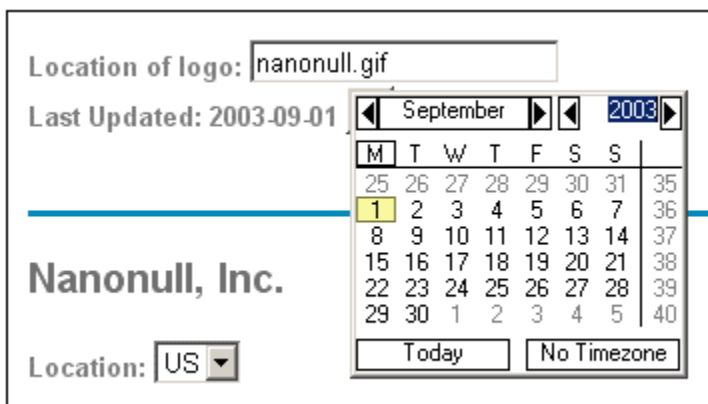


Organization Chart

Location of logo:

Last Updated: 2003-09-01 

Um die Datumswahl (siehe unten) anzuzeigen, klicken Sie auf das Datumswahl-Symbol.



Location of logo:

Last Updated: 2003-09-01

Nanonull, Inc.

Location:

September 2003

M	T	W	T	F	S	S
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

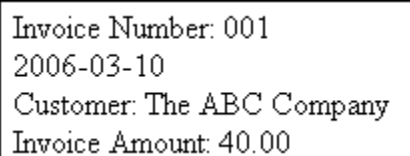
Today No Timezone

Um ein Datum auszuwählen, klicken Sie auf das gewünschte Datum, den gewünschten Monat bzw. das gewünschte Jahr. Das Datum wird in das XML-Dokument eingegeben und das angezeigte Datum wird entsprechend geändert. Falls nötig können Sie auch eine Zeitzone angeben.

4.5.2 Texteintrag

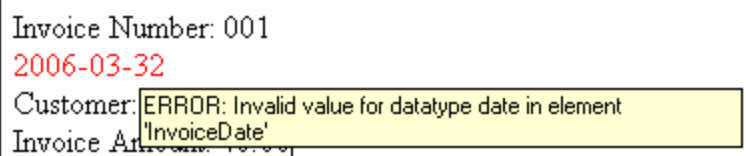
In Datumsfeldern, die keine Datumswahl haben (*siehe Screenshot*) können Sie das Datum direkt durch Eingabe des neuen Werts bearbeiten.

Bitte beachten Sie: Beim Bearbeiten des Datums darf das Format nicht geändert werden.



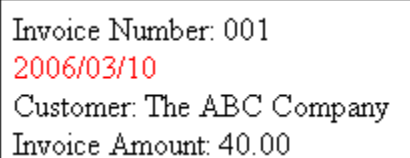
Invoice Number: 001
2006-03-10
Customer: The ABC Company
Invoice Amount: 40.00

Wenn Sie ein Datum ändern und es sich daraufhin außerhalb des gültigen Bereichs für Datumsangaben befindet, wird das Datum rot angezeigt, um den Fehler anzuzeigen. Wenn Sie den Mauszeiger über das ungültige Datum platzieren, erscheint eine Fehlermeldung (*siehe Screenshot*).



Invoice Number: 001
2006-03-32
Customer: ERROR: Invalid value for datatype date in element
Invoice Amount: 40.00

Wenn Sie versuchen, das Datumsformat zu ändern, wird das Datum rot angezeigt (*siehe Screenshot*).



Invoice Number: 001
2006/03/10
Customer: The ABC Company
Invoice Amount: 40.00

4.6 Definieren von Entities

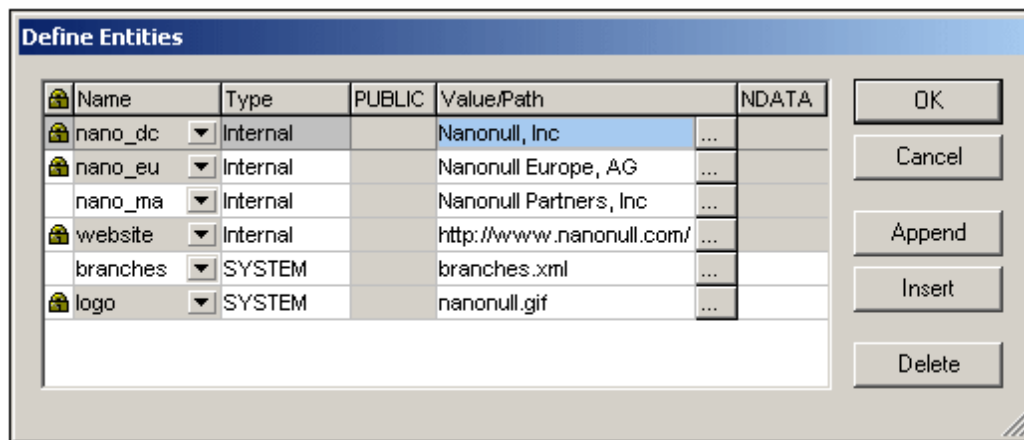
Unabhängig davon, ob Ihr Dokument auf einer DTD oder einem XML-Schema basiert, können Sie für die Verwendung in der Authentic-Ansicht Entities definieren. Diese Entities werden anschließend in der Entities-Eingabehilfe und im Kontextmenü-Untermenü von **Entity einfügen** angezeigt. Wenn Sie in der Entities-Eingabehilfe auf eine Entity doppelklicken, wird diese an der Cursorposition eingefügt.

Eine Entity eignet sich v.a. dann, wenn Sie einen Textstring, ein XML-Fragment oder einige andere externe Ressourcen an mehreren Stellen in Ihrem Dokument verwenden möchten. Die Entity, bei der es sich im Grunde um eine Kurzbezeichnung für die benötigten Daten handelt, wird im Dialogfeld "Entities definieren" definiert. Nachdem Sie die Entity definiert haben, können Sie diese an mehreren Stellen in Ihrem Dokument verwenden und dadurch Zeit sparen und effizienter arbeiten.

Sie können in Ihrem Dokument zwei grundlegende Arten von Entities verwenden: **geparste Entities**, wobei es sich um XML-Daten handelt (entweder einen Textstring oder ein Fragment aus einem XML-Dokument) oder **ungeparste Entities**, wobei es sich um nicht-XML-Daten handelt wie z.B. eine Binärdatei (normalerweise eine Grafik, Ton oder ein Multimedia-Objekt). Jede Entity hat einen Namen und einen Wert. Im Fall von geparsten Entities ist die Entity ein Platzhalter für die XML-Daten. Der Wert der Entity sind entweder die XML-Daten selbst oder eine URI, die auf eine .xml-Datei verweist, die die XML-Daten enthält. Bei ungeparsten Entities ist der Wert der Entity eine URI, die auf eine nicht-XML-Datei verweist.

So definieren Sie eine Entity:

1. Klicken Sie auf **Authentic | XML-Entities definieren....** Daraufhin wird das Dialogfeld "Entities definieren" geöffnet. (Abbildung unten)



2. Geben Sie den Namen Ihrer Entity im Feld "Name" ein. Dies ist der Name, der in der Entities-Eingabehilfe angezeigt werden wird.
3. Wählen Sie den Typ der Entity aus der Dropdown-Liste im Feld "Type" aus. Es stehen die folgenden Typen zur Auswahl. Eine **interne** (internal) Entity ist eine Entity, für die der zu verwendende Text im XML-Dokument selbst gespeichert ist. Wenn Sie **PUBLIC** oder **SYSTEM** auswählen, wird angegeben, dass sich die Ressource außerhalb der XML-Datei befindet und mittels eines "Public Identifier" oder eines "System Identifier" gefunden wird. Ein System Identifier ist eine URI, die den Pfad der Ressource definiert. Ein Public Identifier ist ein pfadunabhängiger Identifier, der es manchen Prozessoren ermöglicht, die Ressource zu identifizieren. Wenn Sie sowohl einen Public als auch einen System Identifier angeben, wird der Public Identifier zum System Identifier aufgelöst und es wird der System Identifier verwendet.
4. Wenn Sie als Type PUBLIC ausgewählt haben, geben Sie den Public Identifier der

Ressource im Feld PUBLIC ein. Wenn Sie Internal oder SYSTEM als Type ausgewählt haben, ist das Feld PUBLIC deaktiviert.

5. Im Feld "Value/Path" haben Sie folgende Eingabemöglichkeiten:
 - Wenn die Entity "Internal" ist, geben Sie den Textstring ein, der als Wert der Entity verwendet werden soll. Setzen Sie den Textstring nicht in Anführungszeichen. Alle Anführungszeichen werden als Teil des Textstrings behandelt.
 - Wenn der Entity-Typ SYSTEM ist, geben Sie die URI der Ressource ein oder wählen Sie mit Hilfe der Durchsuchen-Schaltfläche eine Ressource auf Ihrem lokalen Netzwerk aus. Wenn die Ressource gepackte Daten enthält, muss es sich um eine XML-Datei handeln (d.h. die Datei muss die Erweiterung `.xml` haben). Als Alternative dazu kann die Ressource eine Binärdatei sein, z.B. eine GIF-Datei.
 - Wenn der Entity-Typ PUBLIC ist, müssen Sie zusätzlich einen System Identifier in dieses Feld eingeben.
6. Der Eintrag NDATA sagt dem Prozessor, dass diese Entity nicht gepackt, sondern an den entsprechenden Prozessor gesendet werden soll. Das Feld "NDATA" sollte daher nur mit ungepackten Entities verwendet werden.

Dialogfeldfunktionen

Im Dialogfeld "Entities definieren" können Sie Folgendes tun:

- Entities anhängen
- Entities einfügen
- Entities löschen
- Entities spaltenweise alphabetisch sortieren, indem Sie auf die entsprechende Spaltenüberschrift klicken; bei einem einmaligen Klick wird aufsteigend sortiert, bei zweimaligem Klick absteigend
- die Größe des Dialogfelds und die Spaltenbreite anpassen.
- Sperren. Sobald eine Entity im XML-Dokument verwendet wird, ist sie gesperrt und kann im Dialogfeld "Entities definieren" nicht bearbeitet werden. Gesperrte Entities werden durch das Schlosssymbol in der ersten Spalte gekennzeichnet. Durch das Sperren einer Entity wird sichergestellt, dass das Dokument in Bezug auf Entities gültig ist. (Das Dokument wäre ungültig, wenn eine Entity referenziert aber nicht definiert ist.)
- Doppelt vorhandene Entities werden markiert.

Einschränkungen

- Eine Entity innerhalb einer anderen Entity wird nicht aufgelöst - weder im Dialogfeld, noch in der Authentic-Ansicht, noch in der XSLT-Ausgabe, und das kaufmännische Und-Zeichen einer derartigen Entity wird in seiner Escape-Form, also `&` angezeigt.
- Externe Entities werden in der Authentic-Ansicht nicht aufgelöst, es sei denn es handelt sich bei der Entity um eine Bilddatei und sie wurde als der Wert eines Attributs vom Typ `ENTITY` oder `ENTITIES` eingegeben. Derartige Entities werden aufgelöst, wenn das Dokument mit einem anhand des SPS generierten XSLT verarbeitet wird.

4.7 Grafiken in der Authentic-Ansicht

Sie können in der Authentic-Ansicht Bilder definieren, die im Ausgabedokument (HTML, RTF, PDF und Word 2007) verwendet werden sollen. Beachten Sie jedoch, dass einige Bildformate in manchen Formaten oder von manchen Applikationen eventuell nicht unterstützt werden. So wird das SVG-Format z.B. in PDF unterstützt, nicht aber in RTF und für die Anzeige in HTML würden Sie ein Browser-Zusatzprogramm benötigen. Stellen Sie daher bei Auswahl eines Bildformats sicher, ein Format zu wählen, das in den Ausgabeformaten Ihres Dokuments auch unterstützt wird. Die meisten Bildformate werden in allen Ausgabeformaten unterstützt (*siehe Liste unten*).

Die Authentic-Ansicht basiert auf dem Internet Explorer und kann die meisten der Grafikformate, die von Ihrer Internet Explorer-Version unterstützt werden, anzeigen. Unterstützt werden die folgenden gebräuchlichen Grafikformate:

- GIF
- JPG
- PNG
- BMP
- WMF (Microsoft Windows Metafile)
- EMF (Enhanced Metafile)
- SVG (nur für PDF-Ausgabe)

Relative Pfade

Relative Pfade werden relativ zur SPS-Datei aufgelöst.

4.8 Verwendung von Tasten in der Authentic-Ansicht

Eingabetaste

In der Authentic-Ansicht wird die **Eingabetaste** an bestimmten Cursorpositionen verwendet, um zusätzliche Elemente anzuhängen. Wenn das Kapitel eines Buchs z.B. (laut Schemadefinition) mehrere Absätze enthalten kann, wird bei Drücken der **Eingabetaste** innerhalb des Texts des Absatzes ein neuer Absatz direkt hinter dem aktuellen Absatz angehängt. Wenn ein Kapitel einen Titel und mehrere Kapitel enthalten kann, wird bei Drücken der **Eingabetaste** innerhalb des Kapitels aber außerhalb eines Absatzelements (auch innerhalb des Titel-Elements) ein neues Kapitel hinter dem aktuellen Kapitel angehängt (vorausgesetzt es sind laut Schemadefinition mehrere Kapitel zulässig).

Bitte beachten Sie: Bei Drücken der **Eingabetaste** wird keine Zeilenschaltung, kein Zeilenvorschub eingefügt, d.h. es wird keine neue Zeile angelegt. Dies ist auch dann der Fall, wenn sich der Cursor innerhalb eines Textnode wie z.B. eines Absatzes befindet.

Verwendung der Tastatur

Verwenden Sie die Tastatur wie bei allen Anwendungen zum Eingeben von Text/Daten und zum Navigieren. Beachten Sie dabei die folgenden Punkte:

- Mit der **Tabulatortaste** springen Sie mit dem Cursor weiter. Der Cursor hält vor und hinter Nodes an und markiert den Inhalt von Nodes; statischer Inhalt wird übersprungen.
- Die Hyperlinks `hinzufügen...` und `Node hinzufügen` werden als Node-Inhalt betrachtet und markiert, wenn Sie mit der Tabulatortaste navigieren. Durch Drücken der Leertaste oder der **Eingabetaste** werden sie aktiviert.

5 Browser-Ansicht

Die Browser-Ansicht wird normalerweise zur Ansicht folgender Dateien verwendet:

- XML-Dateien, die mit einer XSLT-Datei verknüpft wurden. Wenn Sie in die Browser-Ansicht wechseln, wird die XML-Datei mit Hilfe eines XSLT Stylesheet on-the-fly transformiert. Das Ergebnis wird direkt in der Browser-Ansicht angezeigt.
- HTML-Dateien, die direkt im HTML-Format erstellt wurden oder mittels einer XSLT-Transformation einer XML-Datei erzeugt wurden.

Um XML- und HTML-Dateien in der Browser-Ansicht anzuzeigen, klicken Sie auf das Register **Browser**.

Anmerkung zu Microsoft Internet Explorer und XSLT

Für die Browser-Ansicht wird Microsoft Internet Explorer 5.0 oder höher benötigt. Zur Anzeige von XML-Dateien, die mit Hilfe eines XSLT Stylesheet transformiert wurden, sollten Sie unbedingt Internet Explorer 6.0 oder höher verwenden, da diese Versionen MSXML 3.0 verwenden, einen XML-Parser, die den XSLT 1.0 Standard vollständig unterstützt. Eventuell sollten Sie auch MSXML 4.0 installieren. Nähere Informationen dazu erhalten Sie in unserem [Download Center](#). (Beachten Sie, dass die Unterstützung von XSL in IE 5 nicht zu 100 % kompatibel mit der offiziellen XSLT Recommendation ist. Wenn Sie daher in der Browser-Ansicht Probleme mit IE 5 haben, sollten Sie auf IE 6 upgraden.) Außerdem sollten Sie überprüfen, welche XSLT-Version Ihre Version des Internet Explorer unterstützt.

Funktionen der Browser-Ansicht

In der Browser-Ansicht stehen die folgenden Funktionen zur Verfügung. Sie können über das [Menü Browser](#) und das [Menü Bearbeiten](#) aufgerufen werden.

- **Öffnen in einem separaten Fenster:** Wenn die Browser-Ansicht ein separates Fenster ist, können Sie die Browser-Ansicht Seite an Seite mit der Bearbeitungsansicht des selben Dokuments anzeigen. Dieser Befehl befindet sich im Menü **Browser** und ist ein Ein/Aus-Befehl, mit dem Sie zwischen dem Browser-Fenster als separatem Fenster und als Registerkarte des Fensters wechseln können. Im Dialogfeld "Optionen" können Sie auf dem [Register "Ansicht"](#) festlegen, ob die Browser-Ansicht in einem separaten Fenster angezeigt werden soll.
- **Vorwärts und Zurück:** die allgemeinen Browser-Befehle zum Navigieren durch Seiten, die in der Browser-Ansicht gefunden wurden. Diese Befehle befinden sich im Menü **Browser**
- **Schriftgröße:** Kann über den **Browser** Menübefehl angepasst werden.
- **Abbrechen, Aktualisieren, Drucken:** Weitere Browser-Standardbefehle. Diese Befehle befinden sich in den Menüs **Browser** und **Datei**.
- **Suchen:** Dient zum Suchen von Textstrings. Dieser Befehl befindet sich im Menü **Bearbeiten**.

6 Globale Altova-Ressourcen

Bei den globalen Altova-Ressourcen handelt es sich um eine Sammlung von Aliassen für Datei-, Ordner- und Datenbankressourcen. Jeder Alias kann mehrere Konfigurationen haben, wobei jede Konfiguration genau einer Ressource zugeordnet wird.

Bei Verwendung einer globalen Ressource als Input können Sie mittels der Schaltflächen auf der Benutzeroberfläche jederzeit zwischen den verschiedenen Konfigurationen für die globale Ressource wechseln.

Eine globale Ressource kann nicht nur verwendet werden, um Ressourcen innerhalb einer Altova-Applikation zu wechseln, sondern auch, um Ressourcen in anderen Altova-Applikationen zu generieren und zu verwenden. So können z.B. Dateien on-the-fly in einer Altova-Applikation generiert werden, um in einer anderen Altova-Applikation verwendet zu werden. Damit lassen sich Entwicklungs- und Testabläufe enorm vereinfachen und beschleunigen.

Um globale Altova-Ressourcen verwenden zu können, sind zwei Schritte erforderlich:

- [Definieren globaler Ressourcen](#): Ressourcen werden definiert und die Definitionen werden in einer XML-Datei gespeichert. Diese Ressourcen können mehreren Altova-Applikationen zur Verfügung gestellt werden.
- [Verwendung von globalen Ressourcen](#): In einer Altova-Applikation können Dateien über eine globale Ressource anstatt über einen Dateipfad zugewiesen werden. Der Vorteil ist, dass die verwendete Ressource durch Wechseln der Konfiguration in Authentic Desktop sofort gewechselt werden kann.

Globale Ressourcen in anderen Altova-Produkten

Derzeit können globale Ressourcen in den folgenden Altova-Produkten definiert und verwendet werden: XMLSpy, StyleVision, MapForce, und DatabaseSpy.

6.1 Definieren globaler Ressourcen

Globale Altova-Ressourcen werden im Dialogfeld **Globale Ressourcen verwalten** definiert, welches auf zwei Arten aufgerufen werden kann:

- Klicken Sie in der Menüleiste auf "Extras", um das Menü **Extras** anzuzeigen (*Abbildung unten*) und wählen Sie den Befehl **Globale Ressourcen**. Daraufhin wird das Dialogfeld "Globale Ressourcen verwalten" angezeigt.
- Daraufhin wird das Dialogfeld **"Globale Ressourcen verwalten"** angezeigt.
- Wählen Sie den Menübefehl **Extras | Anpassen**, um das Dialogfeld "Anpassen" aufzurufen. Klicken Sie anschließend auf das Register **Symbolleiste** und aktivieren Sie die Option "Globale Ressourcen". Daraufhin wird die Symbolleiste "Globale Ressourcen" eingeblendet (*Abbildung unten*).



Sobald die Symbolleiste angezeigt wird, klicken Sie auf die Schaltfläche "Globale Ressourcen verwalten". Daraufhin wird das Dialogfeld "Globale Ressourcen verwalten" aufgerufen.

Die XML-Datei für globale Ressourcen

Die Informationen über globale Ressourcen, die definiert wurden, werden in einer XML-Datei gespeichert. Standardmäßig hat diese XML-Datei den Namen `GlobalResources.xml` und ist im Ordner `C:\Dokumente und Einstellungen\<Benutzername>\Eigene Dateien\Altova\` gespeichert. Diese Datei ist bei allen Altova-Applikationen als Standard-XML-Datei für globale Ressourcen definiert. Infolgedessen steht eine globale Ressource, die in einer beliebigen Altova-Applikation definiert wurde, allen Altova-Applikationen zur Verfügung - vorausgesetzt, dass diese Datei in allen Applikationen verwendet wird.

Sie können die Datei umbenennen und in einem beliebigen Ordner speichern. Sie können daher mehrere XML-Dateien für globale Ressourcen haben; es kann jedoch immer nur eine dieser Dateien aktiv sein und nur die in dieser Datei enthaltenen Definitionen stehen der Applikation zur Verfügung.

Um eine XML-Datei für globale Ressourcen als aktive Datei auszuwählen, suchen Sie im Dialogfeld "Globale Ressourcen verwalten" (*Abbildung unten*) im Feld "Definitionsdatei" danach und wählen Sie sie aus.

Verwalten globaler Ressourcen: Hinzufügen, Bearbeiten, Löschen

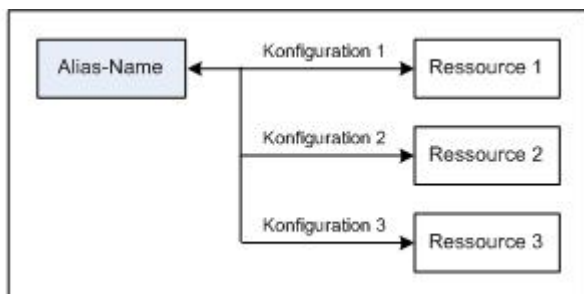
Im Dialogfeld "Globale Ressourcen verwalten" (*Abbildung oben*) können Sie eine globale Ressource zur ausgewählten XML-Datei für globale Ressourcen hinzufügen oder eine ausgewählte globale Ressource löschen oder bearbeiten. In der XML-Datei für globale Ressourcen werden die von Ihnen hinzugefügten Aliasse in Abschnitten aufgelistet: Dateien, Ordner und Datenbanken (*siehe Abbildung oben*).

Um eine globale Ressource hinzuzufügen, klicken Sie auf die Schaltfläche **Hinzufügen** und definieren Sie die globale Ressource im Dialogfeld **Globale Ressource**, das angezeigt wird (*Beschreibung siehe oben*). Nachdem Sie eine globale Ressource definiert und gespeichert haben, wird die globale Ressource (Alias) zur Bibliothek der globalen Definitionen in der ausgewählten XML-Datei für globale Ressourcen hinzugefügt. Um eine globale Ressource zu bearbeiten, wählen Sie sie aus und klicken Sie auf **Bearbeiten**. Daraufhin wird das Dialogfeld **Globale Ressource** angezeigt, in dem Sie die nötigen Änderungen vornehmen können (*siehe dazu die Beschreibungen in den Unterabschnitten [Dateien](#), [Ordner](#) und [Datenbanken](#) in den Unterabschnitten dieses Abschnitts*). Um eine globale Ressource zu löschen, wählen Sie sie aus und klicken Sie auf **Löschen**.

Nachdem Sie mit dem Hinzufügen, Bearbeiten oder Löschen globaler Ressourcen fertig sind, klicken Sie im Dialogfeld **Globale Ressourcen verwalten** auf **OK**, um Ihre Änderungen an der XML-Datei für globale Ressourcen zu speichern.

Hinzufügen einer globalen Ressource

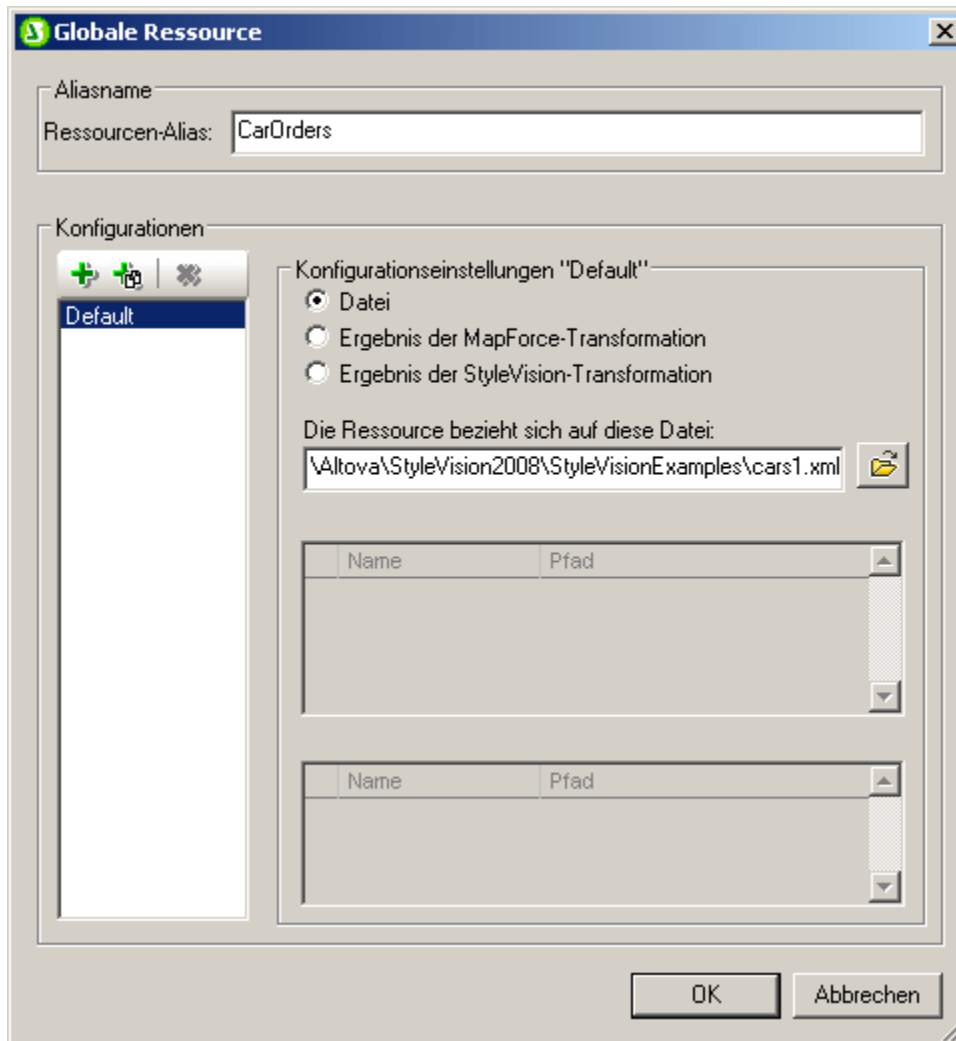
Beim Erstellen einer globalen Ressource wird ein Aliasname auf eine oder mehrere Ressourcen (Datei, Ordner oder Datenbank) gemappt. Jedes Mapping wird als Konfiguration bezeichnet. Ein einziger Aliasname kann daher mittels unterschiedlicher Konfigurationen mit mehreren Ressourcen verknüpft werden (*Abbildung unten*).



Wenn Sie im Dialogfeld **Globale Ressourcen verwalten** (*Abbildung oben*) auf die Schaltfläche **Hinzufügen** klicken, können Sie auswählen, ob Sie eine Ressource vom Typ Datei, Ordner oder Datenbank hinzufügen möchten. Wie man die einzelnen Ressourcenarten hinzufügt, ist in den Unterabschnitten dieses Abschnitts beschrieben.

6.1.1 Dateien

Im Dialogfeld **Globale Ressource** (Abbildung unten) können Sie auf folgende Art eine Dateiressource hinzufügen.



1. Geben Sie einen Aliasnamen ein.
2. Im Fenster "Konfigurationen" finden Sie eine Konfiguration mit dem Namen "Default" (Abbildung oben). Diese Standardkonfiguration kann nicht gelöscht werden und der Name kann nicht geändert werden. Sie können beliebig viele zusätzliche Konfigurationen für den ausgewählten Alias hinzufügen. Klicken Sie dazu auf die Schaltfläche **Konfiguration hinzufügen**  und geben Sie im angezeigten Dialogfeld "Konfiguration hinzufügen" den Konfigurationsnamen ein. Klicken Sie auf **OK**. Die neue Konfiguration wird im Fenster "Konfigurationen" aufgelistet. Wiederholen Sie diesen Schritt, um so viele Konfigurationen für diesen Alias (globale Ressource) hinzuzufügen, wie erforderlich. Sie können eine Konfiguration auch (mit Hilfe der Schaltfläche "Konfiguration als Kopie hinzufügen") kopieren und anschließend ändern.
3. Wählen Sie eine Konfiguration im Bereich "Konfigurationen" aus und definieren Sie anschließend die Ressource, auf die die Konfiguration gemappt werden soll. Im Fenster für die Einstellungen der Konfiguration X können Sie auswählen, ob es sich bei der Ressource um eine Datei oder das Ergebnis einer Altova MapForce- oder Altova StyleVision-Transformation handelt. Nachdem Sie den Ressourcentyp durch Klicken

auf das entsprechende Optionsfeld ausgewählt haben, suchen Sie die Datei, die MapForce-Datei oder die StyleVision-Datei. Wo mehrere Inputs oder Outputs für die Transformationen möglich sind, stehen Optionen zur Auswahl. Wenn z.B. das Ergebnis einer StyleVision-Transformation als Ressourcentyp ausgewählt wurde, werden, je nachdem welche Edition installiert ist (*in der Abbildung unten sehen Sie die Ausgabeoptionen für die Enterprise Edition*), die verschiedenen Ausgabeoptionen angezeigt.



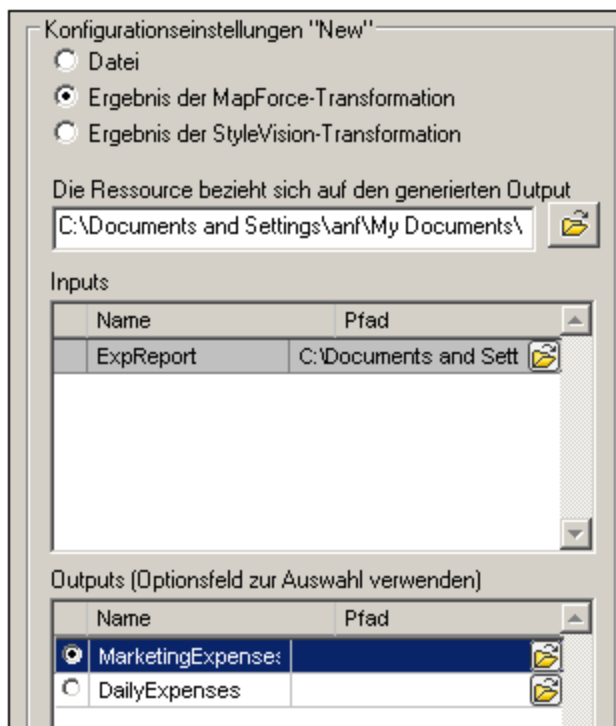
Aktivieren Sie das gewünschte Optionsfeld (in der Abbildung oben ist "HTML-Ausgabe" ausgewählt). Das Ergebnis einer Transformation kann selbst als globale Ressource oder als Dateipfad (über die Schaltfläche und anschließendes Auswählen von "Globale Ressource" oder "Durchsuchen") gespeichert werden. Wenn keine dieser beiden Speicheroptionen ausgewählt wird, wird das Transformationsergebnis bei Aufruf der globalen Ressource als temporäre Datei geladen.

4. Definieren Sie eine Dateiressource für jede Konfiguration (d.h. wiederholen Sie Schritt 3 für die verschiedenen von Ihnen erstellten Konfigurationen).
5. Klicken Sie im Dialogfeld **Globale Ressource** auf **OK**, um den Alias und alle seine Konfigurationen als globale Ressource zu speichern. Die globale Ressource wird im Dialogfeld **Globale Ressourcen verwalten** unter "Dateien" aufgelistet.

Auswählen des Ergebnisses einer MapForce-Transformation als globale Ressource

Altova MapForce mappt ein oder mehrere (bestehende) Schemas auf ein oder mehrere (neue) vom MapForce-Benutzer erstellte Schemas. XML-Dateien, die den Input-Schemas entsprechen, werden als Datenquellen verwendet und MapForce kann eine XML-Ausgabedatei generieren, die auf dem vom Benutzer erstellten Schema basiert. Diese generierte Ausgabedatei (Ergebnis der MapForce-Transformation) ist die Datei, die als globale Ressource verwendet wird.

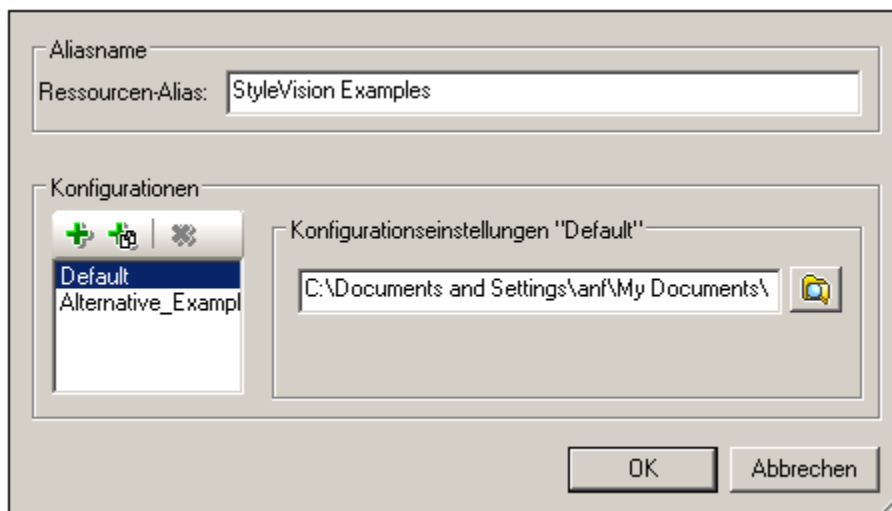
In einer MapForce-Transformation, die mehrere Ausgabeschemas hat, können Sie durch Klicken auf das entsprechende Optionsfeld auswählen, welches der Ausgabeschemas für die globale Ressource verwendet werden soll (*Abbildung unten*). Die XML-Datei, die für dieses Schema generiert wird, kann als globale Ressource oder (über die Schaltfläche) als Datei gespeichert werden. Wenn keine dieser Optionen ausgewählt ist, wird bei Verwendung der globalen Ressource eine temporäre XML-Datei erstellt.



Beachten Sie, dass auch jeder Input als globale Ressource oder als Datei (klicken Sie auf die Schaltfläche  und wählen Sie "Globale Ressource" bzw. "Durchsuchen") gespeichert werden kann.

6.1.2 Ordner

Im Dialogfeld **Globale Ressource** für Ordner (*Abbildung unten*) können Sie eine Ordnerressource auf folgende Art hinzufügen:

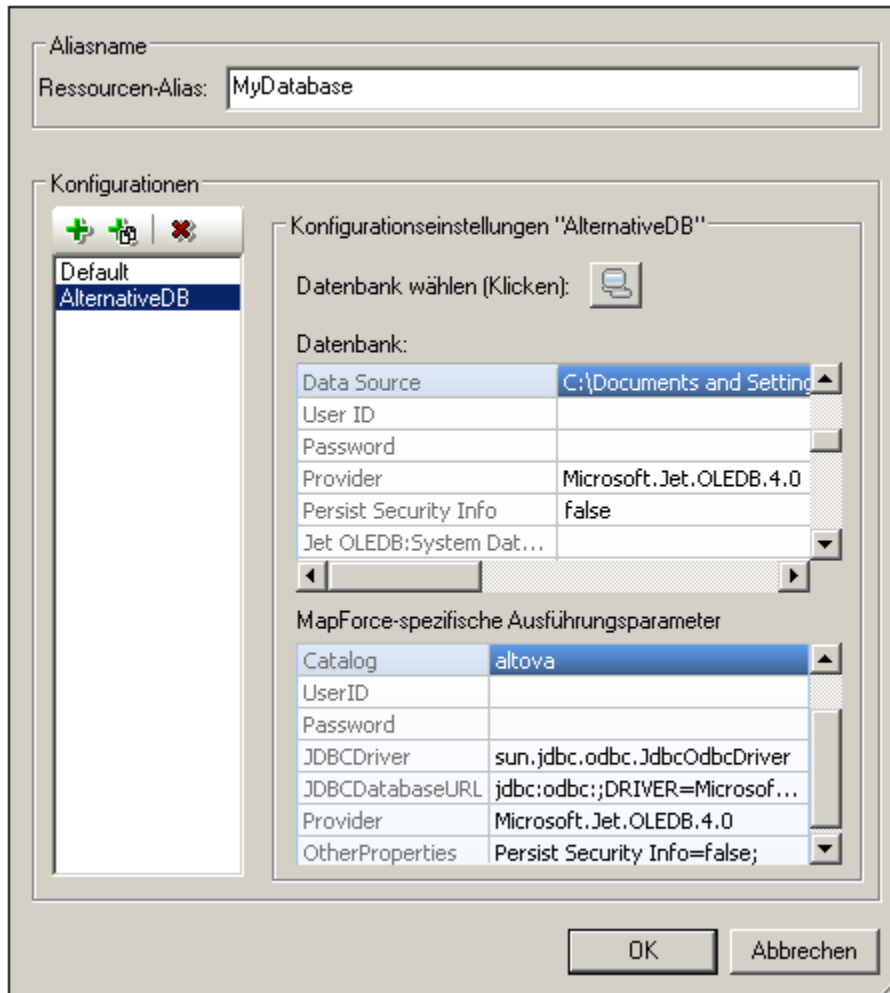


Geben Sie einen Aliasnamen ein.

1. Im Fenster "Konfigurationen" finden Sie eine Konfiguration mit dem Namen "Default" (*Abbildung oben*). Diese Standardkonfiguration kann nicht gelöscht werden und der Name kann nicht geändert werden. Sie können beliebig viele zusätzliche Konfigurationen für den ausgewählten Alias hinzufügen. Klicken Sie dazu auf die Schaltfläche **Konfiguration hinzufügen**  und geben Sie im angezeigten Dialogfeld **Konfiguration hinzufügen** den Konfigurationsnamen ein. Klicken Sie auf **OK**. Die neue Konfiguration wird im Fenster "Konfigurationen" aufgelistet. Wiederholen Sie diesen Schritt um so viele Konfigurationen für diesen Alias (globale Ressource) hinzuzufügen, wie erforderlich.
2. Wählen Sie im Fenster "Konfigurationen" eine der Konfigurationen aus und suchen Sie den Ordner, den Sie als globale Ressource erstellen möchten.
3. Definieren Sie eine Ordnerressource für jede Konfiguration (d.h. wiederholen Sie Schritt 3 für die verschiedenen von Ihnen erstellten Konfigurationen).
4. Klicken Sie im Dialogfeld **Globale Ressource** auf **OK**, um den Alias und alle seine Konfigurationen als globale Ressource zu speichern. Die globale Ressource wird im Dialogfeld **Globale Ressourcen verwalten** unter "Ordner" aufgelistet.

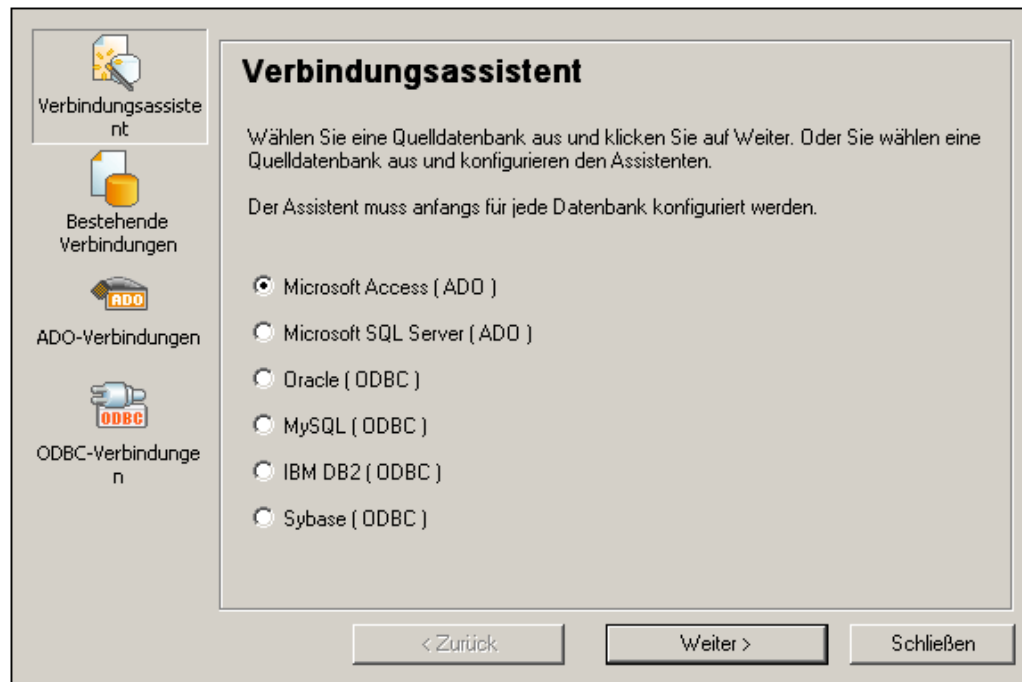
6.1.3 Datenbanken

Im Dialogfeld **Globale Ressource** für Datenbanken (*Abbildung unten*) können Sie auf folgende Weise eine Datenbankressource hinzufügen:



Geben Sie einen Aliasnamen ein.

1. Im Fenster "Konfigurationen" finden Sie eine Konfiguration mit dem Namen "Default" (*Abbildung oben*). Diese Standardkonfiguration kann nicht gelöscht werden und der Name kann nicht geändert werden. Sie können beliebig viele zusätzliche Konfigurationen für den ausgewählten Alias hinzufügen. Klicken Sie dazu auf die Schaltfläche **Konfiguration hinzufügen**  und geben Sie im angezeigten Dialogfeld **Konfiguration hinzufügen** den Konfigurationsnamen ein. Klicken Sie auf **OK**. Die neue Konfiguration wird im Fenster "Konfigurationen" aufgelistet. Wiederholen Sie diesen Schritt um so viele Konfigurationen für diesen Alias (globale Ressource) hinzuzufügen, wie erforderlich.
2. Wählen Sie im Fenster "Konfigurationen" eine der Konfigurationen aus und klicken Sie auf die Schaltfläche **Datenbank wählen**. Daraufhin wird das Dialogfeld **Verbindung zu Globaler Ressource erstellen** angezeigt (*Abbildung unten*).
- 3.



Wählen Sie aus, ob Sie die Verbindung zur Datenbank mit dem Verbindungsassistenten herstellen, eine bestehende Verbindung verwenden möchten oder eine ADO-Verbindung oder eine ODBC-Verbindung herstellen möchten.

6.1.4 Kopieren von Konfigurationen

Im Dialogfeld **Globale Ressourcen verwalten** können Sie bestehende Konfigurationen für alle Konfigurationstypen duplizieren. Dazu wählen Sie eine Konfiguration und klicken auf das

Symbol **Konfiguration kopieren** . Dann wählen Sie einen Konfigurationsnamen aus oder geben einen neuen ein und klicken auf **OK**. Dadurch wird eine Kopie der ausgewählten Konfiguration erstellt, die Sie nun entsprechend bearbeiten können.

Siehe auch:

Extras | Globale Ressourcen: Beschreibung zum Menübefehl zum Aufrufen des Dialogfelds "Globale Ressourcen verwalten".

Extras | Aktive Konfiguration: Hier wird der Menübefehl beschrieben, mit dem die aktive Konfiguration der Applikation geändert werden kann.

Symbolleisten und Statusleiste: Hier finden Sie Informationen über die Symbolleiste "Globale Ressourcen".

6.2 Verwendung globaler Ressourcen

Es gibt mehrere Arten von globalen Ressourcen (Datei, Ordner und Datenbank). Die verschiedenen Szenarios, in denen Sie globale Ressourcen in Authentic Desktop verwenden können, sind im Unterabschnitt dieses Abschnitts aufgelistet: Dateien und Ordner.

Auswahloptionen, die festlegen, welche Ressource verwendet wird

Es gibt zwei applikationsweite Auswahlmöglichkeiten, um festzulegen, welche globalen Ressourcen verwendet werden können und welche globalen Ressourcen zu einem bestimmten Zeitpunkt tatsächlich verwendet werden:

- *Die aktive XML-Datei für globale Ressourcen* wird im Dialogfeld "Globale Ressource" ausgewählt. Die globalen Ressource-Definitionen, die in der aktiven XML-Datei für globale Ressourcen enthalten sind, stehen zur Verfügung. Nur die Definitionen in der aktiven XML-Datei für globale Ressourcen sind verfügbar. Sie können jederzeit eine andere XML-Datei für globale Ressourcen verwenden. In diesem Fall werden die globalen Ressourcen der vorher aktiven Datei sofort durch die der neuen aktiven Datei ersetzt. Die aktive XML-Datei für globale Ressourcen legt also Folgendes fest: (i) welche globale Ressourcen zugewiesen werden können und (ii) welche globalen Ressourcen angezeigt werden können (Wenn z.B. eine globale Ressource in einer XML-Datei für globale Ressourcen zugewiesen ist, es aber in der aktiven XML-Datei für globale Ressourcen keine globale Ressource dieses Namens gibt, dann kann die zugewiesene globale Ressource (der Alias) nicht angezeigt werden).
- *Die aktive Konfiguration* wird über den Menübefehl **Extras | Aktive Konfiguration** oder über die Symbolleiste "Globale Ressourcen" ausgewählt. Bei Auswahl dieses Befehls wird eine Liste der Konfigurationen für alle Aliasse angezeigt. Bei Auswahl einer Konfiguration wird diese in der gesamten Applikation aktiv. D.h. immer wenn eine globale Ressource (oder ein Alias) verwendet wird, wird die Ressource, die der aktiven Konfiguration der einzelnen verwendeten Aliasse entspricht, geladen. Die aktive Konfiguration wird auf alle verwendeten Aliasse angewendet. Wenn ein Alias keine Konfiguration mit dem Namen der aktiven Konfiguration hat, wird die Standardkonfiguration dieses Aliasses verwendet. Die aktive Konfiguration spielt beim Zuweisen von Ressourcen keine Rolle; Sie ist nur dann von Bedeutung, wenn die Ressourcen tatsächlich verwendet werden.

6.2.1 Zuweisen von Dateien und Ordnern

In diesem Abschnitt wird beschrieben, wie globale Ressourcen vom Typ Datei und Ordner zugewiesen werden. Die Zuweisung für diese beiden Arten erfolgt auf unterschiedliche Weise. In jedem der unten beschriebenen Szenarios erscheint bei Auswahl des Befehls **Zu globalen Ressourcen wechseln** das Dialogfeld "Globale Ressource öffnen" (*Abbildung unten*).



Bei Auswahl einer *globalen Ressource vom Typ Datei* wird die Datei zugewiesen. Bei Auswahl einer *globalen Ressource vom Typ Ordner* wird ein Öffnen-Dialogfeld angezeigt, in dem Sie nach der erforderlichen Datei suchen können. Der Pfad zur ausgewählten Datei wird relativ zur Ordnerressource eingegeben. Wenn also eine *globale Ressource vom Typ Ordner* zwei Konfigurationen hat, von denen jede auf einen anderen Ordner verweist, so könnten Dateien mit demselben Namen, die jedoch in unterschiedlichen Ordnern liegen, über diese zwei Konfigurationen aktiviert werden. Dies kann vor allem für Testzwecke nützlich sein.

Im Dialogfeld "Globale Ressource öffnen" können Sie durch Klicken auf die entsprechende Schaltfläche am unteren Rand des Dialogfelds zur Datei- bzw. URL-Dialogfeld wechseln. Mit der Schaltfläche **Globale Ressourcen verwalten**  im rechten oberen Eck erscheint das Dialogfeld [Globale Ressourcen verwalten](#) dialog.

Verwendungsszenarios

Globale Ressourcen vom Typ Datei und Ordner können in den folgenden Szenarios verwendet werden:

- [Öffnen globaler Ressourcen](#)
- [Speichern als globale Ressource](#)
- [Zuweisen von Dateien zu XSLT-Transformationen](#)
- [XSLT-Transformationen und XQuery-Ausführungen](#)
- [Zuweisen eines SPS](#)

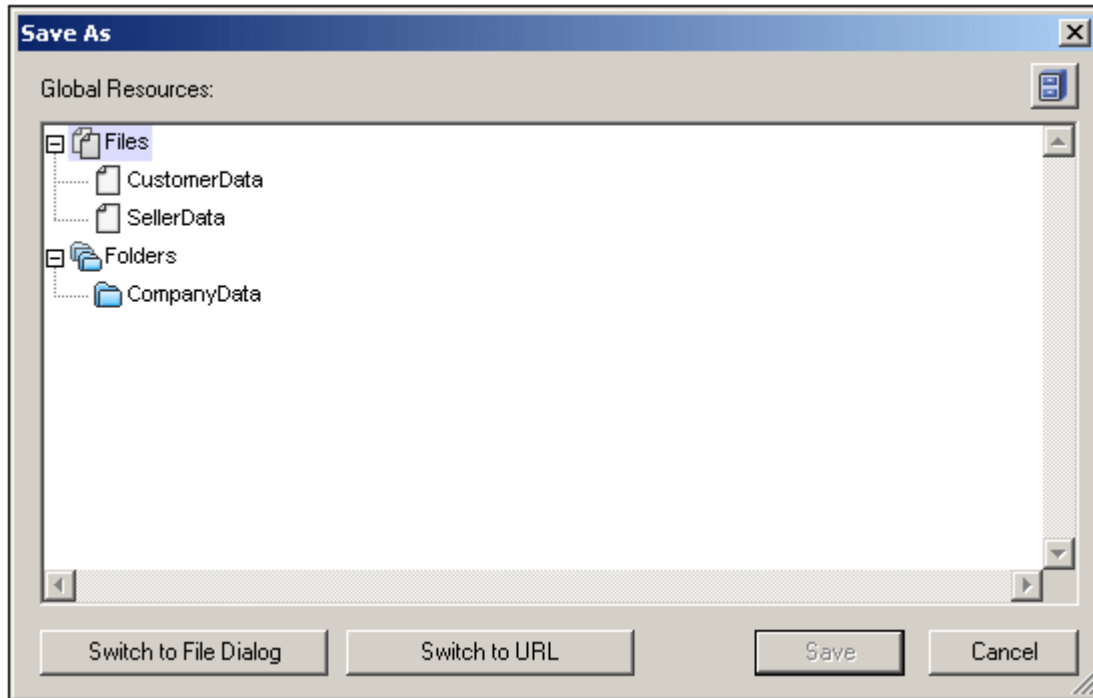
Öffnen von globalen Ressourcen

Eine globale Ressource kann in Authentic Desktop mit dem Befehl [Datei | Öffnen \(Zu globaler Ressource wechseln\)](#) geöffnet und bearbeitet werden. Im Fall einer globalen Ressource vom Typ Datei wird die Datei direkt geöffnet. Im Fall einer globalen Ressource vom Typ Ordner wird ein Öffnen-Dialogfeld aufgerufen, in dem der verknüpfte Ordner ausgewählt ist. Sie können nun in den untergeordneten Ordnern nach der gewünschten Datei suchen. Ein Vorteil der Definition von Dateien mittels globaler Ressourcen für die Bearbeitung ist, dass miteinander in Zusammenhang stehende Dateien unter unterschiedlichen Konfigurationen einer einzigen globalen Ressource gespeichert und nur durch Wechseln der Konfigurationen

aufgerufen werden können. Alle Bearbeitungen müssten vor dem Ändern der Konfiguration gespeichert werden.

Speichern als globale Ressource

Eine neu erstellte Datei kann als globale Ressource gespeichert werden. Sie können auch eine vorhandene Datei öffnen und anschließend als globale Ressource speichern. Wenn Sie auf den Befehl **Datei | Speichern** oder **Datei | Speichern unter** klicken, wird das Dialogfeld "Speichern" aufgerufen. Klicken Sie auf die Schaltfläche **Zu globaler Ressource wechseln**, um die verfügbaren globalen Ressourcen (*Abbildung unten*) aufzurufen. Dies sind die Aliasnamen, die in der aktuellen XML-Datei für globale Ressourcen gespeichert sind.



Wählen Sie einen Alias aus und klicken Sie auf "Speichern". Wenn es sich beim Alias um einen [Dateialias](#) handelt, wird die Datei direkt gespeichert. Wenn es sich um einen [Ordneralias](#) handelt, wird ein Dialogfeld angezeigt, in dem Sie den Namen angeben müssen, unter dem die Datei gespeichert werden soll. In beiden Fällen wird die Datei unter dem Pfad gespeichert, der für die [gerade aktive Konfiguration](#) definiert wurde.

Anmerkung: Jede Konfiguration verweist auf einen bestimmten Dateipfad, der in der Definition dieser Konfiguration festgelegt ist. Wenn die Datei, die Sie als globale Ressource speichern, nicht dieselbe Dateityperweiterung wie die Datei unter dem aktuellen Pfad der Konfiguration hat, kann es beim Öffnen der globalen Ressource in Authentic Desktop zu Bearbeitungs- und Validierungsfehlern kommen. Der Grund dafür ist, dass Authentic Desktop die Datei unter der Annahme öffnet, dass die Datei den in der Konfigurationsdefinition angegebenen Dateityp aufweist.

Zuweisen von Dateien für XSLT-Transformationen

XSLT-Dateien können XML-Dokumenten und XML-Dateien können XSLT-Dokumenten über globale Ressourcen zugewiesen werden. Bei Auswahl der Befehle zum Zuweisen von XSLT-Dateien (**XSL/XQuery | XSL zuweisen** und **XSL/XQuery | XSL:FO zuweisen**) und XML-Dateien (**XSL/XQuery | XML-Beispieldatei zuweisen**) wird das Zuweisungsdialogfeld aufgerufen. Wenn Sie auf die Schaltfläche **Durchsuchen** klicken, wird das Dialogfeld "Öffnen" angezeigt, in dem Sie zum Dialogfeld "Globale Ressource öffnen" wechseln können. Ein großer

Vorteil bei Verwendung einer globalen Ressource zur Definition einer XSLT-Transformation ist, dass die XSLT-Datei (oder die XML-Datei) für eine Transformation einfach durch Wechseln der aktiven Konfiguration in Authentic Desktop geändert werden kann; Sie müssen nicht jedes Mal, wenn Sie eine Transformation mit einer anderen Datei durchführen wollen, eine neue Dateizuweisung vornehmen. Wenn eine XSLT-Transformation gestartet wird, wird/werden dafür die Datei/die Dateien verwendet, die mit der aktiven Konfiguration verknüpft sind.

XSLT-Transformationen und XQuery-Ausführungen

Bei Auswahl des Befehls [XSL/XQuery | XSL-Transformation](#), [XSL/XQuery | XSL:FO-Transformation](#) oder [XSL/XQuery | XQuery-Ausführung](#) wird ein Dialogfeld angezeigt, in dem Sie nach der gewünschten XSLT-, XQuery- oder XML-Datei suchen können. Klicken Sie auf die Schaltfläche **Durchsuchen** und anschließend auf **Zu globaler Ressource wechseln** um das Dialogfeld "Globale Ressource öffnen" aufzurufen ([Abbildung ganz oben im Abschnitt](#)). Für die Transformation wird die Datei verwendet, die mit der aktuellen aktiven Konfiguration der ausgewählten globalen Ressource verknüpft ist.

Zuweisen eines SPS

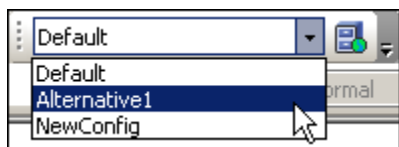
Bei der Zuweisung eines StyleVision Stylesheet zu einer XML-Datei (**Authentic | StyleVision Stylesheet zuweisen**) können Sie eine globale Ressource als Pfad für das Stylesheet auswählen. Klicken Sie auf die Schaltfläche **Zu globaler Ressource wechseln**, um das Dialogfeld "Globale Ressource öffnen" aufzurufen ([Abbildung ganz oben im Abschnitt](#)). Wenn für die Zuweisung eine globale Ressource ausgewählt ist, kann die Authentic-Ansicht des XML-Dokuments einfach durch Wechseln der aktiven Konfiguration in Authentic Desktop geändert werden.

6.2.2 Wechseln von Konfigurationen

Es kann immer nur eine globale Ressourcenkonfiguration aktiv sein und diese ist applikationsweit aktiv. D.h. die aktive Konfiguration ist für alle Aliasse in den derzeit offenen Dateien aktiv. Wenn ein Alias keine Konfiguration mit dem Namen der aktiven Konfiguration hat, wird die Standardkonfiguration für diesen Alias verwendet.

Betrachten wir als Beispiel, wie man eine Konfiguration wechselt, einen Fall, in dem einem XML-Dokument mehrere XSLT-Transformationsdateien über eine globale Ressource mit mehreren Konfigurationen zugewiesen wurde. gewechselt werden, indem Sie einfach die Konfiguration der globalen Ressource ändern. Dies kann auf zwei Arten erfolgen:

- Wenn Sie die Maus über den Menübefehl **Extras | Aktive Konfiguration** platzieren, wird ein Untermenü mit einer Liste aller Konfigurationen in der XML-Datei für globale Ressourcen angezeigt. Wählen Sie die gewünschte Konfiguration aus.
- Wählen Sie in der Auswahlliste der Symbolleiste "Globale Ressourcen" (*Abbildung unten*) die gewünschte Konfiguration aus. (Die Symbolleiste "globale Ressourcen" kann über den Menübefehl **Extras | Anpassen | Symbolleisten | Globale Ressourcen** ein- und ausgeblendet werden.)



Die XSLT-Datei wird sofort gewechselt.

Auf diese Art können Sie durch Ändern der aktiven Konfiguration Quelldateien auswechseln, die über eine globale Ressource zugewiesen sind.

7 Authentic Desktop in Visual Studio

Sie können Authentic Desktop in die Microsoft Visual Studio IDE-Versionen 2003, 2005 und 2008 integrieren. Auf diese Weise stehen Ihnen sowohl die XML-Bearbeitungsfunktionen als auch die professionelle Entwicklungsumgebung von Visual Studio zur Verfügung:

In diesem Abschnitt wird Folgendes beschrieben:

- der [allgemeine Installationsvorgang](#) und die Integration des Authentic Desktop Plugin in Visual Studio.
- [Unterschiede](#) zwischen der Visual Studio -Version und der Standalone-Version.

7.1 Installieren des Authentic Plugin

Um das Authentic Desktop Plugin für Visual Studio zu installieren, müssen Sie Folgendes tun:

- Installieren Sie Microsoft Visual Studio
- Installieren Sie Authentic Desktop
- Downloaden Sie das Authentic Desktop Integrationspaket für Microsoft Visual Studio. Dieses Paket steht auf der Authentic Desktop Download-Seite unter www.altova.com zur Verfügung. (**Bitte beachten Sie:** Sie müssen das richtige Integrationspaket für Ihre jeweilige Authentic Desktop Version (die aktuelle Version ist 2009) verwenden.

Nach Installation des Integrationspakets können Sie Authentic Desktop in der Visual Studio-Umgebung verwenden.

So aktivieren Sie das Plug-In:

Wenn das Plugin nicht automatisch bei der Installation aktiviert wurde, gehen Sie folgendermaßen vor:

1. Gehen Sie zum **Verzeichnis**, in dem die ausführbare Visual Studio IDE Datei installiert wurde, z.B. `C:\Program Files\MS Visual Studio\Common7\IDE`
2. Geben Sie in die Befehlszeil den folgenden Befehl ein: `devenv.exe /setup`.
3. Warten, Sie bis der Vorgang abgeschlossen ist, bevor Sie die Applikation in Visual Studio öffnen.

7.2 Unterschiede zur Standalone-Version

In diesem Abschnitt werden die Unterschiede aufgelistet, die zwischen den Visual Studio-Versionen und den Standalone-Versionen von Authentic Desktop bestehen.

Eingabehilfen (Tool-Fenster in Visual Studio)

Die Eingabehilfen von Authentic Desktop stehen in Visual Studio als Tools-Fenster zur Verfügung. Beachten Sie dazu die folgenden Punkte. (Eine Beschreibung der Eingabehilfen und der Benutzeroberfläche von Authentic Desktop finden Sie im Abschnitt [Einführung](#)).

- Sie können Eingabehilfenfenster an jede beliebige Stelle in der Entwicklungsumgebung ziehen.
- Wenn Sie mit der rechten Maustaste auf das Register einer Eingabehilfe klicken, können Sie die Benutzeroberfläche weiter anpassen. Die Konfigurationsoptionen für Eingabehilfen sind: andockbar, ausblenden, frei schwebend und automatisch ausblenden.

Authentic Desktop Befehle als Visual Studio-Befehle

Einige Authentic Desktop Befehle stehen auf der Benutzeroberfläche von Visual Studio als Visual Studio-Befehle zur Verfügung:

- **Rückgängig, Wiederholen:** Diese Visual Studio-Befehle wirken sich auf alle Aktionen in der Visual Studio Entwicklungsumgebung aus.
- **Projekte:** Authentic Desktop Projekte werden als Visual Studio-Projekte behandelt.
- **Symbolleisten anpassen, Befehle anpassen:** Die Register "Symbolleisten" und "Befehle" im Dialogfeld "Anpassen" (**Extras | Anpassen**) enthalten sowohl Befehle aus Visual Studio als auch Authentic Desktop.
- **Ansichten:** Im Menü **Ansicht** enthält der Befehl **Authentic Desktop** Optionen zum Ein- und Ausblenden der Eingabehilfenfenster und anderer Seitenleisten und zum Wechseln zwischen den Bearbeitungsansichten und zum Ein- und Ausblenden von Bearbeitungshilfen.
- **Authentic Desktop Hilfe:** Dieses Authentic Desktop Menü, wird als Untermenü im Hilfemenü von Visual Studio angezeigt.

8 Authentic Desktop in Eclipse

Eclipse 3.x ist eine Open-Source-Umgebung, in die verschiedene Arten von Applikationen in Form von Plug-Ins integriert werden können.

Das Authentic Desktop Plugin für die Eclipse-Plattform ist ein Eclipse Plug-In, das Ihnen gestattet, auf die Funktionalitäten von Authentic Desktop über die Eclipse 3.2 / 3.3 / 3.4-Plattform zuzugreifen. Es steht auf Windows-Plattformen zur Verfügung. In diesem Abschnitt wird beschrieben, wie man das Authentic Desktop Plugin für Eclipse [installiert](#) und die [Authentic-Perspektive einrichtet](#). Anschließend stehen die Komponenten der Benutzeroberfläche von Authentic Desktop und die Menübefehle von Authentic Desktop über die Benutzeroberfläche von Eclipse zur Verfügung.

Anmerkung: Die OLE-to-Java Bibliothek von Eclipse unterstützt derzeit 64-Bit Windows Systeme nicht.

8.1 Installation des Authentic Desktop Plugin für Eclipse

Stellen Sie vor Installation des Authentic Desktop Plugin für Eclipse sicher, dass die folgenden Komponenten bereits installiert sind:

- Authentic Desktop Edition.
- Java Runtime Environment (JRE) Version 1.5 oder höher, welches für Eclipse benötigt wird. Empfohlen wird JRE5. Nähere Informationen dazu finden Sie auf der [Eclipse Website](#).
- Eclipse-Plattform 3.2, 3.3 oder 3.4.

Nachdem Sie diese Komponenten installiert haben, können Sie das Authentic Plugin für Eclipse, das im Authentic Integrationspaket (*siehe unten*) enthalten ist, installieren.

Anmerkung zu JRE

Wenn beim Öffnen eines Dokuments die unten stehende Fehlermeldung angezeigt wird,

```
java.lang.UnsupportedClassVersionError: com/altova/...  
(Unsupported major.minor version 49.0)
```

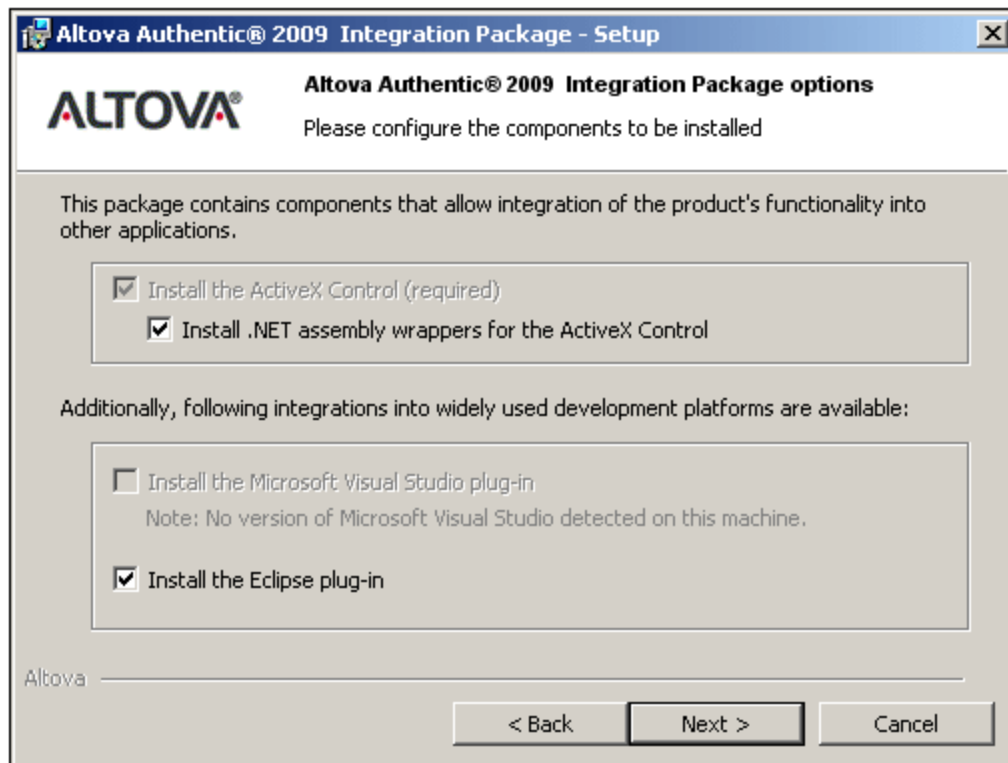
dann weist das darauf hin, dass Eclipse ein älteres JavaRTE verwendet. Eclipse verwendet zum Suchen der Datei `javaw.exe` die Umgebungsvariable `PATH`.

Das Problem kann gelöst werden, indem Sie die `PATH`-Umgebungsvariable ändern, sodass zuerst eine neuere Version gefunden wird. Alternativ dazu können Sie Eclipse auch mit dem Befehlszeilenparameter `-vm` starten und den Pfad zu einer `javaw.exe` Version 1.5 oder höher angeben.

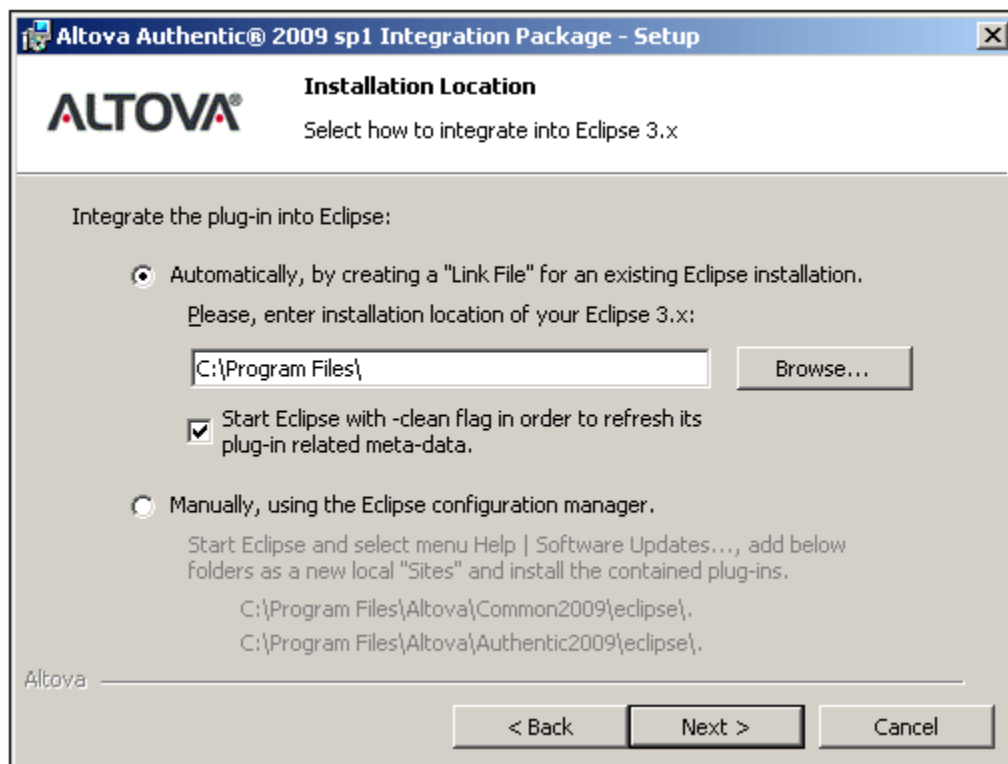
Authentic Integrationspaket

Das Authentic Plugin für Eclipse ist im Authentic-Integrationspaket enthalten und wird mit diesem installiert. Führen Sie die Installation folgendermaßen durch:

1. Stellen Sie sicher, dass Authentic Desktop, JRE und Eclipse bereits installiert sind (*siehe oben*).
2. Laden Sie das Authentic Integrationspaket von der [Komponenten-Download-Seite](#) der [Altova Website](#) herunter und installieren Sie es. Bei der Installation gibt es zwei wichtige Schritte, die weiter unten in Schritt 3 und 4 näher beschrieben sind.
3. Während der Installation des Authentic Integrationspakets erscheint ein Dialogfeld, in dem Sie gefragt werden, ob das Authentic Plugin für Eclipse installiert werden soll (*siehe Abbildung unten*). Überprüfen Sie die Option und klicken Sie auf **Weiter**.



4. Im nächsten Dialogfeld (Eclipse)-Installationspfad (*Abbildung unten*), können Sie auswählen ob der Installationsassistent das Authentic Plugin während der Installation in Eclipse integrieren soll (Option "Automatisch") oder ob Sie das Plug in später (über die Benutzeroberfläche von Eclipse) installieren möchten.



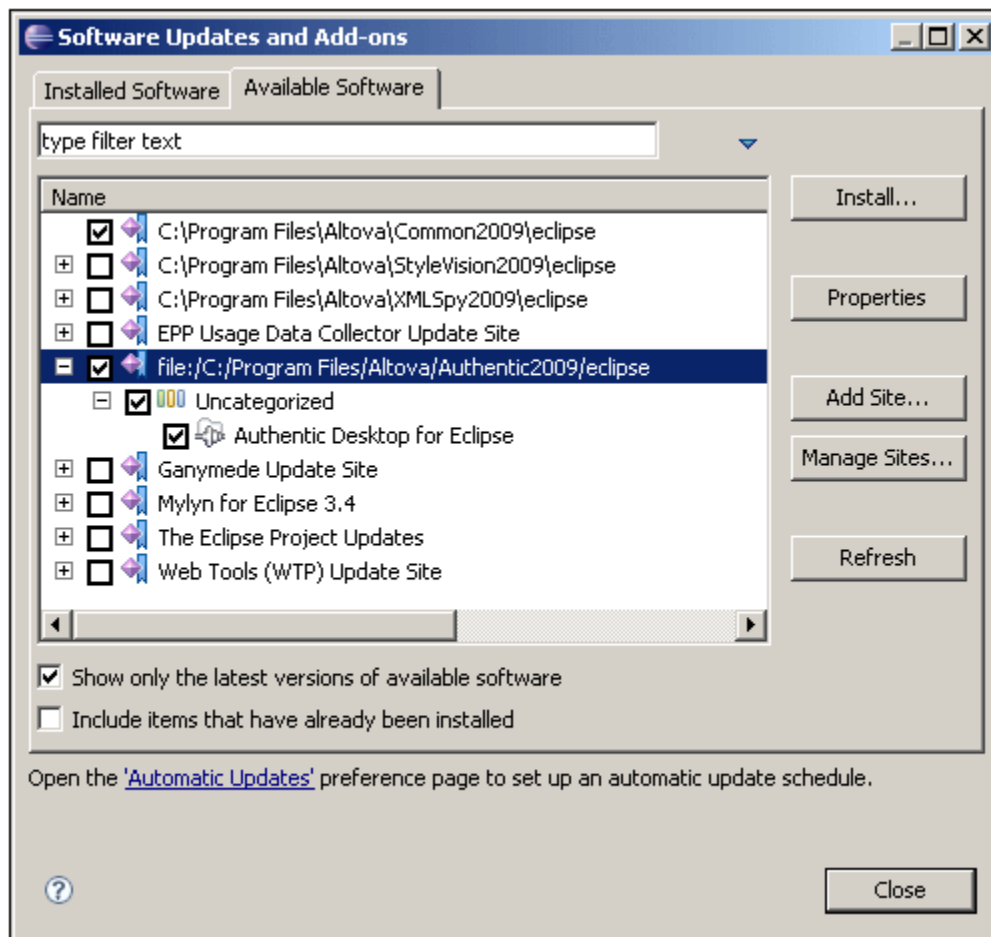
Wir empfehlen die Integration über den Installationsassistenten vorzunehmen. Aktivieren Sie dazu die Option "Automatisch" und suchen Sie nach dem Ordner, in dem sich die ausführbare Datei von Eclipse befindet (`eclipse.exe`). Klicken Sie anschließend auf **Weiter**.

5. Stellen Sie die Installation fertig. Wenn Sie die automatische Integration wählen, wird das Authentic Plugin für Eclipse in Eclipse integriert und steht beim nächsten Start von Eclipse bereits zur Verfügung.

Manuelle Integration des Authentic Plugin in Eclipse

Um das **Authentic** Plugin für Eclipse manuell zu integrieren, gehen Sie folgendermaßen vor:

1. Klicken Sie in Eclipse auf den Menübefehl **Help | Software Updates**.
2. Klicken Sie im Dialogfeld "Software Updates and Add-Ons", das daraufhin angezeigt wird, auf das Register "Available Updates" (*Abbildung unten*).
3. Klicken Sie auf die Schaltfläche **Add Site**.
4. Klicken Sie im Dialogfeld "Add Site" auf die Schaltfläche **Local**.
5. Navigieren Sie zum Ordner `c:\Program Files\Altova\Common2009\eclipse`, wählen Sie ihn aus und klicken Sie auf **OK**.
6. Wiederholen Sie die Schritte 3 bis 5 und wählen Sie diesmal den Ordner `c:\Program Files\Altova\StyleVision2009\eclipse` aus.
7. Die beiden hinzugefügten Ordner werden auf dem Register "Available Software" (*Abbildung unten*) angezeigt. Aktivieren Sie bei jedem Ordner das Kontrollkästchen auf der obersten Ebene, um die Plug-ins auszuwählen und klicken Sie auf die Schaltfläche **Install**.



8. Daraufhin wird ein Dialogfeld geöffnet, in dem Sie überprüfen können, ob die angehakten Komponenten installiert werden.
9. Klicken Sie zum Fortfahren auf **Next**. Daraufhin wird das Dialogfeld "Review License" angezeigt.
10. Lesen Sie die Lizenzbedingungen und klicken Sie auf *I accept the terms....* , wenn Sie damit einverstanden sind. Klicken Sie anschließend auf **Finish**, um die Installation fertig zu stellen.

Falls es Probleme mit dem Plug-in gibt (z.B. fehlende Symbole), starten Sie Eclipse mit dem `-clean` Flag.

Derzeit installierte Version

Um zu sehen, welche Version des Authentic Plugin für Eclipse installiert ist, wählen Sie die Eclipse-Menüoption **Help | About Eclipse Platform**. Wählen Sie anschließend das Symbol Authentic.

8.2 Einstiegspunkte für Authentic in Eclipse

Um die Funktionalitäten von StyleVision in Eclipse aufzurufen, können Sie die folgenden Einstiegspunkte verwenden::

- [Authentic Desktop-Perspektive](#): Hier stehen Ihnen die Funktionen der Authentic Desktop Benutzeroberfläche über die Benutzeroberfläche von Eclipse zur Verfügung.
- [Authentic Desktop -Symbolleisten-Schaltflächen](#): Hier haben Sie Zugriff auf die Authentic Desktop Hilfe und die Funktion zum Erstellen eines neuen Dokuments.

Authentic Desktop-Perspektive

Eine Perspektive ist in Eclipse eine konfigurierte Ansicht der Benutzeroberfläche mit Funktionalitäten aus verschiedenen Applikationen. Wenn das Authentic Plugin für Eclipse in Eclipse integriert wird, wird automatisch eine Authentic-Standardperspektive erstellt. Diese Perspektive ist eine Benutzeroberfläche, die die Elemente der Benutzeroberfläche von Authentic Desktop enthält: Bearbeitungsansichten, Eingabehilfen und andere Seitenleisten.

Wenn eine Datei geöffnet wird, deren Dateityp (`.sps`) mit Authentic Desktop verknüpft ist, kann diese Datei in der Authentic Perspektive bearbeitet werden. Ebenso kann eine Datei eines anderen Dateityps in einer anderen Perspektive von Eclipse geöffnet werden. Zusätzlich dazu können Sie die Perspektive für jeden Dateityp wechseln und so die Datei in einer anderen Umgebung bearbeiten bzw. verarbeiten. Perspektiven haben daher zwei wichtige Vorteile:

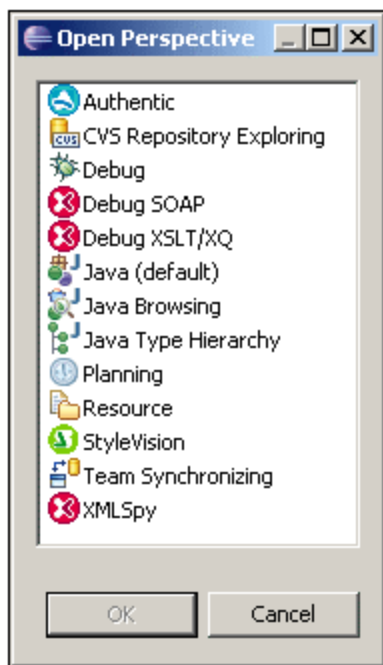
1. Die Arbeitsumgebung für die aktive Datei kann schnell gewechselt werden und
2. Sie können zwischen Dateien wechseln, ohne eine neue Entwicklungsumgebung öffnen zu müssen (die mit der Datei verknüpfte Umgebung steht in einer Perspektive zur Verfügung).

Um mit der Authentic Perspektive zu arbeiten, müssen Sie Folgendes tun:

- Wechseln in die Authentic Perspektive.
- Festlegen der Einstellungen für die Authentic Perspektive.
- Anpassung der Authentic Perspektive.

Wechseln in die Authentic Perspektive

Wählen Sie in Eclipse den Befehl **Window | Open Perspective | Other**. Wählen Sie im Dialogfeld, das jetzt angezeigt wird (*Abbildung unten*), **Authentic** aus und klicken Sie auf **OK**.

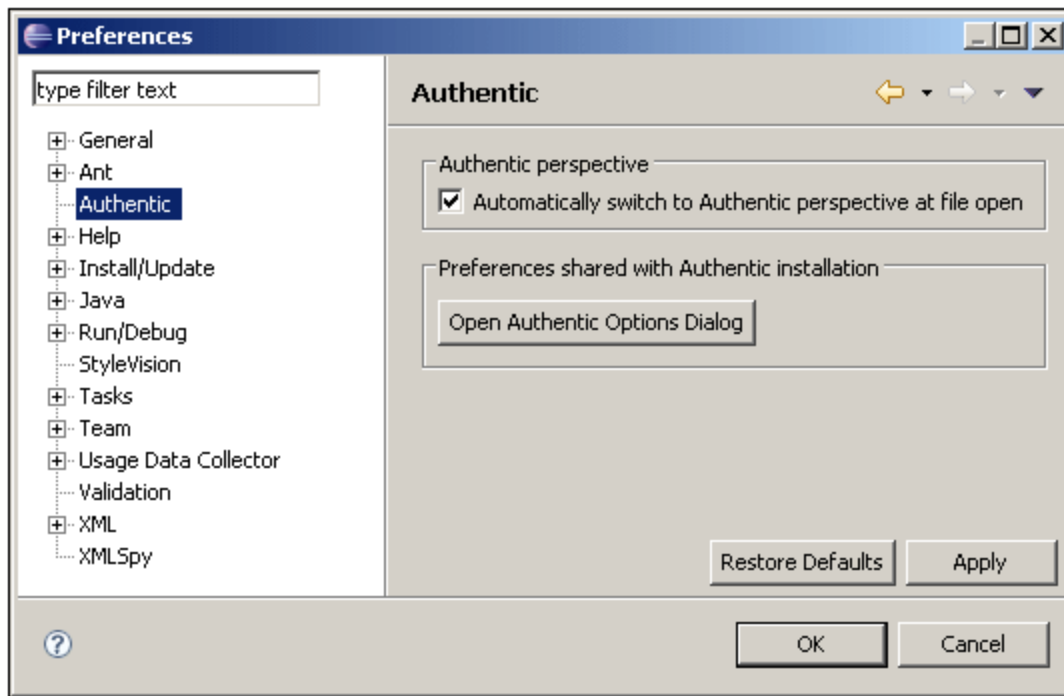


Das leere Fenster bzw. das aktive Dokument hat nun die Authentic Perspektive. Auf diese Art kann der Benutzer die Perspektive über das Menü wechseln. Um eine Perspektive von einer anderen Perspektive aus schneller aufrufen zu können, kann die gewünschte Perspektive im Untermenü **Open Perspective** oberhalb des Menübefehls **Other** aufgelistet werden; diese Einstellung kann im Anpassungsdialogfeld (*siehe weiter unten*) vorgenommen werden.

Perspektiven können auch gewechselt werden, wenn eine Datei geöffnet oder zur aktiven gemacht wird. Die Perspektive der Applikation, die mit dem Dateityp einer Datei verknüpft ist, wird automatisch geöffnet, wenn diese Datei das erste Mal geöffnet wird. Vor dem Wechseln der Perspektive wird ein Dialogfeld angezeigt, in dem Sie gefragt werden, ob die Standardperspektive automatisch mit diesem Dateityp verknüpft werden soll (*Abbildung unten*). Aktivieren Sie die Option *Do Not Ask Again*, wenn die Perspektive mit dem Dateityp verknüpft werden soll, ohne dass Sie jedes Mal gefragt werden, wenn eine Datei dieses Typs geöffnet wird, und klicken Sie anschließend auf **OK**.

Festlegen der Einstellungen für die Authentic Perspektive

Zu den Einstellungen einer Perspektive gehören: (i) eine Einstellung, um die Perspektive automatisch zu ändern, wenn eine Datei eines verknüpften Dateityps geöffnet wird (*siehe oben*) und (ii) Optionen zum Inkludieren oder Ausblenden einzelner Authentic-Symboleisten. Wählen Sie den Menübefehl **Window | Preferences** (*Abbildung unten*), um das Dialogfeld "Preferences" aufzurufen.



Wählen Sie in der Liste der Perspektiven auf der linken Seite **Authentic** aus und wählen Sie dann die gewünschten Einstellungen aus. Klicken Sie anschließend auf **OK**.

Anpassen der Authentic Perspektive

Mit Hilfe der Anpassungsoptionen können Sie festlegen, welche Shortcuts und Befehle in der Perspektive enthalten sein sollen. Um dieses Dialogfeld aufzurufen machen Sie die Perspektive zur aktiven und wählen Sie den Befehl **Window | Customize Perspective** aus.

Auf dem Register "Shortcuts" des Dialogfelds "Customize Perspective" können Sie die Shortcuts für Untermenüs festlegen. Aktivieren Sie in der Untermenü-Auswahlliste das gewünschte Untermenü. Wählen Sie anschließend eine Shortcut-Kategorie aus und aktivieren Sie die Shortcuts, die in der Perspektive inkludiert werden sollen.

Auf dem Register "Commands" können Sie Befehlsgruppen hinzufügen. Um die Befehle in einer Befehlsgruppe anzuzeigen, wählen Sie aus den verfügbaren Befehlsgruppen (die im Bereich "Command Groups" angezeigt werden) die gewünschte Gruppe aus. Die Befehle in dieser Gruppe werden auf der rechten Seite hierarchisch geordnet im Menü, in dem sie zu sehen sein werden, angezeigt. Damit die Befehlsgruppe inkludiert wird, aktivieren Sie ihr Kontrollkästchen.

Klicken Sie auf **OK**, um die Anpassung abzuschließen und damit die Änderungen wirksam werden.

Authentic Desktop Symbolleisten-Schaltflächen

Es werden automatisch zwei <%AUTH-DESK%-Schaltflächen in der Symbolleiste angelegt (Abbildung unten).



Diese dienen zum: (i) Öffnen der Authentic Desktop Hilfe und (ii) Erstellen neuer Authentic-Dokumente.

9 Benutzerreferenz

Im Referenzteil der Dokumentation finden Sie eine Beschreibung aller Authentic Desktop Menübefehle und deren Verwendung. Wir haben versucht, das Handbuch so umfassend wie möglich zu gestalten. Sollten Sie trotzdem Fragen zu Themen haben, die in der vorliegenden Dokumentation nicht behandelt werden, werfen Sie bitte einen Blick in die FAQs und die Diskussionsforen auf der Altova Website. Sollten Ihre Fragen damit noch immer nicht beantwortet sein, kontaktieren Sie uns bitte über unser [Support Center](#) auf der Altova Website.

Bitte beachten Sie, dass im Menü [Datei](#) und [Bearbeiten](#) alle Windows-Standardbefehle und alle zusätzlichen Befehle im Zusammenhang mit XML- und Internetfunktionalitäten unterstützt werden.

9.1 Menü "Datei"

Das Menü "Datei" enthält alle Befehle, die zum Arbeiten mit Dateien benötigt werden. Die Befehle sind in der aus Windows Software-Produkten bekannten Reihenfolge angeordnet.

Zusätzlich zu den Standardbefehlen [Neu](#), [Öffnen](#), [Speichern](#), [Drucken](#), [Druckereinrichtung](#), und [Beenden](#) bietet Authentic Desktop eine Reihe von XML- und applikationsspezifischen Befehlen.

9.1.1 Neu

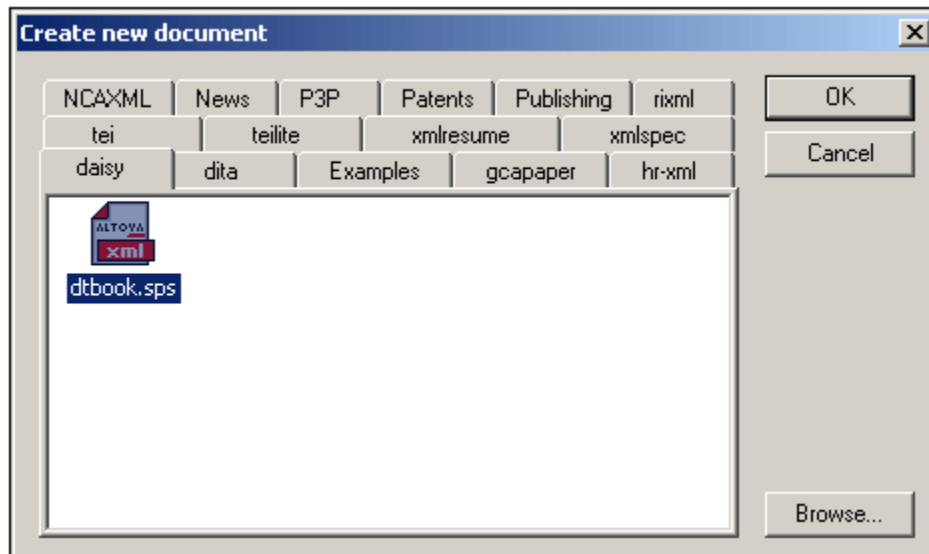


Strg+N

Der Befehl **Neu** wird verwendet, um ein neues Dokument zu erstellen.

Zuweisen eines StyleVision Power Stylesheet beim Erstellen eines neuen Dokuments

Bei Erstellung eines neuen XML-Dokuments Sie es mit einem StyleVision Power Stylesheet (.sps-Datei) verknüpfen, um es in der Authentic-Ansicht anzeigen zu können. Wenn Sie klicken, erscheint das Dialogfeld "Neues Dokument anlegen" (siehe Abbildung unten).



Auf den verschiedenen Ordner-Registern dieses Dialogfelds stehen Ihnen die gewünschten StyleVision Power Stylesheets zur Verfügung. Alternativ dazu können Sie das gewünschte StyleVision Power Stylesheet auch über die Schaltfläche **Durchsuchen...** suchen. Die Register in diesem Dialogfeld entsprechen den Ordnern im Ordner `sps/Template` im Programmordner der Applikation.

9.1.2 Öffnen

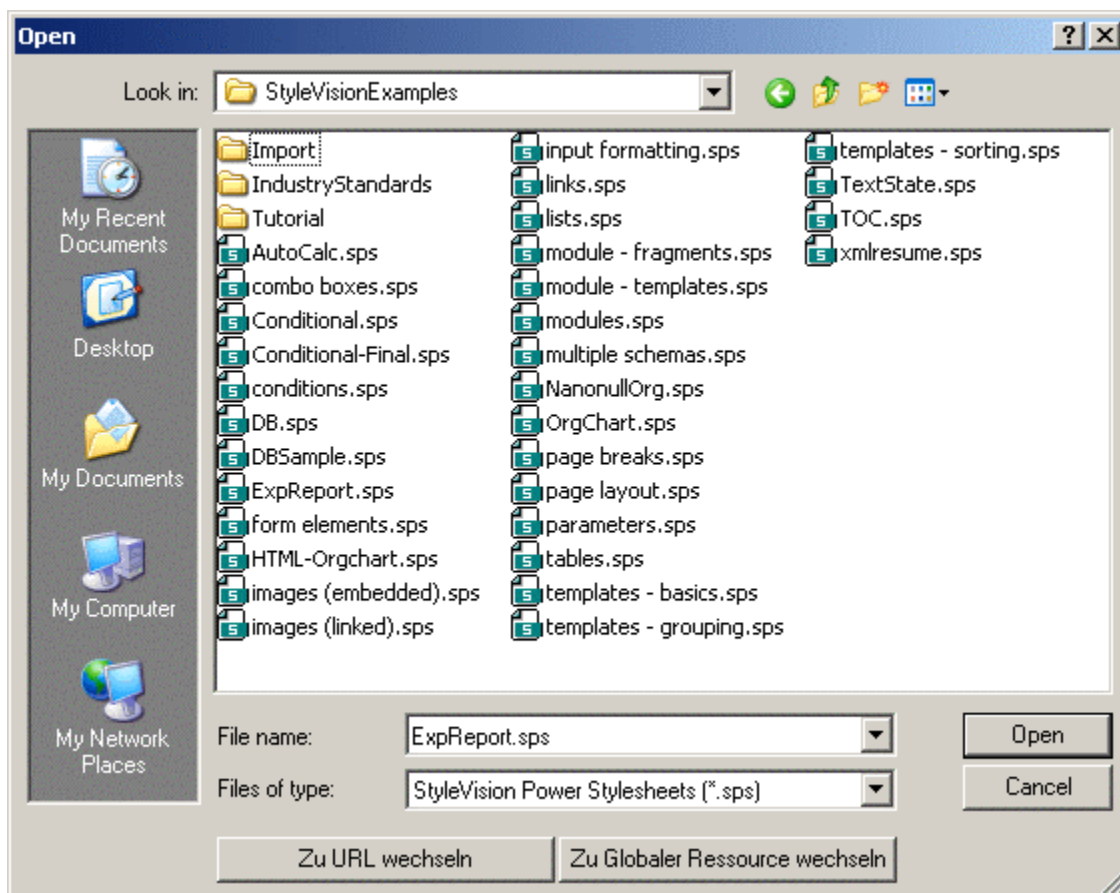


Strg+O

Mit **Öffnen** rufen Sie das bekannte Windows-Dialogfeld "Öffnen" auf, über das Sie jedes XML-Dokument oder jede Textdatei öffnen können. Sie haben auch die Möglichkeit, mehrere Dateien auszuwählen. Im Listenfeld "Dateityp" können Sie auswählen, welche Dateien im Dialogfeld angezeigt werden sollen (die Liste der verfügbaren Dateiformate kann auf dem [Register "Dateiformate" unter Extras | Optionen](#) eingestellt werden). Beim Öffnen einer XML-Datei wird diese auf Wohlgeformtheit überprüft. Ist sie nicht wohlgeformt, wird eine Fehlermeldung angezeigt. Beheben Sie den Fehler und wählen Sie den Menübefehl **Wohlgeformtheit prüfen (F7)**, um die Datei erneut zu überprüfen. Wenn Sie die Option [Validierung beim Öffnen der Datei](#) aktiviert haben und die Datei ungültig ist, wird ebenfalls eine Fehlermeldung angezeigt. Beheben Sie den Fehler und klicken Sie auf **XML | XML validieren (F8)**, um das Dokument erneut zu validieren.

Auswählen von Dateien über URLs und globale Ressourcen

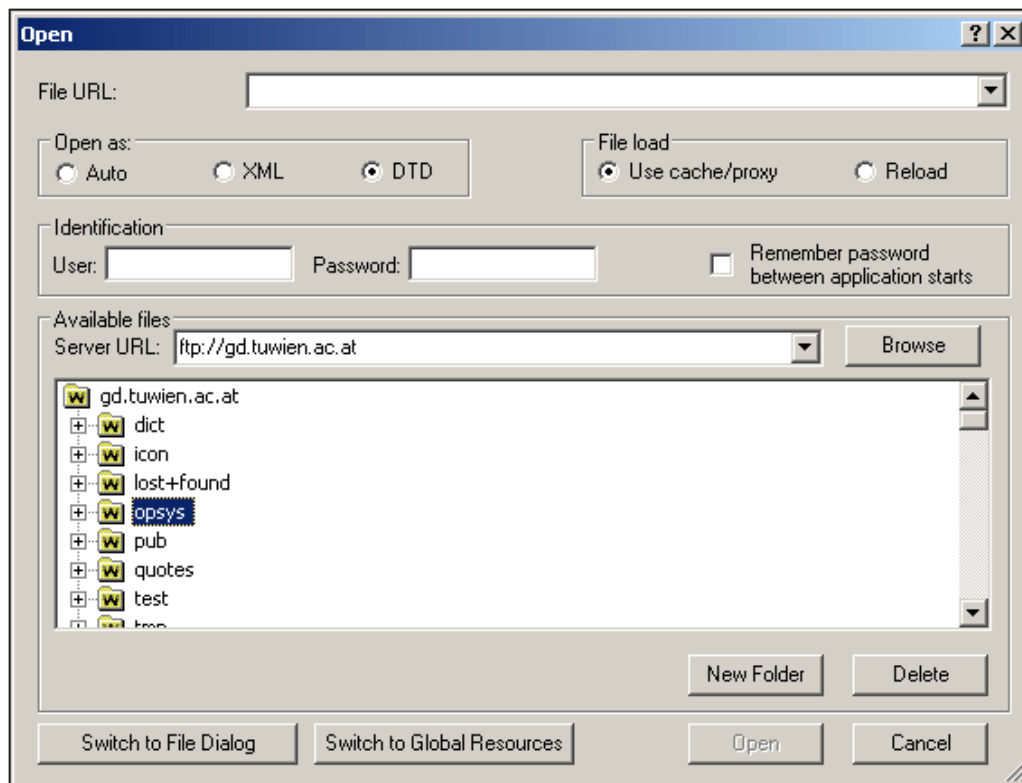
In einigen Dialogfeldern zum Öffnen und Speichern von Dateien können Sie auswählen, ob Sie die gewünschte Datei auswählen oder eine Datei über eine URL oder eine globale Ressource (siehe *Abbildung unten*) speichern möchten. Wählen Sie den Befehl **Zu URL wechseln** oder **Zu globaler Ressource wechseln** um eine dieser Optionen auszuwählen.



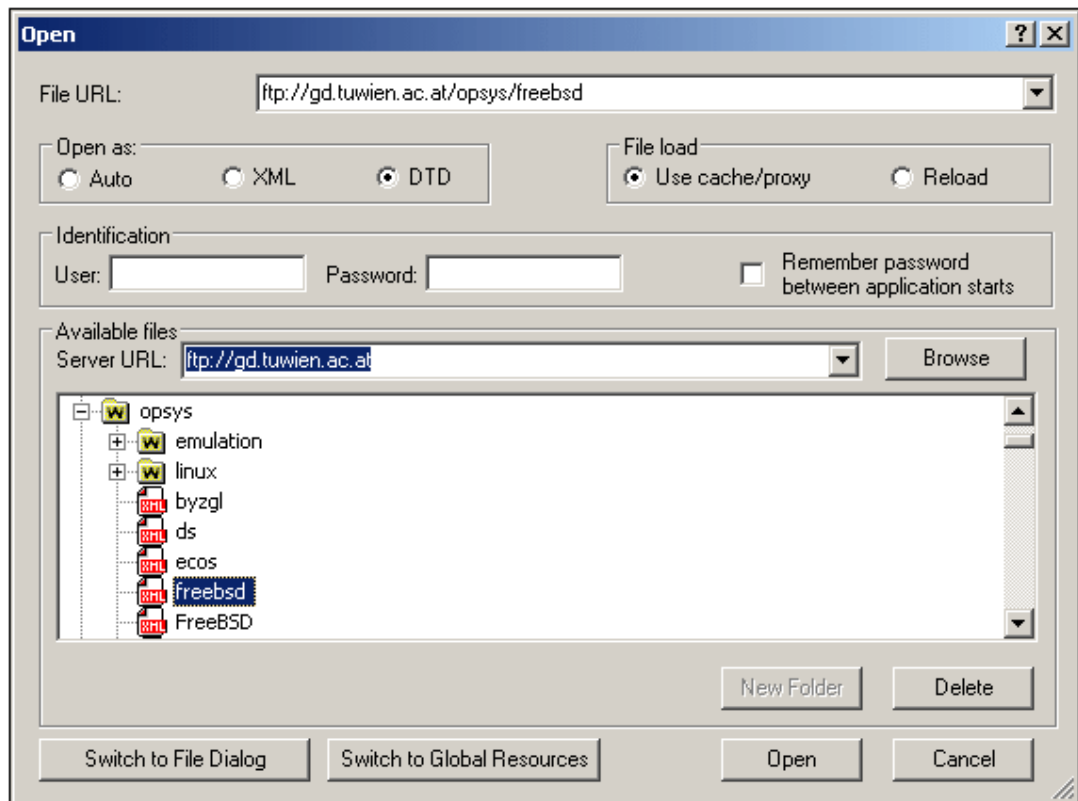
Auswählen von Dateien über URLs

Um eine Datei über eine URL auszuwählen, gehen Sie folgendermaßen vor:

1. Klicken Sie auf den Befehl **Zu URL wechseln**. Daraufhin wird das Dialogfeld "URL öffnen" aufgerufen (*Abbildung unten*).



2. Geben Sie in das Feld *Server URL* die gewünschte URL ein.
3. Geben Sie die Benutzer-ID und das Passwort in die entsprechenden Felder ein, wenn der Server durch ein Passwort geschützt ist.
4. Klicken Sie auf **Durchsuchen**, um die Verzeichnisstruktur des Servers anzuzeigen und darin zu navigieren.
5. Klicken Sie auf die gewünschte Datei.



Die Datei-URL wird im Feld "Datei-URL" angezeigt. Die Schaltfläche OK wird erst jetzt aktiv.

6. Klicken Sie auf **OK**, um die Datei zu laden. Die Datei wird im Hauptfenster angezeigt.

Hinweis: Die Funktion "Durchsuchen" steht nur auf Servern zur Verfügung, die das FTP-, HTTP- und HTTPS-Protokoll (wenn der Server WebDAV unterstützt) unterstützen sowie auf Servern, die WebDAV unterstützen.

Hinweis: Um mehr Kontrolle über den Ladevorgang zu haben, können Sie die Datei über den lokalen Cache oder einen Proxy-Server laden (wodurch der Vorgang erheblich schneller geht, wenn die Datei bereits zuvor einmal geladen wurde). Alternativ dazu können Sie die Datei auch neu laden, wenn Sie mit einem EPS-System oder einem Datenbank-System arbeiten; Wählen Sie in diesem Fall die Option **Neu laden**.

Öffnen und Speichern von Dateien über globale Ressourcen

Um eine Datei über eine globale Ressource zu öffnen oder zu speichern, klicken Sie auf **Zu globaler Ressource wechseln**. Daraufhin wird ein Dialogfeld angezeigt, in dem Sie die globale Ressource auswählen können. Diese Dialogfelder werden im Abschnitt Verwenden globaler Ressourcen näher beschrieben. Eine allgemeine Beschreibung globaler Ressourcen finden Sie in der Dokumentation im Abschnitt [Globale Ressourcen](#)

9.1.3 Neu laden

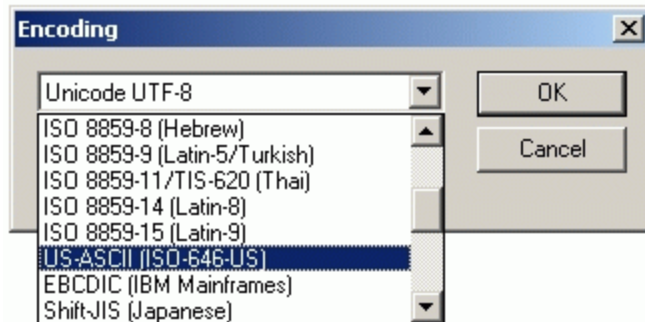


Mit **Neu laden** können Sie geöffnete Dokumente erneut laden.

Dieser Befehl ist nützlich, wenn ein geöffnetes Dokument außerhalb von geändert wurde. In diesem Fall werden Sie von gefragt, ob die Datei erneut geladen werden soll. Wenn Sie dies tun, gehen alle Änderungen, die Sie an der Datei seit der letzten Speicherung vorgenommen haben, verloren. Sie können diese Option im Dialogfeld "Optionen" ([Extras | Optionen](#)) ändern.

9.1.4 Kodierung

Mit **Kodierung...** können Sie die Kodierung der in Verwendung befindlichen Datei (XML oder nicht XML) ansehen und eine andere Kodierung auswählen, mit der das aktive Dokument beim nächsten Mal gespeichert wird.



Wenn Sie eine andere Kodierung als die derzeit verwendete auswählen, wird die Kodierungsspezifikation der XML-Deklaration entsprechend geändert. Bei 2-Byte und 4-Byte-Zeichenkodierungen (UTF-16, UCS-2 und UCS-4) können Sie weiters die zu verwendende Bytefolge in der Datei angeben. Eine andere Methode, um die Kodierung eines XML-Dokuments zu ändern, ist, das Kodierungsattribut direkt in der XML-Deklaration des Dokuments zu ändern.

Die Standardkodierung für vorhandene und neue XML- und nicht-XML-Dokumente kann im Dialogfeld ["Optionen" auf dem Register "Kodierung"](#) festgelegt werden.

Anmerkung: Beim Speichern eines Dokumentes prüft automatisch die Kodierungsspezifikation und öffnet ein Dialogfeld, falls es den vom Benutzer angegebenen Kodierungsnamen nicht erkennt. Sollte Ihr Dokument Zeichen enthalten, die in der gewählten Kodierung nicht angezeigt werden können, erhalten Sie eine Warnmeldung beim Speichern der Datei.

9.1.5 Schließen, Alle schließen

Mit **Schließen** beenden Sie das aktive Dokument. Sollten Sie die Datei verändert haben (in der Titelleiste wird der Dateiname mit einem Sternchen "*" angezeigt), werden Sie gefragt, ob Sie die Datei vor dem Schließen speichern möchten.

Mit **Alle Schließen** beenden Sie alle Dokumente. Sollten Sie Dateien verändert haben (in der Titelleiste wird der Dateiname mit einem Sternchen "*" angezeigt), werden Sie gefragt, ob Sie die Dateien vor dem Schließen speichern möchten.

9.1.6 Speichern, Speichern unter, Alles speichern

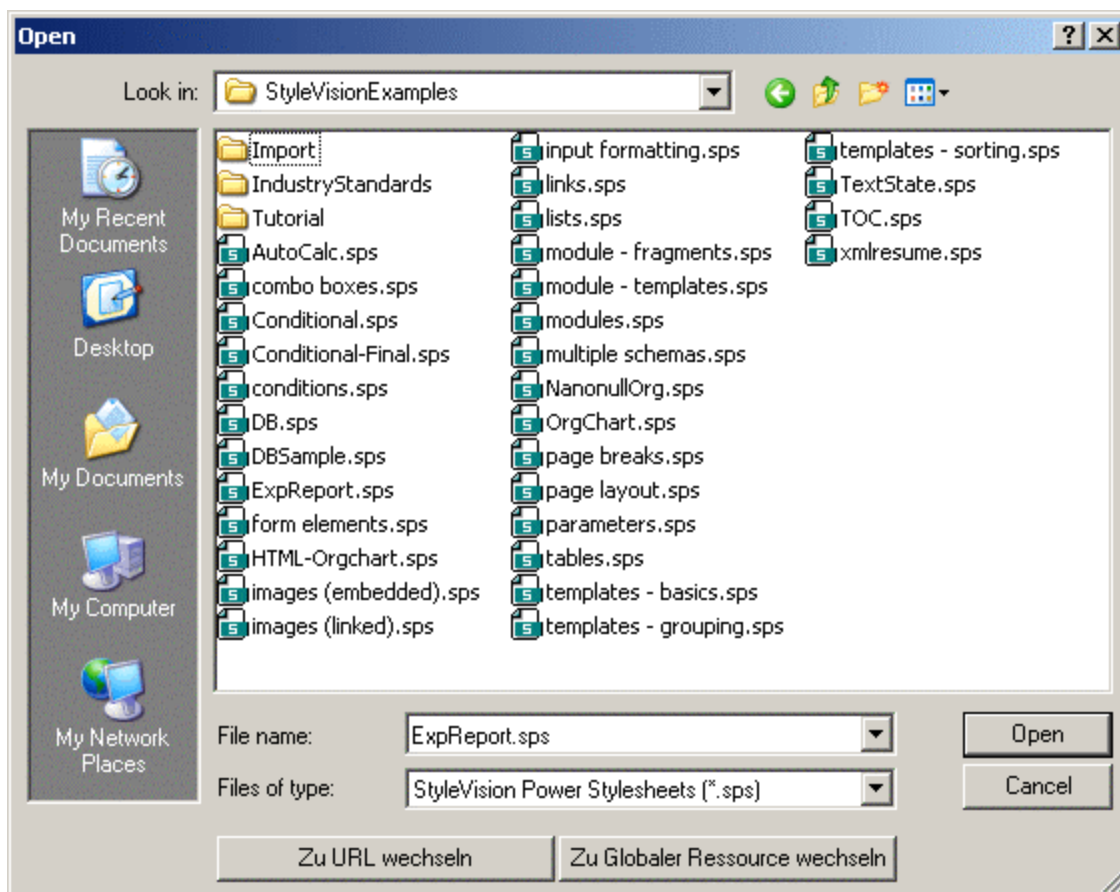
Mit dem Befehl **Speichern**  (**Strg+S**) wird der Inhalt des aktiven Dokuments in der geöffneten Datei gespeichert. Beim Speichern wird das Dokument automatisch auf [Wohlgeformtheit geprüft](#). Außerdem wird das Dokument auch automatisch validiert, wenn diese Option im Dialogfeld "Optionen" ([Extras | Optionen](#)) auf dem Register "Datei" aktiviert wurde. Die XML Deklaration wird auch auf [Kodierungsspezifikationen](#) geprüft. Diese Kodierung wird beim Speichern des Dokuments verwendet.

Mit dem Befehl **Speichern unter** gelangen Sie zu dem aus Windows bekannten Dialogfeld "Speichern unter...". Hier geben Sie den Namen und den Ordner an, in dem die Datei gespeichert werden soll. Bei Verwendung des Befehls **Speichern unter** werden dieselben Validierungen und Prüfungen durchgeführt, wie bei dem Befehl **Speichern**.

Mit dem Befehl **Alles speichern**  werden alle Änderungen gespeichert, die Sie an offenen Dokumenten durchgeführt haben. Dieser Befehl ist hilfreich, wenn Sie gleichzeitig an mehreren Dokumenten arbeiten. Falls ein zu speicherndes Dokument bisher noch nie gespeichert wurde (z.B. nachdem es neu erstellt wurde), wird das Dialogfeld "Speichern unter..." für dieses Dokument angezeigt.

Auswählen von Dateien über URLs und globale Ressourcen

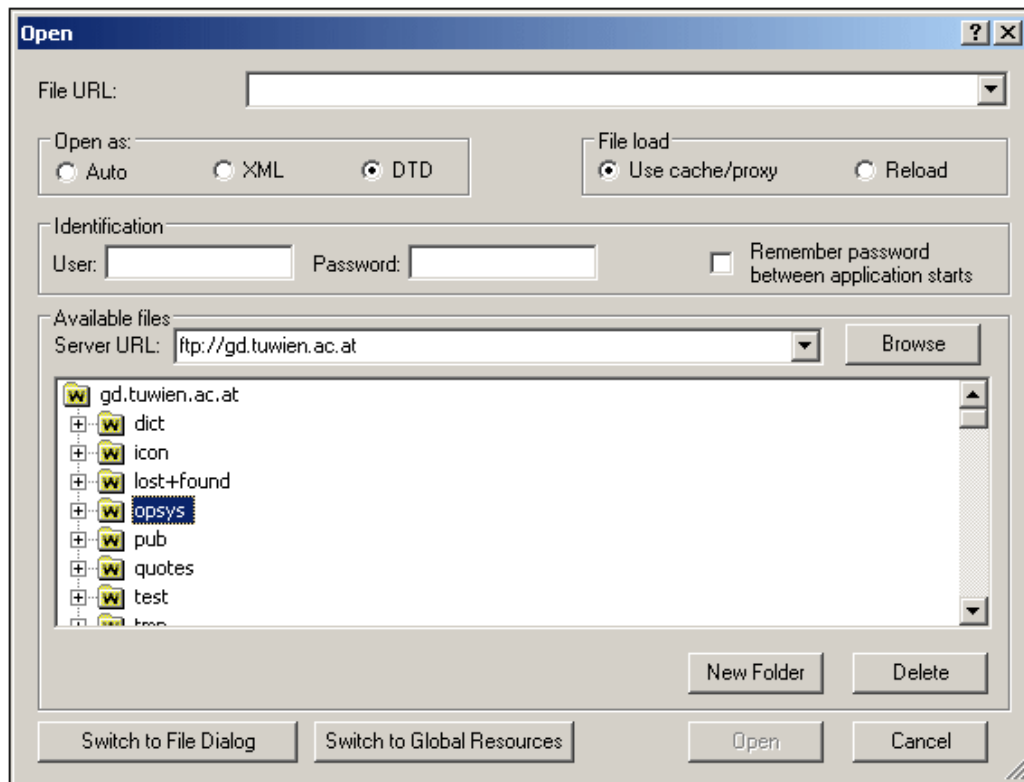
In einigen Dialogfeldern zum Öffnen und Speichern von Dateien können Sie auswählen, ob Sie die gewünschte Datei auswählen oder eine Datei über eine URL oder eine globale Ressource (siehe *Abbildung unten*) speichern möchten. Wählen Sie den Befehl **Zu URL wechseln** oder **Zu globaler Ressource wechseln** um eine dieser Optionen auszuwählen.



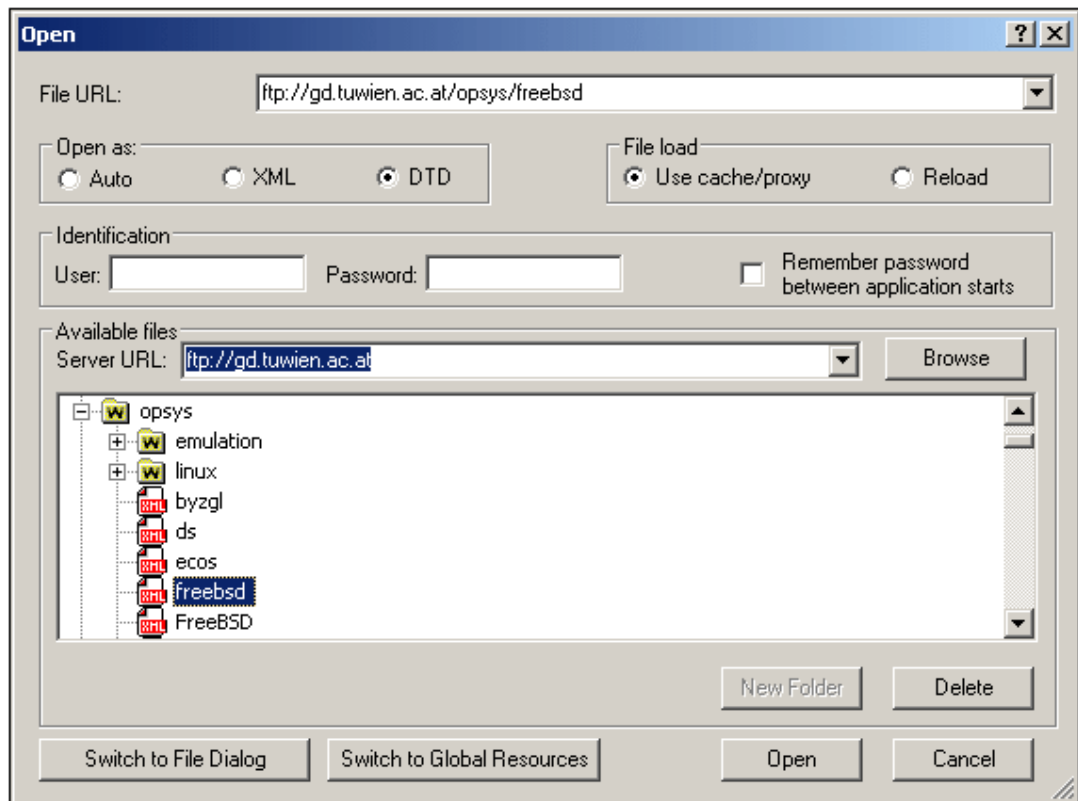
Auswählen von Dateien über URLs

Um eine Datei über eine URL auszuwählen, gehen Sie folgendermaßen vor:

1. Klicken Sie auf den Befehl **Zu URL wechseln**. Daraufhin wird das Dialogfeld "URL öffnen" aufgerufen (*Abbildung unten*).



2. Geben Sie in das Feld *Server URL* die gewünschte URL ein.
3. Geben Sie die Benutzer-ID und das Passwort in die entsprechenden Felder ein, wenn der Server durch ein Passwort geschützt ist.
4. Klicken Sie auf **Durchsuchen**, um die Verzeichnisstruktur des Servers anzuzeigen und darin zu navigieren.
5. Klicken Sie auf die gewünschte Datei.



Die Datei-URL wird im Feld "Datei-URL" angezeigt. Die Schaltfläche OK wird erst jetzt aktiv.

6. Klicken Sie auf **OK**, um die Datei zu laden. Die Datei wird im Hauptfenster angezeigt.

Hinweis: Die Funktion "Durchsuchen" steht nur auf Servern zur Verfügung, die das FTP-, HTTP- und HTTPS-Protokoll (wenn der Server WebDAV unterstützt) unterstützen sowie auf Servern, die WebDAV unterstützen.

Hinweis: Um mehr Kontrolle über den Ladevorgang zu haben, können Sie die Datei über den lokalen Cache oder einen Proxy-Server laden (wodurch der Vorgang erheblich schneller geht, wenn die Datei bereits zuvor einmal geladen wurde). Alternativ dazu können Sie die Datei auch neu laden, wenn Sie mit einem EPS-System oder einem Datenbank-System arbeiten; Wählen Sie in diesem Fall die Option **Neu laden**.

Öffnen und Speichern von Dateien über globale Ressourcen

Um eine Datei über eine globale Ressource zu öffnen oder zu speichern, klicken Sie auf **Zu globaler Ressource wechseln**. Daraufhin wird ein Dialogfeld angezeigt, in dem Sie die globale Ressource auswählen können. Diese Dialogfelder werden im Abschnitt Verwenden globaler Ressourcen näher beschrieben. Eine allgemeine Beschreibung globaler Ressourcen finden Sie in der Dokumentation im Abschnitt [Globale Ressourcen](#)

9.1.7 Als Mail senden



Mit dem Befehl **Als Mail senden...** können Sie jede XML-Datei oder Teile einer XML-Datei als Email versenden. Sie haben folgende Möglichkeiten:

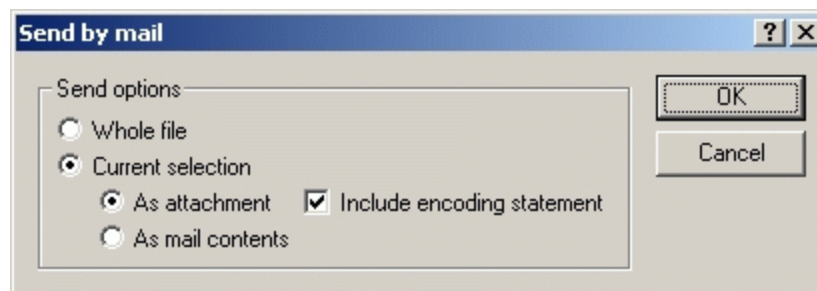
- Versenden eines XML-Dokuments als Anhang oder als Inhalt einer E-Mail.
- Versenden eines Teils eines XML-Dokuments als Anhang oder Inhalt einer E-Mail.
- Versenden einer Gruppe von Dateien (die im Projektfenster ausgewählt wurden) als Anhang einer E-Mail.
- Versenden einer URL (die im Projektfenster ausgewählt wurde) als Anhang oder als Link.

Bitte beachten Sie: Zur Verwendung dieser Funktion benötigen Sie ein MAPI-kompatibles E-Mail-System.

Versenden von Dokumenten und Teilen eines Dokuments

Um ein XML-Dokument zu versenden,

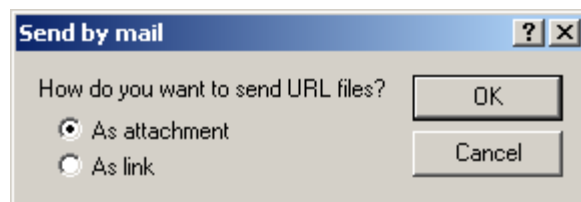
1. Machen Sie es im Hauptfenster zum aktiven Dokument. Wenn Sie nur einen Teil aus einem XML-Dokument oder eine Gruppe von Dateien eines Projekts versenden möchten, markieren Sie den Abschnitt im Dokument oder wählen Sie die gewünschten Dateien im Projekt aus.
2. Klicken Sie nun auf **Als Mail senden...**. Daraufhin wird das folgende Dialogfenster angezeigt.



3. Aktivieren Sie die gewünschten Optionen und klicken Sie auf **OK**. Das/die ausgewählte(n) Dokument(e)/der ausgewählte Inhalt/die ausgewählten URLs werden an die E-Mail angehängt bzw. der Inhalt wird in die E-Mail eingefügt.

URL per Mail versenden

Zum Versenden einer oder mehrerer URLs wählen Sie die URLs im Projektfenster aus und klicken Sie auf **Als Mail senden...**. Es wird das folgende Dialogfenster angezeigt.



Wählen Sie aus, in welcher Form die URL versendet werden soll und bestätigen Sie mit **OK**.

9.1.8 Drucken



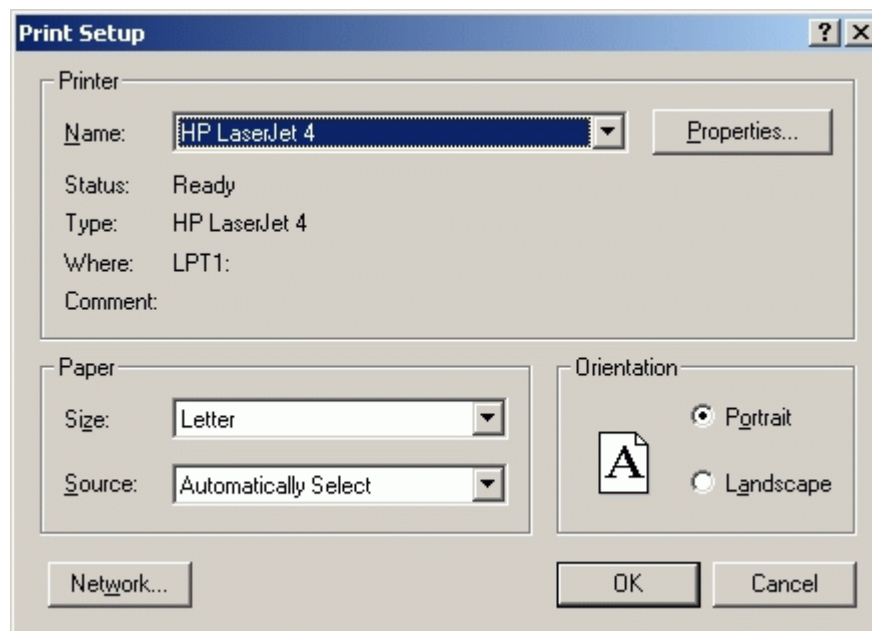
Strg+P

Der Befehl **Drucken** öffnet das Dialogfeld "Drucken". Hier können Sie festlegen, was die Druckausgabe enthalten soll.

9.1.9 Druckereinrichtung

Mit dem Befehl **Druckvorschau** öffnen Sie das Dialogfeld "Drucken". Klicken Sie auf die Schaltfläche "Vorschau", um eine Druckvorschau des aktiven Dokuments zu sehen.

Mit dem Befehl **Druckereinrichtung...** gelangen Sie in das druckerspezifische Dialogfeld, wo Sie Papierformat, Ausrichtung und weitere Einstellungen tätigen können, die für alle zukünftigen Druckaufträge gültig sind.



Das obige Bild zeigt das Dialogfeld, bei dem ein Drucker des Typs HP Laserjet 4 am parallelen Port LPT1 angeschlossen ist.

9.1.10 Zuletzt verwendete Dateien, Beenden

Im Menü **Datei** sehen Sie eine Liste der neun zuletzt verwendeten Dateien, wobei die zuletzt verwendete Datei zuoberst angezeigt wird. Durch Anklicken einer Datei können Sie diese öffnen. Über die Tastatur können Sie mit der Funktionstaste **ALT+A** das Menü **Datei** aufrufen, und dann die der Datei entsprechende Nummer eintippen.

Mit dem Befehl **Beenden** beenden Sie Authentic Desktop, nachdem Sie Ihre Arbeit an Ihren Dokumenten abgeschlossen haben. Wenn noch nicht gespeicherte Dokumente offen sind, werden Sie gefragt, ob Sie die Änderungen speichern möchten. Authentic Desktop speichert weiters Programmeinstellungen, sowie Informationen über die zuletzt verwendeten Dateien.

9.2 Menü "Bearbeiten"

In diesem Menü befinden sich Befehle, die beim Editieren mit verwendet werden.

9.2.1 Rückgängig, Wiederherstellen

Der Befehl **Rückgängig (Strg+Z)**  unterstützt unbegrenztes Rückgängigmachen aller Aktionen! Jede durchgeführte Aktion kann in Einzelschritten rückgängig gemacht werden. Der Verlauf der durchgeführten Aktionen wird beim Schließen des Dokuments gespeichert, sodass Aktionen auch zu einem späteren Zeitpunkt zurückverfolgt werden können.

Mit **Wiederherstellen (Strg+Y)**  (in Visual Studio **Wiederholen**) können Sie zuvor rückgängig gemachte Aktionen wiederherstellen. Sie können den Verlauf des Dokuments sowohl zurück als auch nach vorne verfolgen.

9.2.2 Ausschneiden, Kopieren, Einfügen, Löschen

Mit dem Befehl **Ausschneiden (Umschalt + Entf oder Strg + X)**  wird der markierte Text bzw. das markierte Objekt in die Zwischenablage kopiert und aus der aktuellen Position im Dokument gelöscht.

Mit dem Befehl **Kopieren (Strg+C)**  kopieren Sie markierten Text oder Objekte in die Zwischenablage. So können Sie Daten in Authentic Desktop duplizieren, oder in eine andere Applikation kopieren.

Mit dem Befehl **Einfügen (Strg+V)**  fügen Sie den Inhalt der Zwischenablage an der aktuellen Cursorposition ein.

Mit dem Befehl **Löschen (Entf)**  wird der markierte Text/das markierte Objekt gelöscht, ohne dass es in die Zwischenablage kopiert wird.

9.2.3 Alles markieren

Strg + A

Mit dem Befehl **Alles markieren** (in Visual Studio **Alle auswählen**) wird der Inhalt des gesamten Dokuments markiert.

9.2.4 Suchen, Weitersuchen...

Mit dem Befehl **Suchen (Strg+F)**  (in Visual Studio **Suchen und Ersetzen**) rufen Sie das Dialogfeld "Suchen" auf, in dem Sie den zu suchenden String sowie weitere Suchoptionen spezifizieren können.

Um nach Text zu suchen, geben Sie diesen in das Listenfeld "Suchen nach" ein oder wählen Sie einen der letzten 10 Einträge aus der Dropdown-Liste aus und legen Sie anschließend die Suchoptionen fest.

Mit dem Befehl **Weitersuchen (F3)**  wird der letzte Suchbefehl wiederholt, um nach der nächsten Instanz des gesuchten Texts zu suchen.

9.2.5 Ersetzen...



Strg + H

Mit dem Befehl **Ersetzen** (in Visual Studio **Suchen und Ersetzen**) können Sie einen Textstring durch einen anderen ersetzen. Der Befehl verfügt über dieselben Optionen wie der Befehl [Suchen...](#).

Sie können diese Funktion Schritt für Schritt verwenden, oder alle Begriffe auf einmal ersetzen, indem Sie auf die Schaltfläche "Alle ersetzen" klicken.

9.3 Menü "Projekt"

verwendet die bekannte Baumstruktur, um Dateien und URLs in einem XML-Projekt zu gliedern. [Dateien](#) und [URLs](#) mit gleicher Erweiterung können in [Ordern](#) gesammelt werden, wodurch eine einfache Strukturierung und Batchmanipulation gegeben ist.

Bitte beachten Sie:

Die meisten projektbezogenen Befehle sind auch mittels Rechtsklick auf das gewünschte Objekt über das Kontextmenü verfügbar.

Absolute und relative Pfade

Jedes Projekt ist als Projektdatei gespeichert, die mit der Erweiterung `.spp` versehen ist. Bei diesen Dateien handelt es sich eigentlich um XML-Dokumente, die Sie wie jede reguläre XML-Datei editieren können. In der Projektdatei werden für Dateien/Ordner auf derselben oder einer höheren Ebene absolute Pfade verwendet, während für Dateien/Ordner im aktuellen Ordner oder einem Unterordner relative Pfade definiert werden. Wenn Ihre Verzeichnisstruktur z.B. in etwa so aussieht:

```
| -Folder1
|   |
|   | -Folder2
|   |   |
|   |   | -Folder3
|   |   |   |
|   |   |   | -Folder4
```

...und Ihre `.spp`-Datei in `Folder3` gespeichert ist, so werden Referenzen auf Dateien in `Folder1` und `Folder2` folgendermaßen angegeben:

```
c: \Folder1\NameOfFile.ext
c: \Folder1\Folder2\NameOfFile.ext
```

Referenzen auf Dateien in `Folder3` und `Folder4` würden in etwa folgendermaßen aussehen:

```
.\NameOfFile.ext
.\Folder4\NameOfFile.ext
```

Wenn Sie sicherstellen möchten, dass alle Pfade relativ angegeben werden, speichern Sie die `.spp`-Dateien im Root-Verzeichnis Ihres Rechners.

Drag-and-Drop

Im Projekt-Fenster kann ein Ordner in einen anderen Ordner oder an eine andere Stelle im selben Ordner gezogen werden. Eine Datei kann in einen anderen Ordner, nicht aber innerhalb desselben Ordners (in dem die Dateien alphabetisch geordnet sind) verschoben werden. Zusätzlich dazu können Dateien und Ordner aus dem Windows-Datei-Explorer in das Projektfenster gezogen werden.

Globale Ressourcen im Kontextmenü

Wenn Sie mit der rechten Maustaste im Projektfenster auf einen Ordner klicken, können Sie im daraufhin angezeigten Kontextmenü den Befehl **Globale Ressource hinzufügen** auswählen, um eine hinzuzufügen. Daraufhin wird das Dialogfeld "Globale Ressource wählen" angezeigt, in dem alle globalen Ressourcen vom Typ Datei und Ordner in der derzeit aktiven XML-Datei für globale Ressourcen aufgelistet sind. Wählen Sie die gewünschte globale Ressource und sie wird zum ausgewählten Projektordner hinzugefügt.

Projekte und Versionskontrolle

Wenn Sie vorhaben, ein -Projekt zu einem Versionskontroll-Repository hinzuzufügen, stellen

Sie bitte sicher, dass die Projektdateien in der hierarchischen Dateisystemstruktur in einem Ordner abgelegt werden, von dem aus Dateien nur aus diesem untergeordneten Ordnern hinzugefügt werden können (Nehmen Sie das Root-Verzeichnis als oberste Ebene der Verzeichnisstruktur)

In anderen Worten stellt das Verzeichnis, in dem sich die **Projektdatei** befindet, innerhalb des Versionskontroll-Repository im Grunde das **Root-Verzeichnis** des Projekts dar. Dateien, die dem Projekt aus einem Ordner hinzugefügt werden, der sich oberhalb dieses Root-Verzeichnisses befindet, werden dem -Projekt zwar hinzugefügt, doch ist ihr Speicherort im Repository - falls überhaupt zulässig - manchmal ein unerwarteter.

Nehmen wir als Beispiel die oben gezeigte Verzeichnisstruktur:

Wenn eine Projektdatei in `Folder3` gespeichert wird und einer Versionskontrolle unterliegt:

- so werden Dateien, die zu `Folder1` hinzugefügt werden, unter Umständen nicht von der Versionskontrolle berücksichtigt,
- werden Dateien, die zu `Folder2` hinzugefügt werden, dem Root-Verzeichnis des Repository anstelle des Projektordners hinzugefügt, jedoch von der Versionskontrolle berücksichtigt,
- während die in `Folder3` und `Folder4` gespeicherten Dateien korrekt verarbeitet werden und auch der Versionskontrolle unterliegen.

9.3.1 Neues Projekt



Mit diesem Befehl erstellen Sie ein **neues** Projekt in .
Wenn Sie gerade an einem anderen Projekt arbeiten, werden Sie gefragt, ob Sie alle Dokumente schließen wollen, die zu dem aktuellen Projekt gehören.

9.3.2 Projekt öffnen.



Diese Funktion **öffnet** ein bereits vorhandenes Projekt in.
Falls Sie zuvor an einem anderen Projekt gearbeitet haben, wird dieses Projekt zuerst geschlossen.

9.3.3 Projekt neu laden



Mit dieser Funktion wird das aktuelle Projekt von der Festplatte **neu geladen**. Falls Sie in einer Multiuser-Umgebung arbeiten, kann es oftmals nötig sein, ein Projekt, an dem mehrere Personen gleichzeitig arbeiten, neu zu laden, um deren Änderungen anzusehen.

Bitte beachten Sie: Projektdateien (.spp) sind XML-Dokumente, die Sie wie jede andere XML Datei bearbeiten können.

9.3.4 Projekt schließen

Mit diesem Befehl schließen Sie das **aktuelle** Projekt. Wurde das Projekt geändert, werden Sie vor dem Schließen gefragt, ob die Änderungen gespeichert werden sollen.

Wenn ein Projekt oder eine XML-Datei geändert wurde, wird der Name der Datei in der Titelleiste des Projekts mit einem Stern "*" versehen.

9.3.5 Projekt speichern



Mit diesem Befehl wird das aktuelle Projekt **gespeichert**. Sie können ein Projekt auch speichern, indem Sie das Projektfenster zum aktiven machen und auf das Symbol  klicken.

9.3.6 Versionskontrolle

unterstützt Microsoft Source-Safe und andere kompatible Speicherschnittstellen. In diesem Abschnitt wird anhand von Visual SourceSafe gezeigt, wie die Versionskontrollfunktionen von Authentic Desktop verwendet werden.

Microsoft hat einen Registryeintrag definiert, bei dem alle SCC-kompatiblen Programme sich selbst registrieren. Authentic Desktop liest diesen Eintrag.

HKEY_LOCAL_MACHINE\SOFTWARE\SourceCodeControlProvider\InstalledSCCProviders

Bitte beachten Sie außerdem, dass **Plugins** für Versionskontrollsysteme nicht von allen SSC-Produkten automatisch installiert werden. Nähere Informationen finden Sie in der Dokumentation zu Ihrer Versionskontrollsoftware.

Versionskontrollbefehle können auf verschiedene Arten aufgerufen werden:

- über den Menübefehl **Projekt | Versionskontrolle**
- über das **Kontextmenü** in der Modell-Struktur

Ein Versionskontrollprojekt ist allerdings nicht mit einem Projekt in Authentic Desktop zu vergleichen. Versionskontrollprojekte sind verzeichnisbezogen, wobei Authentic Desktop Projekte logische Konstrukte ohne direkte Verzeichnisabhängigkeit sind.

Unterstützte Versionskontrollserver und Clients

In der folgenden Tabelle sind die von Authentic Desktop unterstützten Source Control Server (SCS) und Clients (SCC) aufgelistet. Weiter unten finden Sie Links zu den entsprechenden Websites.

Altova hat die Microsoft Source Code Control Interface (MSSCCI) v1.1 – v1.3 in XMLSpy und UModel implementiert und die Funktionen mit den folgenden Treibern und Versionskontrollsystemen getestet.

Unterstützte Versionskontroll-Server

Server	Version
AccuRev http://www.accurev.com	AccuRev 4.7.0 Windows
Bazaar http://bazaar-vcs.org/	Bazaar 1.9 Windows
Borland StarTeam 2008 http://www.borland.com/us/products/starteam	StarTeam 2008 Release 2
Codice Software Plastic SCM http://www.codicesoftware.com/xpproducts.aspx	Codice Software Plastic SCM Professional 2.7.127.10 (Server)
Collabnet Subversion 1.5 http://subversion.tigris.org	1.5.4
ComponentSoftware CS-RCS (PRO) http://www.componentsoftware.com/Products/RCS	ComponentSoftware CS-RCS (PRO) 5.1

Dynamsoft SourceAnywhere for VSS http://www.dynamsoft.com/Products/SAW_Overview.aspx	SourceAnywhere for VSS 5.3.2 Standard/Professional Server
Dynamsoft SourceAnywhere Hosted http://www.dynamsoft.com/Products/SourceAnywhere-SourceSafe-VSS.aspx	Server hosted in a Bell Data Center
Dynamsoft SourceAnywhere Standalone http://www.dynamsoft.com/Products/SourceAnywhere-SourceSafe-VSS.aspx	SourceAnywhere Standalone 2.2 Server
IBM Rational ClearCase 7 http://www-01.ibm.com/software/awdtools/clearcase	7.0.1 (LT)
March-Hare CVSNT 2.5 http://www.cvsnt.org	2.5.03.2382
March-Hare CVS Suite 2008 http://www.march-hare.com/cvsnt/en.asp	Server 2008 [3321]
Mercurial http://www.selenic.com/mercurial	Mercurial 1.0.2 for Windows
Microsoft SourceSafe 2005 http://msdn.microsoft.com/en-us/vstudio/aa718670.aspx	2005 with CTP
Microsoft Visual Studio Team System 2008 Team Foundation Server http://msdn.microsoft.com/de-de/vsts2008/products/bb933758.aspx	2008
Perforce 2008 http://www.perforce.com/	P4S 2008.1
PureCM http://www.purecm.com/	PureCM Server 2008/3a
QSC Team Coherence Version Manager http://www.teamcoherence.com	QSC Team Coherence Server 7.2.0.30
Qumasoft QVCS-Enterprise http://www.qumasoft.com/	QVCS-Enterprise 2.1.18
Qumasoft QVCS-Pro http://www.qumasoft.com/	3.10.18
Reliable Software Code Co-Op http://www.relisoft.com/co_op/index.htm	Code Co-Op 5.1a
Seapine Surround SCM http://www.seapine.com/surroundscm.html	Surround SCM Client/Server for Windows 2009.0.0
Serena Dimensions http://www.serena.com/Products/dimensions/	Dimensions Express/CM 10.1.3 for Win32 Server

Softimage Alienbrain http://www.alienbrain.com/	Alienbrain Server 8.1.0.7300
SourceGear Fortress http://www.sourcegear.com/fortress	1.1.4 Server
SourceGear SourceOffsite http://www.sourcegear.com/sos/	SourceOffsite Server 4.2.0
SourceGear Vault http://www.sourcegear.com/vault	4.1.4 Server
VisualSVN Server 1.6 http://www.visualsvn.com	1.6.2

Von bestimmten Servern unterstützte Clients

Server	Client (SCC Plugin)
AccuRev 4.7.0 Windows	AccuBridge for MS-SCC 2008.2
Bazaar 1.9	Aigenta Unified SCC 1.0.5
Borland StarTeam 2008 R2	Borland StarTeam CPCClient 2008 R2
Codice Software Plastic SCM Professional 2.7.127.10	Codice Software Plastic SCM Professional 2.7.127.10
Collabnet Subversion 1.5.4	Aigenta Unified SCC 1.0.5
	PushOK SVN SCC 1.5.1.1
	TamTam SVN SCC 1.2.21
ComponentSoftware CS-RCS (PRO) 5.1	ComponentSoftware CS-RCS (PRO) 5.1
Dynamsoft SourceAnywhere for VSS 5.3.2 Server	Dynamsoft SourceAnywhere for VSS 5.3.2 Client
Dynamsoft SourceAnywhere Hosted Server	Dynamsoft SourceAnywhere Hosted Client (22252)
Dynamsoft SourceAnywhere Standalone 2.2 Server	Dynamsoft SourceAnywhere Standalone 2.2 Client
IBM Rational ClearCase 7.0.1 (LT)	IBM Rational ClearCase 7.0.1 (LT)
March-Hare CVSNT 2.5.03.2382 March-Hare CVS Suite Server 2008 [3321]	Aigenta Unified SCC 1.0.6
	Jalindi Igloo 1.0.3
	March-Hare CVS Suite Client 2008 [3321]
	PushOK CVS SCC NT 2.1.2.5
	TamTam CVS SCC 1.2.36
Mercurial 1.0.2	Sergey Antonovs HgScc 1.0.1

Microsoft SourceSafe 2005	Microsoft SourceSafe 2005
Microsoft Team Foundation Server 2008	Microsoft Team Foundation Server 2008 MSSCCI Provider
Perforce P4S 2008.1	Perforce P4V 2008.1
PureCM Server 2008/3a	PureCM Client 2008/3a
QSC Team Coherence Server 7.2.0.25	QSC Team Coherence Client 7.1.4.30
Qumasoft QVCS-Enterprise 2.1.18	Qumasoft QVCS-Enterprise 2.1.18
Qumasoft QVCS-Pro 3.10.18	Qumasoft QVCS-Pro 3.10.18
Reliable Software Code Co-Op 5.1a	Reliable Software Code Co-Op 5.1a
Seapine Surround SCM 2009.0.0	Seapine Surround SCM Client 2009.0.0
Serena Dimensions 10.1.3	Serena Dimensions 10.1.3
Softimage Alienbrain 8.1 Server	Softimage Alienbrain 8.1 Essentials/Advanced Client
SourceGear Fortress 1.1.4 Server	SourceGear Fortress 1.1.4 Client
SourceGear SourceOffsite 4.2.0 Server	SourceGear SourceOffsite 4.2.0 Client (Windows)
SourceGear Vault 4.1.4 Server	SourceGear Vault 4.1.4 Client
VisualSVN Server 1.6.2	Aigenta Unified SCC 1.0.5
	PushOK SVN SCC 1.5.1.1
	TamTam SVN SCC 1.2.21

Bitte beachten Sie:

- XMLSpy wurde anhand der oben aufgelisteten Versionskontroll-Software getestet. Es wird erwartet, dass XMLSpy auch etwaige zukünftige Versionen dieser Produkte unterstützen wird.
- Versionskontrollsysteme, die in der obigen Tabelle nicht aufgelistet sind, die aber der MSSCCI 1.1-1.3 Spezifikation entsprechen sollten ebenfalls mit UModel funktionieren.

Installationsverfahren für Versionskontrollsysteme

Dieser Abschnitt enthält nähere Informationen zum Einrichten und Installieren der verschiedenen Versionskontrollsysteme.

AccuRev

<http://www.accurev.com>

Unterstützte Versionen: AccuBridge for Microsoft SCC 2008.2

1. Installieren Sie die AccuRev Client Software, führen Sie den Installer aus, definieren Sie den Server, zu dem eine Verbindung hergestellt werden soll, (Hostname und Port) und erstellen Sie anschließend einen Arbeitsbereich.
2. Installieren Sie den AccuBridge SCC Provider. Extrahieren Sie das ZIP-Archiv in das Verzeichnis <AccuRev installation dir>\bin.

Registrieren Sie AccuRev.dll und SccAcc.dll wie folgt:

3. Öffnen Sie ein Befehlseingabefenster (wenn Sie mit Vista arbeiten, starten Sie den Windows Explorer, gehen Sie zu C:\Windows\System32, klicken Sie auf die rechte Maustaste und führen Sie die cmd.exe "As administrator" aus).
4. Gehen Sie zum Verzeichnis <installation AccuRev dir>\bin.
5. Geben Sie an der Eingabeaufforderung den folgenden Befehl ein:
 - Regsvr32 AccuRev.dll
 - Regsvr32 SccAcc.dll
6. Führen Sie das Programm SwitchScc.exe aus und definieren Sie als den Provider AccuRev.
7. Führen Sie in Windows ein Logoff und anschließend ein Login durch.

Aigenta Unified SCC

<http://aigenta.com/products/UnifiedScc.aspx>

Unterstützte Versionen: 1.0.5

Voraussetzungen: Source Control Client. Aigenta Unified SCC funktioniert mit:

- Subversion Befehlszeilen-Client 1.5.4 unter <http://subversion.tigris.org>
- CVSNT 2.5 (Client) unter <http://www.cvsnt.org>
- Bazaar 1.9 Windows (und damit in Zusammenhang stehenden Komponenten) unter <http://bazaar-vcs.org/> (<http://bazaar-vcs.org/WindowsInstall>)

Eine Standard-Installation funktioniert mit allen Altova-Produkten ordnungsgemäß.

Borland StarTeam 2008

<http://www.borland.com/us/products/starteam>

Unterstützte Versionen: StarTeam 2008 Release 2 Cross Platform Client

Um die Borland StarTeam Microsoft SCC Integration zu installieren, führen Sie das Setup-Programm aus und wählen Sie bei der Installation die SCC API Integration.

Altova-Produkte können nun bei Angabe der Server-Adresse, des Endpunkts, des einzuloggenden Benutzernamens und Passworts und Ihres Projekt- und Arbeitspfads eine Verbindung zur Speicherschnittstelle herstellen.

Codice Software Plastic SCM

<http://www.codicesoftware.com/xpproducts.aspx>

Unterstützte Versionen: Codice Software Plastic SCM Professional 2.7.127.10 (SCC Plugin)

Eine Standard-Installation funktioniert mit allen Altova-Produkten ordnungsgemäß. Es müssen nur die Komponenten "client" und das "Visual Studio SCC plug-in" installiert werden.

ComponentSoftware CS-RCS (PRO)

<http://www.componentsoftware.com/Products/RCS>

Unterstützte Versionen: ComponentSoftware CS-RCS (PRO) 5.1

1. Um ComponentSoftware CS-RCS (PRO) zu installieren, starten Sie das Setup-Programm und aktivieren Sie die Option "Workstation Setup".
2. Definieren Sie die Root für die Struktur der Speicherschnittstelle und starten Sie Ihren Rechner nach Abschluss der Installation neu.
3. Über "ComponentSoftware RCS Properties" können Sie ein Projekt auswählen oder erstellen und einen Arbeitsordner definieren.

Dynamsoft SourceAnywhere for VSS

http://www.dynamsoft.com/Products/SAW_Overview.aspx

Unterstützte Versionen: SourceAnywhere for VSS 5.3.2 Standard/Professional Client

Eine Standard-Installation funktioniert mit allen Altova-Produkten ordnungsgemäß.

Für die Integration mit Altova Produkten muss das Plug-in für Adobe DreamWeaver CS3 nicht installiert werden. Stellen Sie nach der Installation eine Server-Verbindung her und definieren Sie einen Arbeitsordner.

Dynamsoft SourceAnywhere Hosted

<http://www.dynamsoft.com/Products/SourceAnywhere-Hosting-Version-Control-Source-Control.aspx>

Unterstützte Versionen: SourceAnywhere Hosted Client (Build 22252)

Eine Standard-Installation funktioniert mit allen Altova-Produkten ordnungsgemäß.

Für die Integration mit Altova Produkten muss das Plug-in für Adobe DreamWeaver CS3 nicht installiert werden.

Dynamsoft SourceAnywhere Standalone

<http://www.dynamsoft.com/Products/SourceAnywhere-SourceSafe-VSS.aspx>

Unterstützte Versionen: SourceAnywhere Standalone 2.2 Client

Eine Standard-Installation funktioniert mit allen Altova-Produkten ordnungsgemäß.

Für die Integration mit Altova Produkten muss das Plug-in für Adobe DreamWeaver CS3 nicht installiert werden. Stellen Sie nach der Installation eine Server-Verbindung her und definieren Sie einen Arbeitsordner.

IBM Rational ClearCase 7

<http://www-01.ibm.com/software/awdtools/clearcase/>

Unterstützte Versionen: 7.0.1 (LT)

Um IBM Rational ClearCase LT zu installieren, führen Sie das Setup-Programm aus.

- Sie werden aufgefordert, die Version des InstallShield Scripting-Prozessors zu aktualisieren, wenn diese älter als Version 10.5 ist. Wählen Sie "Update it if necessary".

Die Aktualisierung wird vor Beginn der Installation durchgeführt.

- Wählen Sie die Standardoption "Enterprise deployment, create a network release area and customize it using Siteprep".

Um die Versionskontrolle mit Altova-Produkten zu integrieren, muss nur der Client installiert werden. Aktivieren Sie nur das Kontrollkästchen "Client".

- Geben Sie einen Server-Namen sowie das/die Lizenzserver Element(e) an. Gehen Sie vor wie in den Beispielen, die vom Installer geliefert werden ([port@server_name](#)).
- Geben Sie einen Namen für die Konfiguration an, indem Sie den Namen entsprechend bearbeiten und den Pfad zu einem Release-Bereich (Release Area) eingeben. Dieser Pfad muss einen freigegebenen Ordner definieren.
- Sie können einen neuen Ordner auf Ihrem Rechner anlegen, ihn freigeben und als

Release-Bereich (Release Area) verwenden.

In Vista müssen Sie "Network discovery" im Network and Sharing Center auf "on" setzen, um diesen Pfad zu definieren.

Der Release-Bereich wird nun erstellt, einige Dateien werden hinein kopiert und ein Shortcut **sitedefs.lnk** wird erstellt.

- Wenn alle Dateien kopiert wurden, fahren Sie fort, indem Sie vom Windows Explorer aus auf den Shortcut klicken.
Ein neues Setup-Programm beginnt mit der Installation des Client.
- Wenn das Setup-Programm beginnt, wählen Sie die Option "Install IBM Rational ClearCase LT".
- Klicken Sie auf "Weiter", akzeptieren Sie das "Software License Agreement" und starten Sie die Installation.

In **Vista** könnte das zweite Setup-Programm den internen Fehler: 2739 ausgeben. Starten Sie in diesem Fall den Windows Explorer und gehen Sie zu C:\Windows\System32.

- Klicken Sie auf die rechte Maustaste und führen Sie "cmd.exe" "As Administrator" aus. Daraufhin erscheint ein Eingabefenster.
- Geben Sie "regsvr32 jscript.dll" ein.
- Starten Sie das Setup-Programm wieder.

Um mit in ClearCase gespeicherten Dateien zu arbeiten, sollten Sie eine Ansicht erstellen, die auf Ihr ClearCase-Projekt verweist.

Jalindi Igloo

<http://www.jalindi.com/igloo/>

Unterstützte Versionen: 1.0.3

Um Jalindi Igloo mit Altova-Produkten zu verwenden, muss nur das Setup-Programm ausgeführt werden, um Jalindi Igloo zu installieren.

Bitte beachten Sie:

Wenn Sie Jalindi Igloo deinstallieren, werden alle anderen installierten SCC Provider Windows Registrierschlüssel (falls vorhanden) ebenfalls gelöscht und stehen nicht mehr zur Verfügung.

Wenn Sie mit Altova-Produkten arbeiten, wird empfohlen den Modus "Auto Commit" einzustellen.

- Der Modus "Auto Commit" befindet sich in den erweiterten Source Control-Optionen.
- Nachdem Sie einen Arbeitsbereich definiert haben, können Sie mit der Arbeit beginnen.

March-Hare CVS Suite 2008

<http://www.march-hare.com/cvsnt/en.asp>

Unterstützte Versionen: Client 2008 [3321]

Eine "typical"-Installation funktioniert mit Altova-Produkten ordnungsgemäß.

Mercurial

http://www.newsupaplex.pp.ru/hgsc_news_eng.html

Unterstützte Versionen: Sergey Antonovs HgSc 1.0.1

Eine Standard-Installation funktioniert mit allen Altova-Produkten ordnungsgemäß.

Microsoft SourceSafe 2005

<http://msdn.microsoft.com/en-us/vstudio/aa718670.aspx>

Unterstützte Versionen: 2005 with CTP

Eine Standard-Installation von Microsoft Source Safe 2005 funktioniert mit Altova-Produkten ordnungsgemäß.

Microsoft Visual Studio Team System 2008 Team Foundation Server MSSCCI Provider

<http://www.microsoft.com/downloads>

Unterstützte Versionen: 2008

Voraussetzungen:

Visual Studio 2008 Team Explorer oder Visual Studio 2008 mit Team Explorer 2008

Eine Standard-Installation funktioniert mit allen Altova-Produkten ordnungsgemäß.

Perforce 2008

<http://www.perforce.com/>

Unterstützte Versionen: P4V 2008.1

Beim Perforce Visual Client (P4V) haben Sie die Wahl:

- alle Client-Funktionen zu installieren (Standardverhalten)
- nur die Funktion "SCC Plug-in (P4SCC)" zu installieren.

Eine Standard-Installation funktioniert mit allen Altova-Produkten ordnungsgemäß.

Wenn die Funktion "SCC Plug-in (P4SCC)" ausgewählt wird:

- funktionieren zwei SCC-Funktionen "Show differences" und "Source control manager" nicht.
- Die Funktion "Show differences" und die Möglichkeit, den Source Control Manager zu starten, funktionieren nicht, weil sie auf den nicht installierten Funktionen "Visual Merge Tool" bzw. "Visual Client (P4V)" basieren.
- Für die Vergleichsfunktion wird Drittanbietersoftware benötigt, während der Source Control Manager nur gestartet werden kann, nachdem der "Visual Client (P4V)" explizit installiert wurde.
- Nachdem Sie die Perforce Visual Client Installation gestartet haben, definieren Sie Ihre eigenen Client-Konfigurationseinstellungen (Server Name, Textbearbeitungsapplikation, Benutzername).
- Vergessen Sie nach Abschluss der Installation nicht, einen neuen Arbeitsbereich zu erstellen oder einen vorhandenen auszuwählen.

PureCM

<http://www.purecm.com/>

Unterstützte Versionen: PureCM Client 2008/3a

Eine Standard-Installation funktioniert mit allen Altova-Produkten ordnungsgemäß.

- Starten Sie nach der Installation den PureCM Client, um einen Server zu registrieren.

PushOK CVS SCC NT

http://www.pushok.com/soft_cvs.php

Unterstützte Versionen: 2.1.2.5

Eine Standard-Installation genügt, um PushOK CVS SCC NT zu verwenden.

- Stellen Sie nach Abschluss der Installation sicher, dass Ihre Version des CVS Proxy Plug-in korrekt registriert ist.
- Nach Definition eines Arbeitsbereichs können Sie anfangen, damit zu arbeiten.

PushOK SVN SCC

http://www.pushok.com/soft_svn.php

Unterstützte Versionen: 1.5.1.1

Eine Standard-Installation genügt, um PushOK CVS SCC mit Altova-Produkten zu verwenden.

- Bei Installation unter Vista kann es vorkommen, dass die COM-Bibliothek **svncom.dll** nicht registriert werden kann.

Stellen Sie in diesem Fall die Installation fertig und registrieren Sie die Bibliothek folgendermaßen manuell:

1. Starten Sie ein Eingabefenster über die Option "Run as administrator".
2. Geben Sie: cd "C:\Program Files\PushOK Software\SVNSCC\svn" ein
3. Geben Sie den Befehl > regsvr32 svncom.dll ein.

Qumasoft QVCS-Enterprise

<http://www.qumasoft.com/>

Unterstützte Versionen: QVCS-Enterprise 2.1.18

Voraussetzungen: J2SE 1.5 oder höher <http://java.sun.com/j2se/downloads.html>

Um den Qumasoft QVCS-Enterprise Client zu installieren, starten Sie den Installer.

- Wenn Ihr Betriebssystem **Vista** ist, müssen Sie das Installationsverzeichnis vom Standardwert "C:\Program Files\QVCS-Enterprise Client" in "C:\QVCS-Enterprise Client" ändern.

Der Grund dafür ist, dass Vista Applikationen nicht in den Bereich C:\Program Files schreiben lässt.

1. Bearbeiten Sie die Datei "**setEnv.cmd**" im Installationsverzeichnis, sodass die Umgebungsvariable JAVA_HOME auf den Ordner Ihrer JVM verweist.
Eventuell kommt es in **Vista** beim Speichern der Datei zu einem Problem.
2. Wenn das der Fall ist, starten Sie den Windows Explorer und gehen Sie zu C:\Windows\System32.
3. Klicken Sie auf die rechte Maustaste und führen Sie "cmd.exe" "As Administrator" aus. Daraufhin erscheint ein Eingabefenster.
4. Geben Sie "cd <Installationsordner des QVCS –Enterprise Client>" ein.
5. Geben Sie "Notepad setEnv.cmd" ein und bearbeiten Sie anschließend die Datei und speichern Sie sie.

6. Führen Sie vom Installationsverzeichnis des Qumasoft QVCS-Enterprise die Batch-Datei "gui.bat" aus.
7. Fügen Sie einen Server aus dem "Server-Menü" hinzu. Geben Sie den erforderlichen Namen, die IP-Adresse und Ports und den Log-in-Namen ein und definieren Sie einen lokalen Arbeitsbereich.

Qumasoft QVCS-Pro

<http://www.qumasoft.com/>

Unterstützte Versionen: 3.10.18

Um Qumasoft QVCS-Pro zu installieren, führen Sie den Installer aus.

- Wenn Ihr Betriebssystem **Vista** ist, müssen Sie das Installationsverzeichnis vom Standardwert "C:\Program Files\QVCSBin" in "C:\QVCSBin" ändern.
Der Grund dafür ist, dass Vista Applikationen nicht in den Bereich C:\Program Files schreiben lässt.
- Starten Sie nach Fertigstellung der Installation den QVCS 3.10 Client, erstellen Sie einen neuen Benutzer und aktivieren Sie die **Ide Integration** durch Auswahl des Untermenüs "Ide Integration" im Menü "Admin" und Hinzufügen von QVCS als Versionskontroll-Tool.
- Erstellen Sie ein Projekt und definieren Sie einen Arbeitsbereich.

Reliable Software Code Co-Op

http://www.relisoft.com/co_op/index.htm

Unterstützte Versionen: Code Co-Op 5.1a

Eine Standard-Installation funktioniert mit allen Altova-Produkten ordnungsgemäß.

Seapine Surround SCM

<http://www.seapine.com/surroundscm.html>

Unterstützte Versionen: Surround SCM Client for Windows 2009.0.0

Eine Standard-Installation funktioniert mit allen Altova-Produkten ordnungsgemäß.

- Nach der Installation muss eine Verbindung zu einem Server hergestellt werden.

Serena Dimensions

<http://www.serena.com/Products/dimensions/>

Unterstützte Versionen: Dimensions Express/CM 10.1.3 for Win32 Client

- Führen Sie eine "Typical" Installation des Serena Dimension Client durch.
- Definieren Sie den WEB Client Hostnamen und die Port-Nummer, wenn Sie dazu aufgefordert werden.

Softimage Alienbrain

<http://www.alienbrain.com/>

Unterstützte Versionen: Alienbrain Essentials/Advanced Client 8.1.0.7300

- Führen Sie eine "Typical" Installation der Alienbrain Client Software aus und installieren Sie anschließend die Alienbrain Microsoft Visual Studio Integration.

Um mit Altova-Produkten zu arbeiten, müssen Sie Microsoft Visual Studio nicht

installieren.

- Beim ersten Öffnen eines Projekts von der Versionskontrolle aus oder beim Hinzufügen eines Projekts zur Versionskontrolle, werden Sie aufgefordert einige Benutzereinstellungen zu definieren, z.B. Ihren Server und die Projektdatenbank, zu der eine Verbindung hergestellt werden soll.

SourceGear Fortress

<http://www.sourcegear.com/fortress>

Unterstützte Versionen: Fortress 1.1.4 Client

Eine Standard-Installation von SourceGear Fortress Client funktioniert mit Altova-Produkten.

SourceGear SourceOffsite

<http://www.sourcegear.com/sos/>

Unterstützte Versionen: SourceOffsite Client 4.2.0 (Windows)

Eine Standard-Installation von SourceOffsite Client funktioniert mit Altova-Produkten.

SourceGear Vault

<http://www.sourcegear.com/vault>

Unterstützte Versionen: 4.1.4 Client

Eine Standard-Installation von SourceGear Vault Client funktioniert mit Altova-Produkten.

TamTam CVS SCC

<http://www.daveswebsite.com/software/tamtam>

Unterstützte Versionen: 1.2.36

Voraussetzungen: CVSNT 2.5 (client) unter <http://www.cvsnt.org>

Eine Standard-Installation funktioniert mit allen Altova-Produkten ordnungsgemäß.

- Um eine Verbindung zur CVS-Speicherschnittstelle herzustellen, müssen Sie CVSNT installieren.
- Öffnen Sie im Altova-Produkt die erweiterten Optionen der "Versionskontrolle" und geben Sie den Pfad zur Datei **cvsexec** ein.

TamTam SVN SCC

<http://www.daveswebsite.com/software/tamtamsvn>

Unterstützte Versionen: 1.2.21

Voraussetzungen: Subversion Command Line Client 1.5.4 unter <http://subversion.tigris.org>

Eine Standard-Installation funktioniert mit allen Altova-Produkten ordnungsgemäß.

- Um eine Verbindung zur SVN-Speicherschnittstelle herzustellen, müssen Sie den Subversion Command Line Client installieren und in den Altova-Versionskontrolloptionen den Pfad zur ausführbaren Datei **svn.exe** definieren.

- Nach Start von UModel/XMLSpy müssen Sie den SCC Provider registrieren.

Auf einem **Vista** Rechner kann die CSS-Registrierung fehlschlagen.

- Falls dies der Fall ist, suchen Sie im Windows Explorer das Verzeichnis, das die ausführbare Altova-Applikationsdatei enthält.
- Klicken Sie auf die rechte Maustaste und führen Sie die Datei "As Administrator" aus. Die SCC-Registrierung sollte jetzt klappen.

QSC Team Coherence Version Manager

<http://www.teamcoherence.com>

Unterstützte Versionen: Team Coherence Client 7.1.4.30

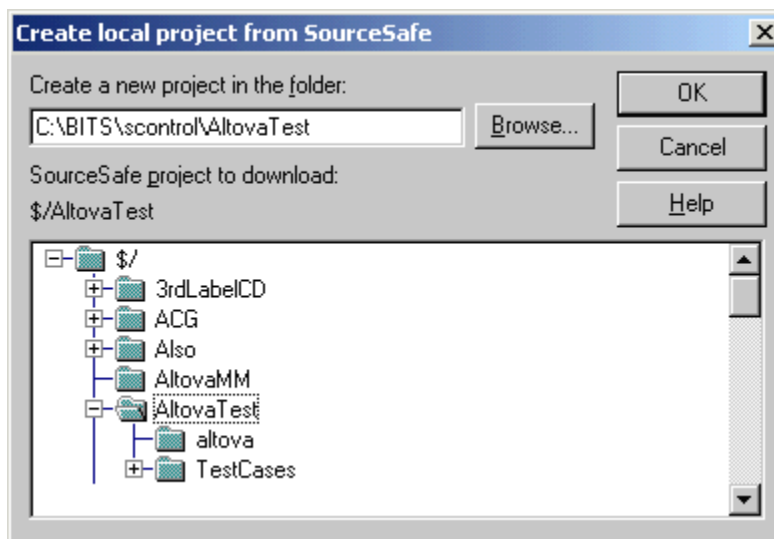
Eine Standard-Installation funktioniert mit allen Altova-Produkten ordnungsgemäß.

- Wenn der Server auf dem Client-Rechner installiert ist, wird nach Installation des Client eine Standardverbindung hergestellt.
- Wenn der Server auf einem anderen Rechner installiert ist, müssen Sie im Dialogfeld "Connection Properties" des Team Coherence Client die Eigenschaft "HOSTNAME" in den entsprechenden Rechner ändern.

Aus Versionskontrolle öffnen

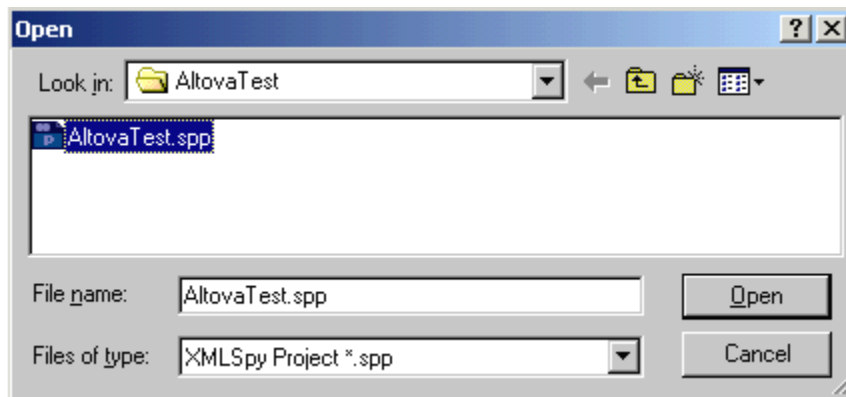
Mit dem Befehl "Aus Versionskontrolle öffnen" wird ein lokales Projekt anhand einer bestehenden Versionskontrolldatenbank erstellt. Dieser Befehl kann nur wirksam werden, wenn vorher über die Menüoption [Zu Versionskontrolle hinzufügen](#) ein Authentic Desktop Projekt zur Versionskontrolle hinzugefügt wurde.

1. Wählen Sie **Projekt | Versionskontrolle | Aus Versionskontrolle öffnen**. Daraufhin wird das Anmeldedialogfeld geöffnet, in dem Sie Ihre Anmeldedaten eingeben, bevor Sie fortfahren können.
2. Daraufhin wird das Dialogfeld "Create local project from SourceSafe" angezeigt (*Abbildung unten*). Definieren Sie das Verzeichnis, das das neue lokale Projekt enthalten soll. Dies wird zum neuen **Arbeitsverzeichnis** oder dem Auscheck-Ordner.



3. Wählen Sie das gewünschte SourceSafe-Projekt aus. Wenn der hier von Ihnen

definierte Ordner noch nicht existiert, erscheint ein Dialogfeld, in dem Sie aufgefordert werden, ihn zu erstellen. Klicken Sie auf **Ja**, um das neue Verzeichnis zu erstellen. Das Dialogfeld "Öffnen" wird angezeigt. (Abbildung unten).



4. Wählen Sie die gewünschte Projektdatei aus und klicken Sie auf "Öffnen". Die Datei wird nun in Authentic Desktop geöffnet und die Datei wird unter Versionskontrolle gestellt. Angezeigt wird dies in der Modell-Struktur durch ein Schlosssymbol im Dateisymbol im Projektordner.

Symbole der Versionskontrolle:

Die Ordner und Dateien sind mit bestimmten Symbolen versehen, deren Bedeutung im Folgenden erklärt wird.



oder

Das Vorhängeschloss-Symbol zeigt an, dass diese Datei oder dieser Ordner **unter Versionskontrolle steht**, aber derzeit nicht ausgecheckt ist.



oder

Das **rote Kontrollhäkchen** zeigt an, dass diese Datei **ausgecheckt** ist, d.h. dass die Authentic Desktop-Projektdatei zur Bearbeitung ausgecheckt wurde. Das Sternchen bedeutet, dass diese Datei verändert wurde. Beim Schließen werden Sie aufgefordert, die Datei zu speichern.

Das Personensymbol zeigt an, dass die Datei(en) von jemand anderem im Netzwerk ausgecheckt wurde(n).

Versionskontrolle aktivieren

Mit Hilfe dieses Befehls können Sie die Versionskontrolle für ein Authentic Desktop-Projekt aktivieren oder deaktivieren. Bei Auswahl dieser Option für eine Datei oder einen Ordner, wird die Versionskontrolle für das gesamte Projekt aktiviert/deaktiviert.

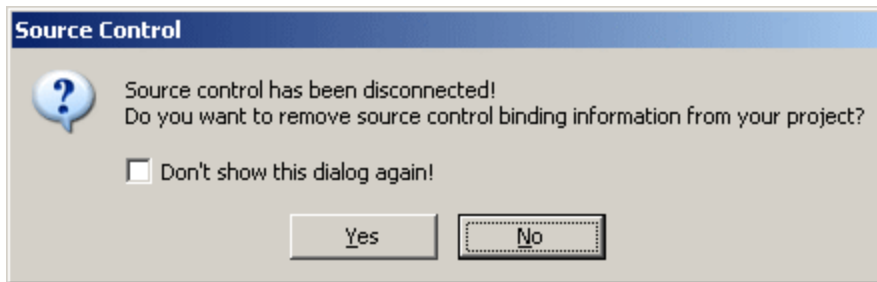
So aktivieren Sie die Versionskontrolle für ein Projekt:

1. Klicken Sie auf jede beliebige Datei oder jeden beliebigen Ordner im Projektfenster.
2. Wählen Sie die Menüoption **Projekt | Versionkontrolle | Versionskontrolle aktivieren**. Der vorherige Ein-/Auscheck-Status der verschiedenen Dateien wird abgerufen und im Projektfenster angezeigt.

So deaktivieren Sie die Versionskontrolle für ein Projekt:

Wählen Sie die Menüoption **Projekt | Versionkontrolle | Versionskontrolle aktivieren**. Sie

werden nun gefragt, ob Sie die Binding-Informationen aus dem Projekt entfernen möchten. (siehe Abbildung unten).



Um die Versionskontrolle **provisorisch** zu deaktivieren, wählen Sie **Nein**.
Um die Versionskontrolle für dieses Projekt **permanent** zu deaktivieren, wählen Sie **Ja**.

Aktuellste Version holen

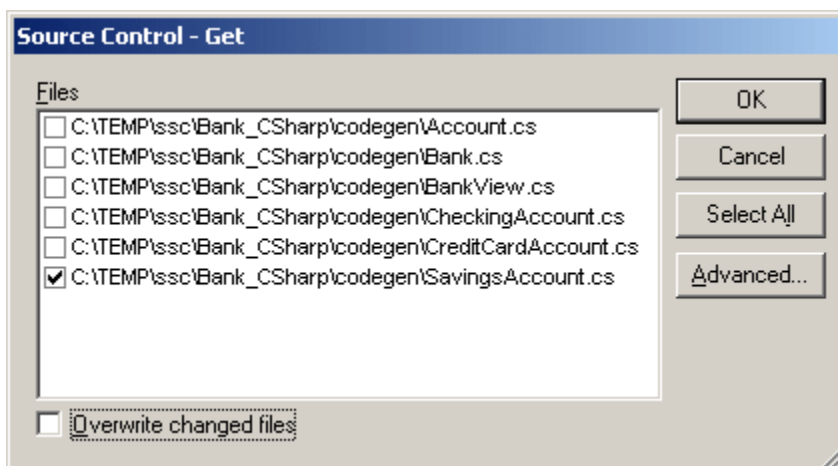
Mit diesem Befehl wird die aktuellste Version der ausgewählten Datei(en) aus dem Versionskontrollspeicher in das Arbeitsverzeichnis geholt. Die Dateien werden als schreibgeschützte Dateien und nicht ausgecheckt abgerufen. Bei Verwendung dieses Befehls stehen keine weiteren Optionen zur Verfügung.

So rufen Sie die neueste Version einer Datei ab:

1. Wählen Sie die Dateien aus, für die Sie die neueste Version holen möchten.
2. Wählen Sie **Projekt | Versionskontrolle | Aktuellste Version holen**.

Abrufen

Ruft eine schreibgeschützte Kopie der ausgewählten Dateien ab und platziert Sie in den Arbeitsordner. Standardmäßig sind die Dateien nicht zur Bearbeitung ausgecheckt. Um eine Datei abzurufen, wählen Sie sie im Projektfenster aus (es können auch mehrere Dateien ausgewählt werden) und wählen Sie dann den Befehl **Abrufen**. Daraufhin wird das Dialogfeld "Abrufen" (Abbildung unten) geöffnet. Versehen Sie die abzurufenden Dateien mit einem Häkchen.



Kontrollkästchen "Geänderte Dateien überschreiben"

Überschreibt lokal geänderte Dateien mit jenen aus der Versionskontrolldatenbank.

Alle auswählen

Wählt alle Dateien im Listenfeld aus.

Erweitert

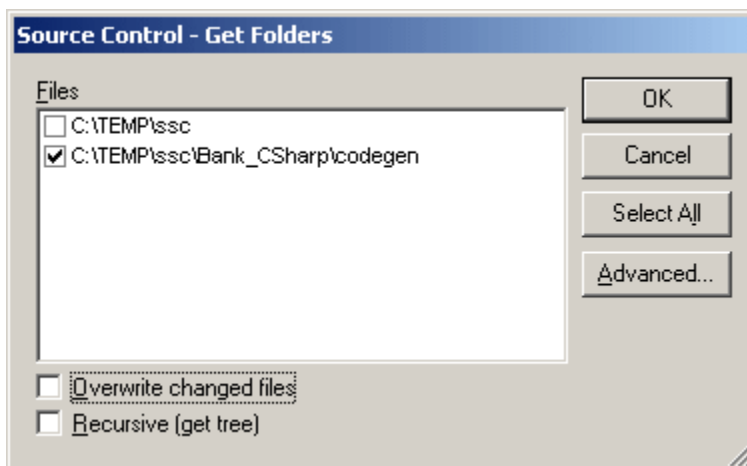
Mit Hilfe dieser Option können Sie die Optionen **Replace writable** und **Set timestamp** in den entsprechenden Auswahllisten auswählen.



Mit dem Kontrollkästchen "Make writable" wird das "read-only" Attribut der abgerufenen Dateien entfernt.

Ordner abrufen

Ruft schreibgeschützte Kopien von Dateien in den ausgewählten Ordnern ab und platziert Sie in den Arbeitsordner. Die Dateien werden standardmäßig nicht zur Bearbeitung ausgecheckt. Um einen Ordner abzurufen, wählen Sie ihn im Projektfenster aus und wählen Sie dann den Befehl **Ordner abrufen**. Daraufhin wird das Dialogfeld "Abrufen" (*Abbildung unten*) geöffnet. Versehen Sie die abzurufenden Ordner mit einem Häkchen.

**Kontrollkästchen "Geänderte Dateien überschreiben"**

Überschreibt lokal geänderte Dateien mit jenen aus der Versionskontrolldatenbank.

Kontrollkästchen Rekursiv (Struktur abrufen)

Ruft alle Dateien der Ordnerstruktur unterhalb des ausgewählten Ordners ab.

Alle auswählen

Wählt alle Dateien im Listenfeld aus.

Erweitert

Mit Hilfe dieser Option können Sie die Optionen **Replace writable** und **Set timestamp** in den entsprechenden Auswahllisten auswählen.



Mit dem Kontrollkästchen "Make writable" wird das "read-only" Attribut der abgerufenen Dateien entfernt.

Auschecken

Mit diesem Befehl checken Sie die aktuellste Version der ausgewählten Datei(en) aus und platzieren beschreibbare Kopien davon in das Arbeitsverzeichnis. Die Dateien werden für andere Benutzer als "ausgecheckt" markiert.

Um Dateien auszuchecken, gehen Sie folgendermaßen vor:

1. Wählen Sie die gewünschte Datei bzw. den gewünschten Ordner in der Modell-Struktur aus.
2. Wählen Sie **Projekt | Versionskontrolle | Auschecken**.
3. Wählen Sie im Dialogfeld "Auschecken" die gewünschten Dateien aus und klicken Sie auf **OK**.

Anmerkung:

- Sie können die Anzahl der auszucheckenden Dateien ändern, indem Sie die einzelnen Kontrollkästchen in der Liste der Dateien aktivieren.
- Bei Auswahl der Option *Lokale Version auschecken* wird nur die lokale Version der Dateien ausgecheckt, nicht die aus der Versionskontrolldatenbank.
- Folgende Objekte können ausgecheckt werden: (i) Einzelne Dateien - (klicken Sie im Projektfenster auf die Datei (oder mehrere Dateien mittels Strg + Klick); (ii) Projektordner - klicken Sie im Projektfenster auf den Ordner (oder mehrere Ordner mittels Strg + Klick)
- Ausgecheckte Dateien und Ordner werden mit einem roten Häkchen markiert.
-

Erweitert

Mit Hilfe dieser Option können Sie die Optionen **Replace writable** und **Set timestamp** in den entsprechenden Auswahllisten auswählen.



Mit dem Kontrollkästchen "Make writable" wird das "read-only" Attribut der abgerufenen Dateien entfernt.

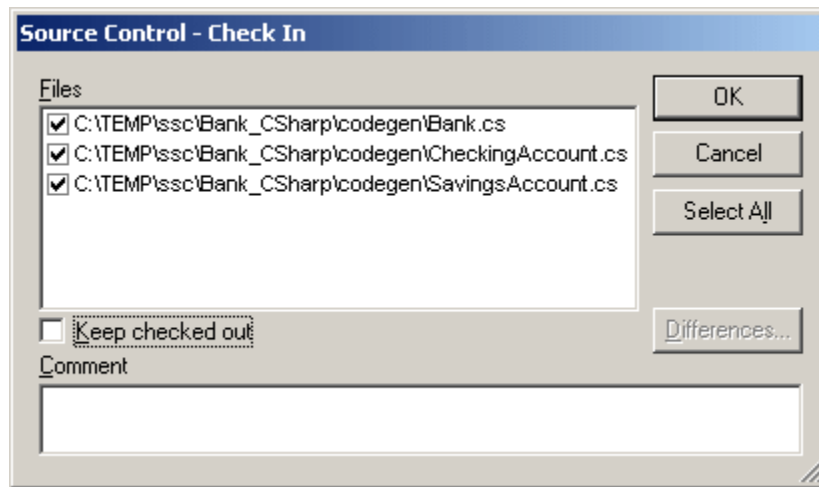
Einchecken

Mit diesem Befehl checken Sie zuvor ausgecheckte Dateien, d.h. Ihre lokal aktualisierten Dateien, wieder ein und stellen die Dateien zurück in das Versionskontrollprojekt.

Um Dateien einzuchecken, gehen Sie folgendermaßen vor:

1. Wählen Sie die gewünschten Dateien in der Modell-Struktur aus.

2. Wählen Sie **Projekt | Versionskontrolle | Einchecken**.
3. Wählen Sie im Dialogfeld "Einchecken" die gewünschten Dateien aus und klicken Sie auf OK.



Anmerkung:

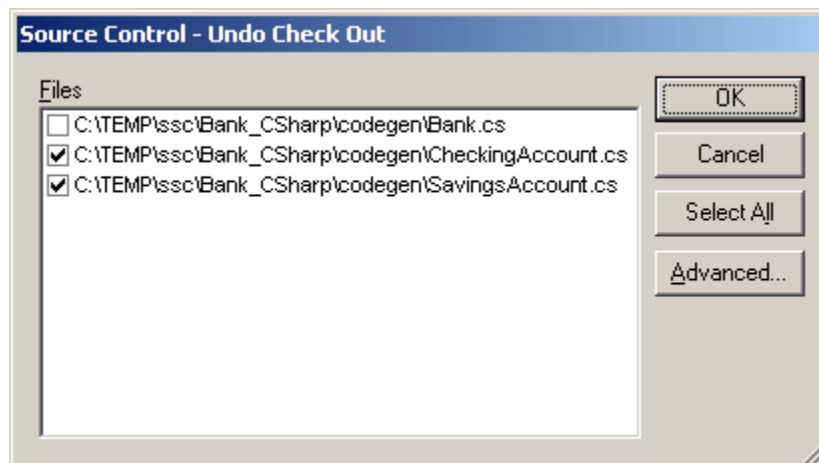
- Rechtsklicken Sie im Projektfenster auf ein ausgechecktes Objekt, und wählen Sie "Einchecken" im Kontextmenü.
- Folgende Objekte können eingchecked werden: (i) Einzelne Dateien - klicken Sie im Projektfenster auf die Datei (oder mehrere Dateien mittels Strg + Klick) (ii) Ordner - klicken Sie im Projektfenster auf die Ordner (mittels Strg + Klick)
- Das Vorhängeschloss-Symbol zeigt an, dass sich die Datei/der Ordner **unter Versionskontrolle** befindet, aber derzeit nicht ausgecheckt ist.

Auschecken rückgängig...

Mit diesem Befehl werden Änderungen in zuvor ausgecheckten Dateien, also lokal aktualisierten Dateien **verworfen**. Es werden die alten Dateien aus der Versionskontrolldatenbank beibehalten.

Um das Auschecken rückgängig zu machen, gehen Sie folgendermaßen vor:

1. Wählen Sie die Dateien im Projektfenster aus
2. Wählen Sie **Projekt | Versionskontrolle | Auschecken rückgängig**.
3. Wählen Sie im Dialogfeld "Auschecken rückgängig" die gewünschten Dateien, für die das Auschecken rückgängig gemacht werden soll und klicken Sie auf OK.



Die Option "Auschecken rückgängig" kann auf die folgenden Dateien angewendet werden: (i) Einzelne Dateien - klicken Sie im Projektfenster auf die Datei (oder mehrere Dateien mittels Strg + Klick), (ii) Ordner - klicken Sie im Projektfenster auf die Ordner (mittels Strg + Klick).

Erweitert

Mit Hilfe dieser Option können Sie die Optionen **Replace writable** und **Set timestamp** in den entsprechenden Auswahllisten auswählen.



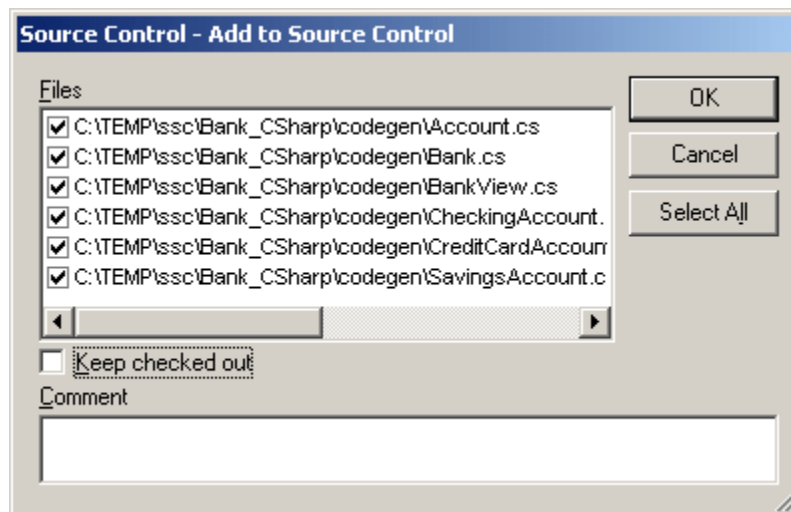
Mit dem Kontrollkästchen "Make writable" wird das "read-only" Attribut der abgerufenen Dateien entfernt.

Zu Versionskontrolle hinzufügen

Fügt die ausgewählten Dateien oder Ordner in einem Projektordner zur Versionskontrolldatenbank hinzu und stellt sie unter Versionskontrolle.

Um Dateien zur Versionskontrolle hinzuzufügen, gehen Sie folgendermaßen vor:

1. Rechtsklicken Sie im Projektfenster auf eine Datei und wählen Sie die Menüoption **Projekt | Versionskontrolle | Zu Versionskontrolle hinzufügen**
2. Wenn Sie einen Ordner ausgewählt haben, werden Sie nach einem Ordnerpfad gefragt, wo die Datei gespeichert werden soll. Geben Sie einen Ordner an und klicken Sie auf OK.
3. Stellen Sie sicher, dass die Dateien, die hinzugefügt werden sollen, im Dialogfeld "Zu Versionskontrolle hinzufügen", das angezeigt wird, angehakt sind. Wenn die hinzuzufügenden Dateien ausgecheckt bleiben sollen, aktivieren Sie das Kontrollkästchen "Ausgecheckt lassen" und klicken Sie anschließend auf OK.



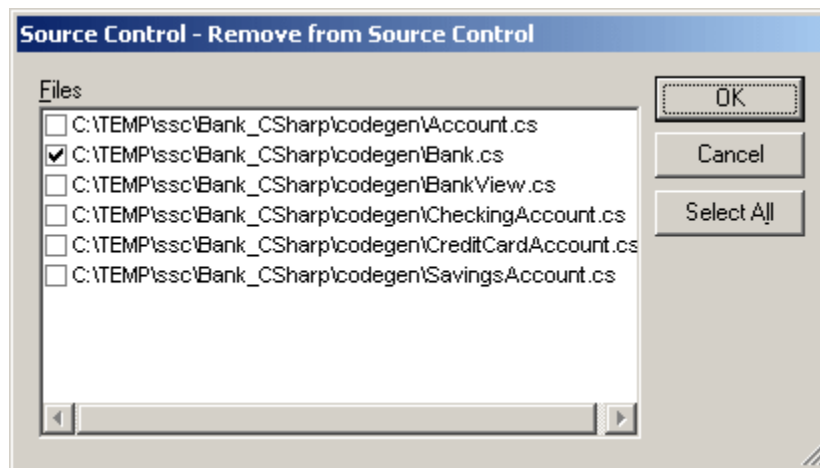
Neben den einzelnen unter Versionskontrolle gestellten Dateien erscheint nun ein Vorhängeschloss. Wenn die Dateien ausgecheckt werden, werden sie mit einem roten Häkchen angezeigt.

Von Versionskontrolle ausgliedern

Mit diesem Befehl **entfernen** Sie zuvor hinzugefügte Dateien aus dem Versionskontrollprodukt. Diese Dateiarten werden zwar im Projektfenster weiterhin angezeigt, können aber nicht ein- oder ausgecheckt werden. Mit Hilfe des Befehls "Zu Versionskontrolle hinzufügen" können Sie die Dateien wieder unter Versionskontrolle stellen.

Um Dateien aus der Versionskontrolle auszugliedern, gehen Sie folgendermaßen vor:

1. Wählen Sie die gewünschten Dateien im Projektfenster aus
2. Wählen Sie **Projekt | Versionskontrolle | Von Versionskontrolle ausgliedern**.
3. Wählen Sie im Dialogfeld "Von Versionskontrolle ausgliedern" die gewünschten Dateien aus, die entfernt werden sollen und klicken Sie auf OK.



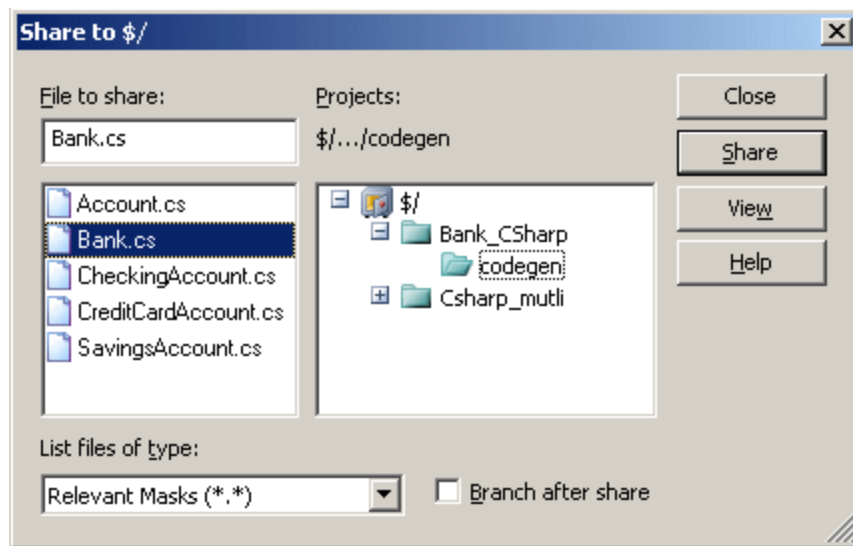
Folgende Objekte können aus der Versionskontrolle ausgegliedert werden: (i) Einzelne Dateien - klicken Sie im Projektfenster auf die Datei (oder mehrere Dateien mittels Strg + Klick), (ii) Ordner - klicken Sie im Projektfenster auf die gewünschten Ordner (**Strg+Klick**)

Aus Versionskontrolle freigeben

Um dem Freigeben-Befehl verwenden zu können, müssen Sie für das Projekt, das Sie freigeben möchten, Ein- und Auscheck-Rechte besitzen.

Um eine Datei aus der Versionskontrolle freizugeben, gehen Sie folgendermaßen vor:

1. Wählen Sie die gewünschte Datei im Projektfenster aus und wählen Sie den Befehl **Projekt | Versionskontrolle | Aus Versionskontrolle freigeben**.
2. Wählen Sie im Listenfeld "Projekte" den Projektordner aus, der die Datei enthält.



3. Wählen Sie im Listenfeld "Files to share" die gewünschte Datei aus und klicken Sie auf die Schaltfläche "Freigeben".
Die Datei wird nun aus der Liste "Files to share" entfernt.
4. Klicken Sie zum Fortfahren auf "Schließen".

Branch after share

Gibt die Datei frei und erstellt einen neuen Branch zum Erstellen einer separaten Version.

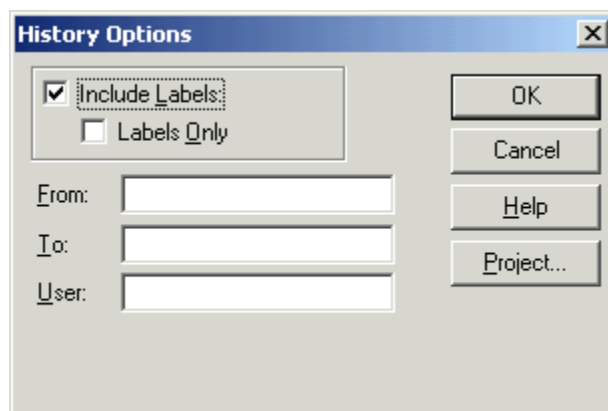
Verlauf anzeigen

Mit diesem Befehl zeigen Sie den Verlauf einer unter Versionskontrolle stehenden Datei an. Sie können damit eine detaillierte Verlaufsliste anzeigen, genauere Verlaufsinformationen anzeigen, einen Vergleich durchführen oder frühere Versionen der Datei abrufen.

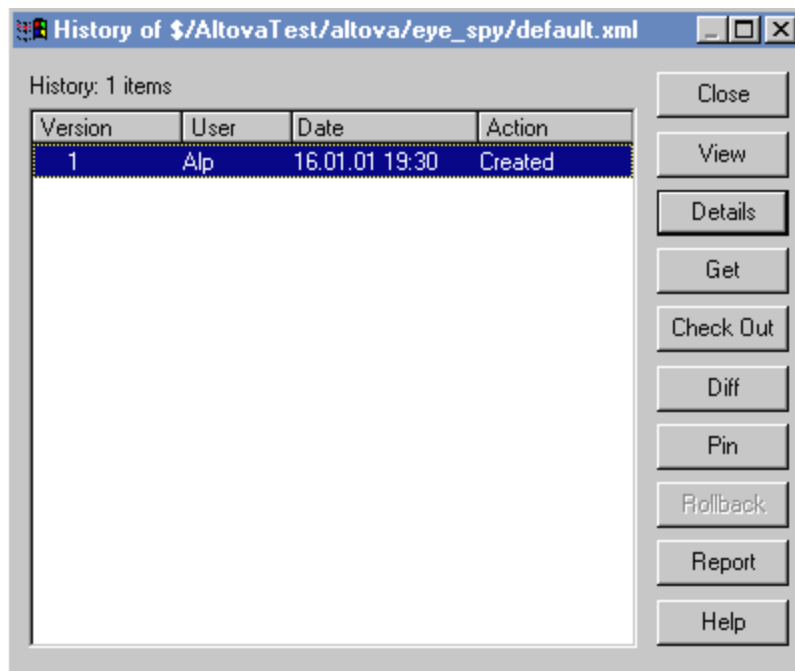
Um den Verlauf einer Datei anzuzeigen, gehen Sie folgendermaßen vor.

1. Klicken Sie Projektfenster auf die Datei, deren Verlauf Sie ansehen möchten.
2. Wählen Sie die Menüoption **Projekt | Versionskontrolle | Verlauf anzeigen**.

Es erscheint eventuell ein Dialogfeld, in dem Sie nähere Informationen eingeben müssen (in diesem Beispiel wird Visual Source Safe verwendet).



3. Wählen Sie den gewünschten Eintrag und bestätigen Sie mit OK.



Dieses Dialogfeld bietet verschiedene Möglichkeiten, um bestimmte Versionen der ausgewählten Datei zu vergleichen und abzurufen. Wenn Sie auf einen Eintrag in der Liste doppelklicken, wird das Dialogfeld "Verlauf" für diese Datei geöffnet. Die Schaltflächen in Dialogfeld haben die folgenden Funktionen:

- **Schließen:** Schließt dieses Dialogfeld.
- **Ansicht:** Öffnet ein weiteres Dialogfeld, in dem Sie die Art des Ansichtsprogramms auswählen können, in dem Sie die Datei anzeigen möchten.
- **Details:** Öffnet ein Dialogfeld, in dem Sie die [Eigenschaften](#) der gerade aktiven Datei sehen.
- **Abrufen:** Damit können Sie eine der vorherigen Versionen der Datei aus der Versionsliste in das Arbeitsverzeichnis holen.
- **Auschecken:** Damit können eine frühere Version der Datei auschecken.
- **Diff:** Öffnet das Dialogfeld [Vergleichsoptionen](#), in dem Sie die Vergleichsoptionen zur Anzeige der Unterschiede zwischen den beiden Dateiversionen definieren können. Mit Hilfe von **Strg + Klick** können Sie zwei Dateiversionen in diesem Fenster markieren. Klicken Sie anschließend auf Diff, um die Unterschiede zwischen den beiden Dateien anzuzeigen.
- **Pin:** Markiert eine Version der Datei. Damit können Sie die Dateiversion definieren, die für den Dateivergleich verwendet werden soll.
- **Rollback:** Führt ein Rollback für die ausgewählte Version der Datei durch.
- **Report:** Generiert einen Verlaufsbericht, den Sie an den Drucken, die Datei oder die Zwischenablage senden können.
- **Hilfe:** Öffnet die Online-Hilfe des Versionskontrollanbieters für das Plugin.

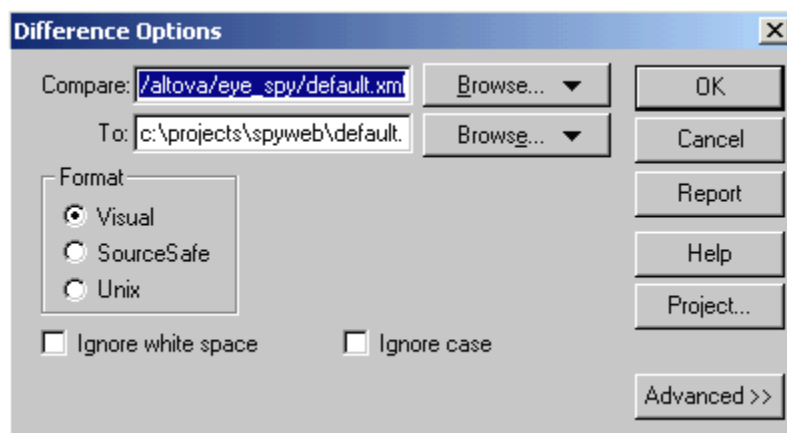
Unterschiede anzeigen

Dieser Befehl ermöglicht die Anzeige von Unterschieden zwischen der in der Versionskontroll-Ablage befindlichen und der gleichnamigen Datei, die Sie **ein-/ausgecheckt** haben. Wenn Sie eine der Dateien im Dialogfeld "Verlauf" mit einem Pin markiert haben, so wird diese Datei im Textfeld "Vergleichen" verwendet. Sie können mit Hilfe der

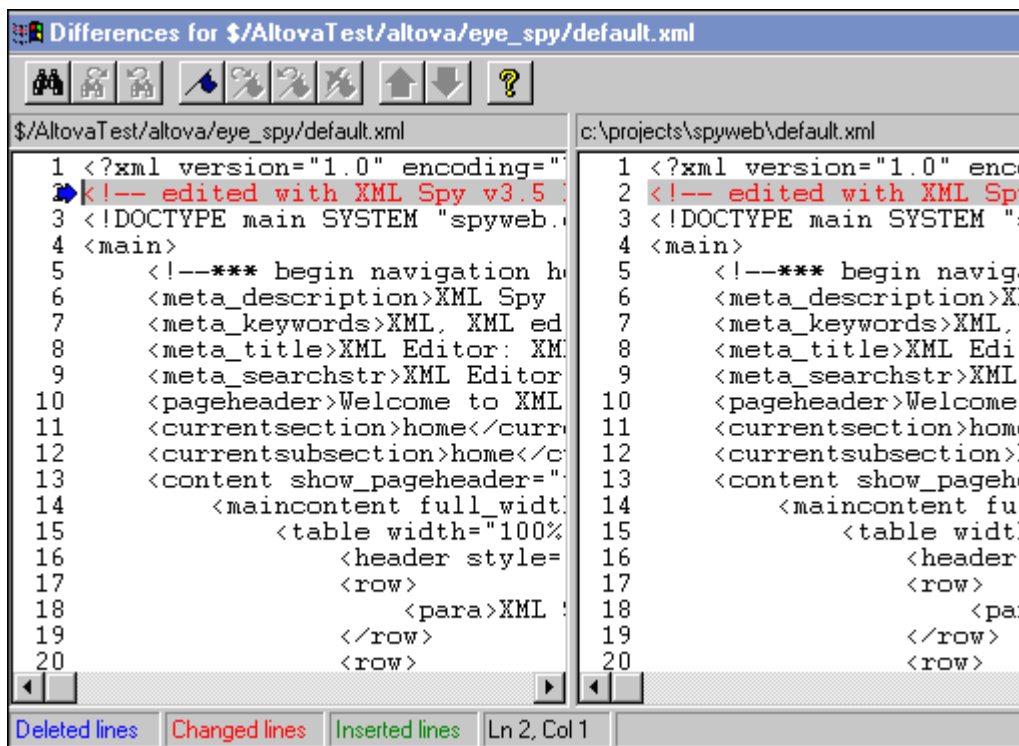
Durchsuchen-Schaltfläche 2 beliebige Dateien auswählen.

So zeigen Sie die Unterschiede zwischen zwei Dateien an:

1. Checken Sie eine Datei aus Ihrem Projekt aus. Klicken Sie im Projektfenster auf die Datei.
2. Wählen Sie die Menüoption **Projekt | Versionskontrolle | Unterschiede anzeigen**. Es erscheint eventuell ein Dialogfeld, in dem Sie weitere Informationen eingeben können.



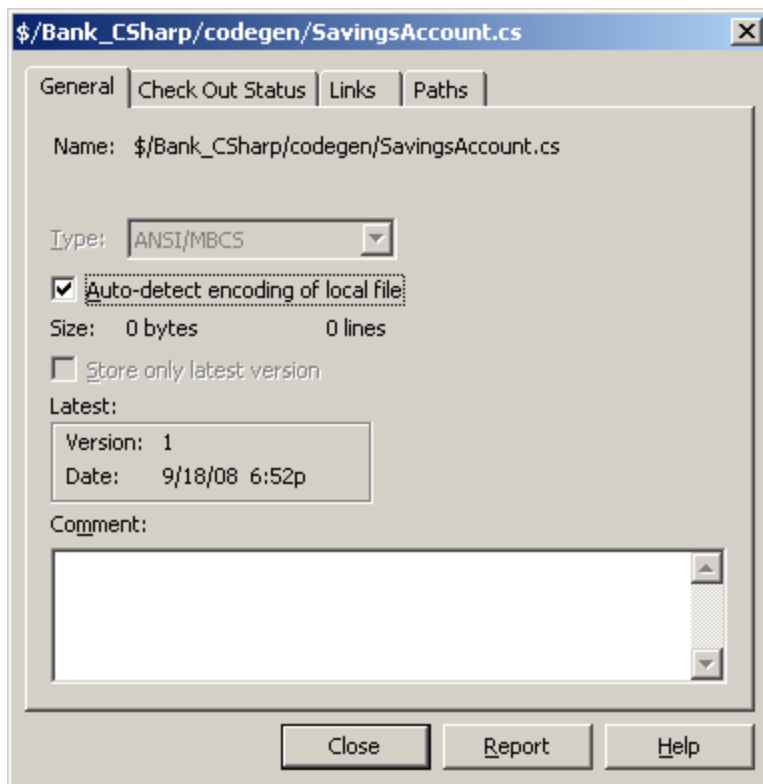
3. Wählen Sie die gewünschten Einträge und bestätigen Sie mit OK.



Die Unterschiede zwischen beiden Dateien werden in beiden Fenstern farblich markiert (in diesem Beispiel wird MS Source-Safe verwendet).

Eigenschaften anzeigen

Mit diesem Befehl können Sie die Eigenschaften der ausgewählten Datei anzeigen (*Abbildung unten*). Die Anzeige kann je nach verwendetem Versionskontrollprodukt unterschiedlich sein.



Beachten Sie, dass dieser Befehl nur für einzelne Dateien verwendet werden kann.

Status aktualisieren

Mit diesem Befehl **aktualisieren** Sie den Status aller Projektdateien unabhängig von ihrem derzeitigen Status.

Versionskontrollmanager

Mit diesem Befehl **starten** Sie die ursprüngliche Oberfläche Ihrer Versionskontroll-Software.

9.3.7 Dateien zu Projekt hinzufügen



Mit diesem Befehl fügen Sie Dateien in das aktuelle Projekt ein. Verwenden Sie diesen Befehl um Dateien in einen beliebigen Ordner Ihres Projekts einzufügen. Sie können einzelne Dateien oder Dateigruppen (Auswahl mit **Strg + Klick**) im Dialogfeld "Öffnen" hinzufügen. Wenn Sie Dateien in ein Projekt einfügen, werden diese automatisch - nach Erweiterung sortiert - in die entsprechenden Projektordner verteilt (Einstellungen hierzu im Dialogfeld [Projekteigenschaften](#)).

9.3.8 Globale Ressource zu Projekt hinzufügen

Mit dem Befehl **Projekt | Globale Ressource zu Projekt hinzufügen...** wird das Dialogfeld "Globale Ressource auswählen" aufgerufen, in dem Sie eine globale Ressource vom Typ Datei oder Ordner auswählen können, um diese zum Projekt hinzuzufügen. Bei Auswahl einer globalen Ressource vom Typ Datei wird eine Datei auf Basis der im Dialogfeld [Projekteigenschaften](#) definierten Dateierweiterung zum entsprechenden Ordner hinzugefügt. Bei Auswahl einer globalen Ressource vom Typ Ordner wird ein Ordner in einem "Datei öffnen" Dialogfeld geöffnet und Sie werden aufgefordert, eine Datei auszuwählen; die ausgewählte Datei wird auf Basis der im Dialogfeld [Projekteigenschaften](#) definierten Dateierweiterung zum entsprechenden Ordner hinzugefügt. Eine Beschreibung globaler Ressourcen finden Sie im Abschnitt "Globale Ressourcen" in dieser Dokumentation.

9.3.9 URL zu Projekt hinzufügen



Mit diesem Befehl fügen Sie eine URL in das aktuelle Projekt ein. Bei Verwendung von URLs in einem Projekt wird das Ziel der URL in das Projekt einbezogen. Wenn eine Batch-Operation an einer URL oder einer Datei mit URL ausgeführt wird, sucht nach dem Dokument aus der URL, und führt die gewünschte Operation aus.

9.3.10 Aktive Datei zu Projekt hinzufügen



Mit diesem Befehl fügen Sie die aktive Datei in das aktuelle Projekt ein. Wenn Sie eine Datei von der Festplatte oder über eine URL geöffnet haben, können Sie die Datei mit diesem Befehl in das aktuelle Projekt aufnehmen.

9.3.11 Aktive und verwandte Dateien zu Projekt hinzufügen



Mit diesem Befehl fügen Sie das aktive XML-Dokument und alle verwandten Dateien in das Projekt ein. Beim Arbeiten mit einem XML-Dokument, das auf einem Schema oder einer DTD basiert, fügt dieser Befehl nicht nur das XML-Dokument zum Projekt hinzu sondern auch alle verwandten Dateien (z.B. DTDs und sämtliche geparste Entitäten, auf die die DTD referenziert).

Bitte beachten Sie: Dateien, die durch Processing Instructions referenziert werden (z.B. XSLT-Dateien) werden nicht als verwandte Dateien behandelt.

9.3.12 Projektordner zu Projekt hinzufügen



Mit diesem Befehl fügen Sie einen neuen Ordner in das aktuelle Projekt ein. Verwenden Sie diesen Befehl, um einen neuen Ordner oder Unterordner im aktuellen Projekt anzulegen. Sie können diesen Befehl auch über das Kontextmenü (Rechtsklick auf einen Ordner im Projektfenster) aufrufen.

Anmerkung

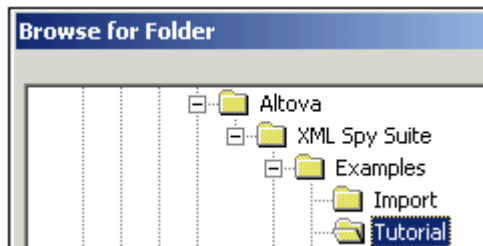
Ein Projektordner in einen anderen Projektordner oder an eine andere Stelle im selben Projekt gezogen werden. Zusätzlich dazu kann ein Ordner aus dem Windows-Datei-Explorer in jeden beliebigen Projektordner gezogen werden.

9.3.13 Externes Verzeichnis zum Projekt hinzufügen

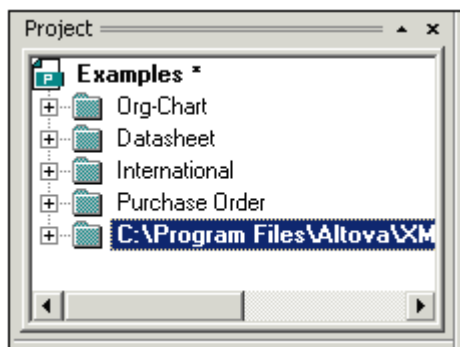
Mit diesem Befehl fügen Sie einen neues **externes Verzeichnis** in das aktuelle Projekt ein. Verwenden Sie diese Funktion, um ein lokales Verzeichnis (oder ein Netzwerkverzeichnis) in das aktuelle Projekt einzufügen. Sie können diesen Befehl auch über das Kontextmenü aufrufen (Rechtsklick auf ein Verzeichnis im Projektfenster). Bitte beachten Sie, dass Dateien in externen Verzeichnissen nicht unter Versionskontrolle gestellt werden können.

So fügen Sie ein externes Verzeichnis in ein Projekt ein:

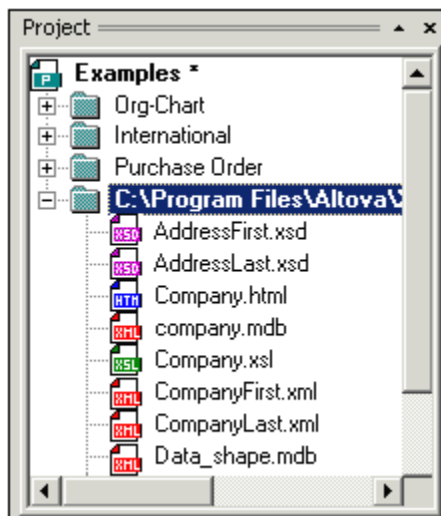
1. Wählen Sie die Menüoption **Projekt | Externes Verzeichnis zu Projekt hinzufügen...**
2. Wählen Sie das Verzeichnis, welches hinzugefügt werden soll (Dialogfeld "Ordner suchen"), und klicken Sie auf OK zum Bestätigen.



Das ausgewählte Verzeichnis wird nun im Projektfenster angezeigt.

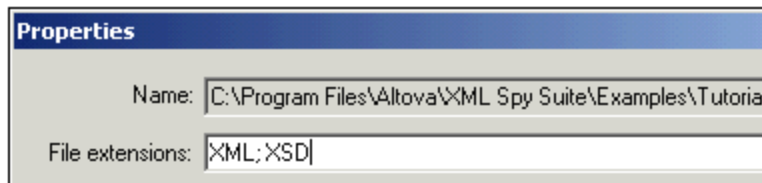


3. Klicken Sie auf das Symbol "+", um den Inhalt des Verzeichnisses anzuzeigen.

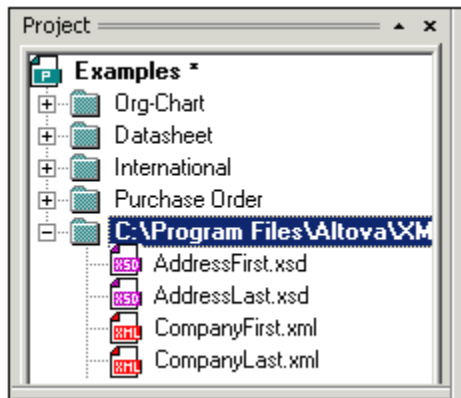


So filtern Sie den Inhalt eines Verzeichnisses:

1. Rechtsklicken Sie auf den lokalen Ordner. Wählen Sie im Popup-Menü die Option **Eigenschaften**.
Das Dialogfeld Eigenschaften wird geöffnet.



2. Klicken Sie auf das Feld **Erweiterungen**, und geben Sie die Erweiterungen der Dateien an, die angezeigt werden sollen. Bei mehreren Erweiterungen trennen Sie diese durch ein **Semikolon** (in diesem Beispiel XML und XSD Schema).
3. Klicken Sie auf OK um Ihre Auswahl zu bestätigen.

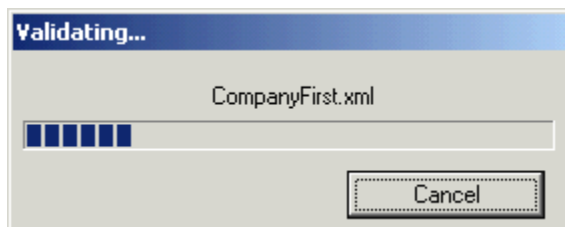


Im Projektfenster werden jetzt ausschließlich XML und XSD Dateien im Verzeichnis "Tutorial" angezeigt.

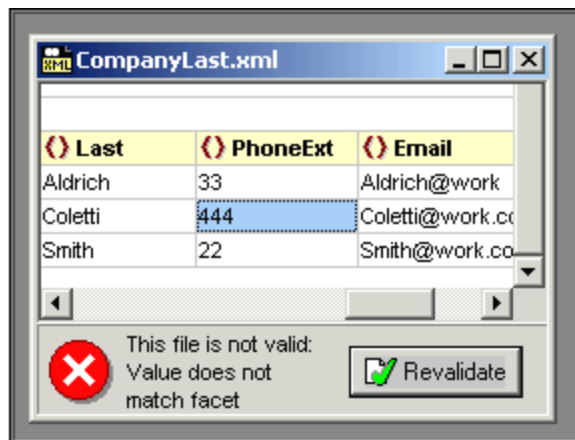
So prüfen Sie ein externes Verzeichnis auf Gültigkeit und Wohlgeformtheit:

So validieren Sie ein externes Verzeichnis und prüfen es auf Wohlgeformtheit:

1. Wählen Sie aus, welche Dateiarnten angezeigt werden sollen oder markieren Sie diese im externen Ordner.
2. Klicken Sie auf das Verzeichnis und anschließend auf das Symbol "Wohlgeformtheit prüfen" oder "Validieren" (Kürzel F7, bzw. F8)
Alle im Ordner angezeigten Dateien werden nun überprüft.



Falls eine Datei nicht wohlgeformt oder ungültig ist, wird diese im Hauptfenster geöffnet. Hier kann der Fehler korrigiert werden. In diesem Beispiel wird die Datei CompanyLast geöffnet, da der Wert PhoneExt nicht mit dem definierten Facet, der im zugrundeliegenden Schema definiert wurde, übereinstimmt. (Das Schema erlaubt nur zweistellige Ziffern für diesen Wert.)



3. Korrigieren Sie den Fehler, und starten Sie die Überprüfung erneut, um das restliche Verzeichnis zu kontrollieren.

Bitte beachten Sie: Sie können mehrere Dateien gleichzeitig im Verzeichnis markieren, wenn Sie die Strg-Taste gedrückt halten, und die gewünschten Dateien anklicken. In diesem Fall werden beim Betätigen der Tasten F7 oder F8 nur die ausgewählten Dateien geprüft.

So aktualisieren Sie den Inhalt des Projektverzeichnisses:

In lokalen oder netzwerkgebundenen Verzeichnissen können jederzeit Dateien gelöscht oder hinzugefügt werden. Um die Verzeichnis-Ansicht zu aktualisieren,

- Rechtsklicken Sie auf das externe Verzeichnis, und wählen Sie im Popup-Menü die Option **Externes Verzeichnis aktualisieren**.

Dateien oder Verzeichnisse löschen:

- Um ein Verzeichnis aus dem Projektfenster zu löschen, klicken Sie auf das **Verzeichnis** und drücken Sie die **Entf**-Taste. Das Verzeichnis wird nur aus der Projekt-Ansicht entfernt; auf der Festplatte oder im Netzwerk ist das Verzeichnis weiterhin vorhanden.
- Durch Klick auf eine **einzelne Datei** und **Drücken der Entf-Taste** wird die Datei **nicht aus dem Projektfenster gelöscht**. Sie müssen sie physisch löschen und dann den Inhalt des externen Ordners aktualisieren.

Anmerkung

Ein Projektordner in einen anderen Projektordner oder an eine andere Stelle im selben Projekt gezogen werden. Zusätzlich dazu kann ein Ordner aus dem Windows-Datei-Explorer in jeden beliebigen Projektordner gezogen werden.

9.3.14 Externes Webverzeichnis zum Projekt hinzufügen

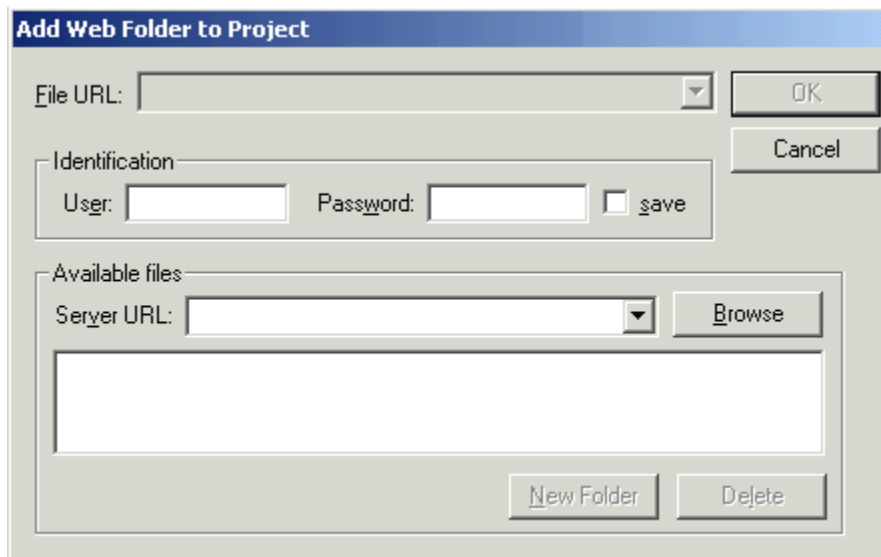
Mit diesem Befehl fügen Sie ein **externes Webverzeichnis** in das aktuelle Projekt ein.

Verwenden Sie diese Funktion, um ein neues Webverzeichnis in das aktuelle Projekt einzufügen. Sie können diesen Befehl auch über das Kontextmenü (Rechtsklick auf ein Verzeichnis im Projektfenster) aufrufen. Bitte beachten Sie, dass Dateien in externen Verzeichnissen nicht unter Versionskontrolle gestellt werden können.

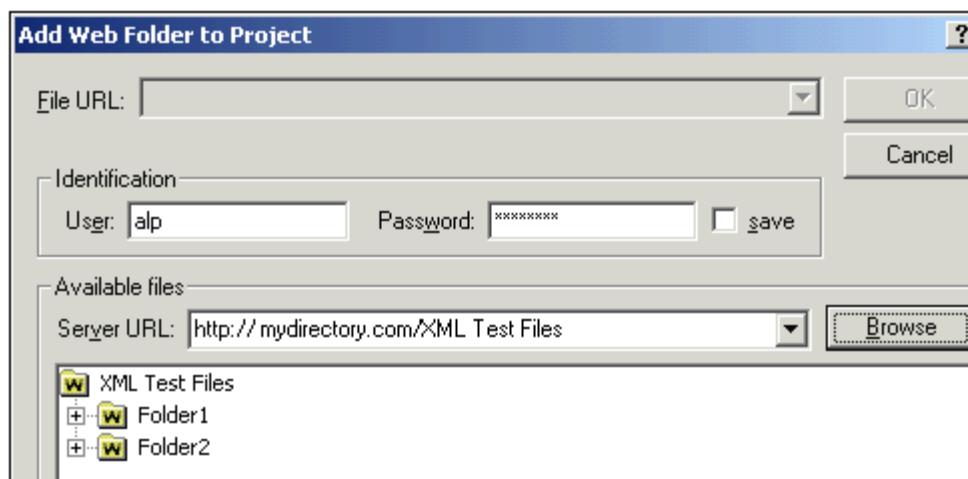
So fügen Sie ein externes Webverzeichnis in ein Projekt ein:

1. Wählen Sie die Menüoption **Projekt | Externes Webverzeichnis zum Projekt hinzufügen**.

Das Dialogfeld "Webordner in Projekt einfügen" wird angezeigt.



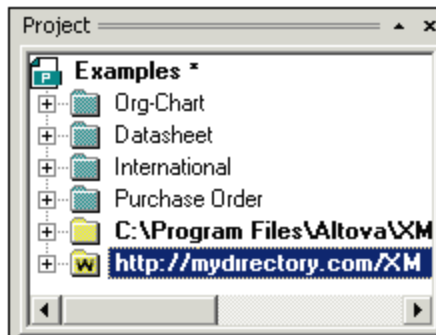
2. Klicken Sie auf das Server URL-Feld, um die URL anzugeben, und geben Sie Benutzernamen und Passwort in die entsprechenden Felder ein.
3. Klicken Sie auf "Durchsuchen" um die Verbindung mit dem Server herzustellen, und die verfügbaren Dateien zu anzeigen.



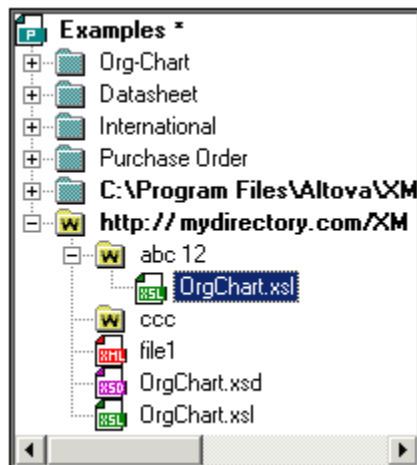
4. Klicken Sie auf das Verzeichnis, welches Sie in die Projektansicht übernehmen möchten. Die OK Schaltfläche wird erst danach aktiv. Verzeichnisname und http:

Serveradresse werden nun im Feld Datei-URL angezeigt.

5. Klicken Sie auf **OK**, um das Verzeichnis in das Projekt zu übernehmen.



6. Klicken Sie auf das Symbol "+" um den Inhalt des Verzeichnisses anzuzeigen.

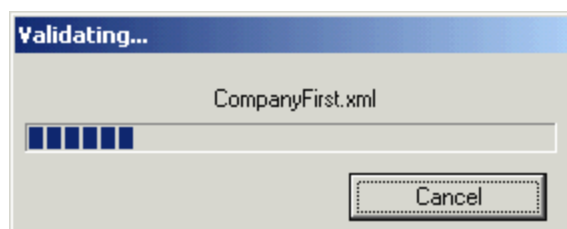


So filtern Sie den Inhalt eines Verzeichnisses:

1. **Rechtsklicken** Sie auf das Webverzeichnis, und wählen Sie im Popup-Menü die Option **Eigenschaften**.
Das Dialogfeld Eigenschaften wird geöffnet.
2. Klicken Sie auf das Feld **Erweiterungen**, und geben Sie die Erweiterungen der Dateien an, die angezeigt werden sollen. Bei mehreren Erweiterungen trennen Sie diese durch ein **Semikolon** (z.B. XML und Schema XSDs).
3. Klicken Sie auf OK um Ihre Auswahl zu bestätigen.
Im Projektfenster werden jetzt ausschließlich die XML und XSD Dateien des Webverzeichnisses angezeigt.

So prüfen Sie ein Verzeichnis auf Gültigkeit und Wohlgeformtheit:

1. Klicken Sie auf das Verzeichnis und anschließend auf das Symbol "Wohlgeformtheit prüfen" oder "Validieren" (Kürzel F7, bzw. F8)
Alle angezeigten Dateien im Verzeichnis werden nun überprüft.



Falls eine Datei nicht wohlgeformt oder ungültig ist, wird diese im Hauptfenster geöffnet. Hier können Sie den Fehler korrigieren.

2. Nach erfolgter Korrektur, starten Sie die Prüfung erneut, um das restliche Verzeichnis zu kontrollieren.

Bitte beachten Sie:

Sie können mehrere Dateien gleichzeitig im Verzeichnis markieren, wenn Sie die Strg-Taste gedrückt halten, und die gewünschten Dateien anklicken. In diesem Fall werden beim Betätigen der Funktionstasten F7 oder F8 nur die markierten Dateien überprüft.

So aktualisieren Sie den Inhalt des Projektverzeichnisses:

Sie können jederzeit Dateien aus dem Webverzeichnis hinzufügen oder löschen. Um die Verzeichnis-Ansicht zu aktualisieren,

- Rechtsklicken Sie auf das externe Verzeichnis, und wählen Sie im Popup-Menü die Option **Externes Verzeichnis aktualisieren**.

Dateien oder Verzeichnisse löschen:

- Um ein Verzeichnis aus dem Projektfenster zu löschen, klicken Sie auf das **Verzeichnis** und drücken Sie die **Entf**-Taste. Das Verzeichnis wird nur aus der Projekt-Ansicht entfernt; auf der Festplatte oder im Netzwerk ist das Verzeichnis weiterhin vorhanden.
- Durch Klick auf eine **einzelne Datei** und Betätigen der **Entf**-Taste wird die Datei **nicht aus dem Projektfenster gelöscht**. Sie müssen sie physisch löschen, und dann den Inhalt des externen Ordners aktualisieren.

9.3.15 Projekteigenschaften



Mit diesem Befehl können Sie wichtige Einstellungen für spezifische Ordner Ihres Projekts definieren.

So definieren Sie Projekteigenschaften für ein Verzeichnis:

1. Rechtsklicken Sie auf das Verzeichnis, für welches Sie Einstellungen vornehmen möchten, und
2. Wählen Sie den Befehl **Eigenschaften...** aus dem Kontextmenü.

Bitte beachten Sie:

Falls die Projektdatei unter Versionskontrolle steht, werden Sie gefragt, ob die Projektdatei (*.spp) ausgecheckt werden soll. Nach dem Ändern der Einstellungen klicken Sie auf OK um diese zu speichern.

Die Zuweisungen in den Feldern "XSL-Vorlage" bzw. "XML-Vorlage" haben Vorrang vor anderen lokalen Zuweisungen innerhalb einer XML-Datei. So wird z.B. die Datei OrgChart.xml ("XSL-Vorlage") immer dann verwendet, wenn XML-Dateien im **XML-Dateiverzeichnis** transformiert werden. Die hier vorgenommene Zuweisung hat auch Vorrang vor anderen Zuweisungen, die für übergeordnete Ordner vorgenommen wurden.

Dateierweiterungen

Die Dateierweiterungen helfen beim Hinzufügen neuer Dateien zum Projekt (im Gegensatz zum Einordnen in einen bestimmten Ordner) bei der automatischen Einordnung der Dateien in

Ordner eines Projekts.

Validierung

Definieren Sie das DTD- oder Schemadokument, welches bei der [Validierung](#) aller Dateien des aktuellen Verzeichnisses verwendet werden soll (in diesem Beispiel die Hauptseiten).

XSL Transformation von XML-Dateien

Definieren Sie das XSL Stylesheet, welches zur [XSL Transformation](#) aller Dateien eines Verzeichnisses verwendet werden soll.

Wenn Sie selbst XSL Stylesheets erstellen, können Sie ein XML-Beispieldokument zuweisen, das als Vorschau für das XSL Stylesheet bei der XML-Transformation dient. Sie können dies anstelle des XML Instanzdokuments verwenden.

XSL:FO Transformation von XML-Dateien

Sie können das XSL Stylesheet mit XSL:FO Markup definieren, das für eine [XSL:FO Transformation](#) aller Dateien eines Verzeichnisses verwendet werden soll.

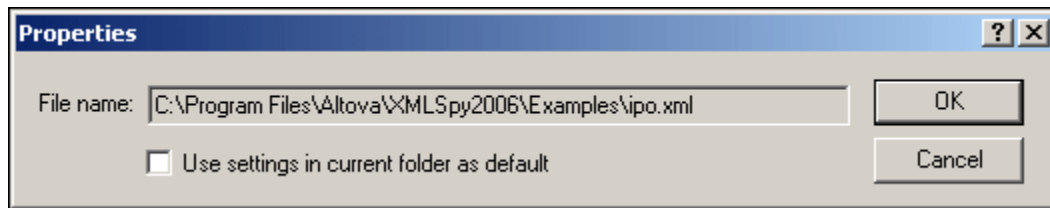
Zieldateien bei XSL Transformation

Für Batchtransformationen in XSL können Sie das Zielverzeichnis angeben, in dem die transformierten Dateien abgelegt werden sollen.

Wenn Sie eine Datei oder URL in mehreren Verzeichnissen Ihres Projekts abgelegt haben, können Sie über das Dialogfeld "Eigenschaften" das Standardverzeichnis definieren, mit dessen Einstellungen die Validierung oder Transformation im non-batch Modus erfolgen soll. Aktivieren Sie dazu das Kontrollkästchen **Einstellungen des aktuellen Ordners als Standard verwenden** (siehe Screenshot).

So rufen Sie das Dialogfeld "Eigenschaften" auf und aktivieren das Kontrollkästchen:

1. Kopieren Sie eine XML-Datei in einem Projekt in einen anderen Ordner.
2. Rechtsklicken Sie im Projektfenster auf die kopierte Datei und wählen Sie im Kontextmenü den Befehl **Eigenschaften**.



Authentic

Die Option "Konfig. verwenden" ermöglicht es Ihnen, das StyleVision Power Stylesheet (SPS Datei) im aktuellen Verzeichnis auszuwählen, welche beim Editieren von XML Dateien in der Authentic-Ansicht zugrunde gelegt werden soll. Nachdem Sie die Schema-, SPS- und XML-Dateien miteinander verknüpft und in ein Projekt eingegeben haben, kann ein Verschieben dieser Dateien in einen anderen Ordner Fehler bei den Verknüpfungen verursachen.

Um solche Fehler zu vermeiden, sollten Sie am besten einen endgültigen Speicherort für Ihre Schema-, SPS- und XML-Dateien festlegen, bevor Sie sie miteinander verknüpfen und einem Projekt zuweisen. Weitere Informationen zur Verwendung der Authentic-Ansicht finden Sie im Handbuch zur Authentic-Ansicht.

9.3.16 Zuletzt verwendete Projekte

Mit dieser Funktion werden die neun zuletzt verwendeten Projekte angezeigt um einen schnelleren Zugriff auf diese zu erhalten.

Beachten Sie bitte, dass automatisch das [zuletzt verwendete Projekt](#) beim Starten von öffnen kann. (**Extras | Optionen**| Register **Datei**, Projekt | Letztes Projekt beim Programmstart öffnen).

9.4 Menü "XML"

Das XML-Menü enthält alle nötigen Befehle um mit XML-Dokumenten zu arbeiten.

9.4.1 Einfügen

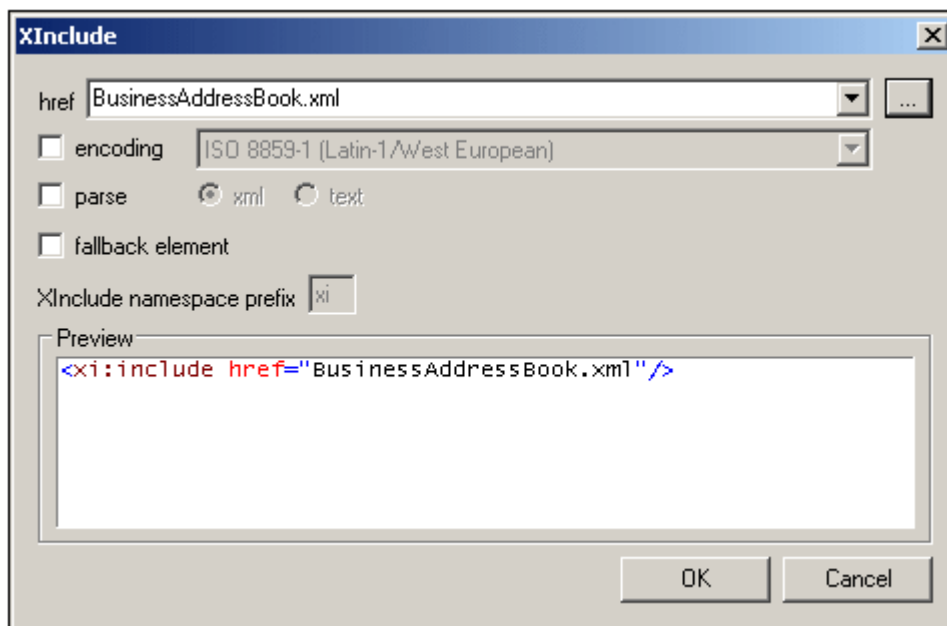
Der Befehl **Einfügen** ist zwar in allen Menüs aktiv, kann aber nur in der Grid-Ansicht verwendet werden. Er enthält ein Untermenü (*siehe Screenshot*), über das Sie folgende Möglichkeiten haben:

- die XML-Deklaration und Node-Typen (Attribut, Element, Text, CDATA, Comment, Processing Instruction) in XML-Dokumente einfügen,
- DOCTYPE-Deklarationen und externe DTD-Deklarationen in XML-Dokumente einfügen und
- DTD-Deklarationen (ELEMENT, ATTLIST, ENTITY und NOTATION) in DTD-Dokumente und interne DTD-Deklarationen von XML-Dokumenten einfügen.

XInclude einfügen



Der Befehl **XML | Einfügen | XInclude** steht nur in der Grid-Ansicht zur Verfügung und fügt ein neues XInclude-Element vor dem ausgewählten Objekt ein. Wenn gerade ein Attribut ausgewählt ist, wird das XInclude-Element nach dem Attribut und vor dem ersten Child-Element des Parent-Elements des Attributs eingefügt. Bei Auswahl dieses Befehls wird das Dialogfeld "XInclude" aufgerufen (*Abbildung unten*).



Die zu inkludierende XML-Datei wird in das Textfeld `href` eingegeben (alternativ dazu können Sie die Datei auch über die Schaltfläche "Durchsuchen" rechts vom Textfeld eingeben). Der Dateiname wird als Wert des `href`-Attributs eingegeben. Auch die Attribute `encoding` und `parse` von `include` und das Child-Element `fallback` von `include` können über das Dialogfeld eingegeben werden. Aktivieren Sie dazu zuerst das entsprechende Kontrollkästchen und wählen Sie anschließend die benötigten Werte aus, bzw. geben Sie sie ein.

Hier sehen Sie ein Beispiel für ein XML-Dokument, in dem zum Inkludieren zweier XML-

Dokumente XInclude verwendet wird:

```
<?xml version="1.0" encoding="UTF-16"?>
<AddressBook xsi:schemaLocation="http://www.altova.com/sv/myaddresses
AddressBook.xsd"
    xmlns="http://www.altova.com/stylevision/tutorials/myaddresses"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xi="http://www.w3.org/2001/XInclude">
    <xi:include href="BusinessAddressBook.xml"/>
    <xi:include href="PersonalAddressBook.xml"/>
</AddressBook>
```

Beim Parsen dieses Dokuments werden die beiden XInclude-Elemente durch die in den entsprechenden href Attributen definierten Dateien ersetzt.

xml: base

Wenn der XML-Validator von XMLSpy ein XML-Dokument liest und im XInclude-Namespaces auf das Element `include` trifft (im Folgenden `xi:include` genannt), ersetzt er dieses Element (`xi:include`) durch das im href-Attribut des `xi:include`-Elements definierte XML-Dokument. Das Dokument-Element (Root-Element) des inkludierten XML-Dokuments (bzw. das durch einen XPointer gekennzeichnete Element) wird mit einem Attribut von `xml:base` inkludiert, um die Basis-URLs des inkludierten Elements beizubehalten. Wenn das XML-Ergebnisdokument (das das/die inkludierte(n) XML-Dokument(e) oder Baumstruktur-Fragment(e) enthält) dem Schema entsprechend gültig sein muss, so muss das Dokument-Element des inkludierten Dokuments (oder das Element auf oberster Ebene des Baumstrukturfragments) mit einem Content Model erstellt werden, das ein Attribut `xml:base` zulässt. Wenn das `xml:base` Attribut laut Schema in diesem Element nicht zulässig ist, ist das Ergebnisdokument ungültig. Eine Anleitung zum Definieren eines `xml:base` Attributs im Content Model eines Elements in der Schema/WSDL-Ansicht von XMLSpy finden Sie in dieser Dokumentation im Abschnitt `xml:base`, `xml:id`, `xml:lang`, `xml:space` des Abschnitts Schema/WSDL-Ansicht.

XPointers

XMLSpy unterstützt XPointers in XInclude. Die relevanten W3C Recommendations dazu sind die [XPointer Framework](#) und die [XPointer element\(\) Scheme](#) Recommendation. Durch die Verwendung eines XPointers in einem XInclude-Element kann ein bestimmter Teil des XML-Dokuments anstatt des gesamten XML-Dokuments inkludiert werden. XPointers werden folgendermaßen in einem XInclude-Element verwendet:

```
<xi:include href="PersonalAddressBook.xml" xpointer="element( usa)"/>
<xi:include href="BusinessAddressBook.xml" xpointer="element(/1/1)"/>
<xi:include href="BobsAddressBook.xml" xpointer="element( usa/3/1)"/>
<xi:include href="PatsAddressBook.xml" xpointer="
element( usa) element(/1/1)"/>
```

Im `element()` Schema von XPointer kann ein NCName oder eine von Ganzzahlen geleitete Child-Sequenz verwendet werden.

- Im ersten oben angeführten `xi:include` Element verwendet das `xpointer` Attribut das Elementschema mit dem NCName `usa`. Laut XPointer Framework identifiziert dieser NCName das Element, das die ID `usa` hat.
- Im zweiten oben angeführten `xi:include` Element identifiziert das `xpointer` Attribut mit dem Wert `element(/1/1)` im ersten Schritt das erste Child-Element der Dokument-Root (wobei es sich um das Dokument (oder Root)-Element handelt, wenn das Dokument wohlgeformt ist. Im zweiten Schritt wird das erste Child-Element des im vorherigen Schritt gesuchten Elements gesucht; in unserem Beispiel wäre das das erste Child-Element des Dokument-Elements.
- Das `xpointer` Attribut des dritten oben angeführten `xi:include` Elements verwendet eine Kombination von NCName und Child-Sequenz. Dieses XPointer-Element sucht

- das erste Child-Element des dritten Child-Elements des Elements mit der ID `usa`.
- Wenn Sie nicht sicher sind, ob Ihr erstes XPointer-Element funktioniert, können Sie zur Sicherheit ein zweites verwenden, wie im vierten oben angeführten `xi:include` Element aufgelistet: `xpointer="element(usa) element(/1/1) "`. Wenn hier kein Element mit der ID `usa` gefunden wird, definiert der Backup-XPointer, dass das erste Child-Element des Dokument-Elements ausgewählt werden soll. Auch zusätzliche Backups sind zulässig. Einzelne XPointer-Elemente müssen nicht getrennt werden, können aber auch durch ein Leerzeichen getrennt werden wie z.B. in `xpointer="element(usa) element(addresses/1) element(/1/1) "`.

Anmerkung: Der Namespace Binding-Kontext wird im `element()` Schema nicht verwendet, das das `element()` Schema qualified Names nicht unterstützt.

9.4.2 Anhängen

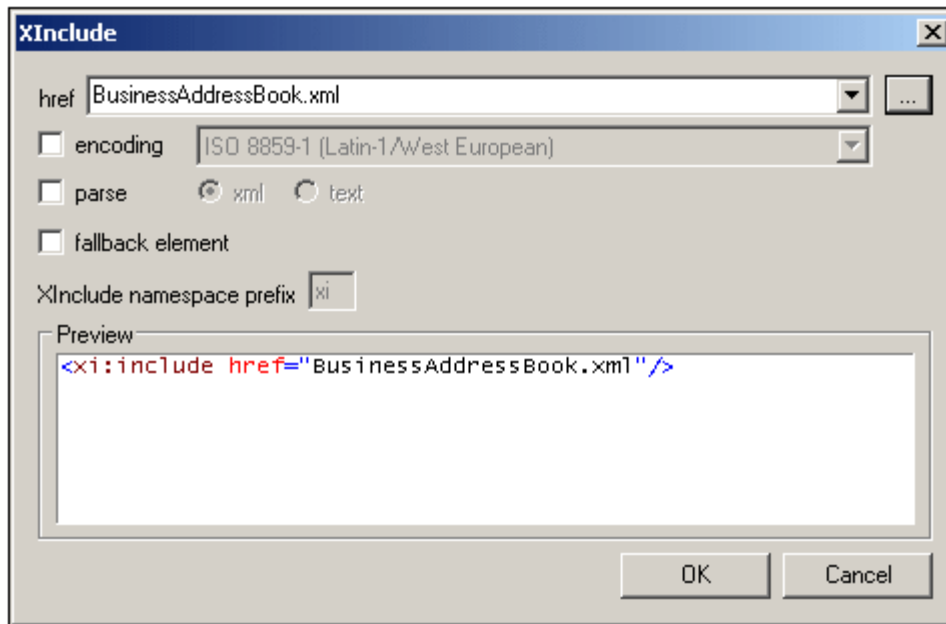
Der Befehl **Anhängen** ist zwar in allen Menüs aktiv, kann aber nur in der Grid-Ansicht verwendet werden. Er enthält ein Untermenü (*siehe Screenshot*), über das Sie folgende Möglichkeiten haben:

- die XML-Deklaration und Node-Typen (Attribut, Element, Text, CDATA, Comment, Processing Instruction) in XML-Dokumenten anhängen,
- DOCTYPE-Deklarationen und externe DTD-Deklarationen in XML-Dokumenten anhängen und
- DTD-Deklarationen (ELEMENT, ATTLIST, ENTITY und NOTATION) in DTD-Dokumenten und internen DTD-Deklarationen von XML-Dokumenten anhängen.

XInclude anhängen



Der Befehl **XML | Anhängen | XInclude** steht nur in der Grid-Ansicht zur Verfügung und dient zum Anhängen eines XInclude-Elements hinter dem letzten gleichrangigen Element eines ausgewählten Node (nicht Attribut-Node). Wenn es sich bei der aktuellen Auswahl um ein Attribut handelt, wird das XInclude-Element hinter dem letzten Child-Element des Elements angehängt, das dem ausgewählten Attribut übergeordnet ist. Bei Auswahl dieses Befehls wird das Dialogfeld XInclude (*Abbildung unten*) angezeigt.



Die zu inkludierende XML-Datei wird in das Textfeld `href` eingegeben (alternativ dazu können Sie die Datei auch über die Schaltfläche "Durchsuchen" rechts vom Textfeld eingeben). Der Dateiname wird als Wert des `href`-Attributs eingegeben. Auch die Attribute `encoding` und `parse` von `include` und das Child-Element `fallback` von `include` können über das Dialogfeld eingegeben werden. Aktivieren Sie dazu zuerst das entsprechende Kontrollkästchen und wählen Sie anschließend die benötigten Werte aus, bzw. geben Sie sie ein.

Hier sehen Sie ein Beispiel für ein XML-Dokument, in dem zum Inkludieren zweier XML-Dokumente XInclude verwendet wird:

```
<?xml version="1.0" encoding="UTF-16"?>
<AddressBook xsi:schemaLocation="http://www.altova.com/sv/myaddresses
AddressBook.xsd"
  xmlns="http://www.altova.com/stylevision/tutorials/myaddresses"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <xi:include href="BusinessAddressBook.xml"/>
  <xi:include href="PersonalAddressBook.xml"/>
</AddressBook>
```

Beim Parsen dieses Dokuments werden die beiden XInclude-Elemente durch die in den entsprechenden href Attributen definierten Dateien ersetzt.

xml: base

Wenn der XML-Validator von XMLSpy ein XML-Dokument liest und im XInclude-Namespace auf das Element `include` trifft (im Folgenden `xi:include` genannt), ersetzt er dieses Element (`xi:include`) durch das im href-Attribut des `xi:include`-Elements definierte XML-Dokument. Das Dokument-Element (Root-Element) des inkludierten XML-Dokuments (bzw. das durch einen XPointer gekennzeichnete Element) wird mit einem Attribut von `xml:base` inkludiert, um die Basis-URLs des inkludierten Elements beizubehalten. Wenn das XML-Ergebnisdokument (das das/die inkludierte(n) XML-Dokument(e) oder Baumstruktur-Fragment(e) enthält) dem Schema entsprechend gültig sein muss, so muss das Dokument-Element des inkludierten Dokuments (oder das Element auf oberster Ebene des Baumstrukturfragments) mit einem Content Model erstellt werden, das ein Attribut `xml:base` zulässt. Wenn das `xml:base` Attribut laut Schema in diesem Element nicht zulässig ist, ist das Ergebnisdokument ungültig. Eine Anleitung zum Definieren eines `xml:base` Attributs im Content Model eines Elements in der Schema/WSDL-Ansicht von XMLSpy finden Sie in dieser Dokumentation im Abschnitt `xml:base`, `xml:id`, `xml:lang`, `xml:space` des Abschnitts Schema/WSDL-Ansicht.

XPointers

XMLSpy unterstützt XPointers in XInclude. Die relevanten W3C Recommendations dazu sind die [XPointer Framework](#) und die [XPointer element\(\) Scheme](#) Recommendation. Durch die Verwendung eines XPointers in einem XInclude-Element kann ein bestimmter Teil des XML-Dokuments anstatt des gesamten XML-Dokuments inkludiert werden. XPointers werden folgendermaßen in einem XInclude-Element verwendet:

```
<xi:include href="PersonalAddressBook.xml" xpointer="element( usa)"/>
<xi:include href="BusinessAddressBook.xml" xpointer="element(/1/1)"/>
<xi:include href="BobsAddressBook.xml" xpointer="element( usa/3/1)"/>
<xi:include href="PatsAddressBook.xml" xpointer="
element( usa) element(/1/1)"/>
```

Im `element()` Schema von XPointer kann ein NCName oder eine von Ganzzahlen geleitete Child-Sequenz verwendet werden.

- Im ersten oben angeführten `xi:include` Element verwendet das `xpointer` Attribut das Elementschema mit dem NCName `usa`. Laut XPointer Framework identifiziert dieser NCName das Element, das die ID `usa` hat.
- Im zweiten oben angeführten `xi:include` Element identifiziert das `xpointer` Attribut mit dem Wert `element(/1/1)` im ersten Schritt das erste Child-Element der Dokument-Root (wobei es sich um das Dokument (oder Root)-Element handelt, wenn das Dokument wohlgeformt ist. Im zweiten Schritt wird das erste Child-Element des im vorherigen Schritt gesuchten Elements gesucht; in unserem Beispiel wäre das das erste Child-Element des Dokument-Elements.
- Das `xpointer` Attribut des dritten oben angeführten `xi:include` Elements verwendet eine Kombination von NCName und Child-Sequenz. Dieses XPointer-Element sucht das erste Child-Element des dritten Child-Elements des Elements mit der ID `usa`.

- Wenn Sie nicht sicher sind, ob Ihr erstes XPointer-Element funktioniert, können Sie zur Sicherheit ein zweites verwenden, wie im vierten oben angeführten `xi:include` Element aufgelistet: `xpointer="element( usa) element( /1/1) "`. Wenn hier kein Element mit der ID `usa` gefunden wird, definiert der Backup-XPointer, dass das erste Child-Element des Dokument-Elements ausgewählt werden soll. Auch zusätzliche Backups sind zulässig. Einzelne XPointer-Elemente müssen nicht getrennt werden, können aber auch durch ein Leerzeichen getrennt werden wie z.B. in `xpointer="element( usa) element( addresses/1) element( /1/1) "`.

Anmerkung: Der Namespace Binding-Kontext wird im `element()` Schema nicht verwendet, das das `element()` Schema qualified Names nicht unterstützt.

9.4.3 Child hinzufügen

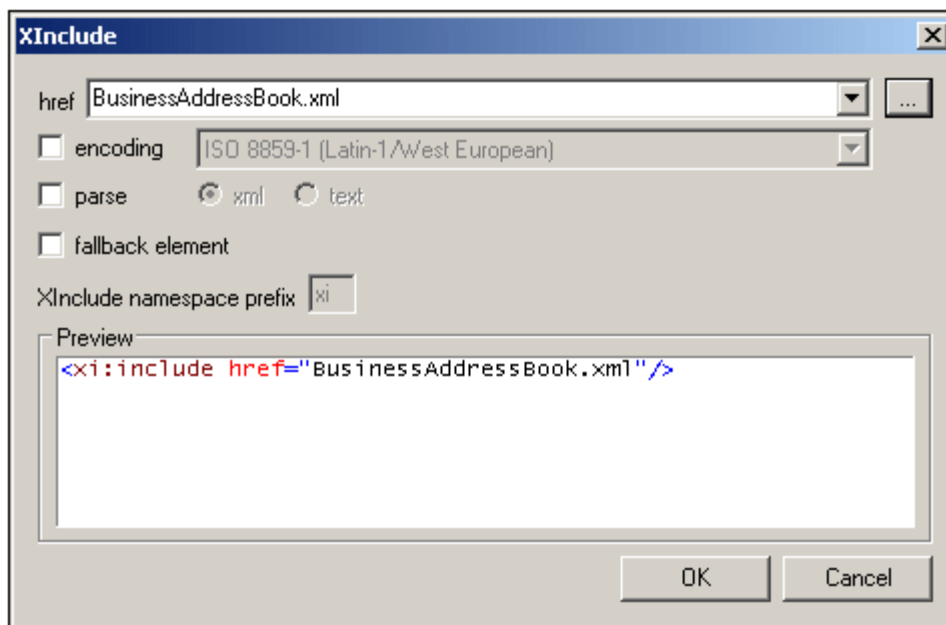
Der Befehl **Child hinzufügen** ist zwar in allen Menüs aktiv, kann aber nur in der Grid-Ansicht verwendet werden. Der Befehl enthält ein Untermenü (*siehe Screenshot*), über das Sie die folgenden Child-Datenelemente zum aktuell ausgewählten Element hinzufügen können:

- die XML-Deklaration und Node-Typen (Attribut, Element, Text, CDATA, Comment, Processing Instruction) in XML-Dokumenten,
- DOCTYPE-Deklarationen und externe DTD-Deklarationen in XML-Dokumenten und
- DTD-Deklarationen (ELEMENT, ATTLIST, ENTITY und NOTATION) in DTD-Dokumenten und internen DTD-Deklarationen von XML-Dokumenten.

Child XInclude anhängen



Der Befehl **XML | Child hinzufügen | XInclude** steht nur in der Grid-Ansicht zur Verfügung und fügt ein XInclude-Element als Child des ausgewählten Datenelements ein. Bei Auswahl dieses Befehls wird das Dialogfeld "XInclude" (*Abbildung unten*) aufgerufen.



Die zu inkludierende XML-Datei wird in das Textfeld `href` eingegeben (alternativ dazu können Sie die Datei auch über die Schaltfläche "Durchsuchen" rechts vom Textfeld eingeben). Der Dateiname wird als Wert des `href`-Attributs eingegeben. Auch die Attribute `encoding` und `parse` von `include` und das Child-Element `fallback` von `include` können über das Dialogfeld eingegeben werden. Aktivieren Sie dazu zuerst das entsprechende Kontrollkästchen und wählen Sie anschließend die benötigten Werte aus, bzw. geben Sie sie ein.

Hier sehen Sie ein Beispiel für ein XML-Dokument, in dem zum Inkludieren zweier XML-Dokumente XInclude verwendet wird:

```
<?xml version="1.0" encoding="UTF-16"?>
<AddressBook xsi:schemaLocation="http://www.altova.com/sv/myaddresses
AddressBook.xsd"
    xmlns="http://www.altova.com/stylevision/tutorials/myaddresses"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```

xmlns:xi="http://www.w3.org/2001/XInclude">
<xi:include href="BusinessAddressBook.xml"/>
<xi:include href="PersonalAddressBook.xml"/>
</AddressBook>

```

Beim Parsen dieses Dokuments werden die beiden XInclude-Elemente durch die in den entsprechenden href Attributen definierten Dateien ersetzt.

xml: base

Wenn der XML-Validator von XMLSpy ein XML-Dokument liest und im XInclude-Namespace auf das Element `include` trifft (im Folgenden `xi:include` genannt), ersetzt er dieses Element (`xi:include`) durch das im href-Attribut des `xi:include`-Elements definierte XML-Dokument. Das Dokument-Element (Root-Element) des inkludierten XML-Dokuments (bzw. das durch einen XPointer gekennzeichnete Element) wird mit einem Attribut von `xml:base` inkludiert, um die Basis-URLs des inkludierten Elements beizubehalten. Wenn das XML-Ergebnisdokument (das das/die inkludierte(n) XML-Dokument(e) oder Baumstruktur-Fragment(e) enthält) dem Schema entsprechend gültig sein muss, so muss das Dokument-Element des inkludierten Dokuments (oder das Element auf oberster Ebene des Baumstrukturfragments) mit einem Content Model erstellt werden, das ein Attribut `xml:base` zulässt. Wenn das `xml:base` Attribut laut Schema in diesem Element nicht zulässig ist, ist das Ergebnisdokument ungültig. Eine Anleitung zum Definieren eines `xml:base` Attributs im Content Model eines Elements in der Schema/WSDL-Ansicht von XMLSpy finden Sie in dieser Dokumentation im Abschnitt `xml:base`, `xml:id`, `xml:lang`, `xml:space` des Abschnitts Schema/WSDL-Ansicht.

XPointers

XMLSpy unterstützt XPointers in XInclude. Die relevanten W3C Recommendations dazu sind die [XPointer Framework](#) und die [XPointer element\(\) Scheme](#) Recommendation. Durch die Verwendung eines XPointers in einem XInclude-Element kann ein bestimmter Teil des XML-Dokuments anstatt des gesamten XML-Dokuments inkludiert werden. XPointers werden folgendermaßen in einem XInclude-Element verwendet:

```

<xi:include href="PersonalAddressBook.xml" xpointer="element( usa)"/>
<xi:include href="BusinessAddressBook.xml" xpointer="element( /1/1)"/>
<xi:include href="BobsAddressBook.xml" xpointer="element( usa/3/1)"/>
<xi:include href="PatsAddressBook.xml" xpointer="
element( usa) element( /1/1)"/>

```

Im `element()` Schema von XPointer kann ein NCName oder eine von Ganzzahlen geleitete Child-Sequenz verwendet werden.

- Im ersten oben angeführten `xi:include` Element verwendet das `xpointer` Attribut das Elementschema mit dem NCName `usa`. Laut XPointer Framework identifiziert dieser NCName das Element, das die ID `usa` hat.
- Im zweiten oben angeführten `xi:include` Element identifiziert das `xpointer` Attribut mit dem Wert `element( /1/1)` im ersten Schritt das erste Child-Element der Dokument-Root (wobei es sich um das Dokument (oder Root)-Element handelt, wenn das Dokument wohlgeformt ist. Im zweiten Schritt wird das erste Child-Element des im vorherigen Schritt gesuchten Elements gesucht; in unserem Beispiel wäre das das erste Child-Element des Dokument-Elements.
- Das `xpointer` Attribut des dritten oben angeführten `xi:include` Elements verwendet eine Kombination von NCName und Child-Sequenz. Dieses XPointer-Element sucht das erste Child-Element des dritten Child-Elements des Elements mit der ID `usa`.
- Wenn Sie nicht sicher sind, ob Ihr erstes XPointer-Element funktioniert, können Sie zur Sicherheit ein zweites verwenden, wie im vierten oben angeführten `xi:include` Element aufgelistet: `xpointer="element( usa) element( /1/1) "`. Wenn hier kein Element mit der ID `usa` gefunden wird, definiert der Backup-XPointer, dass das erste

Child-Element des Dokument-Elements ausgewählt werden soll. Auch zusätzliche Backups sind zulässig. Einzelne XPointer-Elemente müssen nicht getrennt werden, können aber auch durch ein Leerzeichen getrennt werden wie z.B. in
`xpointer="element( usa) element( addresses/1) element( /1/1) "`.

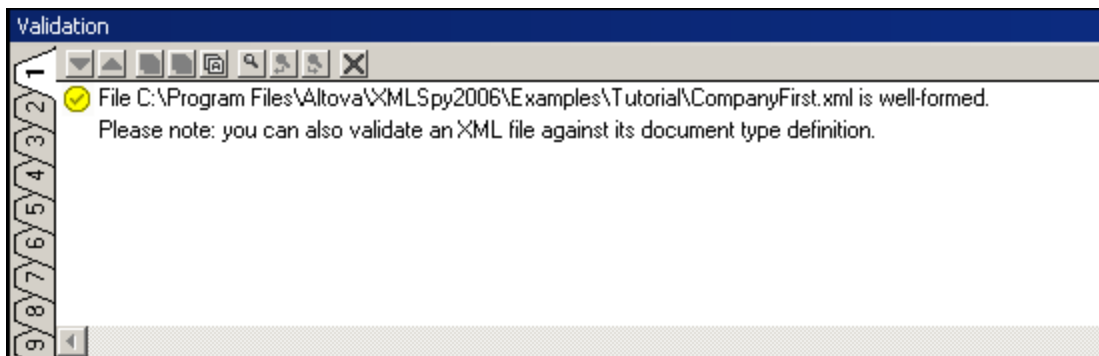
Anmerkung: Der Namespace Binding-Kontext wird im `element()` Schema nicht verwendet, das das `element()` Schema qualified Names nicht unterstützt.

9.4.4 Wohlgeformtheit prüfen

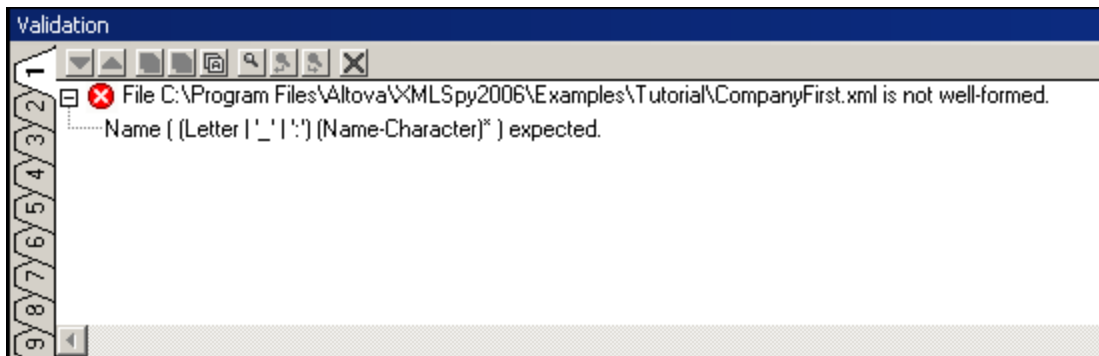
**F7**

Mit diesem Befehl überprüfen Sie das Dokument auf Wohlgeformtheit nach der Definition der XML 1.0 Spezifikation. Jedes XML-Dokument muss wohlgeformt sein; deswegen überprüft ein Dokument automatisch auf Wohlgeformtheit, sobald es geöffnet oder gespeichert wird. Sie können das Dokument auch während der Bearbeitung jederzeit auf Wohlgeformtheit prüfen.

Bei erfolgreicher Wohlgeformtheitsprüfung wird am unteren Rand des Hauptfensters eine Meldung angezeigt:



Wenn bei der Wohlgeformtheitsprüfung ein Fehler entdeckt wurde, wird das Fehler verursachende Objekt markiert und eine entsprechende Fehlermeldung am unteren Rand des Hauptfensters angezeigt.



Beachten Sie, dass das Validierungsfenster neun Ausgaberegister enthält. Das Ergebnis der Validierung wird immer auf dem aktuellen Register angezeigt. Sie können daher die Wohlgeformtheit für eine Schemadatei auf Register 1 überprüfen und das Ergebnis aufbewahren, indem Sie zu Register 2 wechseln, bevor Sie das nächste Schema-Dokument validieren (andernfalls wird Register 1 durch das Validierungsergebnis überschrieben.)

Eigentlich ist es nicht erlaubt, ein schlecht geformtes XML-Dokument zu speichern; enthält die Funktion "Save anyway" (trotzdem speichern), die angezeigt wird, wenn die automatische Wohlgeformtheitsprüfung beim Speichern des Dokument einen Fehler entdeckt. Dies ist nützlich, wenn sie Ihre Arbeit kurzzeitig unterbrechen müssen, und den Fehler nicht sofort beheben können.

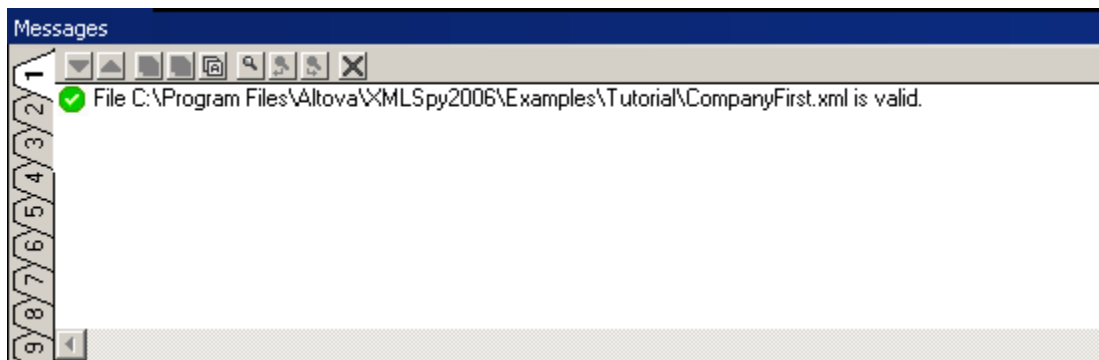
9.4.5 Validieren



F8

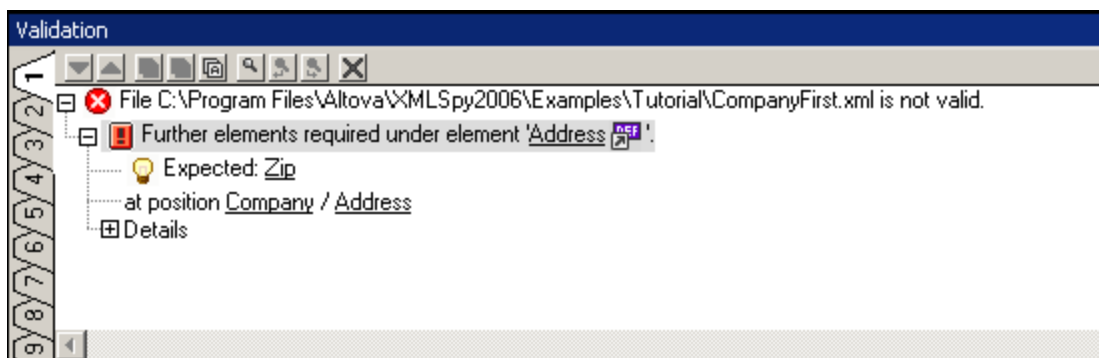
Mit diesem Befehl können Sie ein XML-Dokument gegen DTDs, XML-Schemas und andere Schemas validieren. Sie können einstellen, dass ein Dokument beim Öffnen oder Speichern automatisch validiert wird (**Extras | Optionen | Datei**). Der Befehl **Validieren** führt vor Überprüfung der Gültigkeit gleichzeitig eine Wohlgeformtheitsprüfung durch, sodass diese vorher nicht separat überprüft werden muss.

Bei erfolgreicher Validierung wird am unteren Rand des Hauptfensters folgende Meldung angezeigt:



Wenn bei der Validierung ein Fehler entdeckt wurde, wird das Fehler verursachende Objekt markiert, und eine entsprechende Fehlermeldung wird angezeigt:

Bitte beachten Sie: Wenn Sie auf die Links in der Fehlermeldung klicken, können Sie zu der Stelle in der XML-Datei springen, an der der Fehler gefunden wurde.



Validieren von XML-Dokumenten

Um eine XML-Datei zu validieren, machen Sie das XML-Dokument im Hauptfenster zum aktiven und klicken Sie auf **XML | Validieren** oder drücken Sie **F8**. Das XML-Dokument wird daraufhin gegen das in der XML-Datei referenzierte Schema validiert. Existiert keine Referenz, wird am unteren Rand des Hauptfensters eine Fehlermeldung angezeigt. Solange das XML-Dokument geöffnet ist, bleibt das Schema im Speicher (siehe im DTD/Schema-Menü).

Automatisches Validieren mit Altova XML 2009

Altova XML ist eine kostenlose Applikation, die den Altova XML-Validator sowie einen XSLT 1.0-, XSLT 2.0- und XQuery 1.0-Prozessor enthält. Das Tool kann über die Befehlszeile, über eine COM-Schnittstelle, in Java-Programmen und in .NET-Applikationen zur Validierung von XML-Dokumenten, zum Transformieren von XML-Dokumenten unter Verwendung von XSLT

1.0 und 2.0 Stylesheets sowie zum Ausführen von XQuery-Dokumenten verwendet werden.

Validierungen können daher mittels Altova XML automatisiert werden. So können Sie z.B. eine Batch-Datei erstellen, die Altova XML aufruft, um damit eine Reihe von XML-Dokumenten zu validieren und die Ausgabe an eine Textdatei zu senden. Nähere Informationen dazu finden Sie in der [Altova XML Dokumentation](#).

9.4.6 Eingabehilfen aktualisieren



Mit diesem Befehl aktualisieren Sie die Eingabehilfen-Fenster. Dies erfolgt durch erneutes Laden der zugrunde liegenden DTD / des zugrunde liegenden Schemas. Sollten Sie das Schema oder die DTD, auf dem/auf der ein offenes XML-Dokument basiert, verändert haben, empfiehlt es sich, das Schema oder die DTD erneut zu laden, damit die intelligenten Editierinformationen im XML-Dokument aktualisiert werden.

9.5 Menü "XSL/XQuery"

Mit Hilfe der XSL Transformationssprache können Sie festlegen, wie ein XML-Dokument in andere XML-Dokumente oder Textdateien konvertiert werden soll. Eine Art von XML-Dokument, die mittels eines XSLT-Dokuments generiert wird, ist ein FO-Dokument, das anschließend weiter in ein PDF-Ausgabedokument konvertiert wird.

enthält integrierte XSLT-Prozessoren (für XSLT 1.0 und XSLT 2.0) und kann mit einem FO-Prozessor auf Ihrem System verknüpft werden, um XML-Dateien zu transformieren und verschiedene Arten von Ausgabedokumenten zu generieren. Der Pfad zum FO-Prozessor muss im Dialogfeld "Optionen" auf dem Register XSL ([Extras | Optionen](#)) angegeben werden, um direkt innerhalb der Benutzeroberfläche verwendet werden zu können.

Die Befehle für die oben beschriebenen Transformationen finden Sie im Menü **XSL/XQuery**. Zusätzlich dazu enthält dieses Menü auch Befehle für die Arbeit mit dem Altova XSLT/XQuery Debugger.

9.5.1 XSL-Transformation

**F10**

Mit diesem Befehl führen Sie mittels eines zugewiesenen XML-Dokuments eine **XSL-Transformation** durch. Die Transformation kann entweder mit Hilfe des entsprechenden Altova XSLT-Prozessors (Altova XSLT 1.0-Prozessor für XSLT 1.0 Stylesheets; Altova XSLT 2.0-Prozessor für XSLT 2.0 Stylesheets), mit Hilfe des MSXML-Moduls von Microsoft oder eines externen XSLT-Prozessors durchgeführt werden. Der in Verbindung mit diesem Befehl verwendete Prozessor, wird im Dialogfeld "Optionen" (**Extras | Optionen**) auf dem [Register XSL](#) festgelegt.

Sollte Ihr XML-Dokument eine Referenz zu einem XSLT Stylesheet enthalten, wird dieses Stylesheet automatisch für die XSLT Transformation verwendet. (Ein XSLT Stylesheet kann über den Befehl zugewiesen werden). Wenn das XML-Dokument Teil eines Projekts ist, können Sie im Dialogfeld "[Projekteigenschaften](#)" ein XSLT Stylesheet auch für einzelne Ordner definieren. Klicken Sie mit der rechten Maustaste auf den/die Projektordner oder die zu transformierende Datei(en) und wählen Sie XSL-Transformation. Wurde einer XML-Datei kein XSLT Stylesheet zugewiesen, werden Sie aufgefordert das XSLT Stylesheet anzugeben. Sie können eine Datei auch über eine globale Ressource, eine URL (klicken Sie dazu auf die Schaltfläche [Durchsuchen](#)) oder eines der offenen Fenster in XMLSpy auswählen (Klicken Sie dazu auf die Schaltfläche **Fenster**).

Automatisieren von XSLT-Transformationen mit Altova XML 2009

Altova XML ist eine kostenlose Applikation, die den Altova XML-Validator sowie einen XSLT 1.0-, XSLT 2.0- und XQuery 1.0-Prozessor enthält. Das Tool kann über die Befehlszeile, über eine COM-Schnittstelle, in Java-Programmen und in .NET-Applikationen zur Validierung von XML-Dokumenten, zum Transformieren von XML-Dokumenten unter Verwendung von XSLT 1.0 und 2.0 Stylesheets sowie zum Ausführen von XQuery-Dokumenten verwendet werden.

XSLT-Transformationen können daher mittels Altova XML automatisiert werden. So können Sie z.B. eine Batch-Datei erstellen, die Altova XML aufruft, um damit eine Reihe von XML-Dokumenten zu transformieren. Nähere Informationen dazu finden Sie in der [Altova XML Dokumentation](#).

Transformationen in ZIP-Dateien

Damit die Ausgabe in Form einer ZIP-Datei, z.B. einer Open Office XML-Datei (OOXML) wie `.docx` erfolgt, muss im Dateipfad der Ausgabedatei das ZIP-Protokoll definiert werden. Zum Beispiel:

```
dateiname.zip| zip/dateiname.xxx  
dateiname.docx| zip/dateiname.xxx
```

Anmerkung: Vor Ausführung der Transformation muss eventuell die Verzeichnisstruktur erstellt werden. Wenn Sie Dateien für ein Open Office XML-Archiv generieren, müssen Sie die Archivdateien zippen, um OOXML-Datei der obersten Ebene zu erstellen (z.B. `.docx`).

9.5.2 XSL:FO-Transformation



Strg+F10

FO ist ein XML-Format zur Beschreibung von seitenweise formatierten Dokumenten. Ein FO-Prozessor wie z.B. der FOP-Prozessor von Apache XML Projects verarbeitet eine FO-Datei und generiert anhand dieser Datei ein PDF-Dokument. Die Generierung eines PDF-Dokuments erfolgt also in zwei Schritten.

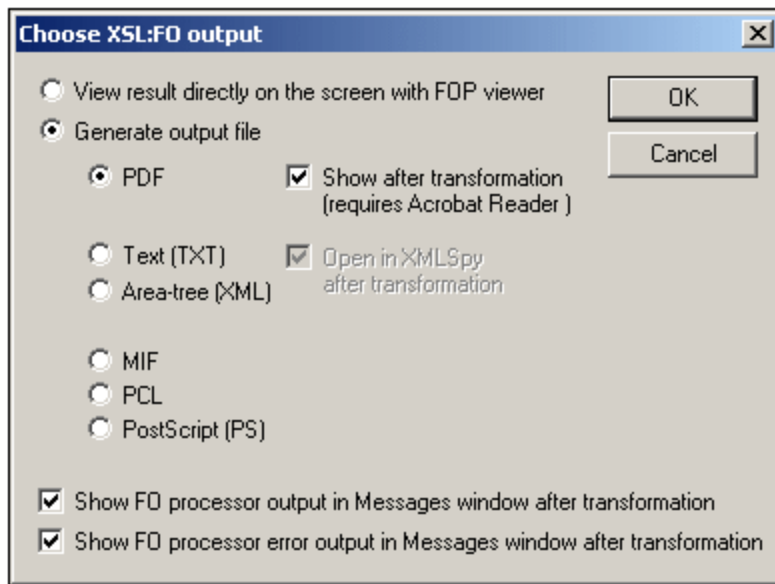
1. Das XML-Dokument wird mittels eines XSLT Stylesheet (auch als XSL-FO bezeichnet) in ein FO-Dokument transformiert.
2. Aus dem FO-Dokument wird von einem FO-Prozessor ein PDF-Dokument (oder ein anderes Ausgabeformat) generiert.

Mit dem Befehl **XSL:FO-Transformation** wird ein XML-Dokument oder ein FO-Dokument ins PDF-Format transformiert.

- Wird die **XSL:FO-Transformation** an einem XML-Quelldokument durchgeführt, werden beide oben beschriebenen Schritte nacheinander durchgeführt. Wenn das für die Transformation in FO benötigte XSLT (oder XSL-FO) Stylesheet im XML-Dokument nicht referenziert ist, werden Sie aufgefordert, eines für die Transformation zuzuweisen (*siehe Abbildung unten*). Sie können eine Datei auch über eine globale Ressource, eine URL (klicken Sie dazu auf die Schaltfläche [Durchsuchen](#)) oder eines der offenen Fenster in XMLSpy auswählen (Klicken Sie dazu auf die Schaltfläche **Fenster**). Die Transformation von XML in XSL-FO wird durch den XSLT-Prozessor durchgeführt, der im Dialogfeld "Optionen" (**Extras | Optionen**) auf dem [Register XSL](#) definiert wurde. Standardmäßig ist der integrierte XSLT-Prozessor von XMLSpy als XSLT-Prozessor ausgewählt. Das erzeugte FO-Dokument wird von dem FO-Prozessor, der im Dialogfeld "Optionen" (**Extras | Optionen**), [Register XSL](#) definiert wurde, direkt verarbeitet.
Wird der Befehl **XSL:FO-Transformation** an einem FO-Dokument ausgeführt, wird das Dokument mit dem FO-Prozessor, der im Dialogfeld "Optionen" (**Extras | Optionen**), [Register XSL](#) definiert wurde, verarbeitet.

XSL:FO-Transformationsausgabe

Wenn Sie den Befehl **XSL:FO-Transformation** aufrufen, wird das Dialogfeld "XSL:FO-Ausgabe wählen" angezeigt (*siehe Abbildung unten*). (Wenn dem aktiven XML-Dokument kein XSLT Stylesheet zugewiesen wurde, werden Sie zuerst nach der XSLT-Datei gefragt).



Sie können das vom FO-Prozessor erzeugte Dokument mit Hilfe des FOP-Viewers direkt am Bildschirm anzeigen oder Sie können eine Ausgabedatei in einem der folgenden Formate erzeugen: PDF, Text, als XML-Baumdiagramm, MIF PCL oder PostScript. Sie können auch Meldungen von FO-Prozessor aktivieren, damit (i) die Standard-Ausgabemeldung des Prozessors im Fenster "Meldungen" angezeigt wird und (ii) die Fehlermeldungen des Prozessors im Fenster "Meldungen" angezeigt werden. Um eine der beiden Optionen zu aktivieren, klicken Sie auf das entsprechende Kontrollkästchen um unteren Rand des Dialogfelds.

Bitte beachten Sie:

- Der Apache FOP-Prozessor kann kostenlos über den Link im [Altova Download Center](#) heruntergeladen werden. Laden Sie FOP herunter und installieren Sie den Prozessor und geben Sie anschließend den Pfad zur FOP Batch-Datei im Dialogfeld "Optionen" (**Extras | Optionen**) auf dem [Register XSL](#) an.

9.5.3 XSL-Parameter/XQuery-Variablen

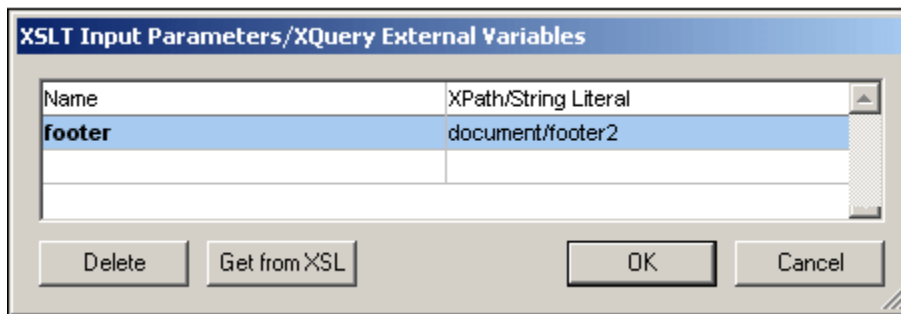
Mit dem Befehl **XSL-Parameter/XQuery-Variablen** öffnen Sie das Dialogfeld "XSLT-Eingabeparameter/externe XQuery-Variablen" (*siehe Screenshot*). Sie können den Namen und die dazugehörigen Werte von einem oder mehreren Parametern eingeben, die Sie an das XSLT Stylesheet übergeben möchten, oder den Namen und die dazugehörigen Werte von einer oder mehreren XQuery-Variablen, die an das XQuery-Dokument übergeben werden sollen. Diese Parameter werden in XMLSpy folgendermaßen verwendet:

- Wenn ein XML-Dokument über den Befehl **XSL-Transformation** aus dem Menü "XSL/XQuery" transformiert werden, werden die derzeit im Dialogfeld gespeicherten Parameterwerte an das ausgewählte XSLT-Dokument übergeben und für die Transformation verwendet.
- Wenn ein XQuery-Dokument über den Befehl **XQuery-Ausführung** aus dem Menü "XSL/XQuery" verarbeitet wird, werden die aktuell im Dialogfeld gespeicherten externen XQuery-Variablen zur Ausführung an das XQuery-Dokument übergeben.

Bitte beachten Sie: Parameter oder Variablen, die Sie im Dialogfeld "XSLT-Eingabeparameter/externe XQuery-Variablen" eingeben, werden nur an den integrierten Altova XSLT-Prozessor übergeben. Wenn Sie daher MSXML oder einen anderen externen Prozessor verwenden, den Sie konfiguriert haben, werden diese Parameter nicht an diesen Prozessor übergeben.

Verwendung von XSLT-Parametern

Bei dem Wert, den Sie für den Parameter eingeben, kann es sich um einen XPath-Ausdruck ohne Anführungszeichen oder einen von Anführungszeichen eingeschlossenen Textstring handeln. Wenn das aktive Dokument ein XSLT-Dokument ist, ist die Schaltfläche **Aus XSL abrufen** aktiv. Wenn Sie darauf klicken, werden die im XSLT-Dokument deklarierten Parameter zusammen mit ihren Standardwerten in das Dialogfeld eingefügt. Auf diese Art können Sie deklarierte Parameter schnell inkludieren und deren Standardwerte gegebenenfalls ändern.



Bitte beachten Sie: Sobald Sie im Dialogfeld "XSLT-Eingabeparameter/externe XQuery-Variablen" die Parameterwerte eingegeben haben, werden diese für alle zukünftigen Transformationen verwendet, bis sie ausdrücklich gelöscht werden oder die Applikation neu gestartet wird. Parameter, die im Dialogfeld "XSLT-Eingabeparameter/externe XQuery-Variablen" eingegeben wurden, werden auf Applikationsebene definiert und werden von diesem Punkt an bei jeder Transformation, die über die IDE durchgeführt wird, an das jeweilige XSLT-Dokument übergeben. Dies bedeutet:

- Parameter sind nicht mit einem bestimmten Dokument verknüpft und
- Alle Parameter, die im Dialogfeld "XSLT-Eingabeparameter/externe XQuery-Variablen" eingegeben wurden, werden gelöscht, sobald die Applikation () geschlossen wird.

Verwendung von XSLT-Parametern

Im folgenden Beispiel wählen wir die für das PDF-Dokument benötigte Fußzeile aus drei Wahlmöglichkeiten im XML-Dokument (footer1, footer2, footer3) aus.

```
<?xml version="1.0" encoding="UTF-8"?>
<document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="C:\workarea\footers\footers.xsd">
  <footer1>Footer 1</footer1>
  <footer2>Footer 2</footer2>
  <footer3>Footer 3</footer3>
  <title>Document Title</title>
  <para>Paragraph text.</para>
  <para>Paragraph text.</para>
</document>
```

Die XSLT-Datei enthält in der Vorlage für das Root-Element einen lokalen Parameter namens footer. Dieser Parameter hat einen Standardwert footer1. Der Parameter wird anschließend in der Vorlage in der Definition des Footer-Blocks mit einem Wert \$footer instanziiert.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fo="http://www.w3.org/1999/XSL/Format">
  ...
  <xsl:param name="footer" select="document/footer1" />
  ...
  <xsl:template match="/">
    <fo:root>
      <xsl:copy-of select="$fo:layout-master-set" />
      <fo:page-sequence master-reference="default-page"
        initial-page-number="1" format="1">
        <fo:static-content flow-name="xsl-region-after"
          display-align="after">
          ...
          <fo:inline color="#800000" font-size="10pt" font-weight="bold">
            <xsl:value-of select="$footer"/>
          </fo:inline>
          ...
        </fo:static-content>
      </fo:page-sequence>
    </fo:root>
  </xsl:template>
</xsl:stylesheet>
```

Sie können im XSLT-Eingabeparameter-Dialogfeld einen neuen Wert für den Parameter footer eingeben, z.B. den XPath: document/footer2 (siehe Abbildung oben) oder einen Textstring. Während der Transformation wird dieser Wert an den Parameter footer in der Vorlage für das Root-Element übergeben und beim Instanzieren des Footer-Blocks verwendet.

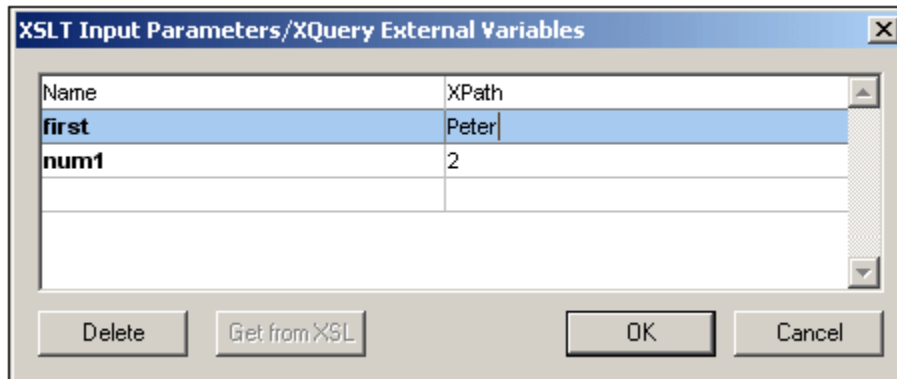
Bitte beachten Sie:

- Wenn Sie den Befehl **XSL:FO-Transformation (XSL/XQuery | XSL:FO-Transformation)** verwenden, werden Parameter, die im Dialogfeld "XSLT-Eingabeparameter/externe XQuery-Variablen" eingegeben werden, **nicht** an das Stylesheet übergeben. Damit diese Parameter in der PDF-Ausgabe verwendet werden, führen Sie mittels des Befehls XSLT-Transformation (**XSL/XQuery | XSL-Transformation**) zuerst eine XSL-FO-Transformation durch und anschließend mit Hilfe des Befehls **XSL:FO-Transformation (XSL/XQuery | XSL:FO-Transformation)** eine FO-PDF-Transformation.
- Falls Sie einen anderen als den integrierten Altova XSLT-Prozessor verwenden, werden Parameter, die Sie mit Hilfe des Befehls "XSLT-Eingabeparameter" eingeben,

nicht an den externen Prozessor übergeben.

Verwendung von externen XQuery-Variablen

Beim Wert, den Sie für eine externe XQuery-Variable eingeben, muss es sich um einen Literalwert ohne Anführungszeichen handeln. Der Datentyp der externen Variable wird in der Variablendeklaration im XQuery-Dokument angegeben.



Bitte beachten Sie: Sobald Sie im Dialogfeld "XSLT-Eingabeparameter/externe XQuery-Variablen" eine Reihe von externen XQuery-Variablen eingegeben haben, werden diese für alle zukünftigen Transformationen verwendet, bis sie ausdrücklich gelöscht werden oder die Applikation neu gestartet wird. Variablen, die im Dialogfeld "XSLT-Eingabeparameter/externe XQuery-Variablen" eingegeben wurden, werden auf Applikationsebene definiert und werden von diesem Punkt an bei jeder Transformation, die über die IDE durchgeführt wird, an das jeweilige XSLT-Dokument übergeben. Dies bedeutet:

- Variablen sind nicht mit einem bestimmten Dokument verknüpft und
- Alle Variablen, die im Dialogfeld "XSLT-Eingabeparameter/externe XQuery-Variablen" eingegeben wurden, werden gelöscht, sobald die Applikation () geschlossen wird.

Anwendungsbeispiel für externe XQuery-Variablen

Im folgenden Beispiel wird im XQuery-Dokument eine Variable `$first` deklariert und anschließend in der Return Clause des FLWOR-Ausdrucks verwendet:

```
xquery version "1.0";
declare variable $first as xs:string external;
let $last := "Jones"
return concat($first, " ", $last )
```

Diese XQuery gibt `Peter Jones` zurück, wenn der Wert der externen Variable (der im Dialogfeld "XSLT-Eingabeparameter/externe XQuery-Variablen" eingegeben wurde) `Peter` ist. Beachten Sie Folgendes:

- Das externe Schlüsselwort in der Variablendeklaration im XQuery-Dokument gibt an, dass es sich bei dieser Variable um eine externe Variable handelt.
- Die Definition des statischen Typs der Variable ist optional. Wenn der Datentyp für die Variable nicht in der Variablendeklaration angegeben wurde, wird dem Variablenwert der Typ `xs:untypedAtomic` zugewiesen.
- Wenn im XQuery-Dokument eine externe Variable deklariert wurde, aber keine externe Variable dieses Namens an das XQuery-Dokument übergeben wird, wird eine Fehlermeldung ausgegeben.
- Wenn eine externe Variable deklariert und im Dialogfeld "XSLT-Eingabeparameter/externe XQuery-Variablen" eingegeben wurde, wird diese als Gültigkeitsbereich für das auszuführende XQuery-Dokument verwendet. Wird eine neue Variable dieses Namens im XQuery-Dokument deklariert, setzt diese die externe Variable für den Gültigkeitsbereich vorübergehend außer Kraft. Das unten gezeigte

XQuery-Dokument gibt z.B. Paul Jones zurück, obwohl die externe in-scope Variable \$first den Wert Peter hat.

```
xquery version "1.0";
declare variable $first as xs:string external;
let $first := "Paul"
let $last := "Jones"
return concat($first, " ", $last )
```

Bitte beachten Sie: Es handelt sich nicht um einen Fehler, wenn im Dialogfeld "XSLT-Eingabeparameter/externe XQuery-Variablen" eine externe XQuery-Variable (oder ein XSLT-Parameter) definiert ist, diese im XQuery-Dokument aber nicht verwendet wird. Auch wenn ein XSLT-Parameter (oder eine externe XQuery-Variable) im Dialogfeld "XSLT-Eingabeparameter/externe XQuery-Variablen" definiert ist, aber in einer XSLT-Transformation nicht verwendet wird, handelt es sich nicht um einen Fehler.

9.6 Menü "Authentic"

Über die Authentic-Ansicht können Sie XML-Dokumente editieren, **die auf StyleVision Power Stylesheets basieren, die mit StyleVision von Altova erstellt worden sind!** Diese Stylesheets enthalten alle Informationen, die benötigt werden, damit eine XML-Datei grafisch in der Authentic-Ansicht dargestellt werden kann. Zusätzlich zu diesen Anzeigeinformationen ermöglichen Ihnen StyleVision Power Stylesheets außerdem, Daten in eine XML-Datei zu schreiben. Diese Daten werden mit Hilfe aller XSLT-Stylesheet-Funktionen dynamisch verarbeitet und sofort in Form einer Authentic-Ansicht ausgegeben.

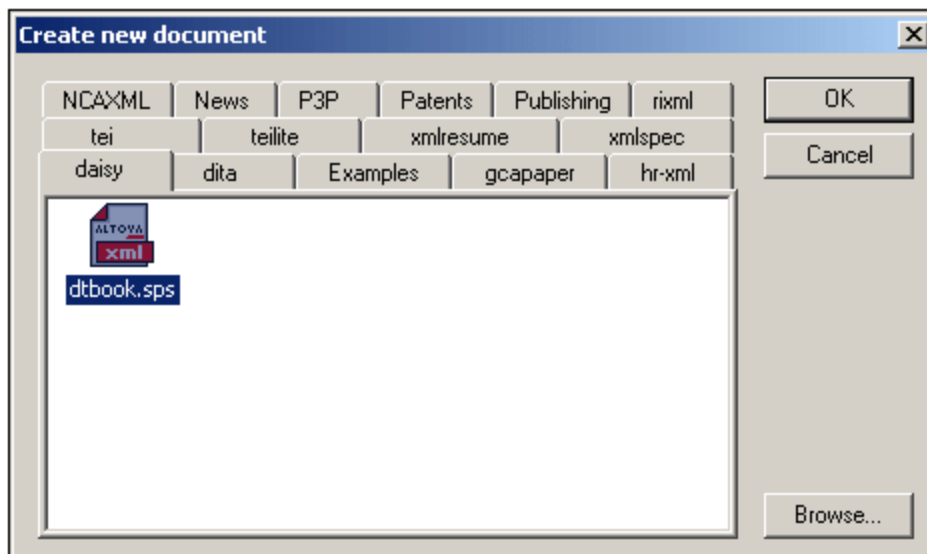
Des Weiteren können Sie StyleVision Power Stylesheets erstellen, um eine editierbare XML-Ansicht einer Datenbank anzuzeigen. Das StyleVision Power Stylesheet enthält Informationen, um eine Verbindung zu einer Datenbank herzustellen, die Datenbankdaten in der Authentic-Ansicht anzuzeigen und wieder zurück in die Datenbank zu schreiben.

Das Menü "Authentic" enthält Befehle zum Editieren von XML-Dokumenten in der Authentic-Ansicht. Ein Tutorial zur Authentic-Ansicht finden Sie im Abschnitt [Tutorials](#).

9.6.1 Neues Dokument

Mit dem Befehl **Neues Dokument...** können Sie eine neue XML-Dokumentenvorlage in der Authentic-Ansicht öffnen. Die XML-Dokumentenvorlage basiert auf einem StyleVision Power Stylesheet (.sps-Datei) und wird durch Auswahl des StyleVision Power Stylesheet geöffnet.

Wenn Sie den Befehl **Neues Dokument...** auswählen, wird das Dialogfeld "Neues Dokument anlegen" angezeigt.



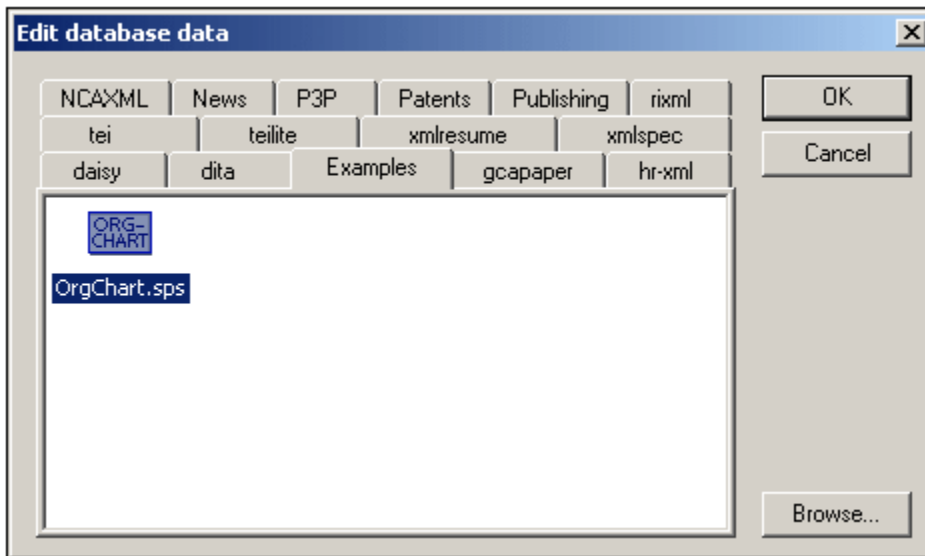
Suchen Sie die gewünschte SPS-Datei und wählen Sie diese aus. Daraufhin wird eine XML-Dokumentenvorlage in der Authentic-Ansicht geöffnet.

Bitte beachten Sie: StyleVision Power Stylesheets werden mit Hilfe von Altova <%SV-PROD%> erstellt. Dem StyleVision Power Stylesheet ist eine XML-Vorlagendatei zugewiesen. Die Daten in dieser XML-Datei liefern die Ausgangsdaten für die neue Dokumentenvorlage, die in der Authentic-Ansicht geöffnet wird.

9.6.2 Datenbankdaten bearbeiten

Mit dem Befehl **Datenbankdaten bearbeiten...** öffnen Sie eine editierbare Ansicht einer Datenbank (DB) in der Authentic-Ansicht. Alle Informationen über die Herstellung einer Verbindung zur Datenbank, die Anzeige der DB und die Übernahme von Änderungen an der Datenbank über die Authentic-Ansicht befinden sich in einem StyleVision Power Stylesheet. Mit dem Befehl **Datenbankdaten bearbeiten...** öffnen Sie ein solches DB-basiertes StyleVision Power Stylesheet. Damit stellen Sie eine Verbindung zur DB her und zeigen die DB-Daten (durch eine XML-Linse) in der Authentic-Ansicht an.

Bei Aufruf des Befehls **Datenbankdaten bearbeiten...** erscheint das Dialogfeld "Datenbankdaten bearbeiten".



Suchen Sie die gewünschte SPS-Datei und wählen Sie diese aus. Diese Datei stellt die Verbindung zur DB her und öffnet eine editierbare Ansicht der Daten in der Authentic-Ansicht.

Warnung: Wenn Sie versuchen, ein StyleVision Power Stylesheet mit dem Befehl **Datenbankdaten bearbeiten...** zu öffnen, das nicht auf einer DB basiert, oder wenn Sie versuchen, ein DB-basiertes StyleVision Power Stylesheet zu öffnen, das in einer StyleVision -Version vor der Version StyleVision 2005 erstellt wurde, erhalten Sie eine Fehlermeldung.

Bitte beachten Sie: StyleVision Power Stylesheets werden mit Altova <%SV-PROD%> erstellt.

9.6.3 StyleVision Stylesheet zuweisen

Mit diesem Befehl weisen Sie einem **XML-Dokument** ein StyleVision Power Stylesheet (SPS) zu, um die Ansicht und Bearbeitung in der Authentic-Ansicht zu ermöglichen. Das StyleVision Power Stylesheet, das der XML-Datei zugewiesen werden soll, muss auf demselben Schema basieren, auf dem auch die XML-Datei basiert.

So weisen Sie einer XML-Datei ein StyleVision Power Stylesheet zu:

1. Machen Sie die XML-Datei zur aktiven und wählen Sie den Befehl **Stylevision Stylesheet zuweisen...**
2. Daraufhin wird ein Dialogfeld geöffnet, in dem Sie angeben können, welche StyleVision Power Stylesheet-Datei zugewiesen werden soll.
3. Klicken Sie auf OK, um die gewünschte SPS-Anweisung in Ihr XML-Dokument einzufügen. Beachten Sie bitte, dass Sie den Pfad der zugewiesenen Datei relativ machen können, indem Sie das Kontrollkästchen "Pfad relativ machen zu ..." aktivieren. Sie können eine Datei auch über eine globale Ressource, eine URL (klicken Sie dazu auf die Schaltfläche [Durchsuchen](#)) oder eines der offenen Fenster in XMLSpy auswählen (Klicken Sie dazu auf die Schaltfläche **Fenster**).

```
<?xml version="1.0" encoding="UTF-8"?>  
<?altova_sps HTML-Orgchart.sps?>
```

Im obigen Beispiel ist der Name des StyleVision Power Stylesheet `HTML_Orgchart.sps`, und die Datei befindet sich im selben Verzeichnis wie die XML-Datei.

Bitte beachten Sie: In früheren Altova-Produktversionen wurde eine Verarbeitungsanweisung mit dem Ziel bzw. Namen `xmlspysps` verwendet. Eine Verarbeitungsanweisung sah daher etwa folgendermaßen aus: `<?xmlspysps HTML-Orgchart.sps?>`. Diese älteren Verarbeitungsanweisungen sind auch in aktuellen Altova-Produktversionen bei Authentic immer noch gültig.

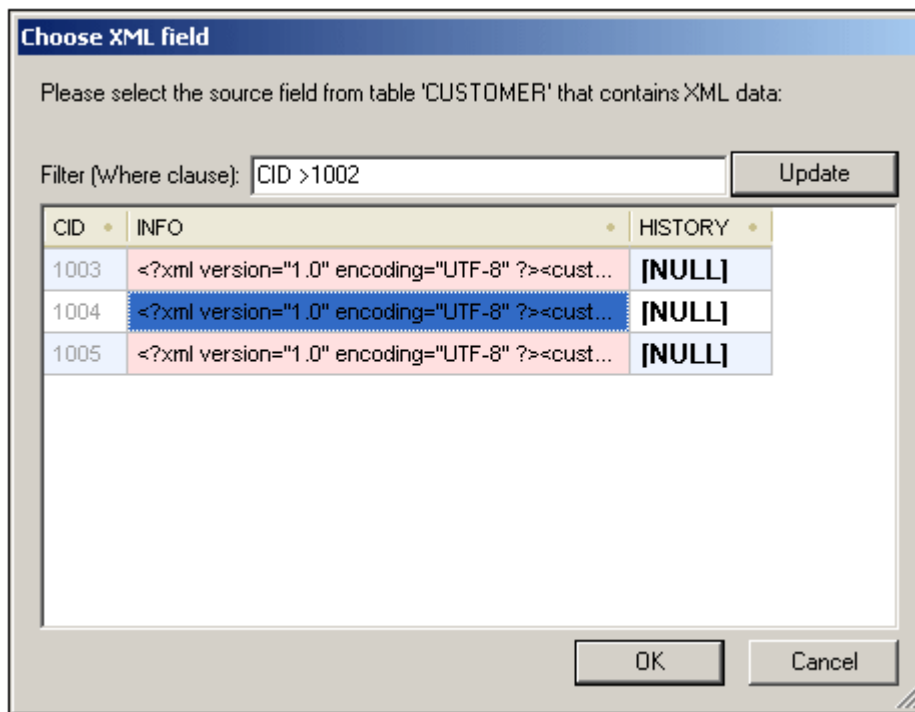
Bearbeiten eines StyleVision Stylesheet

Mit dem Befehl **Authentic | StyleVision Stylesheet bearbeiten** wird StyleVision gestartet. Auf diese Art können Sie das StyleVision Power Stylesheet sofort in StyleVision bearbeiten.

9.6.4 Neue Zeile mit XML-Daten für Bearbeitung auswählen

Mit diesem Befehl können Sie eine neue Zeile in der entsprechenden Tabelle einer XML-Datenbank wie z.B. IBM DB2 auswählen. Diese Zeile wird nun in der Authentic-Ansicht angezeigt, kann hier bearbeitet und wieder in der Datenbank gespeichert werden.

Bei Verwendung einer XML-Datenbank als XML-Datenquelle handelt es sich bei den in der Authentic-Ansicht angezeigten XML-Daten um das XML-Dokument, das in einer der Zellen der XML-Datenspalte enthalten ist. Mit dem Befehl **Neue Zeile mit XML-Daten für Bearbeitung auswählen** können Sie ein XML-Dokument aus einer anderen Zelle (oder Zeile) dieser XML-Spalte auswählen. Bei Auswahl des Befehls **Neue Zeile...auswählen** wird das Dialogfeld "XML-Feld auswählen" aufgerufen (*Abbildung unten*), in dem die Tabelle, die die XML-Spalte enthält angezeigt wird.



Sie können für diese Tabelle einen Filter eingeben. Dabei sollte es sich um eine SQL `WHERE`-Klausel handeln (nur die Bedingung ohne das `WHERE`-Schlüsselwort wie z.B.: `CID>1002`). Klicken Sie auf **Aktualisieren**, um das Dialogfeld zu aktualisieren. In der Abbildung oben sehen Sie das Ergebnis einer gefilterten Ansicht. Wählen Sie als nächstes die Zelle, die das gewünschte XML-Dokument enthält aus, und klicken Sie auf **OK**. Das XML-Dokument in der ausgewählten Zelle (Zeile) wird in die Authentic-Ansicht geladen.

9.6.5 XML Entities definieren

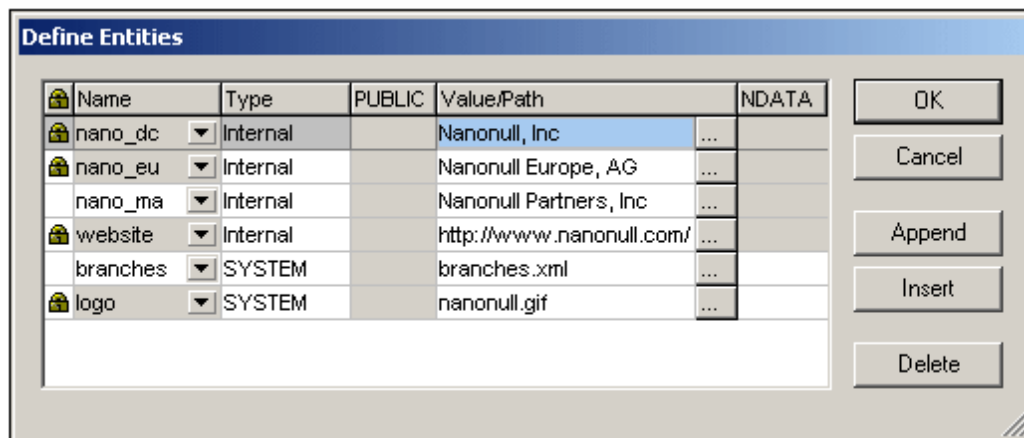
Unabhängig davon, ob Ihr Dokument auf einer DTD oder einem XML-Schema basiert, können Sie für die Authentic-Ansicht Entitäten definieren, die dann in der Eingabehilfe "Entities" und im Kontext-Untermenü **Entity einfügen** angezeigt werden. Wenn Sie in der Eingabehilfe "Entities" auf eine Entität doppelklicken, wird die Entität an der aktuellen Cursorposition eingefügt.

Eine Entität ist nützlich, wenn Sie einen Textstring, ein XML-Fragment oder eine andere externe Ressource an mehreren Stellen in Ihrem Dokument einfügen möchten. Sie definieren die Entität, die im Grund eine Kurzbezeichnung ist, die die gewünschten Daten repräsentiert, im Dialogfeld "Entities definieren". Nachdem Sie die Entität definiert haben, können Sie sie an mehreren Stellen in Ihrem Dokument einfügen. Dies spart Zeit und ist effizient, wenn Sie Änderungen am Dokument vornehmen möchten.

Sie können in Ihrem Dokument zwei Arten von Entitäten verwenden: **geparste** Entitäten, wobei es sich um XML-Daten handelt (entweder Textstrings oder ein Fragment eines XML-Dokuments) oder **nicht geparte** Entitäten, wobei es sich um nicht-XML-Daten wie z.B. eine Binärdatei handelt (gewöhnlich Grafiken, Sound oder Multimedia-Objekte). Jede Entität hat einen Namen und einen Wert. Bei geparten Entitäten ist die Entität ein Platzhalter für die XML-Daten. Der Wert der Entität sind entweder die XML-Daten selbst oder eine URI, die auf eine XML-Datei verweist, die die XML-Daten enthält. Bei nicht geparten Entitäten ist der Wert der Entität eine URI, die auf die nicht-XML-Datendatei verweist.

Um eine Entität zu definieren, gehen Sie vor wie folgt:

1. Wählen Sie **Authentic | XML Entities definieren....** Dadurch wird das Dialogfeld "Entities definieren" geöffnet.



2. Geben Sie in das Feld "Name" den Namen der Entität ein. Dies ist der Name, der in der Eingabehilfe "Entities" angezeigt werden wird.
3. Wählen Sie im Feld "Type" den Entitätstyp aus der Dropdown-Liste aus. Es stehen drei Typen zur Auswahl. Bei einer **internen** Entität handelt es sich um Text, der im XML-Dokument selbst gespeichert ist. Bei Auswahl von **PUBLIC** oder **SYSTEM** geben Sie an, dass sich die Ressource außerhalb der XML-Datei befindet und mit Hilfe eines Public Identifier bzw. eines System Identifier aufgerufen wird. Ein System Identifier ist eine URI, die den Speicherort einer Ressource angibt. Der Public Identifier ist ein pfadunabhängiger Identifier, der es einigen Prozessoren erlaubt, die Ressource zu identifizieren. Wenn Sie sowohl einen Public Identifier als auch einen System Identifier definieren, wird der Public Identifier mit dem System Identifier aufgelöst und der System Identifier wird verwendet.
4. Wenn Sie als Typ PUBLIC ausgewählt haben, geben Sie den Public Identifier Ihrer Ressource im Feld PUBLIC ein. Wenn Sie Internal oder SYSTEM als Typ ausgewählt haben, wird das Feld PUBLIC deaktiviert.

5. Im Feld **Value/Path** haben Sie folgende Eingabemöglichkeiten:
 - Ist der Entitätstyp "Internal", geben Sie den gewünschten Textstring als Wert der Entität ein. Verwenden Sie zur Begrenzung der Entität keine Anführungszeichen, da diese als Teil des Textstrings behandelt werden.
 - Ist der Entitätstyp SYSTEM, geben Sie die URI der Ressource an oder wählen Sie mit Hilfe der Schaltfläche zum Durchsuchen eine Ressource in Ihrem lokalen Netzwerk aus. Wenn die Ressource geparste Daten enthält, muss es eine XML-Datei sein (d.h. sie muss eine XML-Erweiterung haben). Andernfalls kann als Ressource eine Binärdatei wie z.B. eine GIF-Datei verwendet werden.
 - Ist der Entitätstyp PUBLIC, müssen Sie in diesem Feld zusätzlich noch einen System Identifier eingeben.
6. Der Eintrag NDATA sagt dem Prozessor, dass diese Entität nicht geparst, sondern an den entsprechenden Prozessor gesendet werden soll. Das Feld NDATA sollte daher nur bei nicht geparsten Entitäten verwendet werden.

Dialogfeldfunktionen

Sie können Entitäten anhängen, einfügen und löschen, indem Sie auf die entsprechende Schaltfläche klicken. Des Weiteren können Sie Entitäten spaltenweise alphabetisch sortieren, indem Sie auf die Kopfzeile der jeweiligen Spalte klicken; bei einmaligem Klicken werden die Entitäten aufsteigend sortiert, bei zweimaligem Klicken absteigend. Die Größe des Dialogfelds und die Spaltenbreite lassen sich anpassen.

Sobald eine Entity im XML-Dokument verwendet wird, ist sie gesperrt und kann im Dialogfeld "Entities definieren" nicht mehr bearbeitet werden. Gesperrte Entities werden durch ein Schlosssymbol in der ersten Spalte gekennzeichnet. Durch das Sperren einer Entity wird sichergestellt, dass das XML-Dokument hinsichtlich der Entities gültig ist. (Das Dokument wäre ungültig, wenn eine Entity referenziert ist, die nicht definiert ist.)

Einschränkungen

- Eine in einer anderen Entität enthaltene Entität wird weder im Dialogfeld, noch der Authentic-Ansicht noch der XSLT-Ausgabe aufgelöst. Das kaufmännische Und-Zeichen einer solchen Entität wird in seiner Escape-Form also als `&` angezeigt.
- Externe Entitäten werden in der Authentic-Ansicht nicht aufgelöst, außer es handelt sich bei der Entität um eine Bilddatei, die als Wert eines Attributs eingegeben wurde, dessen Typ im Schema als `ENTITY` oder `ENTITIES` definiert wurde. Solche Entitäten werden aufgelöst, wenn das Dokument mit einem in der SPS generierten XSLT verarbeitet wird.

9.6.6 Markup-Symbole ausblenden, Kleine/Große/Gemischte Markup-Symbole einblenden



Mit dem Befehl **Markup-Symbole ausblenden** werden die Markup-Symbole in der Authentic-Ansicht ausgeblendet.



Mit dem Befehl **Kleine Markup-Symbole einblenden** werden kleine Markup-Symbole in der Authentic-Ansicht angezeigt.



Mit dem Befehl **Große Markup-Symbole einblenden** werden große Markup Symbole in der Authentic-Ansicht angezeigt.



Mit dem Befehl **Gemischte Markup-Symbole einblenden** werden gemischte Markup Symbole in der Authentic-Ansicht angezeigt. Der Ersteller des StyleVision Power Stylesheet kann für die einzelnen Elemente/Attribute im Dokument entweder große, kleine oder gar keine Markup-Symbole definieren. Der Benutzer der Authentic-Ansicht sieht diese benutzerdefinierten Markup-Symbole im Ansichtsmodus für gemischte Markup-Symbole.

9.6.7 Zeile anhängen/einfügen/kopieren/löschen



Mit dem Befehl **Zeile anhängen** wird eine Zeile an die Tabelle in der Authentic-Ansicht angehängt.



Mit dem Befehl **Zeile einfügen** wird eine Zeile in die Tabelle in der Authentic-Ansicht eingefügt.



Mit dem Befehl **Zeile kopieren** wird die markierte Zeile in der Tabelle in der Authentic-Ansicht dupliziert.



Mit dem Befehl **Zeile löschen** wird in der Authentic-Ansicht die markierte Zeile aus der Tabelle gelöscht.

9.6.8 Zeile nach oben/unten



Mit dem Befehl **Zeile nach oben** wird in der Authentic-Ansicht die markierte Zeile um eine Zeile nach oben verschoben.



Mit dem Befehl **Zeile nach unten** wird in der Authentic-Ansicht die markierte Zeile um eine Zeile nach unten verschoben.

9.7 Menü "Ansicht"

Im Menü Ansicht kontrollieren Sie die Anzeige für das aktive Hauptfenster. So können Sie sich in Ihre XML-Dokumente verschiedenartig anzeigen lassen.

In diesem Abschnitt finden Sie eine vollständige Beschreibung aller Befehle im Menü "Ansicht".

9.7.1 WSDL-Design-Ansicht

Mit diesem Befehl **wechseln** Sie mit dem aktuellen Dokument in die WSDL-Design-Ansicht. Sie können nur in diese Ansicht wechseln, wenn es sich beim Dokument um ein WSDL-Dokument handelt. Nähere Informationen dazu erhalten Sie im Abschnitt WSDL-Ansicht.

9.7.2 XBRL-Taxonomieansicht

Mit diesem Befehl **wechseln** Sie mit dem aktuellen Dokument in die XBRL-Taxonomieansicht. Sie können nur in diese Ansicht wechseln, wenn es sich beim Dokument um ein XBRL-Dokument handelt. Nähere Informationen dazu erhalten Sie im Abschnitt XBRL-Ansicht.

9.7.3 Authentic-Ansicht



Mit diesem Befehl wechseln Sie mit Ihrem Dokument in die [Authentic](#)-Ansicht.

In der Authentic-Ansicht können Sie XML-Dokumente bearbeiten, die auf in **<%XDESIGN%> erstellten StyleVision Power Stylesheet-Vorlagen basieren!** Die Vorlagen werden in **<%XDESIGN%>** als StyleVision Power Stylesheet (*.sps Dateien) gespeichert, und liefern alle für die Authentic-Ansicht benötigten Informationen.

Öffnen Sie die Vorlagen über den Menübefehl **Datei | Neu** und klicken auf die Schaltfläche "StyleVision Stylesheet auswählen...". Weitere Informationen finden Sie in der Dokumentation zur Authentic-Ansicht.

Bitte beachten Sie: Wenn bei Umschalten in die Authentic-Ansicht eine Meldung angezeigt wird, dass eine temporäre Datei nicht erstellt werden konnte, wenden Sie sich an Ihren Systemadministrator, damit dieser die Standard-Sicherheits-ID für Personen, die nicht als Hauptbenutzer angemeldet sind, ändert, so dass diese Ordner und Dateien erstellen können.

9.7.4 Browser-Ansicht



Mit diesem Befehl wechseln Sie mit dem aktuellen Dokument in die [Browser Ansicht](#).

Diese Ansicht verwendet einen XML-fähigen Browser um das XML-Dokument mit den aus CSS oder XSL Style-Sheets zur Verfügung gestellten Informationen darzustellen.

Wenn Sie in die Browser Ansicht wechseln, wird das Dokument zuerst auf Gültigkeit geprüft, wenn Sie die Option "Validieren beim Speichern der Datei" unter **Extras | Optionen** auf dem Register **Datei** einschalten.

Weitere Informationen zu dieser Ansicht finden Sie im Abschnitt Hauptfenster.

9.8 Menü "Browser"

Dieses Menü enthält Befehle, die nur verfügbar sind, wenn die [Browser Ansicht](#) aktiviert ist.

9.8.1 Zurück



Taste **Zurück** (**Alt + Pfeil nach links**)

Mit diesem Befehl wird in der Browser-Ansicht eine zuvor angezeigte Seite aufgerufen. Mit der Zurück-Taste auf Ihrer Tastatur erreichen Sie denselben Effekt. Dieser Befehl ist hilfreich, wenn Sie auf einen Link in Ihrem XML-Dokument klicken und danach zu Ihrem XML-Dokument zurückkehren möchten.

In der Schema-Ansicht gelangen Sie mit dem Befehl **Zurück** zur zuvor angezeigten Komponente. Das Tastaturkürzel dafür ist **Alt + Pfeil nach links**. Mit dem Befehl **Zurück** können Sie bis zu 500 zuvor angezeigte Positionen erneut anzeigen.

9.8.2 Vorwärts



(Alt + Pfeil nach rechts)

Dieser Befehl ist nur dann verfügbar, wenn Sie zuvor den Befehl **Zurück** verwendet haben. Mit diesem Befehl gelangen sie (i) in der Browser-Ansicht vorwärts zur vorher angezeigten Seite (ii) und in der Schema-Ansicht zu zuvor angezeigten Komponenten.

9.8.3 Abbrechen



Mit diesem Befehl beenden Sie das Laden eines neuen Dokuments.
Dieser Befehl ist nützlich, wenn Sie große externe Dateien oder graphische Darstellungen über eine langsame Internetverbindung laden.

9.8.4 Aktualisieren



F5

Mit diesem Befehl aktualisieren Sie die Browser Ansicht durch erneutes Laden des XML-Dokuments, sowie potentieller externer CSS oder XSL Stylesheets und DTDs.

9.8.5 Schriftgröße

Mit diesem Befehl können Sie die Schriftgröße wählen, die für Text in Ihrem XML-Dokument verwendet werden soll. Die Funktionsweise ist ähnlich wie die, welche in den meisten Browsern zur Anwendung kommt.

9.8.6 Neues Fenster



Mit diesem Befehl öffnen Sie die Browser Ansicht in einem neuen Fenster, sodass Sie Dokumente nebeneinander ansehen können.

Wenn Sie die Browser Ansicht in einem neuen Fenster geöffnet haben, drücken Sie in der Bearbeitungsansicht die Taste **F5**, um die Browser-Ansicht automatisch zu aktualisieren. Um separate Fenster wieder anzudocken, klicken Sie auf die Maximieren-Schaltfläche in der rechten oberen Ecke des aktiven Fensters.

9.9 Menü "Extras"

Im Menü Extras können Sie:

- Die [Rechtschreibung](#) Ihrer XML-Dokumente überprüfen
- In die Skript-Umgebung von wechseln. Sie können Ihre eigenen Formulare, Makros und Event Handler erstellen, bearbeiten und speichern.
- Derzeit zugewiesene Makros [ansehen](#)
- Skripts einem [Projekt](#) zuweisen oder zugewiesene Skripts löschen
- zwei beliebige Dateien vergleichen und auf Unterschiede überprüfen
- zwei beliebige Ordner vergleichen und auf Unterschiede überprüfen
- [globale Ressourcen definieren](#)
- [die aktive Konfiguration für globale Ressourcen](#) in XMLSpy wechseln
- Ihre Version von XMLSpyAuthentic Desktop [anpassen](#): eigene Menüleisten, Tastaturkürzel, Menüs und Makros erstellen
- Globale Authentic Desktop-[Einstellungen](#) definieren

9.9.1 Rechtschreibung

enthält eine integrierte Rechtschreibprüfung.

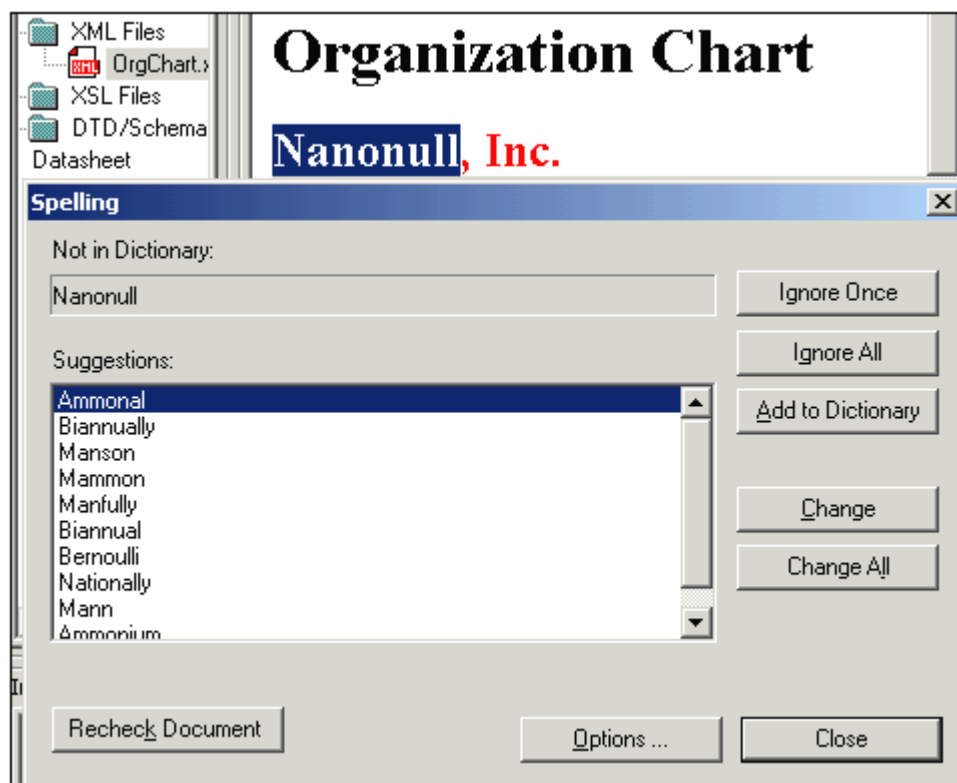
Die Rechtschreibprüfung kann in folgenden Ansichten verwendet werden: Grid, Text und Authentic. Bitte beachten Sie, dass jede Art von XML Datei überprüft werden kann. Die Ansicht, in der Sie die Rechtschreibung überprüfen, beeinflusst die Optionen, die für die Rechtschreibprüfung zur Verfügung stehen. Eine Schemadatei (*.xsd) kann in der Text- und Grid Ansicht, aber nicht in der Schema- oder der WSDL-Ansicht überprüft werden.

Die Rechtschreibprüfung in der **Text-** und **Grid-Ansicht** bietet Ihnen die meisten Optionen für die Überprüfung. Sie können zusätzlich die Rechtschreibung von folgenden Objekten überprüfen:

- Inhalt von Elementen
- Inhalt von Attributen
- Inhalt von CDaten
- Kommentartexten

Weiters können Sie definieren, welche Inhalte von Elementen oder Attributen überprüft oder übersprungen werden sollen. Die Einstellungen werden im Dialogfeld "[Rechtschreibkontext](#)" vorgenommen.

Mit dem Befehl **Extras | Rechtschreibung...** (Umschalt+F7) öffnen Sie das Dialogfeld für die Rechtschreibprüfung. Das aktive XML-Dokument wird automatisch auf Rechtschreibfehler geprüft (in diesem Screenshot das OrgChart.XML-Dokument in der Authentic-Ansicht).



Nicht im Wörterbuch

Dieses Textfeld enthält ein Wort aus dem Text, welches in keinem der existierenden Wörterbücher (Standardwörterbuch und benutzerdefinierte Wörterbücher) vorkommt. Sie

können den Text hier bearbeiten und auf **Ändern** klicken, um ihn im XML-Dokument zu ändern. Das Wort wird im XML-Dokument zur besseren Übersichtlichkeit markiert angezeigt. Die Schaltflächen werden nun aktiviert, und Sie können entscheiden, wie Sie fortfahren möchten.

Vorschläge

Dieses Listefeld zeigt eine Reihe von Wörtern an, die dem unbekannten Wort am nächsten kommen (aus allen verfügbaren Wörterbüchern). Durch Doppelklicken auf einen Eintrag wird dieser anstelle des unbekannten Wortes eingefügt und die Rechtschreibprüfung wird fortgesetzt.

Einmal ignorieren

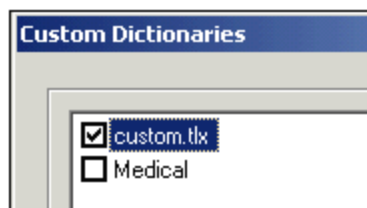
Mit diesem Befehl können Sie das unbekannte Wort bei seinem ersten Auftreten ignorieren und die Rechtschreibprüfung fortsetzen. Sollte das Wort ein weiteres Mal im Dokument vorkommen, wird die Rechtschreibprüfung es dann wieder als unbekannt markieren.

Immer ignorieren

Mit diesem Befehl wird das Wort durch das ganze Dokument ignoriert.

Ins Wörterbuch

Mit diesem Befehl können Sie unbekannte Wörter in das aktive **benutzerdefinierte Wörterbuch** einfügen. Das aktive benutzerdefinierte Wörterbuch ist das Wörterbuch, das im Dialogfeld "[Benutzerdefinierte Wörterbücher](#)" mit einem Häkchen versehen ist. Zum Öffnen dieses Dialogfelds klicken Sie auf **Optionen**, anschließend **Rechtschreiboptionen** und **Benutzerdefinierte Wörterbücher**.



Ändern

Mit diesem Befehl ersetzen Sie das im XML-Dokument markierte Wort durch den markierten Eintrag aus der Auswahlliste der Vorschläge.

Alle ändern

Mit diesem Befehl ersetzen Sie das Wort an allen Stellen im XML-Dokument, an denen es vorkommt.

Dokument erneut kontrollieren

Das Dokument wird vom Anfang an neu überprüft.

Schließen

Mit diesem Befehl beenden Sie die Rechtschreibprüfung.

Einstellungen...

Mit diesem Befehl öffnen Sie ein Dialogfeld in der aktuellen Ansicht:

- In der Text oder Enhanced Grid-Ansicht und in der Authentic-Ansicht wird das Dialogfeld [Rechtschreibkontext](#) geöffnet.

9.9.2 Rechtschreibung - Optionen

Mit diesem Befehl öffnen Sie das Dialogfeld **Optionen** oder **Rechtschreibkontext**, je nachdem, in welcher Ansicht Sie Ihr XML-Dokument derzeit bearbeiten.

Dialogfeld "Optionen" (Browser-Ansicht und Schema/WSDL-Ansicht)

Bei der Ansicht eines XML-Dokuments in der Browser-Ansicht oder der Schema/WSDL-Ansicht wird mit diesem Befehl das Dialogfeld "Optionen" geöffnet. Über dieses Dialogfeld können Sie Optionen für die Rechtschreibprüfung zu definieren.

Korrektur immer vorschlagen:

Wenn Sie dieses Kontrollkästchen aktivieren, werden alle Vorschläge (aus allen Wörterbüchern) in der Vorschlagsliste angezeigt. Wenn Sie diese Option deaktivieren, werden keinen Vorschläge angezeigt.

Korrekturen nur aus dem Hauptwörterbuch übernehmen:

Bei Aktivierung dieser Option werden nur Vorschläge angezeigt, die im Hauptwörterbuch zur Verfügung stehen. Weiters wird die Schaltfläche "Benutzerdefinierte Wörterbücher" deaktiviert.

Großschreibung ignorieren:

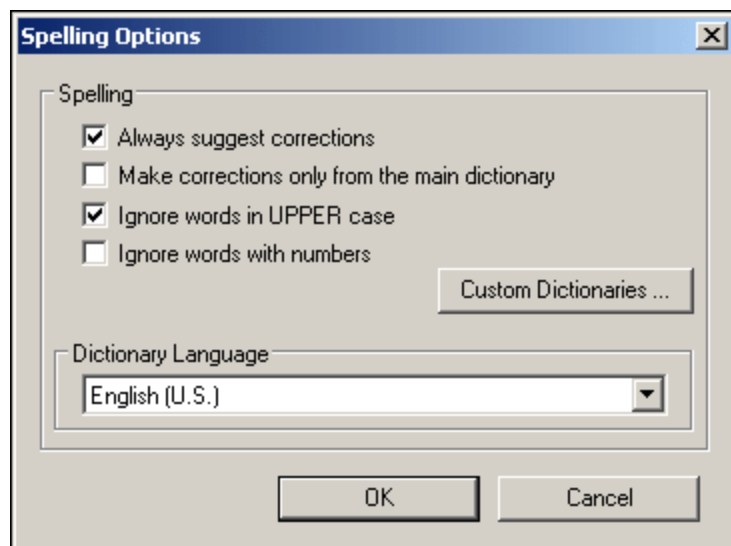
Großgeschriebene Wörter werden in der Rechtschreibprüfung ignoriert.

Wörter mit Ziffern ignorieren:

Wörter, die Ziffern enthalten, werden in der Rechtschreibprüfung ignoriert.

Sprache Wörterbuch

Wählen Sie eine Sprache im Dropdown-Listenfeld, in der die Rechtschreibprüfung erfolgen soll. Die standardmäßig ausgewählte Sprache ist derzeit Englisch. Weitere Sprachen werden demnächst zum Download von Altovas Webseite zur Verfügung stehen.



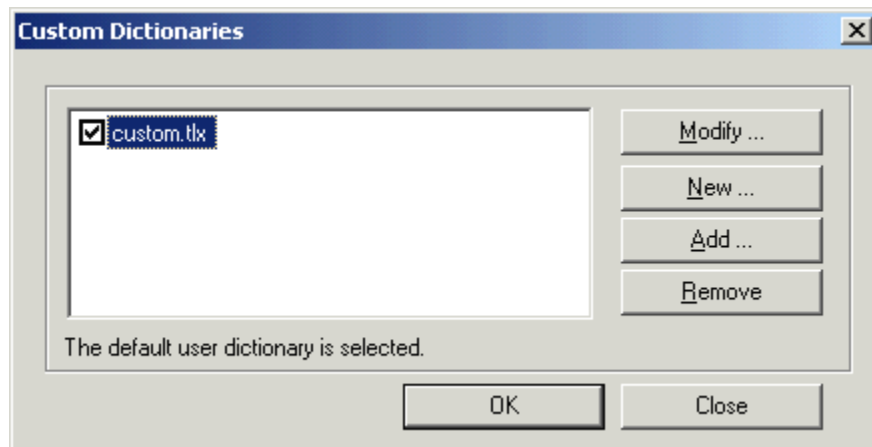
Benutzerdefinierte Wörterbücher

Mit dieser Schaltfläche können Sie:

- existierende Wörterbücher bearbeiten (Einträge hinzufügen oder löschen)
- neue Wörterbücher erstellen
- ein existierendes Wörterbuch hinzufügen
- ein existierendes Wörterbuch löschen

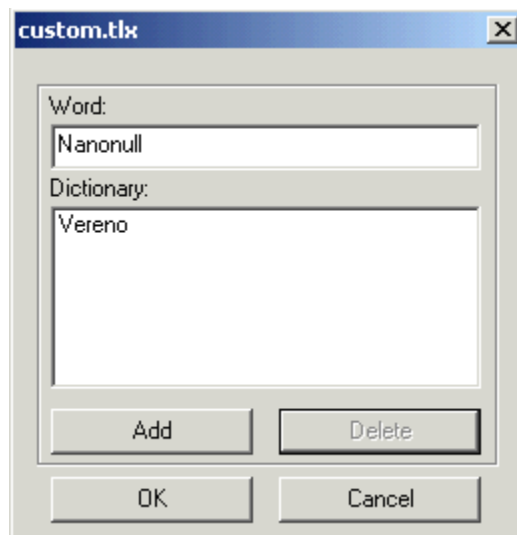
Wenn Sie die Rechtschreibprüfung starten, werden alle im Listefeld angezeigten

benutzerdefinierten Wörterbücher durchsucht. Mit dem Befehl **Entfernen** können Sie bestimmte Wörterbücher deaktivieren. Diese werden während der Rechtschreibprüfung nicht verwendet.



So fügen Sie einen neuen Wörterbucheintrag hinzu:

1. Klicken Sie im Dialogfeld "Benutzerdefinierte Wörterbücher" auf den Namen des benutzerdefinierten Wörterbuches, zu dem Sie einen Eintrag hinzufügen möchten, und klicken Sie auf **Ändern...**
Das ausgewählte Wörterbuch wird geöffnet (in unserem Beispiel, *custom.tlx*). Falls Sie kein Wörterbuch ausgewählt haben, werden Sie dazu aufgefordert.



2. Klicken Sie auf das Feld "Wort:" und geben Sie den neuen Wörterbucheintrag ein (die Schaltfläche "Hinzufügen" wird hierdurch automatisch aktiviert).
3. Klicken Sie auf "Hinzufügen", um den Eintrag in das Wörterbuch einzufügen.
4. Klicken Sie auf OK.

So löschen Sie einen Eintrag aus dem Wörterbuch:

1. Klicken Sie im Dialogfeld "Benutzerdefinierte Wörterbücher" auf den Namen des Wörterbuches aus dem Sie einen Eintrag löschen möchten und klicken Sie auf **Ändern...** Daraufhin wird das im Listefeld markierte Wörterbuch geöffnet (in diesem Fall *custom.tlx*). Wenn kein Wörterbuch ausgewählt wurde, werden Sie aufgefordert eines auszuwählen.
2. Klicken Sie auf das zu löschende Wort im Listefeld, und klicken Sie auf **Löschen**.

3. Klicken Sie auf **OK**.

So fügen Sie ein neues Wörterbuch ein:

1. Klicken Sie im Dialogfeld "Benutzerdefinierte Wörterbücher" auf **Neu**, und geben Sie den Namen des neuen benutzerdefinierten Wörterbuches im Feld "**Dateiname...**" ein.
2. Klicken Sie auf **Speichern** um das Wörterbuch zu speichern.
3. Über die Schaltfläche "Hinzufügen" oder "Ins Wörterbuch" können Sie nun während der Rechtschreibprüfung neue Einträge in das neue Wörterbuch einfügen.

So fügen Sie ein existierendes Wörterbuch ein:

Verwenden Sie diese Option um zuvor deaktivierte Wörterbücher (oder Wörterbücher von anderen Anbietern) wieder einzufügen.

1. Klicken Sie im Dialogfeld "Benutzerdefinierte Wörterbücher" auf "Hinzufügen" und geben Sie den Namen des Wörterbuchs in das Feld **Dateiname** ein. Wörterbücher erkennen Sie an der Erweiterung *.tlx.
2. Klicken Sie auf "Öffnen", um das Wörterbuch zu aktivieren.

Bitte beachten Sie:

Wörterbücher müssen nicht unbedingt eine *.tlx Erweiterung haben. Das Wörterbuch kann eine beliebige oder auch keine Erweiterung haben und trotzdem in die Liste der Wörterbücher aufgenommen werden.

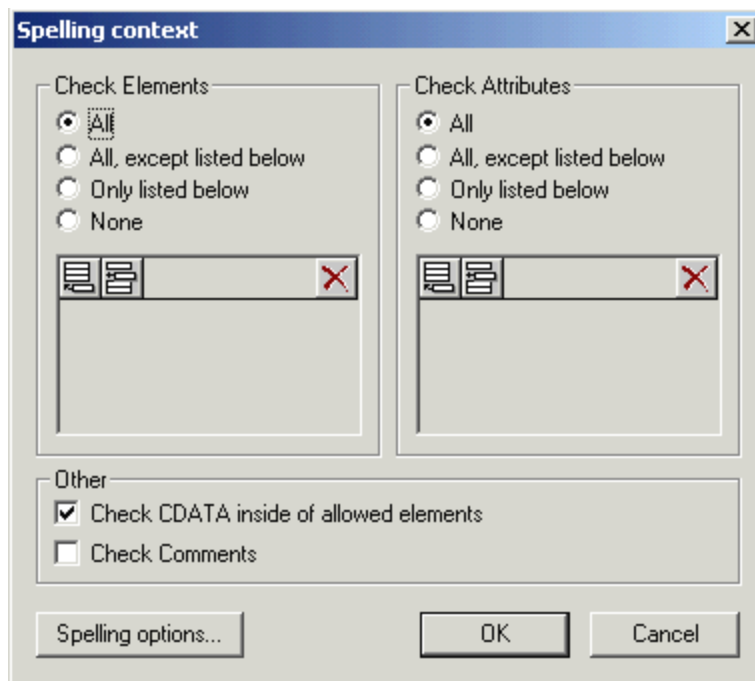
So entfernen Sie ein Wörterbuch:

- Klicken Sie im Dialogfeld "Benutzerdefinierte Wörterbücher" auf den Namen des Wörterbuchs und dann die Schaltfläche "Entfernen". Das Wörterbuch wird aus der Liste, aber nicht von der Festplatte gelöscht.

Dialogfeld Rechtschreibkontext - **Text**-und **Enhanced Grid-Ansicht**

Wenn Sie Ihr XML-Dokument in der Text- oder Enhanced Grid-Ansicht bearbeiten, können Sie mit diesem Befehl das Dialogfeld "Rechtschreibkontext" öffnen. Wenn Sie die Rechtschreibung in der **Text**- und **Enhanced Grid-Ansicht** starten, haben Sie die meisten Optionen zur Verfügung. So können Sie zum Beispiel folgende Inhalte auf Rechtschreibung überprüfen:

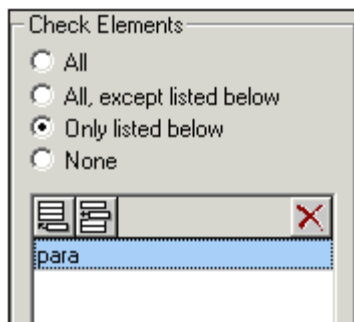
- Inhalte von Elementen
- Inhalte von Attributen
- Inhalte von CDaten
- Kommentartexte



Die Schaltfläche Rechtschreiboptionen... öffnet das Dialogfeld Einstellungen, wo Sie Einstellungen für die Rechtschreibprüfung definieren können.

So definieren Sie, welcher Inhalt von Attributen oder Elementen überprüft werden soll:

1. Klicken Sie auf das Optionsfeld "Alle außer unten aufgelistete" oder "Nur unten aufgelistete".
2. Klicken Sie auf das Symbol "Element/Attribut anhängen" , und geben sie den Tag-Namen ein, dessen Inhalt überprüft werden soll (z.B. para). Bestätigen Sie mit der Eingabetaste.



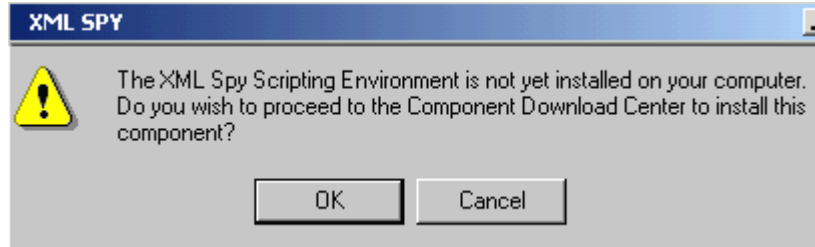
3. Klicken Sie auf die OK Schaltfläche um Ihre Einstellungen zu bestätigen, und bestätigen Sie diese nochmals im Dialogfeld "Rechtschreibung" um die Rechtschreibprüfung mit den neuen Einstellungen zu starten.
In diesem Fall wird **nur der Text innerhalb der Para-Tags**, sowie dessen Attribute auf Rechtschreibfehler überprüft.

So löschen Sie einen Tag-Namen:

- Wählen Sie den Tag-Namen, den Sie löschen möchten, und drücken Sie die Entfernen- oder die Rücktaste Ihrer Tastatur.

9.9.3 Skripting-Umgebung

Die Skripting-Umgebung ist derzeit eine downloadbare Komponente. Wenn Sie die Komponente noch nicht heruntergeladen und installiert haben, werden Sie dazu aufgefordert, sobald Sie diesen Menübefehl wählen.



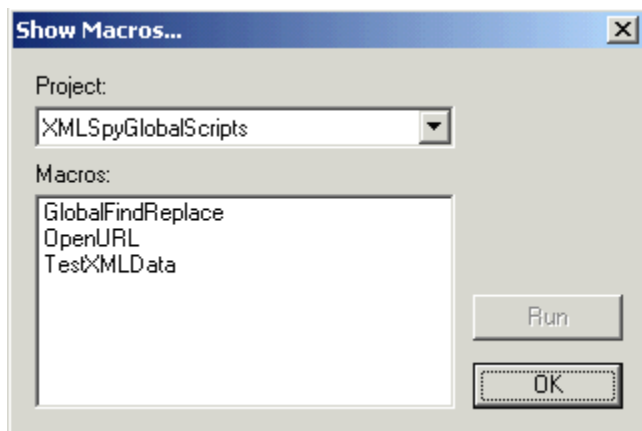
Klicken Sie auf OK um mit dem Download-Center auf Altovas Webseite zu verbunden zu werden. Hier wählen Sie dann die Komponente Skript-Umgebung.

Nachdem Sie die Komponente heruntergeladen haben, doppelklicken Sie auf **Spyscripting.exe** um diese zu installieren.

Die Scripting-Umgebung, ist ein vordefiniertes globales Skript-Projekt, und wird beim ersten Starten des Formular-Editors aktiviert. Weitere Informationen zu diesem Thema finden Sie im Skript-Abschnitt der vorliegenden Dokumentation.

9.9.4 Makros

Mit diesem Befehl wird das Dialogfeld mit der Liste aller Makros angezeigt, die mit dem globalen Skriptprojekt und mit dem aktuellen Projekt in assoziiert sind.



Die Schaltfläche "Starten" im Dialogfeld ruft das markierte Makro auf.

9.9.5 Projekt

Mit diesem Befehl können Sie Skripts einem bestimmten zuweisen (oder löschen).

Die Befehle der Untermenüs sind:

- *Skripts zuweisen*: Dient zum [Zuweisen](#) von Skripts zu einem Scripting-Projekt.
- *Projektskripts entkoppeln*: Damit [heben Sie die Zuweisung](#) von Skripts zu einem Scripting-Projekt auf.
- *Projekt-Skripts aktivieren*: Dient zum temporären Aktivieren oder Deaktivieren von Skripts.

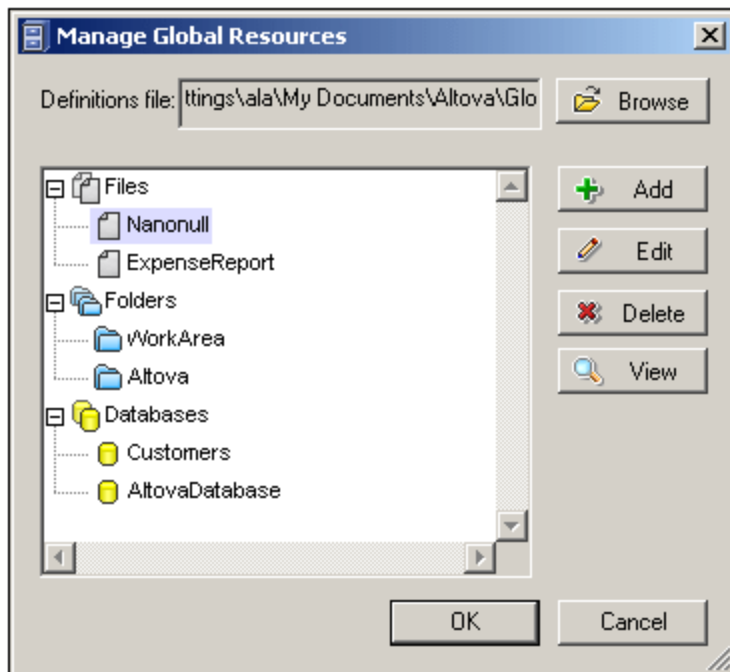
Bitte beachten Sie:

Diese Menüoptionen sind nur aufrufbar, wenn Sie ein Projekt in Authentic Desktop geöffnet haben.

9.9.6 Globale Ressourcen

Mit dem Befehl **Globale Ressourcen** wird das Dialogfeld "Globale Ressourcen" aufgerufen (*Abbildung unten*), in dem Sie Folgendes tun können:

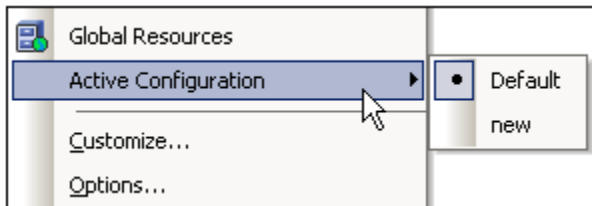
- Angabe der XML-Datei für globale Ressourcen.
- Hinzufügen von globalen Ressourcen (oder Aliassen) vom Typ Datei, Ordner und Datenbank
- Angabe verschiedener Konfigurationen für die einzelnen globalen Ressourcen (Aliasse). Jede Konfiguration wird einer bestimmten Ressource zugeordnet.



Eine Beschreibung, wie man globale Ressourcen definiert, finden Sie im Abschnitt

9.9.7 Aktive Konfiguration

Wenn Sie die Maus über den Menübefehl **Aktive Konfiguration** platzieren, wird ein Untermenü ausgerollt, das alle in der aktiven [XML-Datei für globale Ressourcen](#) definierten Konfigurationen enthält. (*Abbildung unten*).



Die gerade aktive Konfiguration ist durch einen Punkt gekennzeichnet. In der Abbildung oben ist die gerade aktive Konfiguration `Default`. Um die aktive Konfiguration zu wechseln, wählen Sie die Konfiguration aus, die aktiv gemacht werden soll.

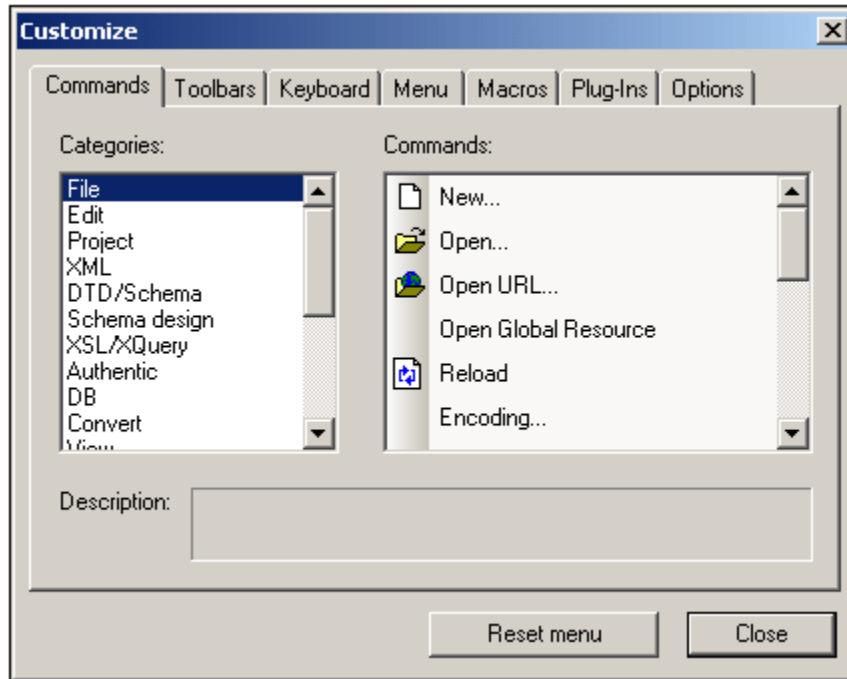
Anmerkung: Die aktive Konfiguration kann auch über die [Symbolleiste "Globale Ressourcen"](#) ausgewählt werden (**Ansicht | Anpassen | Symbolleisten | Globale Ressourcen**).

9.9.8 Anpassen

Mit diesem Befehl können Sie an Ihre eigenen Bedürfnisse anpassen.

Befehle

Auf dem Register Befehle können Sie Ihre Menüs oder Symbolleisten anpassen.



So fügen Sie einen Befehl zu einer Symbolleiste oder einem Menü hinzu:

1. Öffnen Sie das Dialogfeld über die Menübefehle **Extras | Anpassen**. Daraufhin wird das Dialogfeld "Anpassen" angezeigt.
 2. Wählen Sie die Befehlskategorie **Alle Befehle** im Kategorie-Listenfeld. Die verfügbaren Befehle werden im entsprechenden Listenfeld angezeigt.
 3. Markieren Sie einen Befehl aus dem Listenfeld, und ziehen Sie ihn in ein existierendes Menü oder eine Symbolleiste.
 4. Ein **I**-Zeichen wird angezeigt, wenn Sie den Cursor an eine Stelle bewegen, an die der ausgewählte Befehl gestellt werden kann.
 5. Lassen Sie die Maustaste los, um den Befehl an der gewünschten Stelle einzufügen.
- An der Spitze des Mauszeigers wird beim Ziehen eines Befehls eine kleine Schaltfläche angezeigt. Das Zeichen unterhalb des Zeigers zeigt an, dass der Befehl nicht an diese Position gestellt werden kann.
 - Das Zeichen verschwindet, sobald Sie eine Position erreichen, an die der Befehl gestellt werden kann (innerhalb eines Menüs oder einer Symbolleiste).
 - Wenn Sie den Mauszeiger über ein Menü halten, wird dieses geöffnet, so dass Sie den Befehl an einer beliebigen Stelle im Menü einfügen können.
 - Befehle können in Menüs oder Symbolleisten eingefügt werden. Wenn Sie eine eigene Symbolleiste erzeugt haben, können Sie nun die gewünschten Befehle einfügen.

Bitte beachten Sie:

Sie können mit dieser Methode auch die Befehle der [Kontextmenüs](#) bearbeiten (rechtsklicken Sie auf eine beliebige Stelle im Fenster, und das Kontextmenü wird

geöffnet). Klicken Sie auf das Register Menü und wählen Sie aus dem Dropdown-Listenfeld das gewünschte Kontextmenü aus.

So löschen Sie einen Befehl oder ein Menü:

1. Öffnen Sie dieses Dialogfeld über die Menübefehle **Extras | Anpassen**.
2. Klicken Sie auf den Menüeintrag oder das Symbol, welches Sie löschen möchten, und ziehen Sie es mit der Maus an eine Stelle, an der die kleine Schaltfläche am Zeiger angezeigt wird.
3. Lassen Sie die Maustaste los, und der Befehl wird entfernt.

Symbolleisten

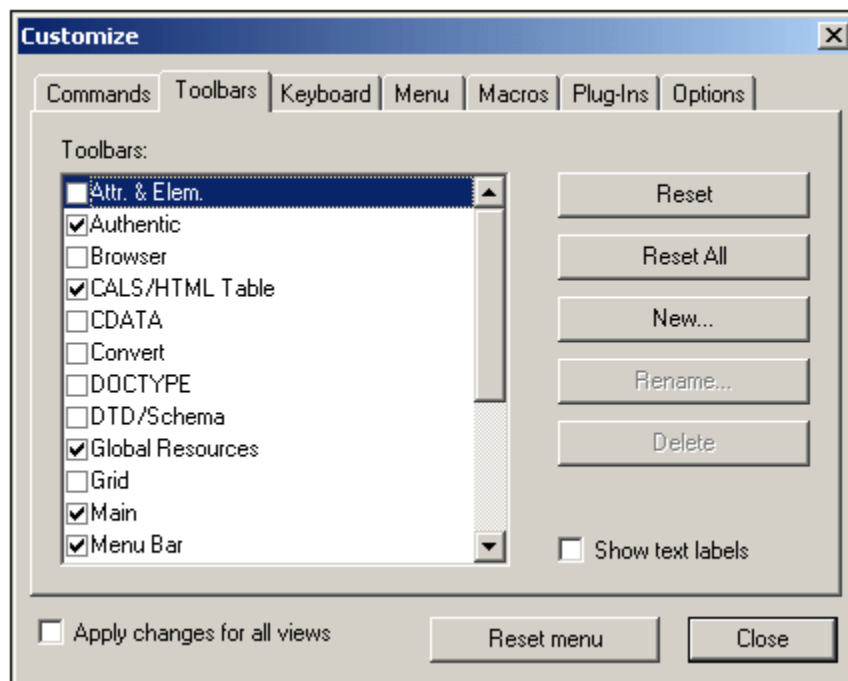
Auf dem Register Symbolleisten können Sie spezifische Symbolleisten aktivieren oder deaktivieren. Weiters können Sie eigene Symbolleisten erstellen.

Symbolleisten enthalten Symbole für die meistverwendeten Befehle.

Zu jedem Symbol gehört ein kurzer Tooltip, der angezeigt wird, wenn Sie mit dem Cursor auf das Symbol gehen. In der Statusleiste wird eine detailliertere Beschreibung zum jeweiligen Symbol angezeigt.

Sie können die Symbolleisten aus Ihrer Standardposition an eine andere beliebige Stelle am Bildschirm verschieben, sowie diese an den rechten oder linken Rand des Hauptfensters andocken.

- Symbolleisteneinstellungen, die in der Grid-, Schema/WSDL Design- oder Text-Ansicht definiert wurden, stehen in den jeweiligen Ansichten zur Verfügung. Die Symbolleisten der Browser-Ansicht sind unabhängig von der jeweiligen Ansicht immer verfügbar.



So aktivieren oder deaktivieren Sie eine Symbolleiste:

1. Klicken Sie auf das Kontrollkästchen um die jeweilige Symbolleiste zu aktivieren oder deaktivieren.

So erstellen Sie eine neue Symbolleiste:

1. Klicken Sie auf die Schaltfläche **Neu...** und geben Sie der neuen Symbolleiste einen Namen (Dialogfeld Symbolleistenname).
2. Fügen Sie Befehle in die neue Symbolleiste ein (Register [Befehle](#) im Dialogfeld **Anpassen**).

So setzen Sie die Menüleiste zurück

- Klicken Sie auf den Menüleisteneintrag und
- Klicken Sie auf die Schaltfläche **Zurücksetzen**, um die Menübefehle auf Werkseinstellung zurückzusetzen.

So setzen Sie alle Symbolleisten und Menübefehle zurück

- Klicken Sie auf die Schaltfläche **Alles Zurücksetzen**, um die Befehle der Symbolleisten auf Werkseinstellung zurückzusetzen. Sie erhalten eine Meldung, dass nun alle Symbolleisten und Menüs zurückgesetzt werden.
- Klicken Sie auf Ja um das Zurücksetzen zu bestätigen.

So ändern Sie den Namen einer Symbolleiste:

- Klicken Sie auf die Schaltfläche **Umbenennen...** um den Namen der Symbolleiste zu ändern.

So löschen Sie eine Symbolleiste:

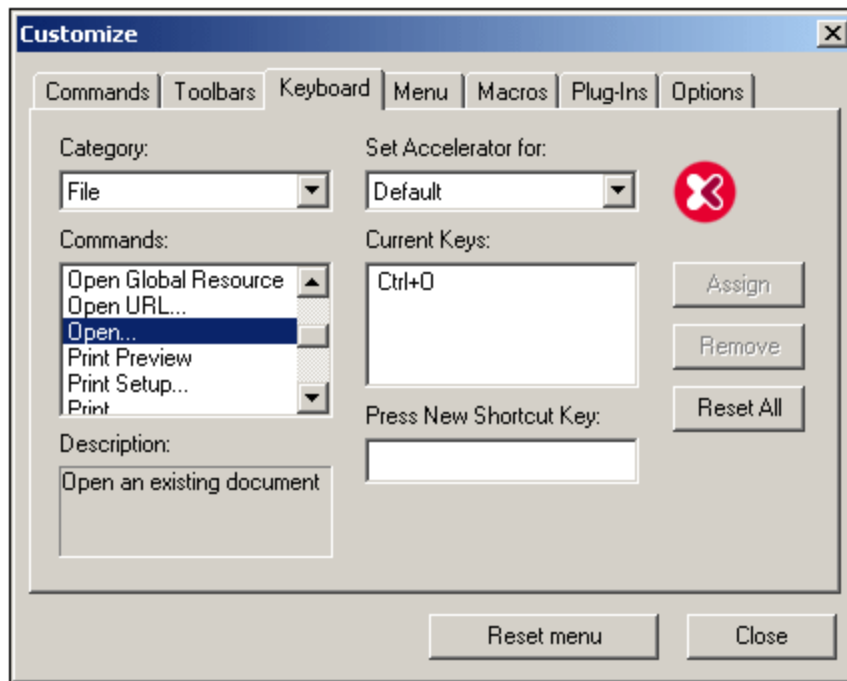
- Klicken Sie auf die Schaltfläche **Löschen** um die im Listefeld markierte Symbolleiste zu löschen.
- Sie erhalten eine Meldung, in der Sie gefragt werden, ob die markierte Symbolleiste tatsächlich gelöscht werden soll. Bestätigen Sie den Löschvorgang mit Ja.
- oder, rechtsklicken Sie auf die Menü- oder Symbolleiste und wählen Sie Löschen aus dem Popup-Menü.

Textlabels zeigen:

Wenn Sie diese Option aktivieren, werden Hilfetexte zu den Symbolen angezeigt.

Tastatur

Auf dem Register Tastatur können Sie Tastaturkürzel für jeden Befehl in definieren (oder löschen).



So weisen Sie einem Befehl ein neues Kürzel zu:

1. Wählen Sie die Befehlskategorie in der Auswahlliste **Kategorie**. Beachten Sie: Wenn [ein Makro als verknüpfter Befehl ausgewählt wurde](#), können über die Auswahlliste auch Makros ausgewählt werden und es kann für das Makro ein Tastenkürzel definiert werden.
2. Wählen Sie den **Befehl** aus dem Listefeld, welchem Sie ein neues Kürzel zuweisen möchten.
3. Klicken Sie in das "Tastaturkürzel eingeben:" Textfeld und geben Sie das Kürzel ein. Die Kürzel werden sofort im Textfeld angezeigt. Wenn das Kürzel schon einem anderen Befehl zugeordnet war, wird diese Funktion im Textfeld angezeigt.
4. Klicken Sie auf die Schaltfläche **Zuweisen**.
5. Das neue Kürzel wird in die Liste der aktuellen Kürzel aufgenommen.
(um das Textfeld zu **löschen** drücken Sie die Strg-, ALT- oder Umschalttaste).

So heben Sie eine Zuweisung auf (oder löschen ein Kürzel):

1. Klicken Sie auf den Kürzel in der Liste der aktuellen Kürzel, der gelöscht werden soll und
2. Klicken Sie auf die Schaltfläche **Entfernen**, die jetzt aktiv ist.
3. Klicken Sie auf die Schaltfläche Schließen um die Änderungen zu speichern.

Kürzel setzen für:

Derzeit noch keine Funktion.

Zugewiesene Tastatur-Kürzel:**Kürzel nach Tasten**

F1	Hilfe-Menü
F3	Weitersuchen
F5	Aktualisieren
F7	Wohlgeformtheit prüfen
F8	Validieren
F10	XSL Transformation
Strg+F10	XSL:FO Transformation
F11	Einsteigen
Strg+F11	Überspringen
Umschalt + F11	Aussteigen
Num +	Erweitern
Num -	Reduzieren
Num *	Alles erweitern
Strg+Num-	Unselektierte reduzieren
Strg + G	Gehe zu Zeichen/Zeile
Strg+TAB	Wechselt zwischen offenen Dokumenten
Strg+F6	Offene Fenster durchblättern
Pfeiltasten oben/unten	Laufleiste auf/ab
Esc.	Dialogfeld schließen/Editieren abbrechen
Eingabetaste/Leerzeilentaste	Bestätigen
Alt + F4	Schließt
Strg + F4	Schließt aktives Fenster
Alt + F, 1	Öffnet letzte Datei
Strg + Doppelklick Element (Schema-Ansicht)	Elementdefinition anzeigen
Strg + N	Datei Neu
Strg + O	Datei Öffnen
Strg + S	Datei Speichern
Strg + P	Datei Drucken
Strg + A	Alles auswählen
Umschalt + Del	Ausschneiden (oder Strg + X)
Strg + C	Kopieren
Strg + V	Einfügen
Strg + Z	Rückgängig
Strg + Y	Wiederherstellen
Del	Löschen (Objekt in der Schema-Ansicht/Grid Ansicht)

Strg + F	Suchen
F3	Weitersuchen
Strg + H	Ersetzen
Strg + I	Attribut anhängen
Strg + E	In der Grid-Ansicht: Element anhängen. In der Textansicht: Gehe zum Star/End-Tag, wenn sich der Cursor auf dem Gegenstück des Tags befindet
Strg + T	Text anhängen
Strg + D	CDATA anhängen
Strg + M	Comment anhängen
Strg + Umschalt + I	Attribut einfügen
Strg + Umschalt + E	Element einfügen
Strg + Umschalt + T	Text content einfügen
Strg + Umschalt + D	CDATA einfügen
Strg + Umschalt + M	Comment einfügen
Strg + ALT + I	Child Attribute hinzu
Strg + ALT + E	Child Element hinzu
Strg + ALT + T	Child Text hinzu
Strg + ALT + D	Child CDATA hinzu
Strg + ALT + M	Child Comment hinzu
Tastaturkürzel für die Textansicht	
Strg + "+"	Vergrößern
Strg + "-"	Verkleinern
Strg + 0	Zoom zurücksetzen
Strg + Mausrad nach vorne	Vergrößern
Strg + Mausrad nach hinten	Verkleinern

Kürzel nach Funktion

Editieren abbrechen	Esc.
Child Attribute hinzu	Strg + ALT + I
Child CDATA hinzu	Strg + ALT + D
Child Comment hinzu	Strg + ALT + M
Child Element hinzu	Strg + ALT + E
Child Text hinzu	Strg + ALT + T
Attribut anhängen	Strg + I
CDATA anhängen	Strg + D
Comment anhängen	Strg + M
Element anhängen	Strg + E (Grid-Ansicht)
Text anhängen	Strg + T
Wohlgeformtheit prüfen	F7
Aktives Fenster schließen	Strg + F4
Schließt	Alt + F4
Reduzieren	Num -
Unselektierte reduzieren	Strg + Num-
Bestätigen	Return / Space bar
Kopieren	Strg + C
Ausschneiden	Umschalt + Del (or Strg + X)
Offenen Fenster durchblättern	Strg + TAB und Strg + F6
Löschen	Del

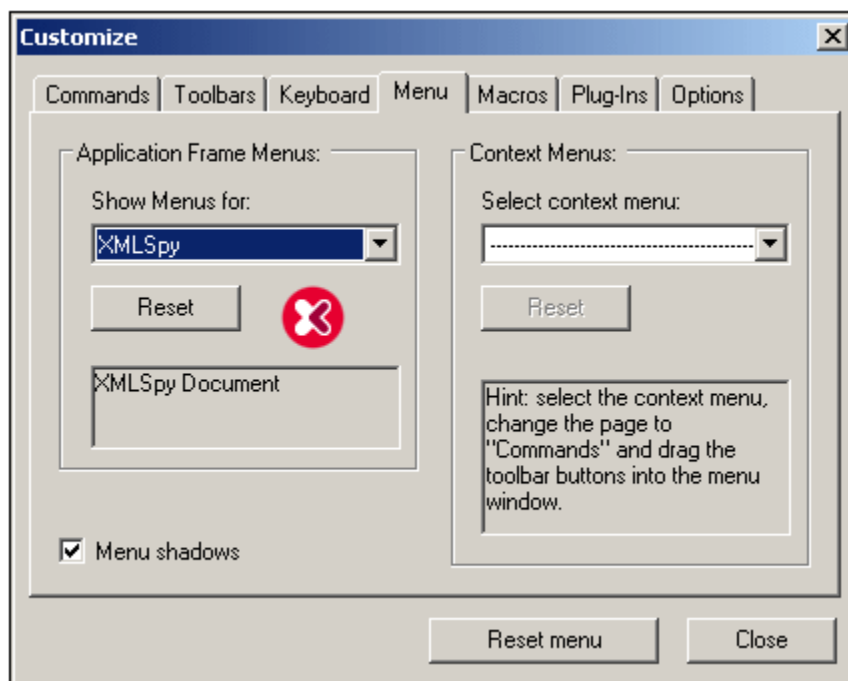
Erweitern	Num +
Alles erweitern	Num *
Datei Neu	Strg + N
Datei Öffnen	Strg + O
Datei Drucken	Strg + P
Datei Speichern	Strg + S
Suchen	Strg + F
Weitersuchen	F3
Gehe zu Zeichen/Zeile	Strg + G
Hilfemenü	F1
Den anderen Tag des Tag-Paars markieren, wenn sich der Cursor innerhalb eines Element-Start- oder End-Tags befindet	Strg + E (Textansicht)
Attribut einfügen	Strg + Umschalt + I
CDATA einfügen	Strg + Umschalt + D
Comment einfügen	Strg + Umschalt + M
Element einfügen	Strg + Umschalt + E
Breakpoint einfügen/löschen	F9
Tracepoint einfügen/löschen	Umschalt+F9
Text content einfügen	Strg + Umschalt + T
Laufleiste auf/ab	Pfeiltasten oben/unten
Letzte Datei öffnen	Alt + F, 1
Einfügen	Strg + V
Wiederherstellen	Strg + Y
Aktualisieren	F5
Ersetzen	Strg + H
Alles markieren	Strg + A
Start Debugger	Alt + F11
Einsteigen	F11

Aussteigen	Umschalt + F11
Überspringen	Strg + F11
Definition anzeigen	Strg + Doppelklick Element
Rückgängig	Strg + Z
Validieren	F8
XSL Transformation	F10
XSL:FO Transformation	Strg + F10

In der Applikation sehen Sie im Dialogfeld "Tastaturhilfe" ([Hilfe | Tastaturzuordnungen](#)) eine Liste der Befehle mit Ihren Tastaturkürzeln und Beschreibungen.

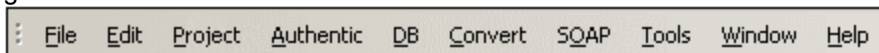
Menü

Auf dem Register Menü können Sie die Menüleisten und die Rechtsklick-Kontextmenüs anpassen.



Sie können sowohl die standardmäßigen Menüleisten, als auch die Menüleisten anpassen.

Das **Standardmenü** (Abbildung unten) wird angezeigt, wenn Sie kein XML-Dokument in geöffnet haben.



Das -Menü ist die Menüleiste, die dargestellt wird, wenn mindestens ein XML-Dokument geöffnet ist.

So passen Sie ein Menü an:

1. Wählen Sie die Menüleiste aus dem Dropdown-Listefeld "Menüs anzeigen für:", welche Sie anpassen wollen.
2. Klicken Sie auf das Register [Befehle](#) und ziehen Sie die gewünschten Befehle in das ausgewählte Menü.

So löschen Sie Befehle aus einem Menü:

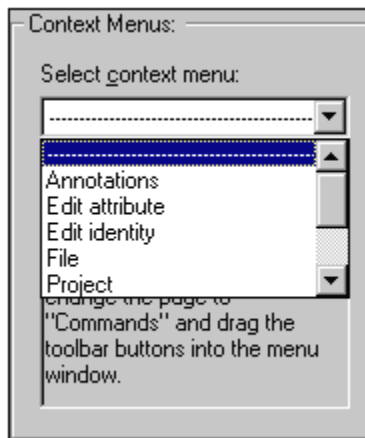
1. Rechtsklicken Sie bei geöffnetem Dialogfeld "Anpassen" auf den Befehl, oder das Symbol, welches für den Befehl steht, und
2. Wählen Sie die Option Löschen aus dem Popup Menü, oder
1. Wählen Sie **Extras | Anpassen** um das Dialogfeld Anpassen zu öffnen und
2. Ziehen Sie den Befehl aus dem Menü bis am Mauszeiger die kleine Schaltfläche angezeigt wird.

So setzen Sie eine der Menüleisten zurück:

1. Wählen Sie entweder den Eintrag Standard oder aus dem Dropdown-Listenfeld und
2. Klicken Sie auf die Schaltfläche **Zurücksetzen** unter dem Menünamen.
Mit einer Meldung werden Sie gefragt, ob Sie die Menüleiste tatsächlich zurücksetzen möchten.

So passen Sie ein Kontextmenü an (Rechtsklick-Menü):

1. Wählen Sie das entsprechende Kontextmenü aus dem Dropdown-Listenfeld.
2. Klicken Sie auf das Register **Befehle** und ziehen Sie die Befehle in das offene Kontextmenü.



So löschen Sie Befehle aus einem Kontextmenü:

1. Rechtsklicken Sie bei geöffnetem Dialogfeld "Anpassen" auf den Befehl, oder das Symbol, welches für den Befehl steht, und
2. Wählen Sie die Option Löschen aus dem Popup Menü, oder
- oder
1. Wählen Sie **Extras | Anpassen** um das Dialogfeld Anpassen zu öffnen und
2. Ziehen Sie den Befehl aus dem Menü heraus und lassen Sie die Maustaste los, sobald am Mauszeiger die kleine Schaltfläche angezeigt wird.

So setzen Sie eines der Kontextmenüs zurück:

1. Wählen Sie das Kontextmenü aus dem Dropdown-Listenfeld und
2. Klicken Sie auf die Schaltfläche **Zurücksetzen** unter dem Menünamen.
Mit einer Meldung werden Sie gefragt, ob Sie das Kontextmenü tatsächlich zurücksetzen möchten.

So schließen Sie ein Kontextmenüfenster:

1. Klicken Sie auf das Symbol **Schließen** rechts oben in der Titelleiste oder
2. Klicken Sie auf die Schaltfläche "Schließen" im Dialogfeld "Anpassen".

Menüanimationen (vor Windows 2000)

- Wählen Sie eine der Menüanimationen aus dem Dropdown-Listenfeld, wenn Sie

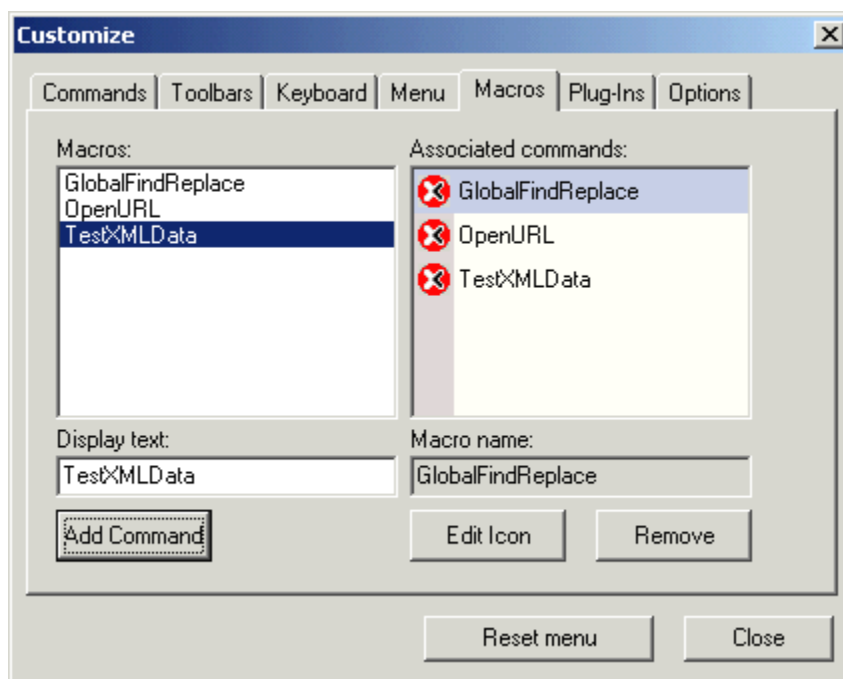
animierte Menüs sehen möchten. Bitte beachten Sie, dass das Dropdown-Listefeld nur für Windows-Versionen vor Windows 2000 verfügbar ist. Ab Windows 2000 können die Einstellungen auf dem Register Effekte im Dialogfeld "Eigenschaften von Anzeige" geändert werden. Doppelklicken Sie in der Systemsteuerung auf das Symbol **Anzeigen**, um das Dialogfeld zu öffnen.

Menüschatten

- Klicken Sie auf das Kontrollkästchen Menüschatten, wenn Menüs mit Schatten dargestellt werden sollen.

Makros

Auf dem Register Makros können Sie Makros, die Sie zuvor in der XML Skript-Umgebung erzeugt wurden, in eine Symbolleiste oder ein Menü einfügen.



So fügen Sie das Makrosymbol in eine Symbolleiste oder ein Menü ein:

1. Starten Sie die Skriptumgebung über **Extras | Skripting-Umgebung**.
2. Doppelklicken Sie auf den Eintrag XMLSpyMacros im Ordner Module des Projekts **XMLSpyGlobalScripts**.
Zuvor definierte Makros werden im rechten Fenster angezeigt.
3. Wechseln Sie zurück in und wählen Sie **Extras | Anpassen** und klicken Sie auf das Register Makros.
Die in der Skriptumgebung definierten Makros werden nun im Makros Listenfeld links angezeigt.
4. Klicken Sie auf den Namen des Makros und dann die Schaltfläche "Befehl hinzufügen".
Der Makroname wird im Listenfeld "Zugewiesene Befehle" angezeigt.
5. Klicken Sie im Listenfeld "Zugewiesene Befehle" auf den Namen des Makros und ziehen Sie ihn in die Symbolleiste oder das Menü.

So bearbeiten Sie ein Makrosymbol:

- Klicken Sie auf die Schaltfläche **Symbol bearbeiten**.

So löschen Sie ein Makro aus der Liste der zugewiesenen Befehle:

- Klicken Sie auf die Schaltfläche **Entfernen**.

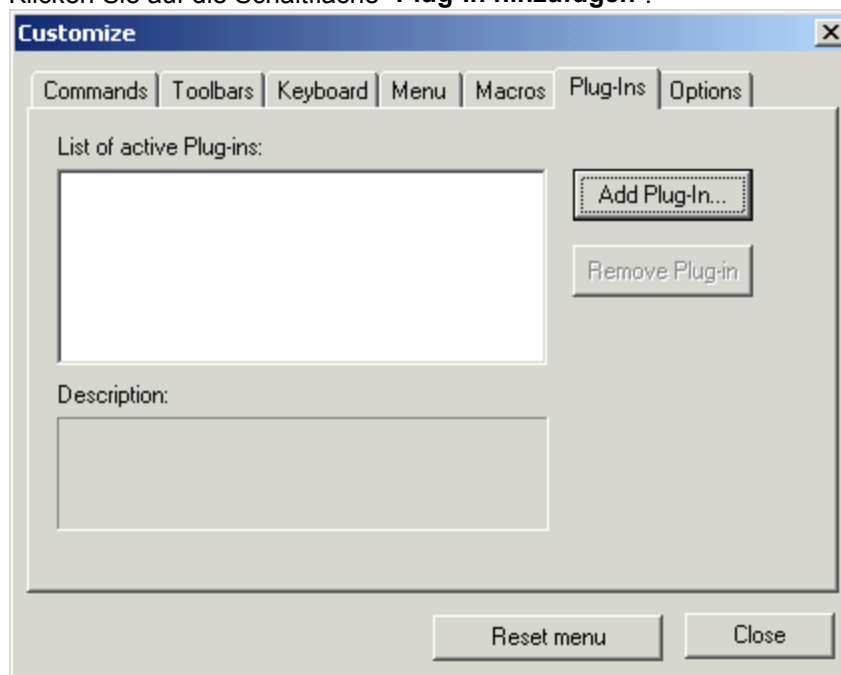
Anmerkung: Wenn ein Makro als "zugewiesener Befehl" ausgewählt wurde, können Sie dafür [ein Tastaturkürzel definieren](#). Wählen Sie auf dem Register "Tastatur" des Dialogfelds "Anpassen" in der Auswahlliste "Kategorie" die Kategorie "Makros" aus, klicken Sie auf das gewünschte Makro und definieren Sie das Tastenkürzel. Sie sollten ein Makro als "zugewiesenen Befehl" definieren, damit für Makros Tastenkürzel definiert werden können.

Plug-Ins

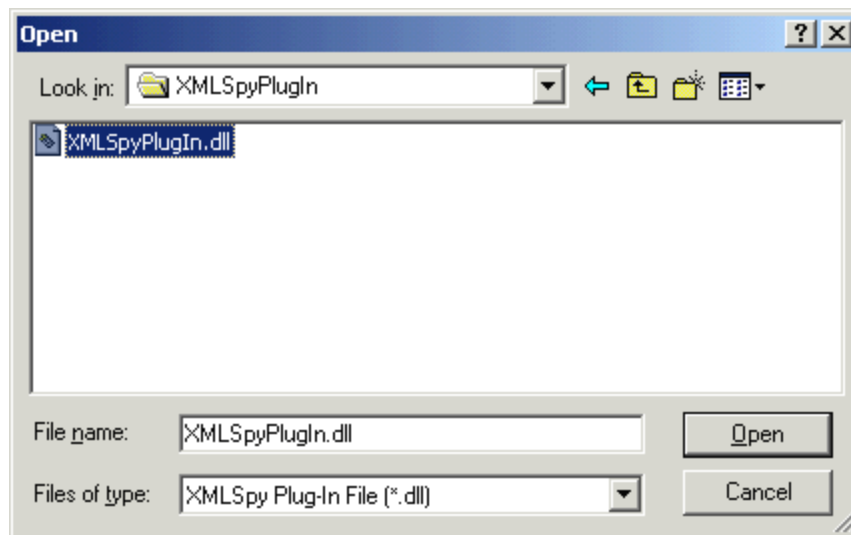
Auf dem Register Plug-Ins können Sie Plug-Ins in eine Symbolleiste oder ein Menü einfügen.

So fügen Sie das Symbol für ein Plug-In in eine Symbolleiste oder Menü ein:

1. Klicken Sie auf die Schaltfläche **"Plug-In hinzufügen"**.



2. Wählen Sie den Ordner aus und klicken Sie auf die Plug-In Datei um diese zu markieren (in diesem Beispiel: XMLSpyPlugIn.dll).



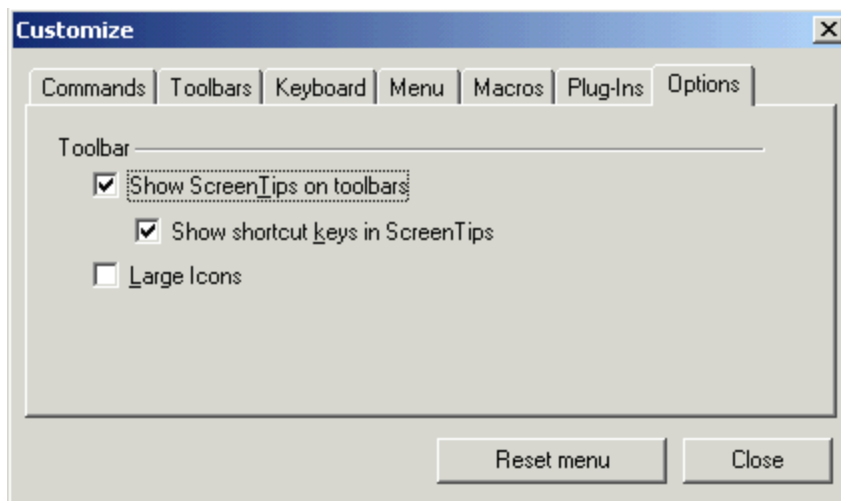
3. Klicken Sie auf die Schaltfläche **Öffnen** um das Plug-In zu installieren.
Der Name des Plug-Ins wird in der Liste "Derzeit aktive Plug-Ins" angezeigt und das Symbol des/der Plug-Ins wird in einer neuen Symbolleiste angezeigt.

So entfernen Sie ein Plug-In:

1. Klicken Sie auf den Namen des Plug-Ins in der Liste "Derzeit aktive Plug-Ins" und klicken Sie auf die Schaltfläche "Plug-In löschen".

Optionen

Auf dem Register Optionen können Sie verschiedene, allgemeine Einstellungen vornehmen.



Symbolleiste

Wenn das Kontrollkästchen **Tooltip in Symbolleiste einblenden** aktiviert ist, wird ein Popup beim Ziehen des Mauszeigers über das entsprechende Symbol in der Symbolleiste angezeigt. Das Popup enthält eine Kurzbeschreibung der Funktion, sowie, falls vorhanden, ein Tastaturkürzel.

Das Kontrollkästchen "Tastaturkürzel in Tooltips anzeigen" ermöglicht es Ihnen, auszuwählen, ob die Tastaturkürzel (wo vorhanden) angezeigt werden sollen.

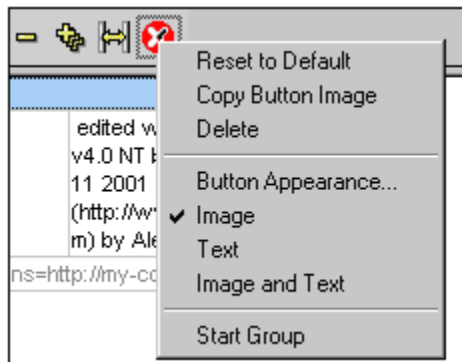
Mit dem Kontrollkästchen Große Symbole können Sie die Größe der angezeigten Symbole beeinflussen.

Kontextmenü 'Anpassen'

Über das Kontextmenü "Anpassen" können Sie Symbole und Menüobjekte Ihren Ansprüchen entsprechend anpassen.

So öffnen Sie das Kontextmenü "Anpassen":

1. Zuerst öffnen Sie das Menü Anpassen über den Menübefehl **Extras | Anpassen**.
2. Dann setzen Sie den Mauszeiger auf das Symbol (oder Menü) und **rechtsklicken Sie**, um das Kontextmenü zu öffnen.



Zurücksetzen

Derzeit keine Funktion.

Schaltflächensymbol kopieren

Das Symbol wird in die Zwischenablage kopiert.

Löschen

Mit dieser Option wird das Symbol oder Menü gelöscht, auf welches Sie zuvor rechtsgeklickt haben. Wenn Sie ein Menü löschen, werden alle enthaltenen Menüoptionen gelöscht!

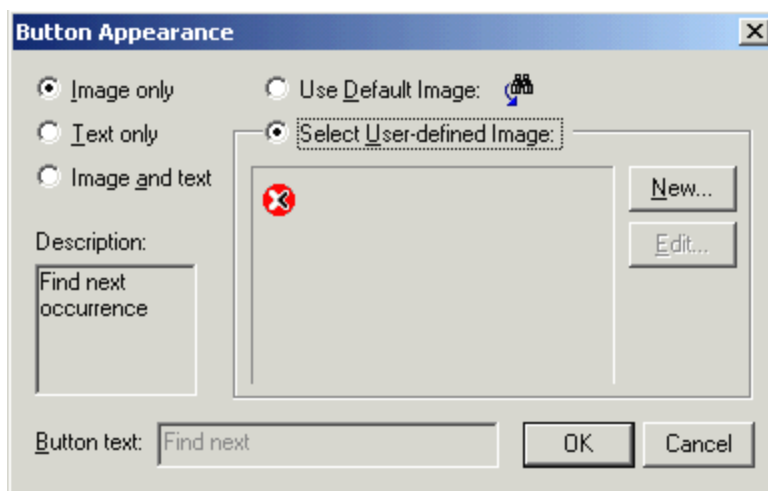
So stellen Sie ein gelöscht Menü wieder her:

- Wählen Sie das Register Menü
- Wählen Sie das Menü, welches wiederhergestellt werden soll (oder Standardmenü).
- Klicken Sie auf die Schaltfläche Zurücksetzen unter dem Menü im Dropdown-Listenfeld.

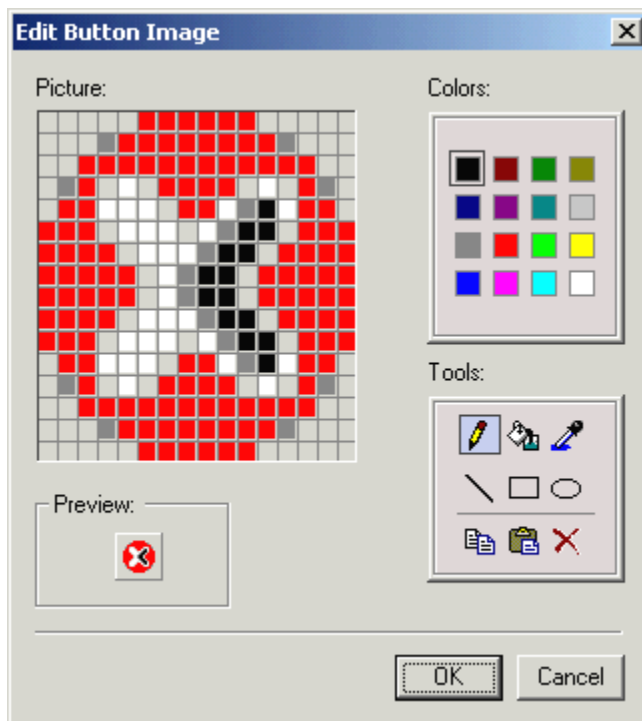
Schaltflächendarstellung

Mit dieser Option können Sie Schaltflächensymbole und Text bearbeiten. Derzeit kann nur das Symbol für Makros bearbeitet werden (Standardsymbol).

Mit den Optionsfeldern "Nur Symbol", "Nur Text" oder "Symbol und Text" können Sie definieren, was Sie editieren möchten. Klicken Sie auf das Optionsfeld **Benutzerdefiniertes Symbol** und klicken Sie auf eines der verfügbaren Symbole.



Klicken Sie auf die Schaltfläche "Bearbeiten" um das Dialogfeld **Schaltflächensymbol editieren** zu öffnen. Der Schaltflächentext kann bearbeitet werden, wenn Sie entweder die Option "Nur Text" oder "Symbol und Text" ausgewählt haben.



Symbol

Mit dieser Option ändern Sie das graphische Design des Symbols.

Text

Mit dieser Option ändern Sie die Textbeschreibung.

Symbol und Text

Mit dieser Option wird die gleichzeitige Darstellung des Symbols und des Textes aktiviert.

Gruppe starten

Mit dieser Option wird links neben dem Symbol, auf welches Sie rechtsgeklickt haben, ein

vertikaler Trennungsstrich angezeigt.

9.9.9 Optionen

Mit dem Befehl "Optionen" können Sie globale Programmeinstellungen in einem mit Registern versehenen Dialogfeld vornehmen. Diese Einstellungen werden in der Registry gespeichert und auf alle aktuellen und zukünftigen Dokumentfenster angewendet. Die Schaltfläche "Übernehmen" zeigt die Änderungen in den aktuell geöffneten Dokumenten an und speichert die Einstellungen. Die Änderungen sehen Sie sofort in den Fenstern im Hintergrund.

In diesem Abschnitt werden die einzelnen Register des Dialogfelds "Optionen" beschrieben.

Datei

Das Register "**Datei**" definiert die Art, wie Authentic Desktop Ihre XML-Dokumente öffnet und speichert. Weitere Einstellungen hierzu finden Sie auf dem Register [Kodierung](#).

Auto-laden von geänderten Dateien

Wenn Sie in einer Multi-User Umgebung, oder mit Dateien arbeiten, die von Servern dynamisch erstellt werden, können Sie die Änderungen während des Bearbeitens direkt verfolgen. Authentic Desktop wird Sie fragen, ob die Datei neu geladen werden soll, sobald eine Änderung in einem offenen Dokument entdeckt wurde.

Validierung

Wenn Sie Dokument Type Definitions (DTDs) oder Schemas zur Definition der Struktur Ihrer XML-Dokumente verwenden, können Sie das Dokument beim Öffnen oder Speichern automatisch auf Gültigkeit überprüfen lassen. Sie haben beim Speichern und Öffnen die Option, Dateien nur zu validieren, wenn die Dateigröße eine von Ihnen definierte Größe nicht übersteigt. Wenn das Dokument ungültig ist, wird eine Fehlermeldung angezeigt und die Operation wird ohne weitere Benachrichtigung fortgesetzt. Mit Authentic Desktop können Sie die Dateien auch in den Speicher cachieren, sodass unnötiges Neuladen entfällt (z.B. wenn das referenzierte Schema über eine URL aufgerufen wird). Falls in Ihrer schemaLocation-Deklaration eine URL verwendet wird, deaktivieren Sie die Option "DTD/Schema Dateien im Cachespeicher", um am Schema vorgenommene Änderungen sofort anzuzeigen und nicht die im Cache gespeicherte Version des Schemas zu verwenden.

Datei speichern

Beim Speichern eines XML-Dokuments fügt Authentic Desktop im oberen Bereich des Dokuments den kurzen Kommentar `<!-- Mit Authentic Desktop bearbeitet`
`http://www.altova.com -->` ein. Dieser Kommentar kann nur von lizenzierten Benutzern deaktiviert werden und wird beim Bearbeiten oder Speichern von Dateien in der Grid-Ansicht oder der Schema Design-Ansicht verwendet.

Wenn ein StyleVision Power Stylesheet mit einer XML-Datei verknüpft ist, wird der Link zum StyleVision Power Stylesheet bei Auswahl der Option 'Authentic: Link in Design-Datei speichern' mit der XML-Datei gespeichert.

Zeilenumbrüche

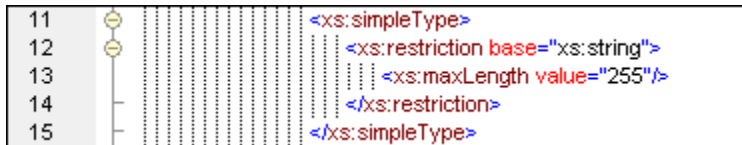
Wenn Sie eine Datei öffnen, werden die Zeichen für Zeilenumbrüche beibehalten, wenn Sie die Option **Alte beibehalten** gewählt haben. Alternativ dazu können Zeilenumbrüche in **CR&LF** für PCs, **CR** für MacOS, oder **LF** für Unix kodiert werden.

Keine Ausgabe-Formatierung für

In der Textansicht können Sie Child-Elemente aus Gründen der besseren Übersichtlichkeit einrücken (siehe **Datei speichern**). Sie können jedoch den definierten Einzug für einzelne Elemente auch außer Kraft setzen, indem Sie den Namen des Elements in das Feld **Keine Ausgabe-Formatierung für** eingeben. Alle in diesem Feld definierten Elemente werden ohne

Whitespaces zwischen aufeinanderfolgenden untergeordneten Elementen formatiert. (siehe *Abbildung*).

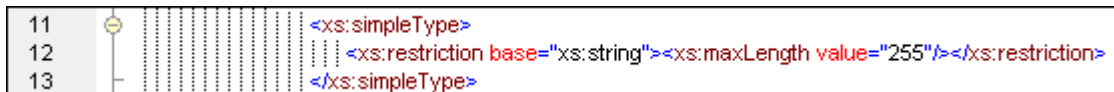
Hierarchische Einrückung für alle Elemente:



```

11  <xs:simpleType>
12    <xs:restriction base="xs:string">
13      <xs:maxLength value="255"/>
14    </xs:restriction>
15  </xs:simpleType>
  
```

Für das Element `xs:restriction` wurde die Einstellung **Keine Ausgabeformatierung** definiert.



```

11  <xs:simpleType>
12    <xs:restriction base="xs:string"><xs:maxLength value="255"/></xs:restriction>
13  </xs:simpleType>
  
```

Klicken Sie anschließend auf **OK**, um die Einstellung zu bestätigen und das Dialogfeld "Optionen" zu schließen.

Dateiarten

Auf dem Register "Dateiarten" können Sie für Authentic Desktop verschiedene Einstellungen für die einzelnen Dateiarten festlegen.

Wählen Sie eine Dateiart im Listenfeld "Dateiarten" aus, um folgende Funktionen für die ausgewählte Dateiart anzupassen:

Windows Explorer Einstellungen

Sie können die Beschreibung der Dateiart und den MIME-konformen Inhaltstyp, der von Windows Explorer verwendet wird, ändern. Weiters können Sie definieren, ob Authentic Desktop als der standardmäßige Editor für diese Dateiart verwendet werden soll.

XML Konformität

Authentic Desktop bietet für verschiedene Dateiarten spezifische Editierfunktionen und andere Funktionen, die Sie durch Einstellung der Konformität in dieser Option festlegen können. Für viele Dateiarten ist bereits eine standardmäßige Konformitätseinstellung festgelegt, die am besten beibehalten werden sollte, es sei denn Sie fügen eine neue Dateiart hinzu oder möchten für eine bestimmte Dateiart die Konformität explizit ändern.

Standard Ansicht

In dieser Gruppe definieren Sie, welche Standardansicht für welche Dateiart verwendet werden soll.

Textansicht

In diesem Textfeld können Sie die Syntaxfärbung für bestimmte Dateiarten bestimmen.

Automatische Validierung deaktivieren

Mit dieser Option können Sie für bestimmte Dateiarten die automatische Validierung deaktivieren. Die automatische Validierung wird normalerweise beim Öffnen oder Speichern einer Datei oder beim Wechseln der Ansicht durchgeführt.

Leere Elemente im Kurzformat "<E/>" speichern

Manche XML-Tools haben Probleme, die Kurzform `<Element/>` für leere Elemente zu verstehen (die in der XML 1.0 Spezifikation definiert wurde). Sie können Authentic Desktop anweisen, diese Elemente in der Langform zu speichern `<Element></Element>`, die gleichermaßen gültig ist.

Schaltfläche "Dateierweiterung hinzufügen"

Eine neue Dateierweiterung wird in die Liste der Dateiartern aufgenommen. Mit den Optionen in diesem Dialogfeld definieren Sie die Einstellungen für die neue Dateiartern.

Schaltfläche "Markierte Dateierweiterung löschen"

Löscht die Dateiartern und die dazugehörigen Einstellungen.

Klicken Sie anschließend auf **OK**, um die Einstellung zu bestätigen und das Dialogfeld "Optionen" zu schließen.

Ansicht

Auf dem Register "Ansicht" können Sie die Darstellung der XML-Dokumente in Authentic anpassen.

Pretty-Print

Bei Auswahl des Befehls **Bearbeiten | Pretty-Print** wird das Dokument mit Pretty-Print formatiert und zwar mit oder ohne Einrückungen, je nachdem, ob die Option "Einrückungen verwenden" in diesem Dialogfeld aktiviert ist oder nicht. Wird ein Pretty-Print mit Einrückungen vorgenommen, werden die Einrückungen gemäß den Einstellungen im Bereich "Tabulatoren" des Dialogfelds "Einstellungen für Textansicht" vorgenommen. Bei Pretty-Print wird jede Zeile im Dokument mit Null-Einrückung angezeigt.

Programmlogo

Das Programmlogo beim Starten kann deaktiviert werden, um die Applikation schneller starten zu können.

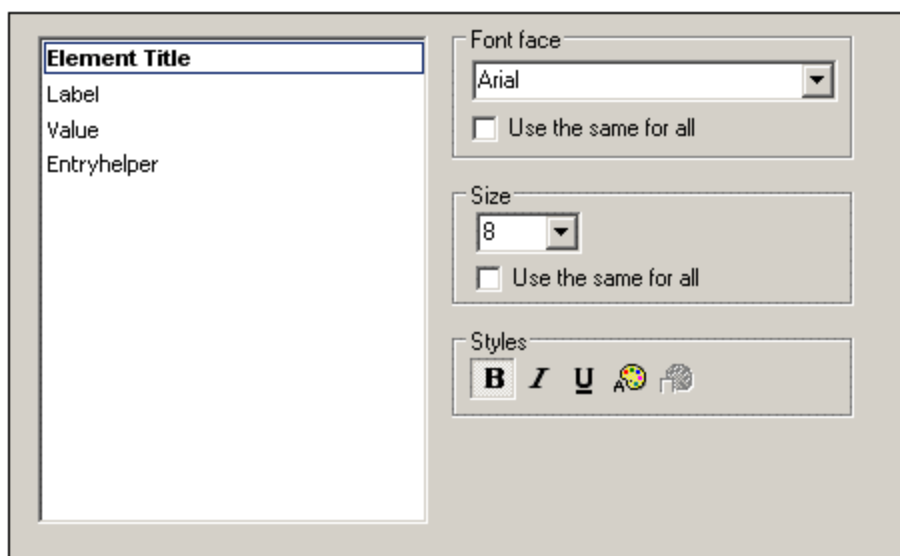
Titelleiste-Inhalt

Im Fenstertitel jedes Dokumentfensters kann entweder nur der Dateiname oder der komplette Pfad angezeigt werden.

Klicken Sie anschließend auf **OK** um die Einstellungen zu bestätigen und das Dialogfeld zu schließen.

XBRL-Schriftarten

Auf dem Register **XMBRL-Schritarten** wird das Aussehen von Text in der XBRL-Ansicht konfiguriert.



Schriftart

Sie können die Schriftart und die Schriftgröße auswählen, in der die verschiedenen Objekte in der XBRL-Ansicht dargestellt werden. Dieselben Schriftarten werden auch beim Drucken und Erstellen von Dokumentation verwendet, daher sollten nur True Type-Schriftarten verwendet werden.

Größe

Wählen Sie die gewünschte Größe aus. Wenn Sie dieselbe Größe für alle Einträge verwenden möchten, klicken Sie auf das Kontrollkästchen *Auf alle anwenden*.

Stile

Sie können über die Optionen in diesem Bereich Stile und Farben auswählen. Die aktuellen Einstellungen werden sofort in der Liste auf der linken Seite angezeigt, so dass Sie sehen, wie Ihr Dokument aussehen wird.

Klicken Sie anschließend auf **OK** um die Einstellungen zu bestätigen und das Dialogfeld zu schließen.

Kodierung

Auf dem Register **Kodierung** definieren Sie die Einstellungen für die Dateikodierung.

Standardkodierung für neue XML-Dateien

Die Standardkodierung für neue Dateien kann im Dialogfeld "Optionen" durch Auswahl einer Option aus der Dropdown-Liste definiert werden, sodass jedes neue Dokument automatisch mit der entsprechenden XML-Deklaration und der hier gewählten Kodierungsspezifikation erstellt wird. Wird eine 2- oder 4-Byte-Kodierung als standardmäßige Kodierung ausgewählt (z.B. UTF-16, UCS-2 oder UCS-4) können Sie weiters wählen, ob die Byteordnung für XML-Dokumente Little Endian oder Big Endian sein soll.

Die Kodierung für existierende Dateien wird immer beibehalten und kann nur mit dem Befehl [Datei | Kodierung](#) geändert werden.

Dateien mit unbekannter Kodierung öffnen als

Sie können in dieser Auswahlliste auswählen, mit welcher Kodierung eine Datei geöffnet werden soll, die keine Kodierungsspezifikation enthält oder deren Kodierung nicht automatisch bestimmt werden konnte.

Nicht-XML-Dateien öffnen in

In dieser Auswahlliste können Sie auswählen, welche Kodierung für nicht XML-Dateien verwendet werden soll. Die können die Kodierung des Dokuments mit dem Befehl [Datei | Kodierung](#) ändern.

BOM (Byte Order Mark)

Bei Speicherung eines Dokuments mit einer 2-Byte- oder einer 4-Byte-Zeichenkodierung kann es entweder mit (i) der Bytefolge Little Endian und einer Little Endian BOM gespeichert werden (*BOM immer erstellen, wenn nicht UTF-8*); oder (ii) mit der ermittelten Bytefolge und der ermittelten BOM (*Festgestellten BOM beim Speichern beibehalten*).

Klicken Sie anschließend auf **OK** um die Einstellungen zu bestätigen und das Dialogfeld zu schließen.

XSL

Auf dem Register **XSL** (*Abbildung unten*) werden die Optionen für [XSLT-Transformationen](#) und über die Applikation ausgeführte [XSL-FO-Transformationen](#) definiert.

XSLT-Transformationen

Authentic Desktop enthält den Altova XSLT 1.0-Prozessor und den Altova XSLT 2.0-Prozessor, die Sie für Ihre XSLT-Transformationen verwenden können. Je nach dem Wert des `version`-Attributs im Element `xsl:stylesheet` oder `xsl:transform` wird immer der jeweilige XSLT-Prozessor für Transformationen verwendet.

Zur Transformation von XML-Dokumenten mittels XSLT können Sie die folgenden Prozessoren verwenden:

- den integrierten Altova XSLT-Prozessor (einschließlich Altova XSLT 1.0- und Altova XSLT 2.0-Prozessor).
- den MSXML 3.0-, 4.0- oder 6.0-Parser (welcher bereits vorinstalliert ist). Wenn Sie wissen, welche Version des MSXML Parsers auf Ihrem Rechner läuft, könnten Sie ihn auswählen. Andernfalls wird die Version automatisch von der Applikation ausgewählt. (Die Option *Version automatisch auswählen* ist standardmäßig aktiv.) In diesem Fall versucht die Applikation, die neueste verfügbare Version auszuwählen.

- einen externen XSLT-Prozessor Ihrer Wahl. Sie müssen den Befehlszeilenstring für den externen XSLT-Prozessor definieren. Dazu stehen die folgenden Variablen zur Verfügung:

%1 = zu verarbeitendes XML-Dokument

%2 = zu generierende Ausgabedatei

%3 = zu verwendendes XSLT Stylesheet (wenn das XML-Dokument keine Referenz auf ein Stylesheet enthält)

Der Befehl zur Ausführung einer einfachen Transformation mit dem Saxon (XSLT 1.0) Prozessor lautet z.B.:

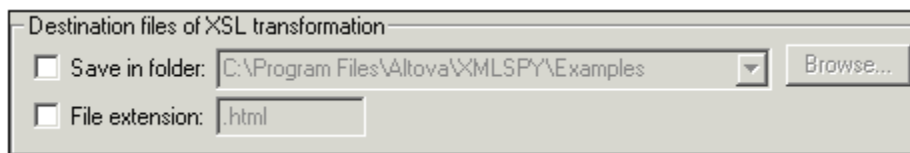
```
saxon.exe -o output.xml input.xml stylesheet.xslt
parameter-name=parameter-value
```

Um diesen Befehl von der Applikation aus zu starten, aktivieren Sie das Optionsfeld "Externes XSL-Transformationsprogramm" und geben Sie im Textfeld die folgende Zeile ein:

```
c:\saxon\saxon.exe -o %2 %1 %3 parameter-name=parameter-value
```

Aktivieren Sie die entsprechenden Kontrollkästchen, um die Ausgabe- und Fehlermeldungen des externen Programms im Fenster "Meldungen" in Authentic Desktop anzuzeigen.

- Bei Auswahl der Option *Ausgabefenster wiederverwenden* wird das Ergebnisdokument bei nachfolgenden Transformationen im selben Ausgabefenster angezeigt. Wenn die XML-Datei zu einem Projekt gehört und die Option *Ausgabefenster wiederverwenden* deaktiviert ist, wird diese Einstellung nur dann aktiv, wenn auch der unter dem Feld *In Ordner speichern* definierte Ausgabedateipfad (*Abbildung unten*) in den entsprechenden [Projekteigenschaften](#) ebenfalls deaktiviert ist.



XSL-FO-Transformationen

FO-Dokumente werden mit Hilfe des FO-Prozessors verarbeitet. Der Pfad zur ausführbaren Datei des FO-Prozessors muss im Textfeld für den XSL-FO-Transformationsprozessor definiert werden. Die Transformation wird mit Hilfe des Menübefehls [XSL/XQuery | XSL-FO-Transformation](#) ausgeführt. Wenn es sich bei der Quelldatei (dem Dokument, das bei Ausführung des Befehls in der IDE aktiv ist) um ein XSL-FO-Dokument handelt, wird für die Transformation der FO-Prozessor aufgerufen. Wenn es sich beim Quelldokument um ein XML-Dokument handelt, muss das XML-Dokument zuerst mittels einer XSLT-Transformation in ein XSL-FO-Dokument konvertiert werden. Diese XSLT-Transformation kann entweder von dem für die Applikation als Standardprozessor definierten XSLT-Prozessor ausgeführt werden ([siehe oben](#)) oder vom XSLT-Prozessor, der möglicherweise bereits in den FO-Prozessor integriert ist, den Sie als Standard-FO-Prozessor für die Applikation definiert haben. Klicken Sie zur Auswahl zwischen den beiden Optionen auf das entsprechende Optionsfeld.

Klicken Sie anschließend auf **OK** um die Einstellungen zu bestätigen und das Dialogfeld zu schließen.

Skripting-Umgebung

Auf dem Register **Skripting-Umgebung** können Sie die [Skripting-Umgebung](#) beim Start der Applikation aktivieren sowie weitere spezifische Skriptparameter definieren.

Skripting-Umgebung:

Definieren Sie, ob die Skriptumgebung beim Starten von aktiviert werden soll. Klicken Sie auf die Schaltfläche "Durchsuchen...", um die Skripting-Projektdatei auszuwählen.

Standard Skriptsprache:

Wählen Sie die Skriptsprache, die verwendet werden soll: JavaScript oder VBScript.

Automatische Skriptverarbeitung:

In diesem Abschnitt definieren Sie, ob automatische Makros laufen dürfen, und ob die Abarbeitung von Events erlaubt sein soll.

Klicken Sie anschließend auf **OK** um die Einstellungen zu bestätigen und das Dialogfeld zu schließen.

Versionskontrolle

Auf dem Register **Versionskontrolle** (*Abbildung unten*) können Sie die Standard-Anmelde-IDs für verschiedene Versionskontrollsysteme definieren und die Benutzereinstellungen konfigurieren.

Versionskontroll-Plugin

Das Versionskontroll-Plugin kann aus der Dropdown-Liste der Auswahlliste ausgewählt werden. In dieser Liste werden die installierten Versionskontroll-Plugins aufgelistet. Definieren Sie nach Auswahl der gewünschten Versionskontrolle im nächsten Textfeld die Anmelde-ID dafür. Über die Schaltfläche **Erweitert** rufen Sie ein Dialogfeld zum jeweiligen Versionskontroll-Plugin auf, in dem Sie die Einstellungen für dieses Plugin definieren können. Diese Einstellungen unterscheiden sich je nach Plugin.

Benutzereinstellungen

Es steht eine Reihe von Benutzereinstellungen zur Verfügung, darunter die folgenden:

- Statusaktualisierungen können im Hintergrund durchgeführt werden.
- Dateien können beim Öffnen und Schließen von Projekten aus- bzw. eingchecked werden.
- Die Anzeige der Dialogfelder "Einchecken" und "Auschecken" kann unterdrückt werden.
- Die Schaltfläche **"Zurücksetzen"** steht zur Verfügung, wenn Sie in einem der Dialogfelder die Option *"Wenn Dialogfelder mit..."* aktiviert haben. Wenn Sie auf die Schaltfläche **Zurücksetzen** klicken, wird diese Option wieder aktiviert.

Klicken Sie anschließend auf **OK** um die Einstellungen zu bestätigen und das Dialogfeld zu schließen.

9.10 Menü "Fenster"

Mit Hilfe dieses Menüs können Sie die Dokumentfenster für Ihre Sitzung anordnen. Das Menü "Fenster" enthält Standardbefehle, die aus den meisten Windows-Applikationen bekannt sind.

Sie können die offenen Dokumente hintereinander oder nebeneinander anzeigen. Weiters können Sie Dokumentsymbole neu anordnen (nachdem Sie diese minimiert haben) und die verschiedenen Eingabehilfenfenster ein- oder ausblenden oder direkt über das Menü in ein beliebiges offenes Dokument-Fenster wechseln.

9.10.1 Überlappend

Mit diesem Befehl arrangieren Sie alle offenen Dokumentfenster, sodass sie übereinander und überlappend angezeigt werden.

9.10.2 Horizontal anordnen

Mit diesem Befehl arrangieren Sie alle offenen Dokumente **horizontal nebeneinander**, sodass alle Dokumentfenster gleichzeitig sichtbar sind.

9.10.3 Vertikal anordnen

Mit diesem Befehl arrangieren Sie alle offenen Dokumente **vertikal nebeneinander**, sodass alle Dokumentfenster gleichzeitig sichtbar sind.

9.10.4 Projekt-Fenster

Mit diesem Befehl schalten Sie das Projekt-Fenster ein oder aus.

Dieses Fenster ist abdockbar, d.h. wenn Sie auf die Titelleiste klicken, können Sie das Fenster an jede beliebige Stelle verschieben. Wenn Sie mit der rechten Maustaste auf die Titelleiste klicken, können Sie das Fenster andocken oder ausblenden.

9.10.5 Info-Fenster

Mit diesem Befehl schalten Sie das Info-Fenster ein oder aus.

Dieses Fenster ist andockbar, d.h. wenn Sie auf die Titelleiste klicken, können Sie das Fenster an jede beliebige Stelle verschieben. Mit Rechtsklick docken Sie das Fenster wieder an, oder verbergen es.

9.10.6 Eingabehilfen

Mit diesem Befehl blenden Sie alle drei Eingabehilfe-Fenster ein oder aus.

Dieses Fenster ist andockbar, d.h. wenn Sie auf die Titelleiste klicken können Sie das Fenster an jede beliebige Stelle verschieben. Mit Rechtsklick docken Sie das Fenster wieder an, oder verbergen es.

9.10.7 Ausgabefenster

Bei den Ausgabefenstern handelt es sich um eine Gruppe von drei mit Registern versehenen Ausgabebereichen: (i) Validierungsergebnisse; (ii) Find in Files; und (iii) Ergebnisse der XPath-Auswertung. In der Anfangseinstellung werden sie unterhalb des Hauptfensters geöffnet. Mit dem Befehl "Ausgabefenster" können Sie die Ausgabefenster ein- oder ausblenden.

Das Fenster für die Ausgabebereich kann an- oder abgedockt werden. Wenn Sie die Titelleiste ziehen, können Sie es verschieben und als freischwebendes Fenster anordnen. Um das Fenster anzudocken oder auszublenden, klicken Sie mit der rechten Maustaste auf die Titelleiste.

Eine ausführliche Beschreibung der Ausgabefenster finden Sie im Abschnitt "Text-Ansicht" unter Ausgabefenster.

9.10.8 Projekt- und Eingabehilfen

Mit diesem Befehl können Sie das Projektfenster und die Eingabehilfen gleichzeitig ein- oder ausblenden.

9.10.9 Alle ein/aus



Mit diesem Befehl werden alle andockbaren Fenster zusammen ein- oder ausgeschaltet:

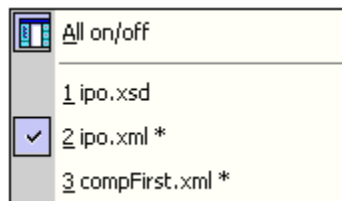
Info-Fenster

- alle drei Eingabehilfen-Fenster

Diese Funktion ist besonders hilfreich, wenn Sie schnell alle anderen Fenster ausblenden möchten, um Ihr Dokument in maximaler Größe anzeigen zu können.

9.10.10 Liste aller offenen Fenster

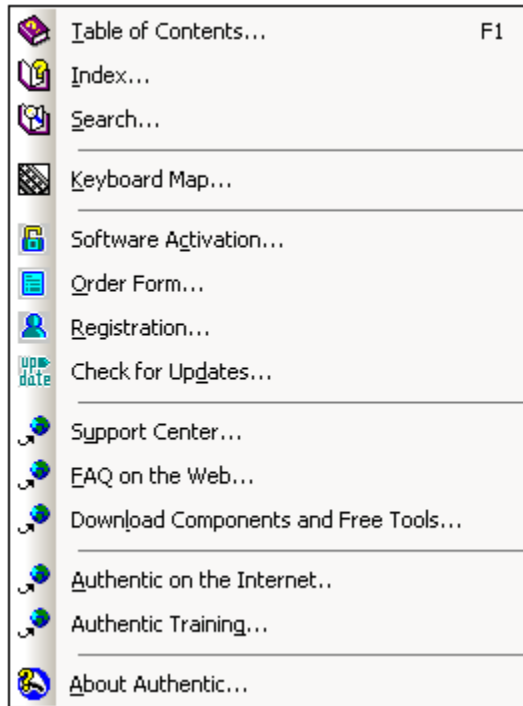
In dieser Liste werden alle derzeit offenen Fenster angezeigt. So können Sie schnell von einem Dokument zum anderen wechseln.



Hierzu können Sie auch die Tastaturkürzel Strg-TAB oder Strg F6 verwenden.

9.11 Menü "Hilfe"

Im Hilfemenü finden Sie alle Befehle, die Sie benötigen, um weitere Informationen zu spezifischen Themen von Authentic Desktop zu erhalten, sowie Links zu Informationen und Supportseiten auf unserem Webserver.



Im Hilfemenü finden Sie weiters den [Registrierungsdialog](#), über den Sie Ihren Keycode eingeben können, sobald Sie das Produkt erworben haben.

9.11.1 Inhaltsverzeichnis

Mit diesem Befehl erhalten Sie eine **hierarchische Darstellung** aller Kapitel und Themen, die in der Onlinehilfe enthalten sind. Verwenden Sie diesen Befehl, um direkt von Authentic Desktop in das Inhaltsverzeichnis zu gelangen.

Sobald das Hilfefenster geöffnet wurde, können Sie zwischen dem Inhaltsverzeichnis, dem [Index](#), und der [Suchfunktion](#) wählen. Auf dem Register "Favoriten" können Sie Lesezeichen zu Kapiteln innerhalb der Hilfe setzen.

9.11.2 Index

Mit diesem Befehl gelangen Sie in das **Stichwortverzeichnis** der Onlinehilfe. Sie können auch das Register Index in der Onlinehilfe verwenden.

Der Index bietet eine Liste von Stichwörtern, mit denen Sie durch Doppelklick zu den jeweiligen Abschnitten gelangen, die das Stichwort behandeln. Wenn es mehrere Abschnitte oder Kapitel gibt, in denen das Stichwort vorkommt, werden Sie aufgefordert, aus einer Liste das gewünschte Kapitel auszuwählen.

9.11.3 Suche

Mit dem Suchbefehl können Sie eine **Volltextsuche** in der gesamten Onlinehilfe starten.

- Geben Sie den Suchbegriff im Suchfeld ein und drücken Sie die Eingabetaste. Die Onlinehilfe zeigt eine Liste von Kapiteln an, in denen der Suchbegriff vorkommt.
- Doppelklicken Sie auf einen Eintrag in der Liste, um das entsprechende Kapitel aufzurufen.

9.11.4 Tastaturzuordnungen

Bei Auswahl des Befehls **Tastenzuordnungen...** wird eine Infobox angezeigt, in der alle Befehle in Authentic Desktop nach Menüs geordnet angezeigt werden. Die Menübefehle werden mit einer Beschreibung und dem entsprechenden Tastaturkürzel für den jeweiligen Befehl angezeigt.

Um die Befehle eines bestimmten Menüs anzuzeigen, wählen Sie den Menünamen in der Dropdown-Liste "Kategorie" aus. Durch Klicken auf das Druckersymbol können Sie den Befehl ausdrucken.

9.11.5 Aktivierung, Bestellformular, Registrieren, Updates

Software-Aktivierung

Nachdem Sie Ihre Altova-Software heruntergeladen haben, können Sie sie entweder mit Hilfe eines kostenlosen Evaluierungs-Keycode oder eines käuflich erworbenen permanenten Lizenzkeycode aktivieren.

- **Kostenloser Evaluierungs-Keycode.** Wenn Sie die Software zum ersten Mal starten, wird das Dialogfeld "Software-Aktivierung" angezeigt. Es enthält eine Schaltfläche, über die Sie einen kostenlosen Evaluierungs-Keycode anfordern können. Geben Sie in das Dialogfeld, das daraufhin angezeigt wird, Ihren Namen, den Namen Ihrer Firma und Ihre E-Mail-Adresse ein und klicken Sie auf "Jetzt anfordern!". Der Evaluierungs-Keycode wird an die von Ihnen angegebene E-Mail-Adresse gesendet und sollte in wenigen Minuten eintreffen. Geben Sie den Schlüssel nun im Dialogfeld "Software-Aktivierung" in das Feld Key-Code ein und klicken Sie auf OK, um Ihr Altova-Produkt zu aktivieren. Die Software wird für einen Zeitraum von 30 Tagen aktiviert.
- **Permanenter Lizenz Keycode.** Das Dialogfeld "Software-Aktivierung" enthält eine Schaltfläche, über die Sie einen permanenten Lizenz-Keycode erwerben können. Wenn Sie auf diese Schaltfläche klicken, gelangen Sie zum Altova Online Shop, wo Sie einen permanenten Lizenz-Keycode für Ihr Produkt erwerben können. Bei Ihrem permanenten Lizenz-Keycode handelt es sich entweder um eine Einzelbenutzerlizenz oder um eine Mehrfachlizenz. Beide Arten werden per E-Mail zugesandt. Eine *Einzelbenutzerlizenz* enthält Ihre Lizenzdaten, Ihren Namen, die Firma, E-Mail-Adresse und Keycode. Eine *Mehrfachlizenz* enthält Ihre Lizenzdaten, ihren Firmennamen und den Keycode. Beachten Sie, dass gemäß Ihrer Lizenzvereinbarung auf den Computern Ihres Unternehmens nur so viele Versionen der Altova-Software installiert werden dürfen, wie lizenziert wurden (Mehrplatzlizenz)

Anmerkung: Stellen Sie bitte sicher, dass die von Ihnen in das Registrierungsdialogfeld eingegebenen Daten genau mit denen in Ihrer Lizenz-E-Mail übereinstimmen. Geben Sie außerdem bei Mehrplatzlizenzen den Namen des jeweiligen Benutzers in das Feld "Name" ein.

Das Dialogfeld "Software-Aktivierung" kann jederzeit über den Befehl **Hilfe | Software-Aktivierung** aufgerufen werden.

Bestellformular

Sobald Sie eine lizenzierte Version der Software bestellen möchten, klicken Sie im Dialogfeld "Software-Aktivierung" (*siehe oben*) auf die Schaltfläche **Permanenter Key-Code erwerben...** im zum sicheren Online-Shop von Altova weitergeleitet zu werden.

Registrierung

Wenn Sie Ihre Altova-Software nach der Aktivierung das erste Mal starten, erscheint ein Dialogfeld, in dem Sie gefragt werden, ob Sie Ihr Produkt registrieren möchten. Es gibt in diesem Dialogfeld drei Schatflächen:

- **OK:** Leitet Sie zum Registrierungsformular weiter
- **Später erinnern:** Ruft ein Dialogfeld auf, in dem Sie auswählen können, wann Sie das nächste Mal erinnert werden möchten.
- **Abbrechen:** Schließt das Dialogfeld und unterdrückt die Anzeige in Zukunft. Wenn Sie das Produkt zu einem späteren Zeitpunkt registrieren möchten, verwenden Sie den Befehl **Hilfe | Registrieren**

Auf Updates überprüfen

Überprüft, ob am Altova Server eine neuere Version Ihres Produkts vorhanden ist und zeigt

eine entsprechende Meldung an.

9.11.6 Support Center, Fragen und Antworten, Komponenten downloaden

Support Center

Falls Sie Fragen zu unseren Produkten haben, kontaktieren Sie über diesen Befehl unser Support-Team. Hier finden Sie Links zu unserer Liste von Fragen und Antworten, Supportformulare und die Email-Adressen, um unser Support-Team zu kontaktieren.

Fragen und Antworten im Web

Um Ihnen den bestmöglichen Support bieten zu können, haben wir eine Liste von Fragen und Antworten im Internet (FAQ) zusammengestellt. Diese Liste wird ständig von unseren Support-Mitarbeitern aktualisiert.

Sollten Sie Probleme mit einem unserer Produkte haben, raten wir Ihnen, zuerst die Fragen und Antworten im Internet zu konsultieren. Oft erhalten Sie hierdurch schnelle Hilfe.

Leider bieten wir derzeit noch keinen telefonischen Support. Ihre E-Mails an das Support-Team werden aber in der Regel innerhalb eines Werktages beantwortet.

Wenn Sie uns Anregungen oder Vorschläge zu

9.11.7 Internet

XMLSpy im Internet

Mit diesem Befehl gelangen Sie zum Altova Webserver auf <http://www.altova.com>. Hier erhalten Sie alle Informationen, Produkt-Updates und weitere Angebote von Altova.

Training

Mit diesem Befehl gelangen Sie zur Online-Training-Seite der Altova Website. Hier können Sie sich für Online-Kurse zu unseren Altova-Produkten anmelden.

9.11.8 Über

Mit diesem Befehl wird der Authentic Desktop Splashscreen und das Dialogfeld mit urheberrechtlichen Informationen angezeigt.

Bitte beachten Sie:

In diesem Dialogfeld wird die Versionsnummer angezeigt. Der Build sowie die komplette Versionsnummer der Version, die Sie verwenden, werden in der Statusleiste angezeigt.

Kapitel 3

Referenz für Programmierer

Referenz für Programmierer

Die Altova® Authentic® Desktop Edition als Automation Server

Authentic Desktop ist ein Automation Server, d.h. es ist eine Applikation, die anderen Applikationen (den so genannten Automation Clients) programmierbare Objekte zur Verfügung stellt, sodass der Automation Client direkt auf die vom Automation Server verfügbar gemachten Objekte zugreifen kann. Dies ist deshalb von Vorteil für einen Automation Client, da er die Funktionalitäten von Authentic Desktop nutzen kann. So kann ein Automation Client z.B. die XML-Validierungsfunktionen von Authentic Desktop verwenden. Entwickler können daher ihre Applikationen verbessern, indem Sie die vorhandenen Funktionalitäten von Authentic Desktop in ihren Applikationen nutzen.

Die programmierbaren Objekte von Authentic Desktop werden den Automation Clients über die **XMLSpy API**, eine COM API, zur Verfügung gestellt. In dieser Dokumentation finden Sie das Objektmodell der API und eine komplette Beschreibung der verfügbaren Objekte (siehe Abschnitt [XMLSpyAPI](#)). Außerdem kann die API auch mit einer Java-Implementierung verwendet werden. Im Kapitel [Using the XMLSpy API with Java](#) finden Sie eine Beschreibung dazu.

Anpassen von Authentic Desktop und Modifizieren der Funktionen

Sie können die Installation von Authentic Desktop durch Ändern und Hinzufügen von Funktionalitäten anpassen. Außerdem können Sie Formulare für Benutzereingaben erstellen und die Benutzeroberfläche so anpassen, dass sie bestimmte neue Menübefehle und Symbolleistenflächen enthält. All diese Funktionen werden in Form von Skripts implementiert, die mit Objekten der XMLSpy API interagieren. Um diese Funktionen effizient umsetzen zu können, bietet Ihnen Authentic Desktop eine integrierte Skripting-Umgebung. Im Abschnitt [Scripting](#) finden Sie eine komplette Beschreibung der Verwendung der Skripting-Umgebung und der darin verfügbaren Funktionen.

Des Weiteren können Sie Authentic Desktop mit Hilfe von externen Skripts manipulieren. So könnten Sie z.B. ein Skript schreiben, das Authentic Desktop zu einem bestimmten Zeitpunkt öffnet, anschließend die XML-Datei in Authentic Desktop öffnet, die Datei validiert und ausdruckt. Diese externen Skripten würden sich zur Durchführung dieser Aufgaben ebenfalls der XMLSpy API bedienen. Eine Beschreibung der XMLSpy API finden Sie im Abschnitt [XMLSpyAPI](#).

Erstellen von Plug-Ins für Authentic Desktop

Authentic Desktop gestattet Ihnen, Ihre eigenen Plug-Ins zu erstellen und in Authentic Desktop zu integrieren. Sie können dies mit Hilfe der speziellen Plug-In-Schnittstelle von Authentic Desktop tun. Eine Beschreibung zur Erstellung von Plug-Ins finden Sie im Abschnitt [XMLSpy Plug-ins](#).

Informationen zur Referenz für Programmierer

Die in dieser Referenz für Programmierer enthaltene Dokumentation besteht aus den folgenden Abschnitten:

- [Anmerkungen zur Release](#), in der die Änderungen seit der letzten Version aufgelistet sind
- [Scripting](#): eine Benutzerreferenz für die Skripting-Umgebung in Authentic Desktop
- [XMLSpy Plug-ins](#): eine Beschreibung zur Erstellung von Plug-Ins für Authentic Desktop
- [XMLSpy API](#): eine Referenz für die XMLSpy API
- [Using XMLSpy API with Java](#): eine Referenz zur Verwendung der XMLSpy API in Verbindung mit Java

1 Anmerkungen zur Release

Nachstehend sind zu jeder Release der XMLSpy API die wichtigsten Änderungen seit der vorhergehenden Release aufgelistet.

Automation Interface für XMLSpy 2007: Typbibliothek Version 1.6

Seit der vorhergehenden Release wurden die folgenden Änderungen implementiert:

- Die [Document](#) Schnittstelle verfügt über die Methode [GenerateDTDOrSchemaEx](#), um automatisch eine DTD oder ein Schema für das aktuelle XML-Dokument zu generieren.
- Die [GenerateSampleXMLDlg](#) Schnittstelle hat die folgenden zusätzlichen Eigenschaften: [FillAttributesWithSampleData](#), [FillElementsWithSampleData](#), [ContentOfNillableElementsIsNonMandatory](#), [TryToUseNonAbstractTypes](#), [Optimization](#), [SchemaOrDTDAssignment](#), [LocalNameOfRootElement](#), [NamespaceURIOfRootElement](#), [OptionsDialogAction](#).
- Die folgenden Eigenschaften der [GenerateSampleXMLDlg](#) Schnittstelle sind veraltet und werden nicht mehr unterstützt: [NonMandatoryElements - veraltet](#), [TakeFirstChoice - veraltet](#), [FillWithSampleData - veraltet](#).
- Die folgenden Enumerationen sind neu: [SPYAttributeTypeDefinition](#), [SPYSampleXMLGenerationOptimization](#), [SPYSampleXMLGenerationSchemaOrDTDAssignment](#).

Automation Interface für XMLSpy 2006sp2: Typbibliothek Version 1.5

Änderungen seit XMLSpy API Ver 2006:

- Die [CodeGeneratorDlg](#) Schnittstelle hat jetzt die Eigenschaften: [CompatibilityMode](#), [CPPSettings_GenerateVC6ProjectFile](#), [CPPSettings_GenerateVSProjectFile](#). Diese Eigenschaften generieren Projektdateien für Visual Studio 2005.

Automation Interface für XMLSpy 2006: Typbibliothek Version 1.5

Wichtige Änderungen seit der vorhergehenden Release (XMLSpy API 2004R4):

- Die [Application](#) Schnittstelle hat die folgenden Methoden: [ReloadSettings](#), [RunMacro](#), [ScriptingEnvironment](#).
- Die [Document](#) Schnittstelle hat die folgenden zusätzlichen Methoden: [GenerateSampleXML](#), [SetExternalIsValid](#), [UpdateXMLData](#).
- Die [Document](#) Schnittstelle hat das zusätzliche Event: [OnBeforeValidate](#).
- Die [XMLData](#) Schnittstelle hat die folgenden zusätzlichen Deklarationen: [CountChildren](#), [CountChildrenKind](#), [HasChildrenKind](#), [GetChild](#), [GetChildKind](#).
- Es wurde eine neue Schnittstelle [GenerateSampleXMLDlg](#) hinzugefügt.

Automation Interface für XMLSpy 2004R4: Typbibliothek Version 1.4

Wichtige Änderungen seit der vorhergehenden Release (XMLSpy API 2004R3):

Authentic-Ansicht

- Die Schnittstellen [AuthenticView](#), [AuthenticRange](#) und [AuthenticDataTransfer](#) ersetzen `DocEdit`, `DocEditSelection` und `DocEditDataTransfer`.
- Die Schnittstellen [AuthenticView](#) und [AuthenticRange](#) treten nun anstelle der alten Schnittstelle `DocEdit`.

- Die `DocEdit` Schnittstellen wurden in `OldAuthentic` umbenannt und werden nicht mehr verwendet.
- Die Schnittstelle `DocEditSelection` wurde in `AuthenticSelection` umbenannt und wird nicht mehr verwendet.
- Die Schnittstelle `DocEditDataTransfer` wurde in `AuthenticDataTransfer` umbenannt.
- Die Funktionalitäten von veralteten Schnittstellen stehen weiterhin zur Verfügung. Es wird jedoch empfohlen, die neuen Schnittstellen zu verwenden.

Verbesserte Events

Alle Events sind als Verbindungspunkt-Events verfügbar, um die einfache und flexible Integration von externen Clients mittels VBA, C++, VBScript, JScript oder Perl zu ermöglichen. Es wurden neue Events, die den Dokumentlebenszyklus unterstützen hinzugefügt. Alle Events übergeben Event-Kontextinformationen als Parameter. Viele Events unterstützen den Abbruch der angekündigten Operationen. Nähere Informationen dazu finden Sie im Abschnitt [Übersicht über Events](#).

Erweiterung der Dokumentschnittstelle

- Unterstützt die [Generierung von Programmcode](#) (C++, C# und Java) für Schemadefinitionen
- Unterstützt die [Generierung von MSWord und HTML-Dokumentation](#) für Schemadefinitionen
- Unterstützt die [XSL-FO-Transformation](#)
- Erweiterung von Standardeigenschaften für [Dokumente](#)
- Direkter Zugriff auf das [Root-Datenelement](#) von XML-Dokumenten

Datenimport und -export

- [Auswahl der Datenbank](#) für den Datenbankimport.
- Für den Import hierarchischer Daten, der früher über eine manuell erstellte SHAPE-Anweisung mit [ImportFromDatabase](#) durchgeführt wurde, ist nun eine Schemadatei und die Verwendung von [ImportFromSchema](#) erforderlich.

Automation Interface für XMLSpy 2004R3 -Typbibliothek Version 1.3

Wichtige Änderungen seit XMLSpy API Ver. 5 Rel. 4:

Authentic-Ansicht

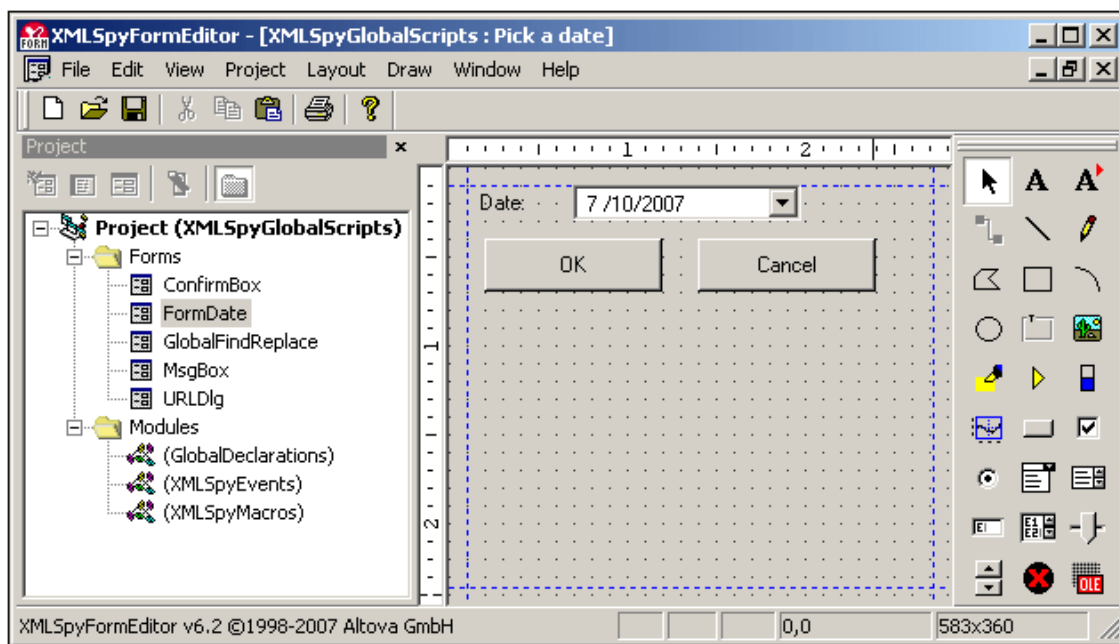
- Komplette neue [AuthenticView](#) und [AuthenticRange](#) Schnittstelle für die Arbeit mit der Authentic-Ansicht.

2 Skripterstellung

Skripts können verwendet werden, um Objekte in der XMLSpy API, wobei es sich um eine COM API handelt, und der API von Authentic Desktop aufzurufen und damit zu interagieren. (Das Objektmodell der XMLSpy API wird in der Dokumentation zur [XMLSpyAPI](#) beschrieben.) Sie können mit einem Skript also Funktionalitäten ändern und zur Installation von Authentic Desktop hinzufügen.

Die Skripting-Umgebung

Die Authentic Desktop IDE bietet Zugriff auf eine Skripting-Umgebung (*Abbildung unten*). Die Skripting-Umgebung wird in einem separaten Fenster geöffnet und kann durch Auswahl des Authentic Desktop Menübefehls **Extras | Skripting-Umgebung** aufgerufen werden.



Die Programmiersprachen, die in der Skripting-Umgebung verwendet werden können, sind **JavaScript** und **VBScript**.

Was können Sie mit der Skripting-Umgebung tun?

Sie können in der Skripting-Umgebung Formulare, Event Handler und Makros erstellen und eine Gruppe davon einem Skripting-Projekt zuweisen. Das Skripting-Projekt wird anschließend einem Authentic Desktop Projekt zugewiesen. Ein Event Handler wird ausgeführt, wenn das dazugehörige Event in Authentic Desktop vorkommt. Alternativ dazu kann ein in der Skripting-Umgebung erstelltes Makro innerhalb von Authentic Desktop ausgeführt werden.

Dokumentation zur Skripting-Umgebung

Die Dokumentation über die Skripting-Umgebung (dieser Abschnitt) ist in die folgenden Teile gegliedert:

- [eine Übersicht](#), in der überblicksmäßig beschrieben wird, wie die Skripting-Umgebung verwendet wird und wie Skripts in Authentic Desktop verwendet werden können.
- [eine Beschreibung, wie man Formulare erstellt und verwendet](#), sowie ein detailliertes Beispiel.

- [eine Erläuterung der Event Handler der Applikation.](#)
- einer [Beschreibung zur Verwendung eines Makros](#) in der Skripting-Umgebung und Authentic Desktop anhand eines Beispiels.
- [einem Referenzteil für die Menübefehle](#) der Skripting-Umgebung.

2.1 Übersicht über die Verwendung

Die Skripting-Umgebung bietet eine Schnittstelle, über die Sie folgendes tun können: (i) Formulare mit grafischen Mitteln erstellen, während Sie Skripts für Komponenten im Formular zuweisen; (ii) Event Handler und (iii) Makros erstellen. Diese Formulare, Event Handler und Makros werden in Skripting-Projekten organisiert, die dann Authentic Desktop -Applikationsprojekten zugewiesen werden und in der Applikation verwendet werden können. Variablen und Funktionen können global (für Formulare, Event Handler und Makros) verfügbar gemacht werden, indem man sie in einer einzigen (globalen Deklarations)-Datei definiert.

In diesem Abschnitt wird in Grundzügen beschrieben, wie die Skripting-Umgebung verwendet werden soll. Er ist in die folgenden Abschnitte gegliedert:

- [Skripting-Projekte und das Projektfenster](#): In diesem Abschnitt wird der Zweck von Skripting-Projekten beschrieben, wie sie in Applikationsprojekte eingebunden werden und wie man in der Skripting-Umgebung mit Skripting-Projekten arbeitet. Auch das Projektfenster wird darin ausführlich beschrieben.
- [Bearbeitung im Hauptfenster](#): In diesem Abschnitt wird das Hauptfenster beschrieben, wie Skripts im Hauptfenster zur Bearbeitung geöffnet werden und wie die Ansicht des Hauptfensters konfiguriert wird.
- [Formulare, Event Handler und Makros](#): Hier wird ein in Grundzügen beschrieben, wie diese Komponenten erstellt und in der Skripting-Umgebung verwendet werden können.
- [Skripting-Einstellungen](#): Hier werden die verschiedenen Optionen für die Skripting-Umgebung beschrieben.

2.1.1 Skripting-Projekte und das Projektfenster

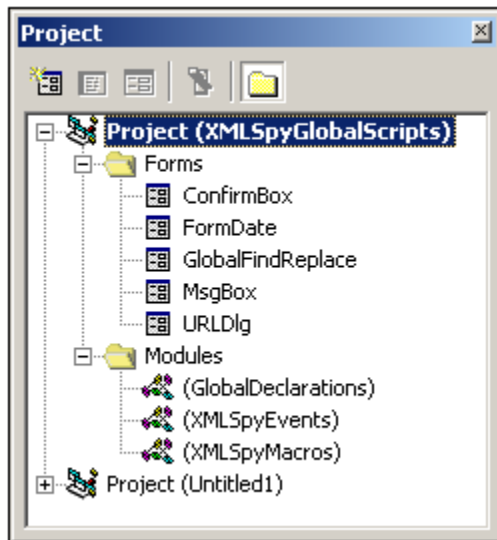
In der Skripting-Umgebung werden alle Skripts und Skripting-Informationen in **Skripting-Projekten** (. prj files) gespeichert. Ihre Applikation wird mit einem globalen Skripting-Projekt namens `GlobalScripts.prj` geliefert, welches globale Standarddeklarationen und Vorlagen-Skripts für einige Standardaufgaben enthält. Diese Skripts werden im Applikationsordner gespeichert.

Skripting-Projekte und Applikationsprojekte

Sie können zusätzliche Skripting-Projekte, die anschließend einem Authentic Desktop Projekt (Applikationsprojekt) zugewiesen werden müssen, erstellen und speichern. Derartige zusätzliche Projekte werden verwendet, wenn das relevante Authentic Desktop-Projekt offen ist. Einem Authentic Desktop -Projekt kann nur ein Skripting-Projekt zugewiesen werden, aber ein Skripting-Projekt kann mehreren Authentic Desktop-Projekten zugewiesen werden. In einem Authentic Desktop-Projekt werden die Skripts (i) im globalen Skripting-Projekt und (ii) im Skripting-Projekt, das diesem zugewiesen ist, verwendet. Wenn Skripts auf die gesamte Applikation angewendet werden können sollen, müssen Sie das globale Skripting-Projekt ändern. Beachten Sie, dass das globale Skripting-Projekt auf die gesamte Applikation angewendet wird, auch wenn kein Applikationsprojekt offen ist.

Inhalt eines Skripting-Projekts

Ein Skripting-Projekt besteht aus zwei Komponenten: einer Komponente "Forms" und einer Komponente "Modules". In der Abbildung unten sehen Sie den Inhalt des globalen Skripting-Projekts im Projektfenster.



Die Komponente "Forms" enthält alle für dieses Skripting-Projekt definierten Formulare. Die Komponente "Modules" enthält drei Module: GlobalDeclarations, XMLSpyEvents und XMLSpyMacros. Das Modul "GlobalDeclarations" enthält Definitionen von in diesem Skripting-Projekt verwendeten Variablen und Funktionen. Das Modul "XMLSpyEvents" enthält Event Handler Skripts für dieses Skripting-Projekt. Das Modul "XMLSpyMakros" enthält die für dieses Projekt definierten Makros.

Aktivieren eines Skripting-Projekts

In der Skripting-Umgebung können gleichzeitig mehrere Skripting-Projekte geöffnet sein. Um das Skripting-Projekt zum aktiven zu machen, klicken Sie mit der rechten Maustaste darauf und

wählen Sie den Befehl **Set as active project**. Beachten Sie: Wenn Sie das Skripting-Projekt deaktivieren, wird die Zuweisung dadurch nicht aufgehoben; es wird nur das Skripting-Projekt in der Skripting-Umgebung deaktiviert.

Zuweisen von Skripting-Projekten und Aufheben der Zuweisung

Ein Skripting-Projekt wird einem Authentic Desktop Projekt von innerhalb der Authentic Desktop -Benutzeroberfläche zugewiesen, wobei das benötigte Authentic Desktop-Projekt geöffnet ist. Um ein Skripting-Projekt zuzuweisen, bzw. dessen Zuweisung aufzuheben, gehen Sie folgendermaßen vor:

1. Wechseln Sie zur GUI der Authentic Desktop-Applikation.
2. Öffnen Sie das erforderliche Applikationsprojekt.
3. Wählen Sie den Menübefehl **Extras | Projekt | Skripts zuweisen**.
4. Daraufhin wird ein Standarddialogfeld zur Dateiauswahl geöffnet, in dem Sie das gewünschte Skripting-Projekt (.prj Datei) auswählen können. Das Skripting-Projekt wird geladen und Sie können die Skripts darin bearbeiten, wenn Sie in die Skripting-Umgebung wechseln.

Um die Zuweisung eines Skripting-Projekts zu einem Applikationsprojekt aufzuheben, öffnen Sie das Applikationsprojekt in Authentic Desktop und wählen Sie den Befehl **Extras | Projekt | Projekt-Skripts entkoppeln**. Die Zuweisung wird aufgehoben und das Skripting-Projekt wird in der Skripting-Umgebung geschlossen.

Anmerkung: Wenn in gerade kein Applikationsprojekt aktiv ist, sind die Menübefehle zum Zuweisen und zur Aufhebung der Zuweisung deaktiviert. .

Erstellen, Öffnen, Schließen und Speichern von Skripting-Projekten

Befehle zum Erstellen, Öffnen, Schließen und Speichern von Skripting-Projekten stehen im Menü **File** der Skripting-Umgebung zur Verfügung. Außerdem stehen im Kontextmenü, das bei Rechtsklick auf ein Projektelement im Projektfenster angezeigt wird, Befehle zum Erstellen eines neuen Projekts, zum Öffnen eines vorhandenen Projekts und zum Schließen des ausgewählten Projekts zur Verfügung.

Schaltflächen im Projektfenster

Die Titelleiste des Projektfensters enthält die folgenden Schaltflächen:



New Form: Fügt ein neues Formular zum Ordner "Forms" hinzu.



View Code: Zeigt das ausgewählte Formular oder den Code eines Moduls an. Der Code des ausgewählten Formulars wird nur angezeigt, wenn **Layout | Edit Script** für dieses Formular aktiviert wurde. Andernfalls wird die editierbare (bei Auswahl von **Layout | Edit View**) bzw. nicht editierbare Ansicht (wenn weder **Layout | Edit Script** noch **Layout | Edit View** ausgewählt ist) des Formulars angezeigt. Die Schaltfläche ist aktiv, wenn ein Formular oder Modul ausgewählt ist.



View Form: Zeigt das ausgewählte Formular an. Die editierbare Ansicht des ausgewählten Formulars wird nur angezeigt, wenn **Layout | Edit View** für dieses Formular aktiviert wurde. Andernfalls wird der Code (bei Auswahl von **Layout | Edit Script**) oder die nicht editierbare Ansicht des Formulars (wenn weder **Layout | Edit View** noch **Layout | Edit Script** ausgewählt ist) angezeigt. Die Schaltfläche ist aktiv, wenn ein Formular ausgewählt ist.



Run Form: Diese Schaltfläche ist aktiv, wenn ein Formular ausgewählt ist.



Toggle Folders: Blendet die Ebene "Forms" und "Modules" der Struktur ein und aus.

2.1.2 Bearbeitung im Hauptfenster

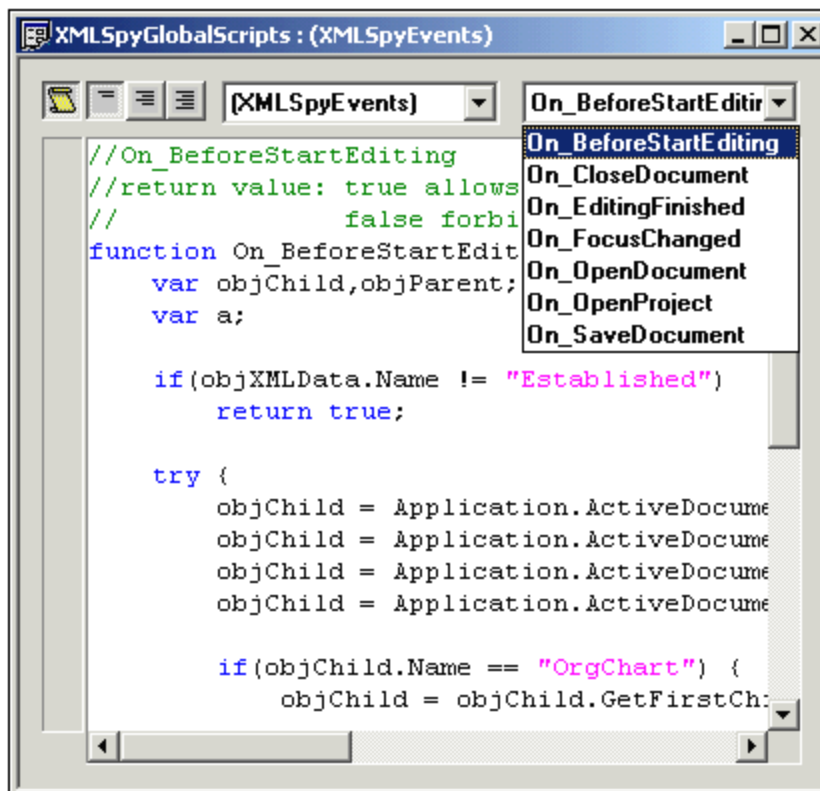
Das Hauptfenster ist jener Bereich der Benutzeroberfläche, in dem einzelne Formulare und Module angezeigt und bearbeitet werden können. Jedes Formular oder Modul wird in einem separaten Fenster geöffnet, wenn Sie auf den entsprechenden Eintrag im Projektfenster klicken.

Formulare

Wenn ein Formular im Hauptfenster geöffnet wird, wird es als grafisches Dialogfeld dargestellt (dies ist die Design-Ansicht des Formulars). Wenn für ein bestimmtes Formular der Befehl **Layout | Edit View** aktiviert ist, kann das Design dieses Formulars bearbeitet werden. Andernfalls kann die Design-Ansicht nicht editiert werden. Wenn der Befehl **Layout | Edit Script** für ein Formular aktiviert wurde, so wird das Skript, das diesem Formular zugrunde liegt, im Hauptfenster angezeigt. Wenn Sie auf die Schaltfläche "Close Script View"  klicken, wechseln Sie wieder in die Design-Ansicht zurück (die editierbar oder nicht editierbar ist, je nachdem ob **Layout | Edit View** aktiviert oder deaktiviert ist). Beachten Sie, dass die Befehle **Layout | Edit View** und **Layout | Edit Script** für jedes Formular separat definiert werden.

Module

Jedes Modul (Globale Deklarationen, (Applikations)-Events und Makros) wird in einem separaten Fenster geöffnet. In einem Modulfenster sehen Sie in der Auswahlliste auf der linken Seite, welches Modul ausgewählt ist. So sehen Sie z.B. in der Abbildung unten, dass das Events-Modul ausgewählt ist. Um ein bestimmtes Element in diesem Modul auszuwählen, klicken Sie im Dropdown-Menü der Auswahlliste auf der rechten Seite auf die gewünschte Option. In der Abbildung unten wird das `On_BeforeStartEditing` Event ausgewählt, wodurch das Skript zur Behandlung dieses Event angezeigt wird.



In Skript-Ansichten, können Sie aus Optionen auswählen, um folgendes anzuzeigen: (i) das ausgewählte Event (Event-Ansicht), (ii) die Objekte im Modul (Objekt-Ansicht) oder (iii) das komplette Modul (vollständige Modulansicht). Klicken Sie zum Auswählen der gewünschten Ansicht auf die entsprechende Schaltfläche (links oben im Hauptfenster).

Kontextmenüs

Wenn Sie mit der rechten Maustaste in die Skript-Ansicht eines Formulars oder in ein Modulfenster klicken, erscheint ein Kontextmenü, über das Sie Funktionen zur Skripterstellung aufrufen können.

2.1.3 Formulare, Event Handler und Makros

In der Skripting-Umgebung werden Formulare, Event Handler und Makros erstellt. In den Skripts, die diesen zugrunde liegen, können Variablen und Funktionen verwendet werden, die in einem globalen Deklarationsmodul definiert sind.

Globale Deklarationen

Das globale Deklarationsmodul enthält Variablen und Funktionen, die von Formularen, Event-Handlern und Makros verwendet werden können. Die Funktionen greifen über die XMLSpy API auf Authentic Desktop-Funktionalitäten zu. Wenn Sie im globalen Deklarationsmodul eine Variable oder Funktion erstellen, können Sie diese von allen Formularen, Event-Handlern und Makros im Skripting-Projekt aus aufrufen. Um eine Variable oder Funktion hinzuzufügen, doppelklicken Sie im Projektfenster auf das globale Deklarationsmodul. Daraufhin wird das globale Deklarationsskript im Hauptfenster geöffnet. Fügen Sie in diesem Skript die gewünschte Variable oder Funktion hinzu.

Formulare

Sie können ein Formular in der Skripting-Umgebung grafisch mittels einer Palette von Formularobjekten wie z.B. Texteingabefeldern und Schaltflächen erstellen. So können Sie z.B. ein Formular erstellen, um die Eingabe eines Elementnamens zu akzeptieren und anschließend alle Instanzen des Elements aus dem aktiven XML-Dokument zu entfernen. Für ein solches Formular kann ein Funktions-Skript mit dem Texteingabefeld verknüpft werden, um eine Eingabevariable aufzunehmen; und ein Event Handler kann mit einer Schaltfläche verknüpft werden, um die Ausführung der in der XMLSpy API verfügbaren Löschfunktion zu starten. Ein Formular wird entweder über einen Funktionsaufruf innerhalb einer Funktion (im globalen Deklarationsmodul) oder direkt in einem Makro aufgerufen. Nähere Informationen zum Erstellen und Bearbeiten von Formularen finden Sie im Abschnitt [Formulare](#).

Behandlung von Events

Event Handler-Skripts können mit den verschiedensten Events verknüpft werden. Sie können damit Events steuern, die sowohl innerhalb von Formularen vorkommen ([Formular-Events](#)) als auch innerhalb der allgemeinen Applikations-Benutzeroberfläche ([Applikations-Events](#)). Das mit einem Event verknüpfte Skript wird sofort bei Auslösung des Event ausgeführt. Die meisten Events haben Parameter, die ausführliche Informationen über das Event enthalten. Der Rückgabewert aus dem Skript enthält normalerweise Anweisungen für die Applikation über dessen weitere Verarbeitung (z.B. lässt die Applikation möglicherweise eine Bearbeitung nicht zu). Ein Event Handler wird ausgeführt, wenn das entsprechende Event im Formular oder in Authentic Desktop vorkommt. Wenn sowohl im globalen Skripting-Projekt als auch im projektspezifischen Skripting-Projekt (Unterschiede siehe [Skripting-Projekte und das Projektfenster](#)) ein Event Handler für ein einzelnes Event definiert ist, wird zuerst der Event Handler für das projektspezifische Skripting-Projekt ausgeführt und gleich anschließend derjenige für das globale Skripting-Projekt. Nähere Informationen zur Erstellung von Event-Handlern finden Sie unter [Event Handler](#).

Makros

Makros werden zum Implementieren komplexer oder wiederholter Aufgaben verwendet. Für Makros werden weder Parameter noch Rückgabewerte verwendet. In einem Makro können Sie alle in den globalen Deklarationen deklarierten Variablen und Funktionen aufrufen und Formulare für die Benutzereingabe anzeigen. Ein einfaches Beispiel zur Erstellung eines Makro finden Sie unter [Schreiben eines Makro](#). Eine Beschreibung dazu, auf welche Arten Makros aufgerufen werden können, finden Sie unter [Aufrufen von Makros](#). Ein Makro wird von der Benutzeroberfläche von Authentic Desktop aus durch Klicken auf **Extras | Makros anzeigen** ausgeführt.

Aufrufen von Formularen, Event-Handlern und Makros

Formulare, Event Handler und Makros werden alle in der Skripting-Umgebung erstellt. Jedoch

ist die Art, wie ein Makro aufgerufen und ausgeführt wird bei jedem Makro anders und hat eine Auswirkung auf die Erstellung Ihrer Skripting-Projekte.

- Ein Formular wird entweder durch einen Aufruf von einer Funktion im globalen Deklarationsmodul aus oder direkt in einem Makro aufgerufen.
- Ein Event Handler wird ausgeführt, wenn das entsprechende Event in Authentic Desktop vorkommt. Wenn sowohl im globalen Skripting-Projekt als auch im projektspezifischen Skripting-Projekt ein Event Handler für ein einzelnes Event definiert ist, wird zuerst der Event Handler für das projektspezifischen Skripting-Projekt ausgeführt und gleich anschließend derjenige für das globale Skripting-Projekt.
- Ein Makro wird von der Benutzeroberfläche von Authentic Desktop aus durch Klicken auf **Extras | Makros anzeigen** ausgeführt. Daraufhin erscheint das Dialogfeld "Makros anzeigen", in dem Sie das gewünschte Makro auswählen können.

2.1.4 Skripting -Einstellungen

Einstellungen für die Skripting-Umgebung können im Dialogfeld "Optionen" auf dem Register "Skripting" (*Abbildung unten*) vorgenommen werden. Die verschiedenen Optionen werden unten beschrieben.

Skript-Umgebung beim Starten von Authentic Desktop starten

Wenn dieses Kontrollkästchen aktiviert ist, wird die Skripting-Umgebung beim Start der Applikation in einem separaten Fenster geöffnet. Andernfalls wird die Skripting-Umgebung durch Auswahl von **Extras | Skripting-Umgebung** aufgerufen.

Globale Skripting-Projektdatei

In diesem Feld ist die globale Skripting-Projektdatei definiert. Mit der Applikation wird eine Beispieldatei für ein globales Skripting-Projekt installiert. Wenn die Datei beim Start der Skripting-Umgebung nicht vorhanden ist, wird unter dem angegebenen Namen und Pfad ein neues Skripting-Projekt erstellt.

Auto-Makros starten

Aktiviert bzw. deaktiviert die Ausführung von Auto-Makros. Derzeit ist `Autorun` aus den globalen Skripts das einzige unterstützte automatische Makro. `Autorun` wird beim Start der Skripting-Umgebung ausgeführt.

Events abarbeiten

Aktiviert bzw. deaktiviert die Ausführung von Event Handlern. Wenn dieses Kontrollkästchen nicht aktiviert ist, werden Event Handler im Skripting-Projekt nicht ausgeführt.

Standard Skriptsprache

Für jedes neue Skripting-Projekt wird die hier ausgewählte Skript-Sprache als Standard Skript-Sprache ausgewählt.

2.2 Formulare

Das Erstellen und Bearbeiten von Formularen in der Skripting-Umgebung erfolgt in folgenden Schritten:

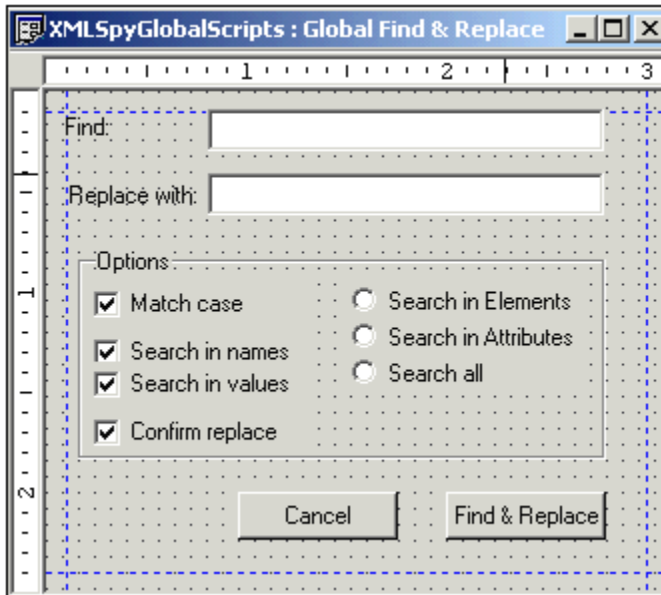
1. [Öffnen des Formulars im Hauptfenster](#). Das Formular kann als editierbares grafisches Design, als nicht editierbares Design, als nicht editierbares fertiges Formular und als Skript angezeigt werden. Diese verschiedenen Editor-Ansichten werden im Unterabschnitt [Editor-Ansichten](#) beschrieben.
2. [Erstellen des Formulars](#) durch Erstellen und Bearbeiten von Formularobjekten. Eine Beschreibung dazu finden Sie im Unterabschnitt [Formulardesign](#).
3. [Zuweisen von Eigenschaften zum Formular und seinen Objekten](#) sowie Zuweisen von Event Handlern für Objekt-Events. Eine Beschreibung dazu finden Sie im Unterabschnitt [Objekteigenschaften und Events](#).

Eine schnelle Kurzanleitung, wie man Formulare erstellt und verwendet, finden Sie unter [Beispiel: Erstellen eines Formulars](#).

2.2.1 Editor-Ansichten

Ein bestehendes Formular können Sie durch Doppelklicken auf das Symbol des Formulars im Projektfenster oder durch Rechtsklick auf das Symbol und Auswahl des Befehls **View Code** oder **View Object** öffnen. Ein neues Formular wird mit dem Befehl **Project | Add Form** oder durch Rechtsklick auf einen Projekteintrag im Projektfenster und Auswahl des Befehls **Add Form** erstellt.

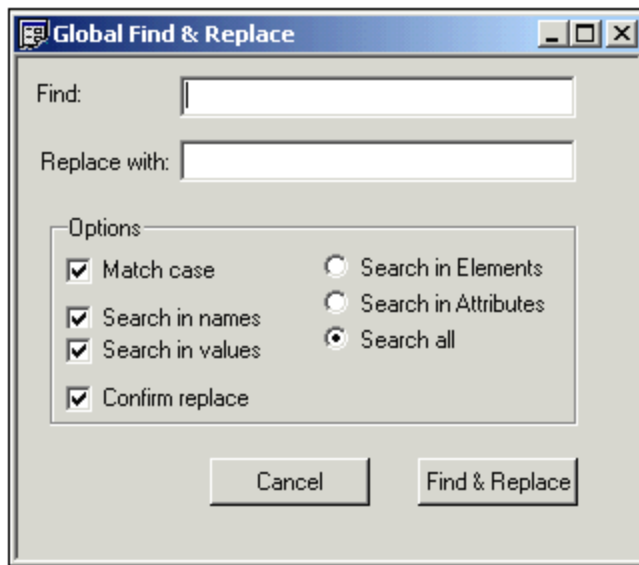
In beiden Fälle (bei Öffnen eines bestehenden oder neuen Formulars) wird das Formular im Hauptfenster geöffnet. Jedes Formular wird in einem separaten Fenster geöffnet (*Abbildung unten*), das in der Größe angepasst und minimiert werden kann.



Formulare können in zwei verschiedenen Ansichten angezeigt werden: der Design-Ansicht und der Skript-Ansicht.

Design-Ansicht

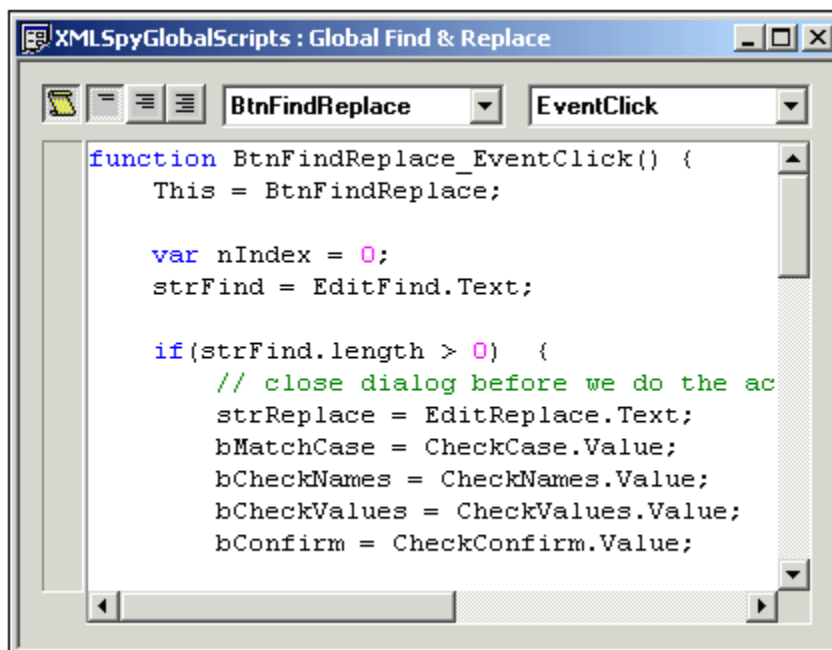
In der Design-Ansicht können Sie zwischen einem editierbaren (*Abbildung oben*) und einem nicht editierbaren Modus wechseln. Klicken Sie dazu auf den Menübefehl **Layout | Edit View**. Wenn der Befehl deaktiviert ist, kann die Design-Ansicht nicht bearbeitet werden und das Formular wird so angezeigt, wie es der Benutzer sehen wird (*Abbildung unten*).



Ein Formular muss im editierbaren Modus der Design-Ansicht entworfen und bearbeitet werden. Die Einstellungen für den Arbeitsbereich des Formulars werden im Dialogfeld "Grid Settings" ([Layout | Grid Settings](#)) festgelegt. Ein Formular wird erstellt, indem man Objekte aus der [Formular-Objektleiste](#) einfügt und dann einzelnen Objekten Eigenschaften und Skripts zuordnet. Visuelle Anpassungen des Layout können Sie über die Befehle des Menüs [Layout](#) vornehmen.

Skript-Ansicht

Die Skript-Ansicht (*Abbildung unten*) wird mit dem Menübefehl [Layout | Edit Script](#) aktiviert. Um in die Design-Ansicht zu wechseln, wählen Sie entweder den Befehl [Layout | Edit View](#) oder klicken Sie im Hauptfenster auf die Schaltfläche "Close Script View" .



Weitere Informationen zur Bearbeitung im Hauptfenster finden Sie unter [Bearbeitung im Hauptfenster](#).

2.2.2 Formulardesign

Die Erstellung eines Formulars erfolgt in den folgenden Schritten:

- Platzieren eines Objekts aus der Formularobjektleiste auf eine Ebene im Formulardesign. Ebenen und die Formularobjektleiste werden in diesem Abschnitt beschrieben.
- Zuweisen von Eigenschaften und Events zu Formularobjekten im Design. Eine Beschreibung dazu finden Sie im nächsten Abschnitt.

Formularebenen

Wenn Sie Formularobjekte in ein Formular platzieren, werden diese Objekte standardmäßig auf einer Ebene mit dem Namen `Default` erstellt. Sie können **unterhalb** der `Default`-Ebene weitere Ebenen hinzufügen und so Hintergrundobjekte erstellen. Wenn Objekte in einem Bereich übereinander zu liegen kommen, so werden die Objekte in den darüber liegenden Ebenen über die Objekte in darunter liegenden gelegt. Ebenen unterhalb der `Default`-Ebene können relativ zueinander verschoben werden (d.h. nach oben oder nach unten), die `Default`-Ebene bleibt jedoch immer die oberste Ebene. Aus diesem Grund liegen Objekte in der `Default`-Ebene immer über allen Objekten in niedrigeren Ebenen. Nähere Informationen zur Verwendung von Ebenen finden Sie unter [Layout](#).

Formularobjekte

Die **Formularobjektleiste** (*Abbildung unten*) ist eine Tool-Palette, über die Sie Steuerelemente zu Ihren Formularen hinzufügen können. Wählen Sie ein Formularobjekt (Steuerelement) aus der Leiste aus und platzieren Sie den Mauszeiger anschließend an die gewünschte Stelle in Ihrem Formular und ziehen Sie ein Rechteck auf, um die Größe des Objekts festzulegen.



Die am häufigsten gebrauchten Objekte sind im Folgenden beschrieben:



Statischer Text: Fügt Textfelder wie z.B. Überschriften oder Feldbeschreibungen hinzu.



Schaltfläche: Fügt eine Schaltfläche hinzu. Auch Bitmaps können diesen Schaltflächen zugewiesen werden.



Kontrollkästchen: Fügt ein Kontrollkästchen hinzu, mit dem Elemente vom Typ

Ja/Nein aktiviert werden können.



Auswahlliste: Fügt eine Auswahlliste hinzu, über die der Benutzer eine Option aus einem Dropdown-Menü auswählen kann.



Listenfeld: Fügt ein Listenfeld hinzu, das eine Liste von Einträgen zur Auswahl anzeigt.



Bearbeitungsfeld: Definiert eine einzelne Textzeile.



Mehrzeiliges Bearbeitungsfeld: Definiert mehrere Textzeilen.



ActiveX: Gestattet die Integration eines ActiveX Control in das Formular.

Formularobjekteigenschaften und Events

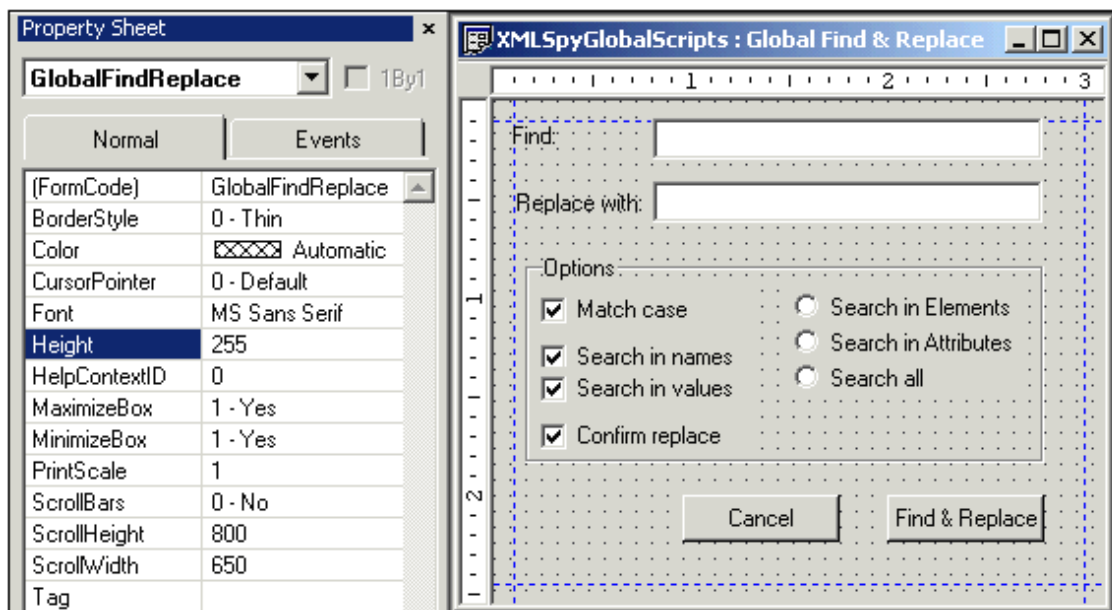
Jedes Formularobjekt verfügt über seine eigenen Eigenschaften und Events. Sie können diese im Eigenschaftsblatt dieses Objekts definieren. Nähere Informationen dazu finden Sie unter [Objekteigenschaften und Events](#).

Festlegen der Tabulatorreihenfolge für Dateneingabeobjekte

Wenn ein Benutzer Daten in ein Formular eingibt, springt er normalerweise durch Drücken der Tabulatortaste von einem Feld zum nächsten. Die Reihenfolge für den Tabulator kann im Objektblatt (Tab Order) festgelegt werden. Eine Beschreibung dazu finden Sie unter [Layout](#).

2.2.3 Objekteigenschaften und Events

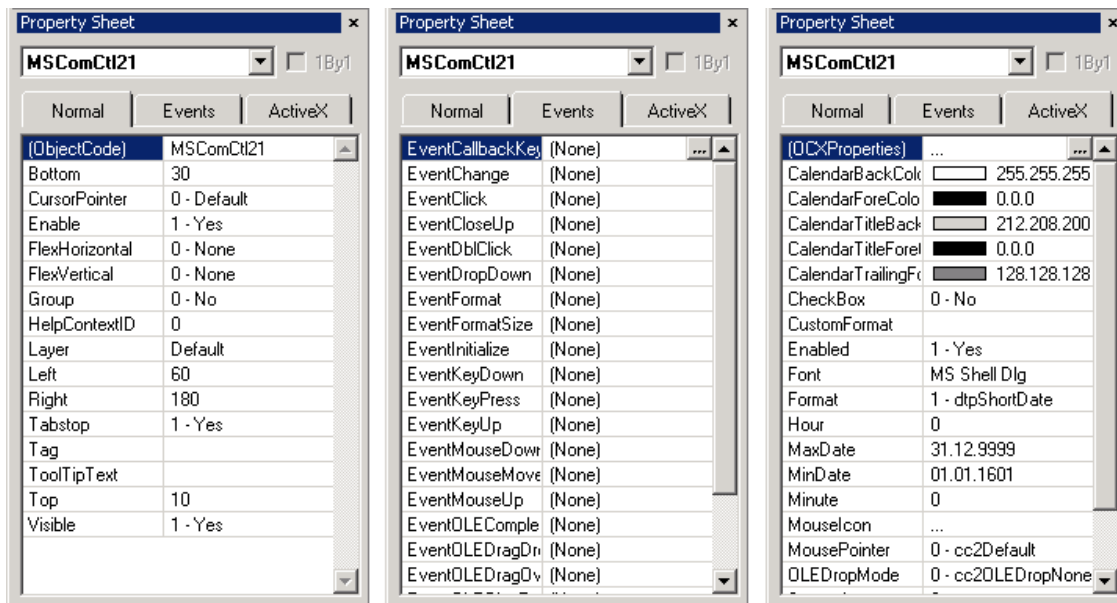
Eigenschaften und Events von Formularen und Formularobjekten werden im Eigenschaftsblatt (siehe Abbildung) dieses Formulars zugewiesen. Um das Eigenschaftsblatt anzuzeigen, während sich die Design-Ansicht in editierbaren Modus befindet, wählen Sie **Layout | Eigenschaften** oder klicken Sie mit der rechten Maustaste in das Formular. (Wenn Sie mit der rechten Maustaste in einen leeren Bereich des Formulars klicken, werden die Eigenschaften des Formulars angezeigt; wenn Sie mit der rechten Maustaste auf ein Formularobjekt rechtsklicken (z.B. auf eine Schaltfläche), werden die Eigenschaften dieses bestimmten Formularobjekts angezeigt. Alle Objekte im Formular, einschließlich der Eigenschaften für das Formular selbst, stehen in der Dropdown-Liste der Auswahlliste des Eigenschaftsblatts zur Auswahl zur Verfügung. Um die Eigenschaften eines beliebigen Objekts im Formular anzuzeigen, wählen Sie dieses Objekt in der Auswahlliste des Eigenschaftsblatts aus oder wählen Sie das Objekt im Design aus.



Um das Eigenschaftsblatt zu schließen, klicken Sie auf die Schaltfläche **Schließen**.

Das Eigenschaftsblatt

Das Eigenschaftsblatt besteht aus zwei Registern, *Normal* und *Events*. Wenn das Formular ein ActiveX Control enthält, gibt es ein drittes Register namens *ActiveX*. In den Abbildungen unten sehen Sie die drei Register des Microsoft DatePicker ActiveX Control. Beachten Sie beim Auswählen unterschiedlicher Objekte, dass unterschiedliche Objekte unterschiedliche Eigenschaften und Events haben.

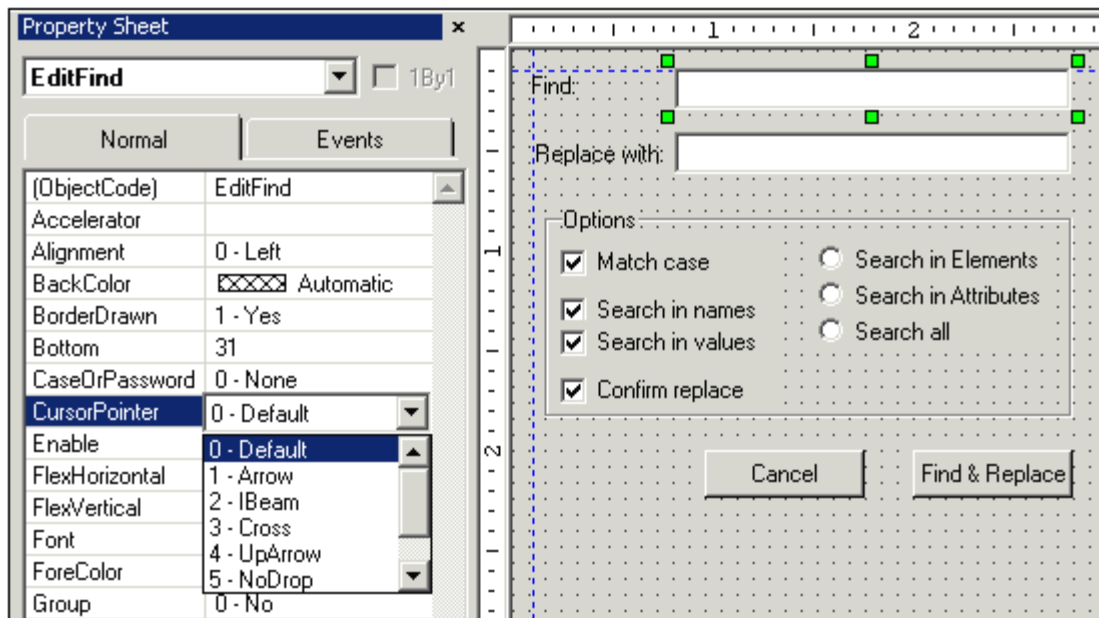


Register "Normal"

Auf dem Register *Normal* sind der Name des Objekts und seine Eigenschaften in Bezug auf das Aussehen des Objekts im Formular definiert. Der Name wird von der Eigenschaft *ObjectCode* definiert (hier `MScomCtl21`). Alternativ dazu könnte der Name des Formulars oder eines Formularobjekts sein. Wenn Sie ein Skript schreiben und Zugriff auf die Methoden und Eigenschaften dieses Objekts benötigen, müssen Sie diesen Namen verwenden. Zum Beispiel:

```
var strTmp;
strTmp = "";
strTmp = MSComCtl21.Year + '-' + MSComCtl21.Month + '-' + MSComCtl21.Day;
```

Wenn eine Eigenschaft eine Aufzählung von Werten hat, stehen diese in einem Dropdown-Menü zur Verfügung, wenn die Eigenschaft ausgewählt wird (siehe Abbildung unten).



In der Abbildung oben ist das `EditFind` Objekt (Das Textfeld "Find") ausgewählt. Wenn die Eigenschaft `CursorPointer` dieses Objekts ausgewählt wird, wird ein Dropdown-Menü mit den möglichen Werten für diese Eigenschaft verfügbar. Auf diese Art können Sie schnell einen passenden und richtigen Wert für diese Eigenschaft auswählen.

Anmerkung: Farbeigenschaften werden in RGB-Werten in der Form `RR, GG, BB` angegeben. So wäre rot z.B. `255, 0, 0`.

Register "Events"

Um auf dem Register *Events* Code für eine Eigenschaft einzugeben, gehen Sie folgendermaßen vor:

1. Wählen Sie die Eigenschaft aus, für die Sie Code eingeben möchten. Rechts vom Eingabefeld erscheint eine Schaltfläche mit einem Auslassungszeichen (...).
2. Klicken Sie darauf, um ein Skript Editor-Fenster für das Codefragment zu öffnen.
3. Geben Sie den Code ein und klicken Sie auf **OK**.

Wiederholen Sie diese Schritte für alle Events, für die Sie Skripte schreiben möchten.

Register "ActiveX"

Die Eigenschaften auf dem Register *ActiveX* erhalten Werte jeweils auf einer der oben für die Register *Normal* und *Events* beschriebenen Arten.

2.2.4 Objekt: TheView

Das Objekt `TheView`, welches für jedes einzelne Formular zur Verfügung steht, enthält Funktionen und Eigenschaften, mit denen das Verhalten des Formulars geändert werden kann..

Methoden

Die folgenden Methoden stehen zur Verfügung:

- `Cancel()`: Schließt das Formular. Mit dem Aufruf `TheView.Cancel()` innerhalb eines button-click Events wird das Formular geschlossen, wenn Sie auf die Schaltfläche klicken.
- `IsFormOpen(Name as String) as Boolean`: Überprüft, ob das angegebene Formular bereits geöffnet ist.
- `FormFind(Name as String) as Object`: Gibt das definierte Formular zurück oder den Wert Null, wenn das Formular nicht gefunden werden kann oder nicht offen ist.
- `OpenDoc(File as String)`: Öffnet das angegebene Dokument in der Applikation.
- `PumpData()`: Löst das Event `EventPumpData` an den einzelnen Controls dieses Formulars aus.
- `RunClick()`: Löst das Event `TheView_EventClick()` aus.
- `RunInitialize()`: Löst das Event `TheView_EventInitialize()` aus. Dieses Event wird auch ausgelöst, bevor das Formular angezeigt wird.
- `RunTerminate()`: Löst das Event `TheView_EventTerminate()` aus. Dieses Event wird auch beim Schließen des Formulars ausgelöst.

Eigenschaften

Die folgenden Eigenschaften stehen zur Verfügung:

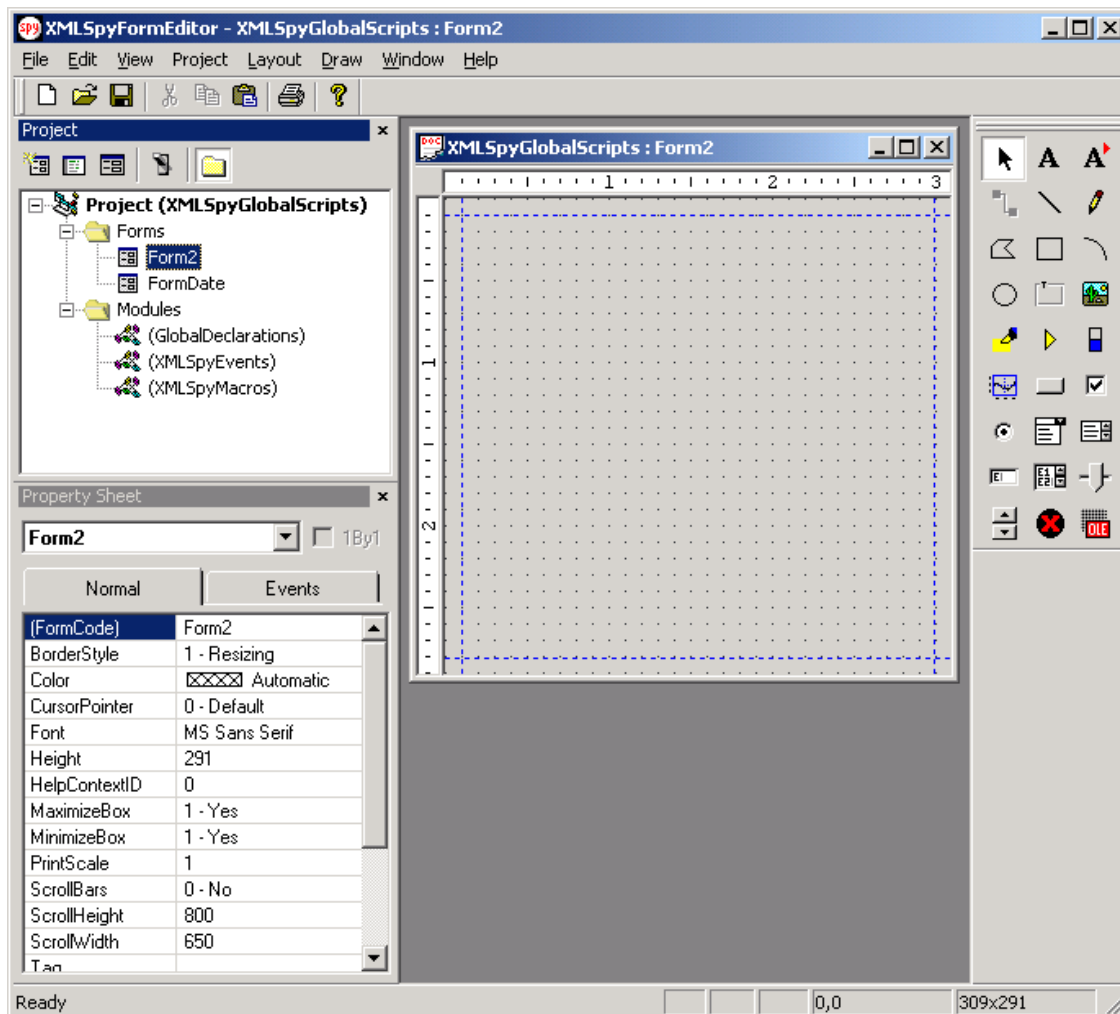
- `ToolTipText as String`: Definiert den Tooltipp des aktuellen Formulars bzw. gibt ihn zurück.
- `Color as Long`: Definiert die Fensterfarbe des Formulars bzw. gibt sie zurück.

2.2.5 Beispiel: Erstellen eines Formulars

Für Formulare sind Benutzereingaben nötig (Daten und/oder Treffen einer Auswahl). Auf Basis dieser Eingabe wird die gewünschte Aktion ausgeführt. In der Skripting-Umgebung können Sie Formulare grafisch entwerfen sowie Skripts zur Durchführung der gewünschten Aktion erstellen. Im Beispiel unten werden die einzelnen Schritte bei der Erstellung eines Formulars beschrieben. Sie erstellen hier ein Formular, in das der Benutzer den Namen der Elemente eingeben kann, die aus der aktiven XML-Datei in Authentic Desktop gelöscht werden sollen.

Erstellen eines neuen Formulars

Um ein neues Formular zu einem Skripting-Projekt hinzuzufügen, wählen Sie **Project | Add Form** oder klicken Sie in der Symbolleiste des Projektfensters auf die Schaltfläche **New Form**. Daraufhin wird das neue Formular sofort hinzugefügt. Es wird als separates Fenster angezeigt (siehe Abbildung unten) und sein Titel wird im Projektfenster angezeigt. Öffnen Sie das Eigenschaftsblatt für das Formular entweder durch Rechtsklick in das Formularfenster oder durch Auswahl des Befehls **Layout | Properties**.



Benennen des Formulars

Geben Sie im Eigenschaftsblatt in die Zeile `FormCode` einen Namen für das Formular ein. In diesem Beispiel wollen wir dem Formular den Namen `RemoveDlg` geben. Drücken Sie

anschließend die **Eingabetaste**. Der Name des Formulars wird nun auch im Titelfeld des Eigenschaftsblatts angezeigt.

Einfügen eines Formularobjekts

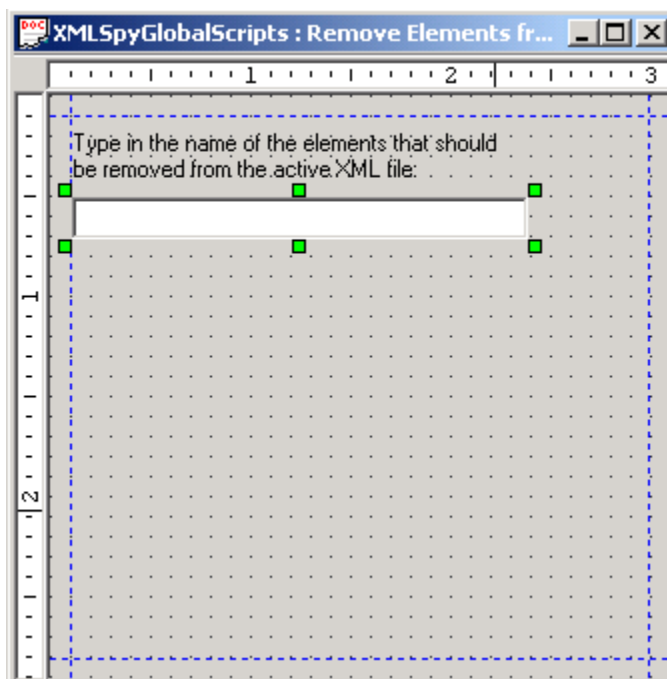
Wir benötigen im erstellten Formular ein Bearbeitungsfeld, in das der Benutzer die Namen der aus der XML-Datei zu entfernenden Elemente eingeben kann. Wählen Sie das Bearbeitungsfeldsymbol  aus der [Formularobjektleiste](#) auf der rechten Seite des Hauptfensters aus. Ziehen Sie anschließend in der Design-Ansicht des `RemoveD1`-Formulars mit einer Maus ein Rechteck für das Bearbeitungsfeld auf. Dieses Rechteck bestimmt die Position und Größe des Bearbeitungsfelds. Das Bearbeitungsfeld wird in der Design-Ansicht erstellt und das Eigenschaftsblatt für das Bearbeitungsfeld wird angezeigt (wenn **Layout | Properties** aktiv ist). (Um das Eigenschaftsblatt für das Bearbeitungsfeld anzuzeigen, wählen Sie entweder das Bearbeitungsfeld aus und aktivieren Sie den Befehl **Layout | Properties** oder klicken Sie mit der rechten Maustaste auf das Bearbeitungsfeld.) Geben Sie in die `ObjectCode` Zeile des Eigenschaftsblatts `EditElements` als den Namen des Bearbeitungsfelds ein.

Position und Größe eines Formularobjekts

Die Position und die Dimensionen des Bearbeitungsfelds werden im Eigenschaftsblatt in den Eigenschaften `Left`, `Right`, `Top` und `Bottom` definiert. Die Werte, die für die Eigenschaften `Left` und `Right` definiert werden, geben den Abstand (in Pixeln) des linken bzw. rechten Rands des Bearbeitungsfelds vom linken Rand des Formulars an. Die Werte für die Eigenschaften `Top` und `Bottom` geben den Abstand (in Pixeln) des oberen und unteren Rands des Bearbeitungsfelds vom oberen Rand des Formulars an.

Einfügen von Text in das Formular

Fügen Sie über die Formularobjektleistenschaltfläche für statischen Text  eine Überschrift für das Bearbeitungsfeld ein. Die Design-Ansicht des Formulars sollte in etwa folgendermaßen aussehen:



Einfügen von Schaltflächen

Wir benötigen nun zwei Schaltflächen in unserem Formular: **Delete** und **Cancel**. Um diese einzufügen, gehen Sie folgendermaßen vor:

1. Wählen Sie das Schaltflächensymbol  aus und ziehen Sie die beiden Schaltflächen in die Design-Ansicht.
2. Benennen Sie die einzelnen Schaltflächen in der (`ObjectCode`) Zeile des entsprechenden Eigenschaftsblatts: `BtnDelete` und `BtnCancel`.
3. Definieren Sie den Schaltflächentext (**Delete** bzw. **Cancel**) entweder durch Bearbeiten des Schaltflächentexts direkt in der Design-Ansicht oder durch Bearbeiten der Eigenschaft `Text` im Eigenschaftsblatt.

Schreiben des erforderlichen Skripts

Nachdem Sie das optische Design für das Formular fertiggestellt haben, müssen Sie nun die Formularobjekte mit den passenden Skripten verknüpfen. In diesem Fall benötigen wir zwei Arten von Skripten:

- Eine Funktion, die eine Schleife durch die XML-Datei bildet und alle `XMLData` Objekte mit einem bestimmten Namen (den der Benutzer im Bearbeitungsfeld definiert) löscht.
- Event Handler, die ausgeführt werden, wenn Sie auf die beiden Schaltflächen klicken.

Erstellen der Funktion

Die Funktion, die die `XMLData` Objekte löscht, muss im globalen Deklarationsmodul des Skripting-Projekts definiert werden. Öffnen Sie das Fenster für globale Deklarationen durch Doppelklick auf das `GlobalDeclarations` Modul im Projektfenster. Geben Sie den folgenden Code in das Fenster für globale Deklarationen ein:

```
var txtElementName;

function DeleteXMLElements( objXMLData )
{
    if( objXMLData == null )
        return;

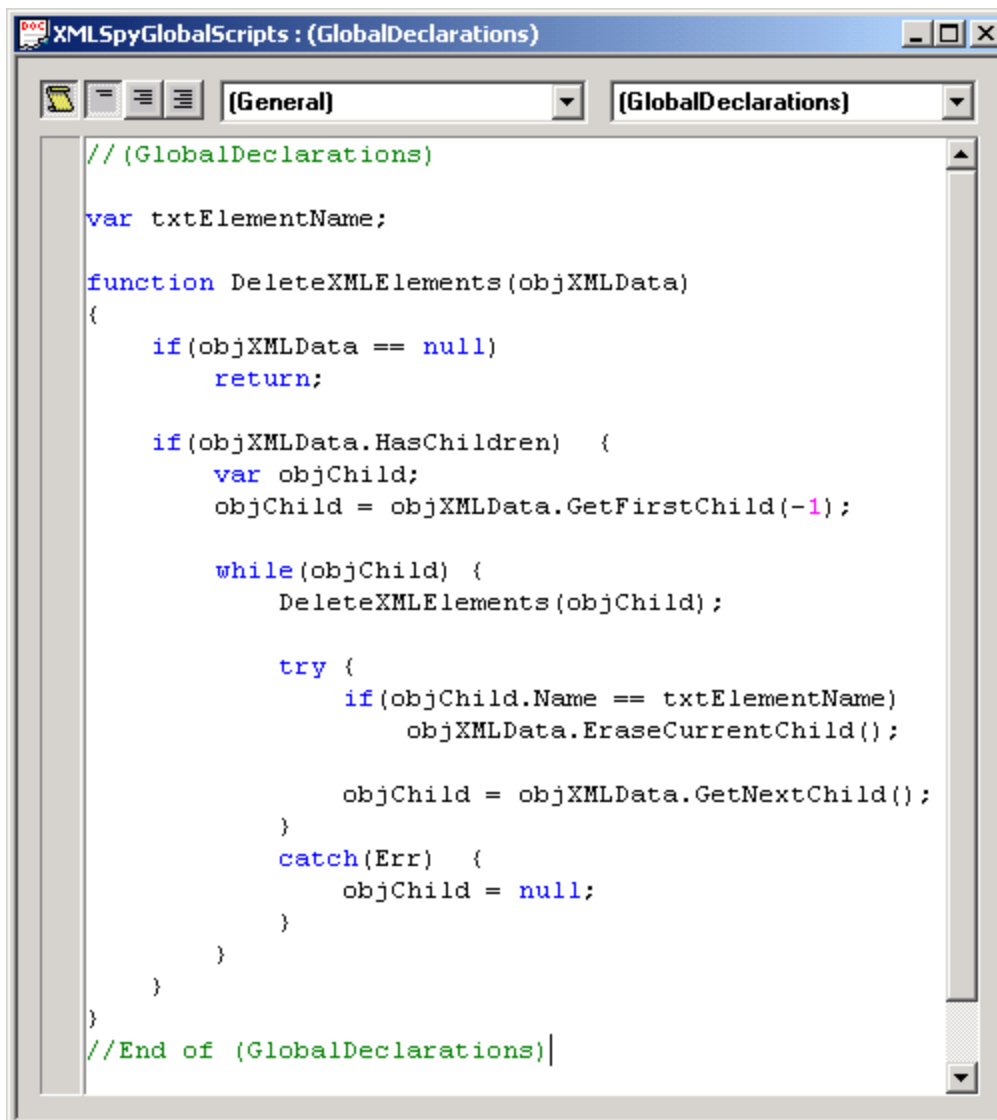
    if( objXMLData.HasChildren ) {
        var objChild;
        objChild = objXMLData.GetFirstChild( -1 );

        while( objChild ) {
            DeleteXMLElements( objChild );

            try{
                if( objChild.Name == txtElementName )
                    objXMLData.EraseCurrentChild();

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

Das Fenster für globale Deklarationen sollte folgendermaßen aussehen:



Erstellen der Schaltflächen-Events

Ein Skript kann in der Skripting-Umgebung ganz einfach mit einem Schaltflächen-Event verknüpft werden. Dies geschieht folgendermaßen:

1. Wählen Sie die Schaltfläche `BtnDelete` aus und klicken Sie im Eigenschaftsblatt auf das Register "Events".
2. Wählen Sie die Eigenschaft `EventClick` aus und klicken Sie auf die Schaltfläche mit dem Auslassungszeichen  auf der rechten Seite. Daraufhin wird ein Script Editor-Fenster mit dem vordefinierten Function Header des Klick-Events geöffnet. Diese Funktion wird bei jedem Klick auf die Schaltfläche **Delete** ausgeführt. Fügen Sie den folgenden Code zum Event hinzu:

```
if(EditElements.Text != "") {
    txtElementName = EditElements.Text;
    DeleteXMLElements(Application.ActiveDocument.RootElement);
}
```

Ausführen des Formulars

Um das Formular auszuführen, erstellen Sie ein Makro. Gehen Sie dazu folgendermaßen vor:

1. Öffnen Sie das Makro-Modul (durch Doppelklick auf `XMLSpyMacros` im Projektfenster).
2. Erstellen Sie ein neues Makro namens `DeleteElements` und geben Sie den folgenden Code ein:

```
var a;  
if( Application.ActiveDocument != null)  
    a = Application.ShowForm( "RemoveDlg" );
```

3. Führen Sie das Makro über die Oberfläche von Authentic Desktop aus. Wählen Sie dazu im Dialogfeld "Makros anzeigen" das Makro `DeleteElements` aus, das Sie durch Klicken auf **Extras | Makros anzeigen** über die Authentic Desktop Oberfläche öffnen.

Anmerkung: Sie können in einem Formular mit der Funktion `Application.ShowForm( )` ein weiteres Formular anzeigen.

2.2.6 Beispiel: Verwenden von Farben

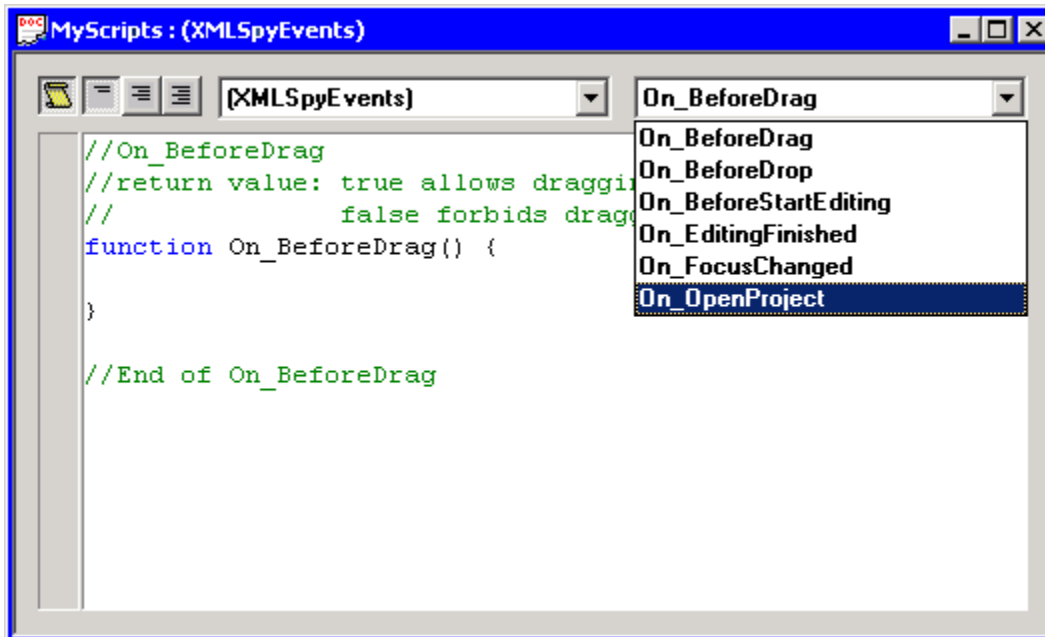
In dem Skript unten werden Farben hinzugefügt, wenn Sie auf eine Schaltfläche klicken. Das Skript wird als Event Handler für das `EventClick` Event der Schaltfläche `Button1` eingefügt (Das Event steht auf dem Register "Events" zur Verfügung, wenn die Schaltfläche im Design ausgewählt wird):

```
function Button1_EventClick() {  
    This = Button1;  
  
    // Sets the font color (red) of the button.  
    Button1.ForeColor(0x0000ff);  
  
    // Sets the button color (blue).  
    Button1.BackColor(0xff0000);  
  
    // Sets the window color (green).  
    TheView.Color(0x00ff00);  
}
```

Beachten Sie, dass die Farben RGB-Werte sind, die in Form hexadezimaler Werte im Format `0xBBGGRR` ausgedrückt werden.

2.3 Application Event Handler

Um einen Event Handler für Applikations-Events - im Gegensatz zu Formular-Events - zu erstellen, doppelklicken Sie im Projektfenster auf das XMLSpyEvents-Modul. Daraufhin wird im Hauptfenster ein Skripting-Fenster für Event Handler angezeigt.



In der Auswahlliste auf der rechten Seite sind alle Applikations-Events aufgelistet. Um ein bestimmtes Event-Handler-Skript aufzurufen, klicken Sie in der Auswahlliste auf der rechten Seite auf das entsprechende Event. Daraufhin wird das Skript angezeigt, das Sie bearbeiten können, oder es erscheint eine Skript-Vorlage, die Sie ausfüllen können. Wenn das Event, für das Sie Code schreiben wollen, in der Auswahlliste der Events keinen Eintrag hat, fügen Sie es folgendermaßen hinzu: (i) Wählen Sie im Projektfenster XMLSpyEvents auf und (II) wählen Sie im Menü **Project** oder im Kontextmenü den Befehl **Add Function**.

Anmerkung:

- Der Name des Event muss im Function Header des Event Handlers richtig geschrieben sein. Andernfalls wird der Event Handler nicht aufgerufen.
- Es können lokale Variablen und Hilfsfunktionen innerhalb von Makros und Event Handlern definiert werden. Beispiel:

```
//return value: true allows editing
//return value: false disallows editing
var txtLocal;
function Helper()
{
    txtMessage = txtLocal;
    Application.ShowForm( "MsgBox" );
}
function On_BeforeStartEditing( objXMLData )
{
    txtLocal = "On_BeforeStartEditing( )";
    Helper();
}
```

- Während der Ausführung eines Makros werden Event Handler aus Authentic Desktop nicht verarbeitet.
- Die beste Art einen gelöschten Event Handler wieder zu erstellen, ist ein neues

Skripting-Projekt zu erstellen und den Namen und die Funktionsdefinition von dort zu kopieren.

- Damit Events verarbeitet werden, muss im Authentic Desktop-Dialogfeld "Einstellungen" die Option "Events abarbeiten" aktiviert sein. Nähere Informationen dazu siehe [Skripting-Einstellungen](#).
- Siehe auch [Hinweise zur Programmierung](#).

Beispiel

In der Abbildung oben sehen Sie das XMLSpyEvents Modul mit einer Liste der vordefinierten Events. Mit Hilfe von `On_OpenProject` können Sie ein Skript hinzufügen, das jedes Mal, wenn Authentic Desktop ein Projekt öffnet, ausgeführt wird. Das Beispielskript unten öffnet der Reihe nach alle XML-Dateien im XML-Ordner des Projekts und validiert sie. Schlägt die Validierung fehl, zeigt das Skript einen Validierungsfehler an und stoppt. Wenn eine Datei gültig ist, wird sie geschlossen und die nächste Datei wird geöffnet.

Geben Sie für das `On_OpenProject()` Event das folgende Skript ein und speichern Sie das Skripting-Projekt.

```
function On_OpenProject()
{
    var bOK;
    var nIndex, nCount;
    var objItems, objXMLFolder = null;

    objItems = Application.CurrentProject.RootItems;
    nCount = objItems.Count;

    // search for XML folder
    for(nIndex = 1; nIndex <= nCount; nIndex++) {
        var txtExtensions;
        txtExtensions = objItems.Item(nIndex).FileExtensions;

        if(txtExtensions.indexOf("xml") >= 0) {
            objXMLFolder = objItems.Item(nIndex);
            break;
        }
    }

    // does XML folder exist?
    if(objXMLFolder) {
        var objChild, objDoc;

        nCount = objXMLFolder.ChildItems.Count;

        // step through associated xml files
        for(nIndex = 1; nIndex <= nCount; nIndex++) {
            objChild = objXMLFolder.ChildItems.Item(nIndex);

            try{
                objDoc = objChild.Open();

                // use JScript method to access out-parameters
                var strError = new Array(1);
                var nErrorPos = new Array(1);
                var objBadData = new Array(1);

                bOK = objDoc.IsValid(strError, nErrorPos, objBadData);

                if(! bOK) {
                    // if the validation fails, we should display the
                    // message from XMLSpy
                }
            }
        }
    }
}
```

```
// of course we have to create the form "MsgBox" and
// define the global txtMessage variable
//
// txtMessage = Position;" + nErrorPos[0] + "\n" + // strError[0];
// txtMessage += "\n\nXML:\n" + objBadData[0].Name + ", " +
//     objBadData[0].TextValue;
//
// Application.ShowForm("MsgBox");

    break;
}

objDoc.Close( true );
objDoc = null;
}
catch( Err ) {
    // displaying the error description here is a good idea

    // txtMessage = Err.Description;
    // Application.ShowForm("MsgBox");

    break;
}
}
}
```

Testen des Event Handlers

Wechseln Sie in Authentic Desktop und öffnen Sie ein Projekt, um zu sehen, wie ein Open Project Event behandelt wird.

2.4 Makros

Makros automatisieren wiederholte oder komplexe Aufgaben. In der Skripting-Umgebung können Sie ein Skript erstellen, das Applikationsfunktionen sowie Ihre eigenen im Modul für globale Deklarationen definierten Funktionen aufruft. Dadurch steht Ihnen eine gute Methode zum Automatisieren von Aufgaben in der Authentic Desktop-IDE zur Verfügung.

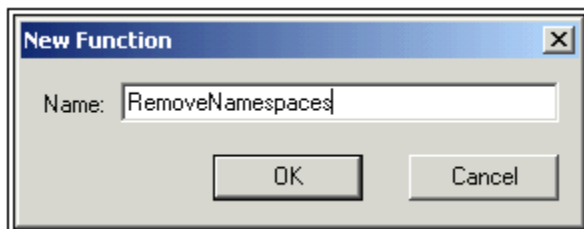
In diesem Abschnitt wird anhand eines Beispiels beschrieben, wie man ein Makro erstellt und verwendet. Mit unserem Beispielmakro werden alle Namespace-Präfixe aus dem aktiven Dokument entfernt. Die einzelnen Schritte werden der Reihe nach in den Unterabschnitten dieses Abschnitts beschrieben:

- [Erstellen eines neuen Makros](#)
- [Erstellen von Funktionen für das Makro](#)
- [Ausführen eines Makros](#)
- [Ändern eines Makros](#)

2.4.1 Erstellen eines neuen Makros

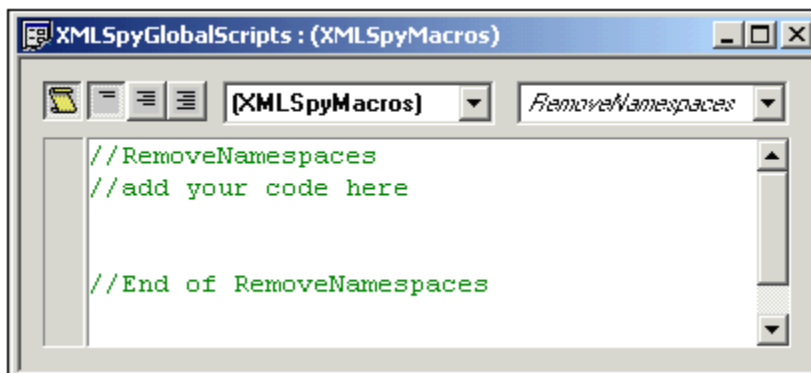
Um ein neues Makro zu erstellen, gehen Sie folgendermaßen vor:

1. Klicken Sie mit der rechten Maustaste im Projektfenster auf XMLSpyMacros und wählen Sie im Kontextmenü den Befehl **Add Function**.
2. Geben Sie im Dialogfeld "New Function", das daraufhin angezeigt wird (*Abbildung unten*), `RemoveNamespaces` für den Namen des Makros ein.



In der Skripting-Umgebung wird ein neues Makro mit diesem Namen erstellt und zum XMLSpyMacros Modul hinzugefügt.

3. Doppelklicken Sie im Projektfenster auf das XMLSpyMacros Modul, um das Makro-Fenster zu öffnen.
4. Wählen Sie das `RemoveNamespaces` Makro in der Auswahlliste auf der rechten Seite aus (*siehe Abbildung unten*).



5. Geben Sie den unten stehenden Code in das Makro-Fenster ein:

```
if( Application.ActiveDocument != null) {  
  
    RemoveAllNamespaces( Application.ActiveDocument.RootElement);  
    Application.ActiveDocument.UpdateViews();  
}
```

Anmerkung: Makros unterstützen keine Parameter oder Rückgabewerte, daher sollte in der Makroimplementierung kein Function Header vorhanden sein.

2.4.2 Erstellen von Funktionen für das Makro

Nun müssen wir die `RemoveAllNamespaces` Funktion erstellen. Wenn wir sie im Modul für globale Deklarationen erstellen, steht sie für alle Makros, Event Handler und Formulare zur Verfügung.

1. Aktivieren Sie das Modul für globale Deklarationen durch Doppelklicken in der Projektfenster.
2. Hängen Sie den folgenden Code für die Funktion `RemoveAllNamespaces()` an den Code für alle bereits zuvor definierten globalen Variablen oder Funktionen an:

```
function RemoveAllNamespaces( objXMLData)
{
    if(objXMLData == null)
        return;

    if( objXMLData.HasChildren)    {
        var objChild;

        // spyXMLDataElement := 4
        objChild = objXMLData.GetFirstChild(4);

        while( objChild) {
            RemoveAllNamespaces( objChild);

            try{
                var nPos,txtName;
                txtName = objChild.Name;

                if(( nPos = txtName.indexOf(":") ) >= 0) {
                    objChild.Name = txtName.substring( nPos+1);
                }

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

Damit ist das neue Makro fertig erstellt.

Anmerkung:

- Sie können in Makros und Event Handlern lokale Variablen und Hilfsfunktionen definieren. Beispiel:

```
//return value: true allows editing
//return value: false disallows editing
var txtLocal;
function Helper()
{
    txtMessage = txtLocal;
    Application.ShowForm( "MsgBox" );
}
function On_BeforeStartEditing( objXMLData)
{
    txtLocal = "On_BeforeStartEditing() ";
    Helper();
}
```

- Rekursive Funktionen werden unterstützt. Die Funktion `DeleteXMLElements()` in

[Beispiel: Erstellen eines Formulars](#) z.B. ist rekursiv.

2.4.3 Ausführen eines Makros

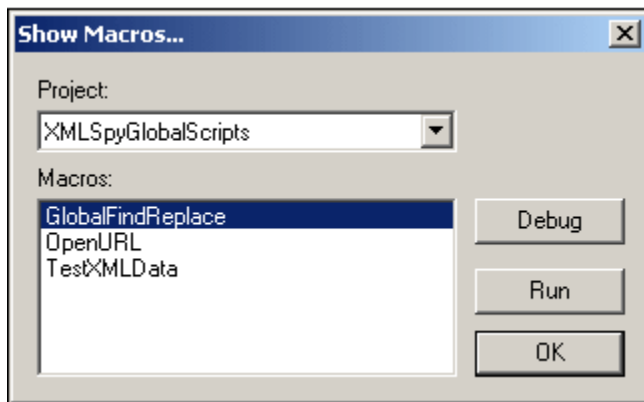
Nun müssen wir das Makro ausführen. Es gibt dafür drei Methoden:

- [Über das Dialogfeld "Makros anzeigen"](#) von Authentic Desktop.
- [Durch Erstellen und Verwenden eines Menübefehls](#) (oder Befehl **Ausführen**) für das Makro im Authentic Desktop-Menü **Extras**.
- [Durch Erstellen und Verwenden einer Symbolleisten-Schaltfläche](#) für ein Makro.

Beachten Sie: Während der Ausführung eines Makros werden Event Handler aus Authentic Desktop nicht verarbeitet.

Dialogfeld "Makros anzeigen"

Das Dialogfeld "Makros anzeigen" (*Abbildung unten*) wird über das Menü **Extras** von Authentic Desktop aufgerufen. Es zeigt alle im globalen Skripting-Projekt und dem aktuell geöffneten Applikationsprojekt definierten Makros.



Wählen Sie in der Auswahlliste der Projekte entweder das globale Skripting-Projekt oder das dem Authentic Desktop-Projekt zugewiesene Skripting-Projekt aus. Wählen Sie anschließend das gewünschte Makro aus und klicken Sie auf **Ausführen**. Bevor das Makro ausgeführt wird, wird das Dialogfeld "Makros anzeigen" geschlossen.

Befehl im Menü "Extras"

Die XMLSpy API enthält eine Funktion `AddMacroMenuItem()`, um Makros als Menübefehle zum Menü **Extras** hinzuzufügen. Mit Hilfe dieser Funktion können Sie ein oder mehrere Makros zur Liste der Makros aus **Extras | Makros anzeigen** hinzufügen. Normalerweise tun Sie dies folgendermaßen:

1. Fügen Sie den Makromenübefehl durch Aufruf der XMLSpy API Funktion `AddMacroMenuItem()` hinzu.

```
Application.AddMacroMenuItem("DeleteElements", "Delete Elements Dialog");
```

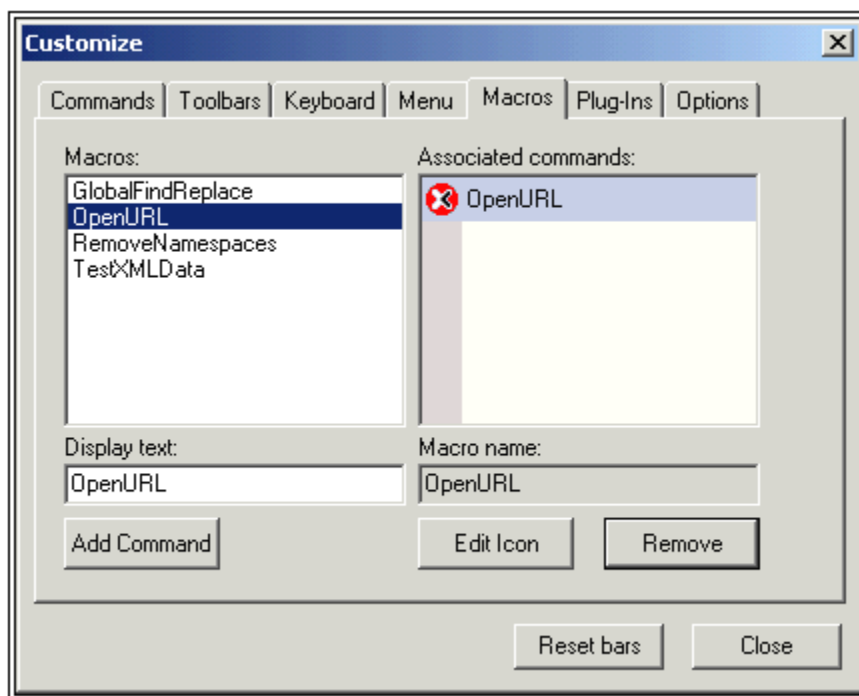
 - Der erste Parameter der Funktion (`DeleteElements` im Beispiel) ist der Name des Makros. Wenn Sie das Makro ausführen und ein Projekt geöffnet ist, mit dem Skripts verknüpft sind, sucht Authentic Desktop zuerst in den Projektskripts nach dem Makro. Wenn keine Projekt-Skripts vorhanden sind, oder wenn Authentic Desktop das Makro nicht finden kann, wird die Suche nach dem Makro in den globalen Skripts fortgesetzt.
 - Der zweite Parameter (`Delete Elements Dialog`) ist Anzeigetext für den Menübefehl.

2. Setzen Sie das Menü **Extras** durch Aufruf von `ClearMacroMenu()` zurück. Damit werden alle zuvor hinzugefügten Menübefehle entfernt.

Am besten lassen sich diese beiden Funktionen über das `Autorun` Makro des globalen Skripting-Projekts oder das `On_OpenProject` Event ausführen.

Symbolleisten-Schaltfläche

Sie können in der Symbolleiste eine Schaltfläche erstellen, mit der bei Anklicken ein Makro ausgeführt wird. Klicken Sie dazu auf **Extras | Anpassen | Makros**. Daraufhin wird das Register "Makros" des Dialogfelds "Anpassen" angezeigt (*Abbildung unten*).



Gehen Sie nun folgendermaßen vor:

1. Wählen Sie auf dem Register "Makros" das gewünschte Makro aus.
2. Geben Sie in das Eingabefeld "Text anzeigen" den Namen der Schaltfläche ein. Dieser Name wird angezeigt, wenn Sie den Cursor in der Symbolleiste über diese Schaltfläche platzieren.
3. Klicken Sie auf **Befehl hinzufügen**, um ihn zur Liste der Befehle hinzuzufügen.
4. Wählen Sie den Befehl aus und klicken Sie auf **Symbol bearbeiten**, um ein neues Symbol zu erstellen.
5. Ziehen Sie das fertige Symbol aus dem Bereich "Zugewiesene Befehle" in die Symbolleiste und lassen Sie die Maustaste los, sobald sich der Cursor-Pfeil in ein I-Symbol verwandelt.
6. Klicken Sie auf **Menüs zurücksetzen**, damit dieses Makrosymbol und der Befehl im Dialogfeld **Extrass | Makros anzeigen** angezeigt werden.

2.4.4 Ändern eines Makros

Das von uns erstellte Makro kann geändert werden, um Namespaces umzubenennen statt sie zu entfernen. Gehen Sie dabei folgendermaßen vor :

1. Erstellen Sie ein Formular (siehe [Formulare](#)), im dem der Benutzer den Namen des alten und des neuen Namespace angeben kann.
2. Ändern Sie im Makro-Code den `try-catch` Block der Funktion `RemoveAllNamespaces()` in Code wie z.B. den folgenden um:

```
try{
    var nPos, txtName;
    txtName = objChild. Name;

    if(( nPos = txtName.indexOf( ":" ) ) >= 0) {
        var txtOld;
        txtOld = txtName.substring( 0, nPos );

        if( txtOld == txtOldNamespace )
            objChild. Name = txtNewNamespace + ":" + txtName.substring( nPos+1 );
    }

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

Bei diesem Code wird angenommen, dass `txtOldNamespace` und `txtNewNamespace` als globale Variablen deklariert sind und mit den richtigen Werten gesetzt werden.

2.5 Menüs

In diesem Abschnitt werden alle in der Skripting-Umgebung verfügbaren Menübefehle aufgelistet. Auf der Benutzeroberfläche der Skripting-Umgebung stehen diese Befehle in den folgenden Menüs zur Verfügung:

- [File](#): Befehle zum Behandeln von Skripting-Projekten, Drucken und Wechseln zwischen der Skripting-Umgebung und Authentic Desktop Fenstern.
- [Edit](#): Standard-Textverarbeitungsbefehle.
- [View](#): Befehle zum Konfigurieren der Benutzeroberfläche der Skripting-Umgebung und der Anordnung von Objekten in einem Formular.
- [Project](#): Befehle, die das aktive Projekt betreffen - zum Hinzufügen und Entfernen von Funktionen
- [Layout](#): Befehle zum Definieren des Formular-Layouts.
- [Draw](#): Befehle zum Verwalten und Organisieren von Objekten in einem Formular.
- [Window](#): Befehle zum Verwalten der Fenster in der GUI der Skripting-Umgebung.

2.5.1 File

New Project

Erstellt ein neues Projekt. Sie werden nach der Skript-Sprache für das neue Projekt gefragt. Sie können entweder JavaScript oder VBScript verwenden. Das Projekt wird als das aktive Projekt geöffnet und der Titel wird fett im Projektfenster angezeigt.

Open Project

Öffnet ein bestehendes Projekt. Das neu geöffnete Projekt wird zum aktiven Projekt. Von allen offenen Projekten kann immer nur eines aktiv sein. Ein Projekt wird automatisch zum aktiven Projekt, wenn es geöffnet oder als neues Projekt erstellt wird. Andernfalls wird ein Projekt aktiv gemacht, indem Sie dies explizit definieren - rechtsklicken Sie auf einen Projekttitel und wählen Sie den Befehl **Set as Active Project**.

Wenn nur eine Instanz des Projekts geöffnet ist, wird der Titel fett angezeigt. Es können mehrere Instanzen desselben Projekts geöffnet sein. In diesem Fall wird der Titel der gerade aktiven Instanz bzw. der Titel der zuletzt aktiven Instanz fett angezeigt.

Close Project

Schließt das ausgewählte Projekt. Wenn ein Projekt nicht das aktive Projekt ist, wird es durch das Schließen automatisch zum aktiven Projekt gemacht. Nachdem es geschlossen wurde, ist kein Projekt aktiv.

Save Project

Speichert das ausgewählte Projekt.

Save Project As

Speichert ein nicht benanntes offenes Projekt unter einem Namen oder ein benanntes Projekt unter einem anderen Namen.

Printing

Mit dem Befehl "Print" können Sie ein Formular oder Skript drucken. Auch eine Druckvorschau steht zur Verfügung. Beachten Sie: Um aus der Druckvorschau wieder in die Skripting-Umgebung zu wechseln, klicken Sie auf die Schaltfläche **Close**. Schließen Sie nicht das Druckvorschau-Fenster, da sonst die gesamte Skripting-Umgebung geschlossen wird.

Switch to XMLSpy

Minimiert die Skripting-Umgebung und stellt das Authentic Desktop Fenster wieder her.

Exit

Schließt das Fenster der Skripting-Umgebung.

2.5.2 Edit

Undo und Redo

Mit Hilfe der Befehle Undo und Redo können Sie mehrere Befehle zum Rückgängigmachen und Wiederherstellen ausführen.

Cut, Copy, Paste, Delete

Zum Ausschneiden, Kopieren, Einfügen und Löschen, wählen Sie ein Objekt oder Textfragment aus und wählen Sie einen dieser Befehle aus. Platzieren Sie den Cursor zum Einfügen an die gewünschte Stelle und wählen Sie den Befehl "Paste" aus.

2.5.3 View

Toolbars

Blendet eine Reihe von Symbolleisten ein/aus.

Project Bar

Blendet die Projektleiste ein/aus.

Status Bar

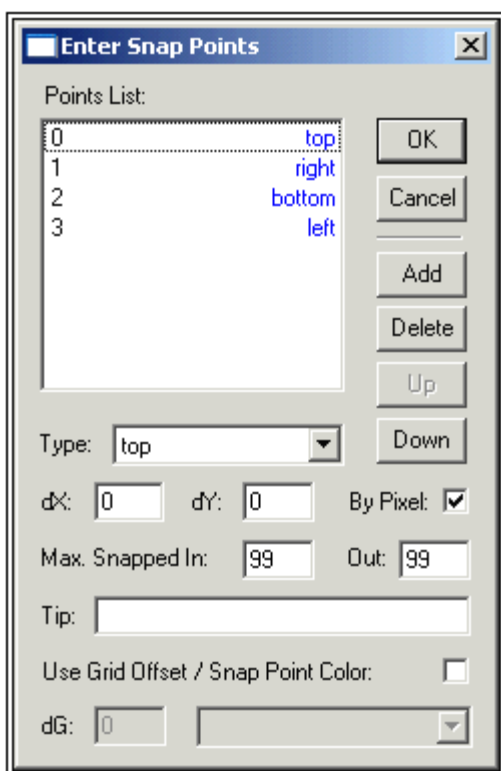
Blendet die Statusleiste am unteren Rand der Skripting-Umgebung ein/aus.

Ruler Bars

Blendet die Linealleisten am linken und oberen Rand der Seite ein/aus.

Snap Points

Blendet die Ausrichtungspunkte eines Objekts ein/aus. Die Ausrichtungspunkte sind die Punkte eines Objekts, anhand derer es an einem anderen Objekt ausgerichtet wird, wenn dieses Objekt verschoben wird. Sie können die Ausrichtungspunkte hinzufügen und löschen, indem Sie das Objekt auswählen und im Eigenschaftsblatt des Objekts auf die Auslassungspunkte im Wertefeld der Eigenschaft AnchorSnaps klicken. Daraufhin wird das Dialogfeld "Enter Snap Points" angezeigt, in dem Sie Ausrichtungspunkte hinzufügen können.



Wenn die Ausrichtungspunkte eingblendet sind, sind sie sichtbar und helfen Ihnen, Objekte exakt zu positionieren.

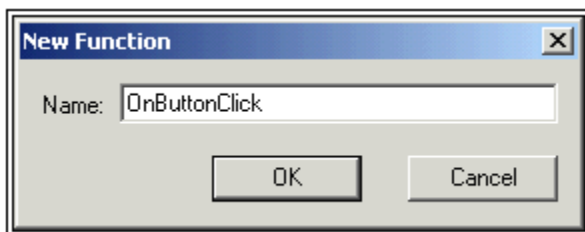
2.5.4 Project

Add Form

Dient zum Hinzufügen eines Formulars zu den aktuellen Formularen im Projekt.

Add Function

Wenn "Events" oder "Macros" im Projektfenster ausgewählt ist, können Sie eine Funktion zur Liste der bereits vorhandenen Funktionen hinzufügen. Bei Auswahl dieses Befehls erscheint das folgende Dialogfeld:



Bei Eingabe des Namens der Funktion und Klicken auf OK wird eine Funktion mit diesem Namen zur Liste der vorhandenen Funktionen hinzugefügt und ein Skripting-Fenster für diese Funktion geöffnet.

Remove Function

Mit diesem Befehl wird das Dialogfeld "Remove Function" geöffnet, in das Sie den Namen der zu entfernenden Events- oder Makros-Funktion eingeben.

Project Information

Mit diesem Befehl erhalten Sie den Pfad zur aktuellen Projektdatei und die Information, ob das Projekt von Authentic Desktop aus geöffnet wurde oder nicht.

2.5.5 Layout

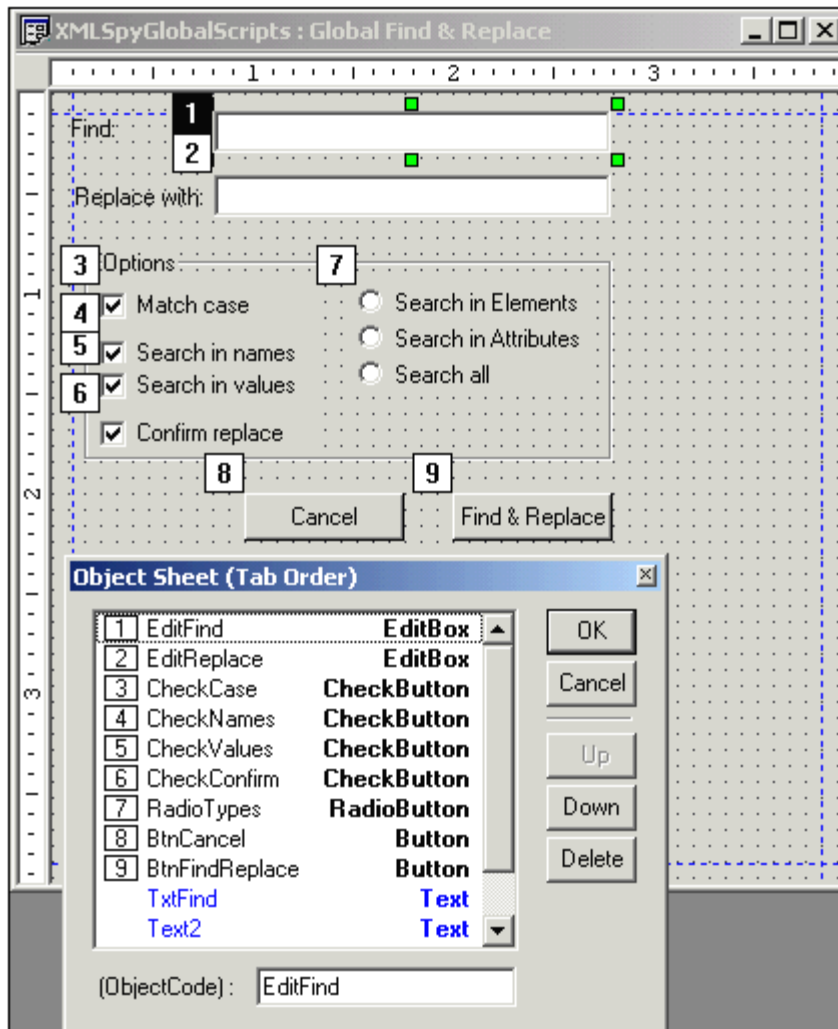
Das Menü "Layout" enthält Befehle für das Formular-Layout und zur Steuerung der Ansicht. Diese Befehle sind für Module nicht relevant.

Einstellen von Abständen, Positionierung und Anpassen der Größe von Objekten

Wenn zwei oder mehr Formularobjekte ausgewählt wurden (durch Drücken der Umschalttaste, während die Objekte angeklickt werden), können sie in gleichem Abstand ausgerichtet werden, in der Größe an das andere Objekt angepasst und relativ zum Rand des Formulars positioniert werden. Verwenden Sie dazu die Befehle **Align Objects**, **Space Evenly**, **Align in View** und **Make Same Size**. Beachten Sie: Um den Befehl **Space Evenly** verwenden zu können, müssen mindestens drei Objekte ausgewählt sein.

Objects

Mit dem Befehl **Objects...** wird das Fenster "Object Sheet (Tab Order)" aufgerufen, in dem Sie die Tabulatorreihenfolge für die Dateneingabefelder festlegen können. Dateneingabefelder entsprechen Dateneingabe-Formularobjekten.



Wenn Sie ein Formularobjekt in Ihr Design einfügen, wird das neue Objekt an den entsprechenden Abschnitt (nummeriert oder nicht nummeriert) in der Objektblatliste angehängt. Die Objektblatliste besteht aus zwei Teilen, einem nummerierten Teil für Dateneingabeobjekte und einem nicht nummerierten Teil für Anzeigeobjekte. Die Nummer

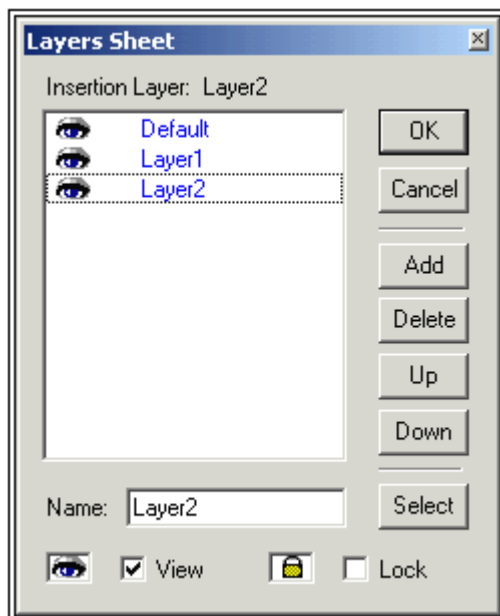
eines Objekts im nummerierten Teil gibt die Position dieses Objekts in der Laufzeit-Tabulatorreihenfolge an. Sie können die Position eines Objekts in der Tabulatorreihenfolge ändern, indem Sie das Objekt auswählen und je nach Bedarf auf die **Up**- oder **Down**-Schaltfläche klicken. Sie können ein Objekt auch löschen, indem Sie auf die Schaltfläche **Delete** klicken.

Anmerkung: Die Tabulatorreihenfolge wird nur zur Laufzeit auf Dateneingabeobjekte angewendet, d.h. wenn das Formular für die Benutzereingabe angezeigt wird. Wenn Sie im Formular-Editor die Tabulatortaste drücken, springen Sie zuerst durch die Anzeigeobjekte (in der Reihenfolge, in der sie im Objektblatt aufgelistet sind) und anschließend durch die Dateneingabeobjekte (in der Tabulatorreihenfolge).

Ebenen

Ein Formular wird in einer "Standardebene" erstellt. Diese Ebene ist immer die oberste Ebene. Sie können unterhalb der Standardebene weitere Ebenen hinzufügen und diese Ebenen relativ zueinander verschieben. Die Standardebene bleibt immer die oberste Ebene. Mit Hilfe von Ebenen können Sie daher Hintergrundeffekte zu Ihrem Formular hinzufügen.

Um Ebenen hinzuzufügen und zu bearbeiten, wählen Sie den Befehl **Layers...** aus. Daraufhin wird das Ebenenblatt (Layers Sheet) angezeigt.



Um eine Ebene hinzuzufügen, klicken Sie auf die Schaltfläche **Add**. Sie können eine Ebene umbenennen, indem Sie sie auswählen und einen neuen Namen in das Eingabefeld **Name** eingeben. Um eine Ebene zu löschen, wählen Sie sie aus und klicken Sie auf die Schaltfläche **Delete**. Um eine Ebene in der Reihenfolge nach oben oder unten zu verschieben, wählen Sie sie aus und klicken Sie je nach Bedarf auf die Schaltfläche **Up** oder **Down**. Die Reihenfolge der Ebenen ist so geordnet, dass die erste Ebene im Ebenenblatt der obersten Ebene des Formulars entspricht.

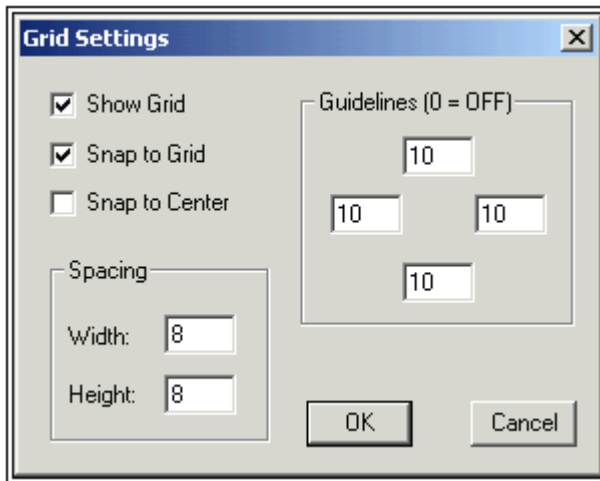
Mit der Schaltfläche **Select** werden in der Design-Ansicht alle Objekte der gerade ausgewählten Ebene ausgewählt. Außerdem können Sie alle Objekte einer Ebene über das Kontrollkästchen **View** ein- oder ausblenden. Wenn Sie eine Ebene mit Hilfe des Kontrollkästchens **Lock** sperren, wird diese Ebene für die Bearbeitung gesperrt und die Objektleiste erscheint ausgegraut, wenn diese Ebene aktiv ist.

Properties

Mit dem Befehl **Properties...** wird das Eigenschaftsblatt für das ausgewählte Formular aufgerufen. Es gibt nicht nur für jedes Formular, sondern auch für jedes Formularobjekt ein Eigenschaftsblatt. Sie können die Eigenschaften eines Formulars oder Formularobjekts über das Eigenschaftsblatt ändern. Eine Beschreibung zur Verwendung des Eigenschaftsblatts finden Sie unter [Editor-Ansichten](#).

Grid Settings

Mit dem Befehl **Grid Settings...** wird das Dialogfeld "Grid Settings" aufgerufen, in dem Sie das Aussehen des Hintergrundrasters für die Design-Ansicht definieren können.



Abstandseinstellungen regeln den horizontalen und vertikalen Abstand zwischen Rasterlinien. Es stehen vier Richtlinien (guidelines) (oben, rechts, unten, links) zur Verfügung. Sie können den Abstand der einzelnen Linien vom entsprechenden Rand des Formulars definieren. Die Rastereinstellungen gelten, sobald die nächste Design-Ansicht aktiviert wird.

Edit View

Dies ist eine Ein/Aus-Einstellung, die für jedes Formular einzeln festgelegt wird. Wenn Sie den Befehl **Edit View** aktivieren, wird das Formular in einer editierbaren Design-Ansicht geöffnet, bzw. es wechselt von der nicht editierbaren Ansicht in die editierbare Design-Ansicht. Wenn Sie den Befehl **Edit View** deaktivieren, wird das Formular in einer nicht editierbaren Ansicht geöffnet bzw. es wechselt von der editierbaren Design-Ansicht in die nicht editierbare Ansicht. Beachten Sie, dass diese Einstellung nur für das aktive Formular und nicht für alle Formulare gilt.

Edit Script

Dies ist ein Ein/Aus-Befehl, der für jedes Formular einzeln festgelegt wird. Wenn Sie den **Edit Script** Befehl aktivieren, wird für dieses Formular ein separates Fenster angezeigt. Wenn Sie den Befehl deaktivieren, wird das Skript-Fenster geschlossen.

2.5.6 Draw

Order

Dies ist ein Form Editor-Befehl, mit dem Sie das ausgewählte Objekt vor oder hinter andere Objekte auf derselben Ebene platzieren können. Wenn Sie Objekte anordnen möchten, die sich auf unterschiedlichen Ebenen befinden, sollten Sie die Ebenen im Ebenenblatt bearbeiten (Aufruf mit dem Befehl **Layout | Layers**).

Zoom

Die ist ein Form Editor-Befehlsmenü über das Sie ein Objekt oder das ganze Formular in der Anzeige vergrößern oder verkleinern können.

Group

Dies ist ein Form Editor-Befehl, der aktiv ist, wenn zwei oder mehr Objekte oder eine Gruppe von Objekten ausgewählt ist. Sie können damit zwei oder mehr Objekte gruppieren und die Gruppierung aufheben. Dieser Befehl ist nützlich, wenn Sie Objekte in einer Gruppe zusammenhalten möchten.

Rotate

Dient zum Drehen bestimmter Objekte. Der Befehl **Free Rotate** ist ein Ein/Aus-Befehl. Sie können damit ein Objekte an einem seiner Ziehpunkte nehmen und um seinen Mittelpunkt drehen. Sie können auch ein Drehung um 90 ° nach rechts durchführen (**Rotate Right**). Der Befehl **Unrotate** hat denselben Effekt wie der Befehl "Rückgängig".

2.5.7 Window

Über das Menü "Window" können Sie die Anordnung der geöffneten Fenster im Hauptfenster festlegen. Sie können Fenster überlappend und nebeneinander anordnen und Symbole anordnen. Über dieses Menü können Sie außerdem das Fenster, das Sie zum aktiven machen möchten, aus einer Liste aller offenen Fenster auswählen.

2.6 Anmerkungen zur Programmierung

Beachten Sie bitte bei der Programmierung die folgenden Punkte:

- Für Out-Parameter aus Methoden der XMLSpy API werden in JavaScript spezielle Variablen benötigt. In der Folge finden Sie einige Beispiele.

```
// use JavaScript method to access out-parameters  
var strError = new Array(1);  
var nErrorPos = new Array(1);  
var objBadData = new Array(1);  
bOK = objDoc.IsValid( strError, nErrorPos, objBadData );
```

3 Plugins

Sie können in Authentic Desktop Ihre eigenen IDE-Plug-Ins erstellen und in Authentic Desktop integrieren.

Plug-Ins können zu folgenden Zwecken verwendet werden:

- zum Konfigurieren Ihrer Version von Authentic Desktop, zum Hinzufügen von Befehlen in Form von Menübefehlen, Symbolleisten-Schaltflächen, Schaltflächen usw.
- zum Reagieren auf Events aus Authentic Desktop.
- zum Ausführen Ihres eigenen Codes in Authentic Desktop, wobei Sie Zugriff auf die komplette Authentic Desktop API haben.

Authentic Desktop erwartet von Ihrem Plug-In, dass es die [IXMLSpyPlugin](#) Schnittstelle implementiert. Ein Beispiel mit C++ finden Sie unter [ATL-Beispieldateien](#). Ein Beispiel, in dem VisualBasic verwendet wird, finden Sie im folgenden Ordner Ihrer Authentic Desktop Installation.

```
C:\Documents and Settings\<username>\My Documents\Altova\Authentic2009  
\AuthenticExamples\XMLSpyPlugIn
```

3.1 Registrierung von IDE Plug-Ins

Authentic Desktop enthält einen speziellen Schlüssel in der Registrierungsdatei, wo alle registrierten IDE-Plug-Ins gespeichert sind:

HKEY_CURRENT_USER\Software\Altova\XML Spy\PlugIns

Alle Werte dieses Schlüssels werden als Referenzen auf registrierte Plug-Ins behandelt und müssen dem folgenden Format entsprechen:

Value name:	ProgID of the plug-in
Value type:	must be REG_SZ
Value data:	CLSID of the component

Bei jedem Start der Applikation werden die Werte des "PlugIns" Schlüssels gelesen und die registrierten Plug-Ins werden geladen.

Manuelle Registrierung von Plug-Ins

Um ein Plug-In manuell zu registrieren, verwenden Sie in Authentic Desktop im Menü "Extras" das Dialogfeld "Anpassen". Klicken Sie auf die Schaltfläche "Plug-in hinzufügen..." um die DLL anzugeben, die Ihr Plug-In implementiert. Authentic Desktop registriert die DLL als einen COM Server und fügt den entsprechenden Eintrag in seinem "PlugIns" Schlüssel hinzu.

Wenn Sie bei der manuelle Registrierung auf Probleme stoßen, können Sie überprüfen, ob die CLSID Ihres Plug-In im "PlugIns" Schlüssel korrekt registriert ist. Wenn dies nicht der Fall ist, war der Name Ihrer Plug-In DLL wahrscheinlich nicht eindeutig genug. Verwenden Sie einen anderen Namen oder führen Sie eine direkte Registrierung durch.

Direkte Registrierung von Plug-Ins

Ein Plug-In kann direkt als IDE Plug-In registriert werden. Dazu muss zuerst die DLL registriert und anschließend müssen z.B. beim Einrichten des Plug-In die entsprechenden Werte zum "PlugIns" Schlüssel von Authentic Desktop hinzugefügt werden. Das neue Plug-In wird beim nächsten Start von Authentic Desktop aktiviert.

3.2 ActiveX Controls

ActiveX Controls werden unterstützt. Jedes IDE PlugIn, das auch ein ActiveX Control ist, wird in einer Dialogfeldsteuerleiste angezeigt. Im Ordner `Examples` Ihres Ordners `XMLSpyPlugInActiveX` des Applikationsordners finden Sie auch ein Beispiel-Plug-In, das auch ein ActiveX Control ist.

3.3 Konfigurations-XML

Mit Hilfe des IDE-Plug-In können Sie die Benutzeroberfläche von Authentic Desktop ändern. Dazu wird jede einzelne Änderung mit Hilfe eines XML-Datenstroms beschrieben. Die XML-Konfiguration wird über die Methode [GetUIModifications](#) der `IXMLSpyPlugIn`-Schnittstelle an Authentic Desktop übergeben.

Die XML-Datei für das IDE Plug-In, die die Änderungen an der Benutzeroberfläche enthält, muss die folgende Struktur haben:

```
<ConfigurationData>
  <ImageFile>path To image file</ImageFile>
  <Modifications>
    <Modification>
      ...
    </Modification>
    ...
  </Modifications>
</ConfigurationData>
```

Sie können Symbole oder Symbolleisten-Schaltflächen für die neuen Menüeinträge, die über das Plug-In zur Benutzeroberfläche von Authentic Desktop hinzugefügt werden, definieren. Der Pfad zur Datei, die das Bild enthält, wird über das Element `ImageFile` definiert. Jedes Bild muss 16 x 16 Pixel haben und darf maximal 256 Farben enthalten. Die Bildreferenzen müssen in einer einzigen (`<ImageFile>...`) Zeile von links nach rechts angeordnet sein. Der am weitesten rechts gelegene Bild-Indexwert ist Null.

Das Element `Modification` kann beliebig viele `Modification` Child-Elemente enthalten. Jedes `Modification`-Element definiert eine bestimmte Änderung an der Standardbenutzeroberfläche von Authentic Desktop. Ab Version 4.3 können Benutzeroberflächenelemente auch aus Authentic Desktop entfernt werden.

Struktur von Modification-Elementen

Alle Modification-Elemente bestehen aus den folgenden beiden Child-Elementen:

```
<Modification>
  <Action>Type of action</Action>
  <UIElement Type="type of UI element">
  </UIElement>
</Modification>
```

Gültige Werte für das Element `Action` sind:

Add - um das darauf folgende Benutzeroberflächenelement zu Authentic Desktop hinzuzufügen
 Hide - um das darauf folgende Benutzeroberflächenelement in Authentic Desktop auszublenden
 Remove - um das Benutzeroberflächenelement im Dialogfeld "Anpassen" aus dem Listenfeld "Befehle" zu entfernen.

Sie können Werte des `Action`-Elements kombinieren z.B. "Hide Remove"

Das Element `UIElement` beschreibt alle neuen oder bestehenden Benutzeroberflächenelemente für Authentic Desktop. Mögliche Elemente sind derzeit: neue Symbolleisten, Schaltflächen, Menüs oder Menübefehle. Das Attribut **type** definiert, welches Benutzeroberflächenelement vom XML-Element beschrieben wird.

Gemeinsame Child-Elemente von UIElement

Die Elemente "ID" und "Name" sind für alle verschiedenen Arten von XML

UIElement-Fragmenten gültig. Es ist jedoch möglich, einen der Werte für einen bestimmten Typ von UIElement zu ignorieren. So wird z.B. "Name" für ein Trennzeichen ignoriert.

```
<ID></ID>
<Name></Name>
```

Wenn **UIElement** ein bestehendes Element der Benutzeroberfläche beschreibt, wird der Wert des Elements "ID" von Authentic Desktop vorgegeben. Normalerweise sind diese ID-Werte dem User nicht bekannt. Wenn das ID-Fragment einen neuen Teil der Benutzeroberfläche beschreibt, so kann die ID beliebig gewählt werden und der Wert sollte weniger als 1000 sein. Das Element **Name** definiert den Textwert. Bestehende UI-Elemente können nur nach ihrem Namen identifiziert werden, z.B. Menüs und Menübefehle mit den dazugehörigen Untermenüs. Bei neuen Benutzeroberflächenelementen definiert das Element **Name** die Überschrift, z.B. den Titel einer Symbolleiste oder den Text eines Menübefehls.

Symbolleisten und Menüs

Um eine Symbolleiste zu definieren, muss die ID und/oder der Name der Symbolleiste definiert werden. Eine bestehende Symbolleiste kann nur durch den Namen oder die ID, unter der sie bekannt ist, definiert werden. Um eine **neue** Symbolleiste zu erstellen, müssen beide Werte definiert werden. Das Attribut **type** muss "ToolBar" sein.

```
<UIElement Type="ToolBar">
  <ID>1</ID>
  <Name>TestPlugIn</Name>
</UIElement>
```

Um ein Authentic Desktop Menü zu definieren, benötigen Sie zwei Parameter:

- die ID der Menüleiste, die das Menü enthält. Wenn im Hauptfenster keine XML-Dokumente geöffnet sind, ist die ID der Menüleiste 128. Wenn ein oder mehrere Dokumente geöffnet sind, ist die ID der Menüleiste 129.
- den Menünamen. Menüs haben keinen dazugehörigen ID-Wert. Im folgenden Beispiel wird das Menü "Edit" (Bearbeiten) der Menüleiste definiert, die aktiv ist, wenn mindestens ein XML-Dokument geöffnet ist:

```
<UIElement Type="Menu">
  <ID>129</ID>
  <Name>Edit</Name>
</UIElement>
```

Wenn Sie ein neues Menü erstellen möchten, wird ein zusätzliches Element verwendet. Das Element **Place** definiert die Position des neuen Menüs in der Menüleiste:

```
<UIElement Type="Menu">
  <ID>129</ID>
  <Name>PlugIn Menu</Name>
  <Place>12</Place>
</UIElement>
```

Ein Wert von -1 für das Element **Place** definiert die neue Schaltfläche oder den neuen Menübefehl am Ende des Menüs oder der Symbolleiste.

Befehle

Wenn Sie einen neuen Befehl in Form einer Symbolleisten-Schaltfläche oder eines Menübefehls hinzufügen, kann das **UIElement**-Fragment jedes beliebige dieser Subelemente enthalten:

```
<MacroName></MacroName>
<Info></Info>
<ImageID></ImageID>
```

Wenn **MacroName** definiert ist, sucht Authentic Desktop in der Skripting-Umgebung nach einem Makro dieses Namens und führt es jedes Mal, wenn dieser Befehl aufgerufen wird, aus. Das Element **Info** enthält einen kurzen Beschreibungsstring, der in der Statusleiste angezeigt wird, wenn sich der Mauszeiger über dem entsprechenden Befehl (Schaltfläche oder Menübefehl) befindet. **ImageID** definiert den Index des Symbols in der externen Bilddatei. Bitte beachten Sie, dass alle Symbole in einer Bilddatei gespeichert sind.

Um einen Symbolleistenbefehl zu definieren, erstellen Sie ein **UIElement** mit der folgenden Struktur:

```
<UIElement Type="ToolBarItem">
  <!--don't reuse local IDs even the commands do the same-->
  <ID>5</ID>
  <Name>Open file from repository...</Name>
  <!--Set Place To -1 If this is the first button To be inserted-->
  <Place>-1</Place>
  <ImageID>0</ImageID>
  <ToolBarID>1</ToolBarID>
  <!--instead of the toolbar ID the toolbar name could be used-->
  <ToolBarName>TestPlugIn</ToolBarName>
</UIElement>
```

Zusätzliche Elemente zum Deklarieren einer Symbolleisten-Schaltfläche sind **Place**, **ToolBarID** und **ToolBarName**. **ToolBarID** und **ToolBarName** dienen dazu, die Symbolleiste zu identifizieren, die die neue oder bestehende Schaltfläche enthält. Die Groß- und Kleinschreibung des Textwerts von **ToolBarName** spielt eine Rolle. Das Attribut **type** von **UIElement** muss "ToolBarItem" sein.

Um einen Menübefehl zu definieren, stehen zusätzlich zu den Standardelementen, die zum Deklarieren eines Befehls verwendet werden, die Elemente **MenuID**, **Place** und **Parent** zur Verfügung. **MenuID** kann 128 oder 129 sein. Nähere Informationen zu diesen Werten finden Sie unter "Symbolleisten und Menüs".

Das Element **Parent** dient dazu, das **Menü** anzugeben, in das der neue Menübefehl eingefügt werden soll. Da die Befehle von Untermenüs keine eindeutige Windows ID haben, benötigen wir eine andere Möglichkeit, um das übergeordnete Element des Menübefehls zu definieren.

Der Wert des Elements **Parent** ist ein Pfad zu einem Menübefehl. Der Textwert des Elements "Parent" muss gleich dem **übergeordneten Menünamen** des Untermenüs sein, wobei der Name des Untermenüs durch einen Doppelpunkt getrennt wird. Wenn das Menü kein übergeordnetes Menü hat, da es kein Untermenü ist, stellen Sie dem Namen am Anfang einen Doppelpunkt voran. Das Attribut **type** muss den Wert "MenuItem" haben. Beispiel für ein **UIElement**, das einen Menübefehl definiert:

```
<UIElement Type="MenuItem">
  <!--the following element is a Local command ID-->
  <ID>3</ID>
  <Name>Open file from repository...</Name>
  <Place>-1</Place>
  <MenuID>129</MenuID>
  <Parent>: PlugIn Menu</Parent>
  <ImageID>0</ImageID>
</UIElement>
```

Sie können in Authentic Desktop Trennlinien für Symbolleisten und Menüs hinzufügen, wenn der Wert des Elements **ID** auf 0 gesetzt ist.

3.4 ATL-Beispieldateien

Auf den folgenden Seiten wird beschrieben, wie Sie mit Hilfe von ATL eine einfache Authentic Desktop IDE Plug-In DLL erstellen. Dazu müssen Sie mit ATL und den Assistenten, die neue ATL-Objekte generieren, sowie MS VisualStudio vertraut sein.

Zum Aufrufen der API importiert die Implementierung die Typbibliothek von Authentic Desktop. Der Code liest verschiedene Eigenschaften aus und ruft mit Hilfe des über die `#import`-Anweisung zur Verfügung gestellten Smart Pointer Methoden auf.

Außerdem werden im Beispielcode der MFC Class String und die ATL-Konvertierungsmakros wie z.B. `W2T` verwendet.

Die grundlegenden Schritte zum Erstellen einer ATL DLL lauten wie folgt:

1. Öffnen Sie VisualStudio und wählen Sie im Menü "File" den Befehl "New...".
2. Wählen Sie das Register "Projects".
3. Wählen Sie "ATL COM AppWizard" und geben Sie einen Projektnamen ein.
4. Aktivieren Sie "Support for MFC", wenn Sie MFC-Klassen verwenden oder ein Projekt für den Beispielcode erstellen möchten.

Nach Erstellung der Projektdateien können Sie ein ATL-Objekt hinzufügen, um die `IXMLSpyPlugIn`-Schnittstelle zu implementieren:

1. Wählen Sie im Menü "Insert" den Befehl "New ATL Object...".
2. Wählen Sie im Assistenten die Option "Simple Object" und klicken Sie auf "Next".
3. Geben Sie einen Namen für das Objekt ein.
4. Wählen Sie auf dem Register "Attributes" als Schnittstellentyp "Custom" aus und deaktivieren Sie die Aggregation.

Damit wird der grundlegende Code für die Implementierung der IDE Plug-In Schnittstelle erzeugt. Wie man den Code ändert und einige grundlegende Funktionalitäten implementiert, wird auf den nächsten Seiten beschrieben.

3.4.1 Schnittstellenbeschreibung (IDL)

Die IDL des neu erstellten ATL-Objekts enthält eine Deklaration für eine COM-Schnittstelle.

- Diese Schnittstellenbeschreibung muss, wie unten gezeigt, durch die Deklaration von `IXMLSpyPlugIn` ersetzt werden.
- Die IDL muss außerdem die Definition der `SPYUpdateAction` Enumeration enthalten.
- Ersetzen Sie in der `coclass`-Deklaration den (vom Assistenten) generierten Standard-Schnittstellennamen durch `"IXMLSpyPlugIn"`. Die IDL sollte anschließend in etwa so aussehen wie im Beispielcode unten:

Nach Erstellung des ATL-Objekts müssen Sie nun die IDE Plug-In-Schnittstelle von Authentic Desktop implementieren.

```
import "oaidl.idl";
import "ocidl.idl";

// ----- please insert the following block into your IDL file -----
typedef enum {
    spyEnable = 1,
    spyDisable = 2,
    spyCheck = 4,
    spyUncheck = 8
} SPYUpdateAction;

// ----- end insert block -----

// ----- E.g. Interface entry automatically generated by the ATL wizard -----
// [
//     object,
//     uuid( AB7CD86A-8145-429A-A1F3-270692EO8AFC ),
//
//     helpstring("IXMLSpyPlugIn Interface")
//     pointer_default(unique)
// ]
// interface IXMLSpyPlugIn : IUnknown
// {
// };

// ----- end automatically generated Interface Entry

// ----- replace the Interface Entry (shown above) generated for you by the
// ATL wizard, with the following block -----

[
    odl,
    uuid( 88F2A622-4B7E-42CD-8D04-3C0E5389DD85 ),
    helpstring("IXMLSpyPlugIn Interface")
]
interface IXMLSpyPlugIn : IUnknown
{
    HRESULT STDMETHODCALLTYPE OnCommand([in] long nID, [in] IDispatch* pXMLSpy);

    HRESULT STDMETHODCALLTYPE OnUpdateCommand([in] long nID, [in] IDispatch* pXMLSpy,
    [out, retval] SPYUpdateAction* pAction);

    HRESULT STDMETHODCALLTYPE OnEvent([in] long nEventID, [in] SAFEARRAY(VARIANT)*
    arrayParameters, [in] IDispatch* pXMLSpy, [out, retval] VARIANT*
    pReturnValue);

    HRESULT STDMETHODCALLTYPE GetUIModifications([out, retval] BSTR* pModificationsXML);
}
```

```
        HRESULT _stdcall GetDescription([out, retval] BSTR* pDescription);
    };

// ----- end replace block -----

// ----- The code below is automatically generated by the ATL wizard and will
// look slightly different in your case -----

[
    uuid( 24FE0D1B-3FC0-494E-B36E-1D4CE412B014 ),
    version(1.0),
    helpstring("XMLSpyIDEPlugInDLL 1.0 Type Library")
]
library XMLSPYIDEPLUGINDLLLib
{
    importlib("stdole32.tlb");
    importlib("stdole2.tlb");

    [
        uuid( 3800E791-7F6B-4ACD-9E32-2AC184444501 ),
        helpstring("XMLSpyIDEPlugIn Class")
    ]
    coclass XMLSpyIDEPlugIn
    {
        [ default ] interface IXMLSpyPlugIn;    // ----- define IXMLSpyPlugIn as the
        default interface -----
    };
};
```

3.4.2 Klassendefinition

In der Klassendefinition des ATL-Objekts müssen einige Änderungen vorgenommen werden. Die Klasse muss von `IXMLSpyPlugIn` abgeleitet sein, die "Interface Map" benötigt einen Eintrag für `IXMLSpyPlugIn` und die Methoden der IDE Plug-In-Schnittstelle müssen deklariert werden:

```
#ifndef __XMLSPYIDEPLUGIN_H_
#define __XMLSPYIDEPLUGIN_H_

#include "resource.h"          // main symbols

////////////////////////////////////
// CXMLSpyIDEPlugIn
class ATL_NO_VTABLE CXMLSpyIDEPlugIn :
public CComObjectRootEx<CComSingleThreadModel>,
public CComCoClass<CXMLSpyIDEPlugIn, &CLSID_XMLSpyIDEPlugIn>,
public IXMLSpyPlugIn
{
public:
    CXMLSpyIDEPlugIn()
    {
    }

    DECLARE_REGISTRY_RESOURCEID(IDR_XMLSPYIDEPLUGIN)
    DECLARE_NOT_AGGREGATABLE(CXMLSpyIDEPlugIn)

    DECLARE_PROTECT_FINAL_CONSTRUCT()

    BEGIN_COM_MAP(CXMLSpyIDEPlugIn)
        COM_INTERFACE_ENTRY(IXMLSpyPlugIn)
    END_COM_MAP()

    // IXMLSpyIDEPlugIn
public:
    virtual HRESULT STDMETHODCALLTYPE OnCommand(long nID, IDispatch* pXMLSpy);

    virtual HRESULT STDMETHODCALLTYPE OnUpdateCommand(long nID, IDispatch* pXMLSpy,
        SPYUpdateAction* pAction);

    virtual HRESULT STDMETHODCALLTYPE OnEvent(long nEventID, SAFEARRAY** arrayParameters,
        IDispatch* pXMLSpy, VARIANT* pReturnValue);

    virtual HRESULT STDMETHODCALLTYPE GetUIModifications(BSTR* pModificationsXML);

    virtual HRESULT STDMETHODCALLTYPE GetDescription(BSTR* pDescription);
};

#endif // __XMLSPYIDEPLUGIN_H_
```

3.4.3 Implementierung

Im Code unten sehen Sie eine einfache Implementierung eines Authentic Desktop IDE Plug-In. Damit wird ein Menübefehl und eine Trennlinie (verfügbar in Authentic Desktop) zum Menü "Extras" hinzugefügt. Der neue Befehl innerhalb der OnUpdateCommand() Methode ist nur aktiv, wenn das aktive Dokument über die Grid-Ansicht angezeigt wird. Der Befehl sucht nach dem XML-Element, das sich gerade im Fokus befindet, und öffnet jede URL, die mit "http://" beginnt am Textwert des Elements.

```

////////////////////////////////////
// CXMLSpyIDEPlugIn

#import "XMLSpy.tlb"
using namespace XMLSpyLib;

HRESULT CXMLSpyIDEPlugIn::OnCommand( long nID, IDispatch* pXMLSpy)
{
    USES_CONVERSION;

    if( nID == 1) {
        IApplicationPtr ipSpyApp;

        if( pXMLSpy) {
            if( SUCCEEDED( pXMLSpy->QueryInterface( __uuidof( IApplication), ( void
**) &ipSpyApp))) {
                IDocumentPtr ipDocPtr = ipSpyApp->ActiveDocument;

                // we assume that grid view is active
                if( ipDocPtr) {
                    IGridViewPtr ipGridPtr = ipDocPtr->GridView;

                    if( ipGridPtr) {
                        IXMLDataPtr ipXMLData = ipGridPtr->CurrentFocus;

                        CString strValue = W2T( ipXMLData->TextValue);

                        if( !strValue.IsEmpty() && ( strValue.Left( 7) == _T( "http://")))
                            ::ShellExecute( NULL, _T( "open")
, W2T( ipXMLData->TextValue), NULL, NULL, SW_SHOWNORMAL);
                    }
                }
            }
        }

        return S_OK;
    }
}

HRESULT CXMLSpyIDEPlugIn::OnUpdateCommand( long nID, IDispatch* pXMLSpy,
SPYUpdateAction* pAction)
{
    *pAction = spyDisable;

    if( nID == 1) {
        IApplicationPtr ipSpyApp;

        if( pXMLSpy) {
            if( SUCCEEDED( pXMLSpy->QueryInterface( __uuidof( IApplication), ( void
**) &ipSpyApp))) {
                IDocumentPtr ipDocPtr = ipSpyApp->ActiveDocument;

                // only enable if grid view is active
                if( ( ipDocPtr != NULL) && ( ipDocPtr->CurrentViewMode == spyViewGrid))

```

```

        *pAction = spyEnable;
    }
}

return S_OK;
}

HRESULT CXMLSpyIDEPlugIn::OnEvent(long nEventID, SAFEARRAY **arrayParameters,
IDispatch* pXMLSpy, VARIANT* pReturnValue)
{
    return S_OK;
}

HRESULT CXMLSpyIDEPlugIn::GetUIModifications(BSTR* pModificationsXML)
{
    CComBSTR bstrMods = _T(" \
        <ConfigurationData> \
        <Modifications> ");
    // add "Open URL..." to Tools menu
    bstrMods.Append(_T(" \
        <Modification> \
        <Action>Add</Action> \
        <UIElement type=\"MenuItem\"> \
        <ID>1</ID> \
        <Name>Open URL...</Name> \
        <Place>0</Place> \
        <MenuID>129</MenuID> \
        <Parent>: Tools</Parent> \
        </UIElement> \
        </Modification> "));
    // add Seperator to Tools menu
    bstrMods.Append(_T(" \
        <Modification> \
        <Action>Add</Action> \
        <UIElement type=\"MenuItem\"> \
        <ID>0</ID> \
        <Place>1</Place> \
        <MenuID>129</MenuID> \
        <Parent>: Tools</Parent> \
        </UIElement> \
        </Modification> "));
    // finish modification description
    bstrMods.Append(_T(" \
        </Modifications> \
        </ConfigurationData>"));

    return bstrMods.CopyTo(pModificationsXML);
}

HRESULT CXMLSpyIDEPlugIn::GetDescription(BSTR* pDescription)
{
    CComBSTR bstrDescr = _T("ATL C++ XMLSpy IDE PlugIn; This PlugIn demonstrates
the implementation of a simple ATL DLL as a IDE PlugIn for XMLSpy.");
    return bstrDescr.CopyTo(pDescription);
}

```


3.5 IXMLSpyPlugIn

Siehe auch

Methoden

[OnCommand](#)

[OnUpdateCommand](#)

[OnEvent](#)

[GetUIModifications](#)

[GetDescription](#)

Beschreibung

Wenn eine DLL als IDE Plug-In zu Authentic Desktop hinzugefügt wird, muss sie als COM-Komponente registriert werden, die auf eine IXMLSpyPlugIn Schnittstelle mit der reservierten UUID(88F2A622-4B7E-42CD-8D04-3C0E5389DD85) antwortet, damit sie als Plug-In erkannt wird.

3.5.1 OnCommand

Siehe auch

Deklaration: `OnCommand( nID als long, pXMLSpy als IDispatch)`

Beschreibung

Die `OnCommand()` Methode der Schnittstellenimplementierung wird jedes Mal aufgerufen, wenn ein zum IDE Plug-In hinzugefügter Befehl (Menübefehl oder Symbolleisten-Schaltfläche) verarbeitet wird. `nID` speichert die durch das `ID`-Element des entsprechenden `UIElement` definierte Befehls-ID.

`pXMLSpy` enthält eine Referenz auf das Dispatch Interface des `Application`-Objekts von Authentic Desktop.

Example

```
Public Sub IXMLSpyPlugIn_OnCommand( ByVal nID As Long, ByVal pXMLSpy As Object)
    If (Not (pXMLSpy Is Nothing)) Then
        Dim objDlg
        Dim objDoc As XMLSpyLib.Document
        Dim objSpy As XMLSpyLib.Application
        Set objSpy = pXMLSpy

        If nID = 3 Or nID = 5 Then
            Set objDlg = CreateObject("MSComDlg.CommonDialog")
            objDlg.Filter = "XML Files (*.xml)|*.xml| All Files (*.*)|*.*||"
            objDlg.FilterIndex = 1
            objDlg.ShowOpen

            If Len(objDlg.FileName) > 0 Then
                Set objDoc = objSpy.Documents.OpenFile(objDlg.FileName, False)
                Set objDoc = Nothing
            End If
        End If

        If nID = 4 Or nID = 6 Then
            Set objDlg = CreateObject("MSComDlg.CommonDialog")
            objDlg.Filter = "All Files (*.*)|*.*||"
            objDlg.Flags = cdIOFNPathMustExist
            objDlg.ShowSave

            If Len(objDlg.FileName) > 0 Then
                Set objDoc = objSpy.ActiveDocument

                If Not (objDoc Is Nothing) Then
                    objDoc.SetPathName objDlg.FileName
                    objDoc.Save
                    Set objDoc = Nothing
                End If
            End If
        End If

        Set objSpy = Nothing
    End If
End Sub
```

3.5.2 OnUpdateCommand

Siehe auch

Deklaration: `OnUpdateCommand( nID als long, pXMLSpy als IDispatch) als SPYUpdateAction`

Beschreibung

Die `OnUpdateCommand()` Methode wird jedes Mal, wenn der Sichtbarkeitsstatus einer Schaltfläche oder eines Menübefehls definiert werden muss, aufgerufen. `nID` speichert die durch das `ID`-Element des entsprechenden `UIElement` definierte Befehls-ID.

`pXMLSpy` enthält eine Referenz auf das Dispatch Interface des `Application`-Objekts.

Mögliche Rückgabewerte zum Definieren des Aktualisierungsstatus sind:

<code>spyEnable</code>	<code>= 1</code>
<code>spyDisable</code>	<code>= 2</code>
<code>spyCheck</code>	<code>= 4</code>
<code>spyUncheck</code>	<code>= 8</code>

Beispiel

```
Public Function IXMLSpyPlugIn_OnUpdateCommand( ByVal nID As Long, ByVal
pXMLSpy As Object) As SPYUpdateAction
    IXMLSpyPlugIn_OnUpdateCommand = spyDisable

    If ( Not ( pXMLSpy Is Nothing) ) Then
        Dim objSpy As XMLSpyLib.Application
        Set objSpy = pXMLSpy

        If nID = 3 Or nID = 5 Then
            IXMLSpyPlugIn_OnUpdateCommand = spyEnable
        End If
        If nID = 4 Or nID = 6 Then
            If objSpy.Documents.Count > 0 Then
                IXMLSpyPlugIn_OnUpdateCommand = spyEnable
            Else
                IXMLSpyPlugIn_OnUpdateCommand = spyDisable
            End If
        End If
    End If
End Function
```

3.5.3 OnEvent

Siehe auch

Deklaration: `OnEvent( nEventID als long, arrayParameters als SAFEARRAY( VARIANT) , pXMLSpy als IDispatch) als VARIANT`

Beschreibung

`OnEvent()` wird jedes Mal, wenn ein Event von Authentic Desktop ausgelöst wird, aufgerufen.

Mögliche Werte für `nEventID` sind:

<code>On_BeforeStartEditing</code>	= 1
<code>On_EditingFinished</code>	= 2
<code>On_FocusChanged</code>	= 3
<code>On_Beforedrag</code>	= 4
<code>On_BeforeDrop</code>	= 5
<code>On_OpenProject</code>	= 6
<code>On_OpenDocument</code>	= 7
<code>On_CloseDocument</code>	= 8
<code>On_SaveDocument</code>	= 9

Events, die ab Authentic Desktop 4r4 verfügbar sind:

<code>On_DocEditDragOver</code>	= 10
<code>On_DocEditDrop</code>	= 11
<code>On_DocEditKeyDown</code>	= 12
<code>On_DocEditKeyUp</code>	= 13
<code>On_DocEditKeyPressed</code>	= 14
<code>On_DocEditMouseMove</code>	= 15
<code>On_DocEditButtonUp</code>	= 16
<code>On_DocEditButtonDown</code>	= 17
<code>On_DocEditContextMenu</code>	= 18
<code>On_DocEditPaste</code>	= 19
<code>On_DocEditCut</code>	= 20
<code>On_DocEditCopy</code>	= 21
<code>On_DocEditClear</code>	= 22
<code>On_DocEditSelectionChanged</code>	= 23

Events, die ab Authentic Desktop 2004 verfügbar sind:

<code>On_DocEditDragOver</code>	= 10

Events, die ab Authentic Desktop 2004r4 (Typbibliothek Version 1.4) verfügbar sind:

On_BeforeOpenProject	= 25
On_BeforeOpenDocument	= 26
On_BeforeSaveDocument	= 27
On_BeforeCloseDocument	= 28
On_ViewActivation	= 29
On_DocEditKeyboardEvent	= 30
On_DocEditMouseEvent	= 31

Events, die ab Authentic Desktop 2006 SP1 (Typbibliothek Version 1.5) verfügbar sind:

On_BeforeValidate	= 32

Events, die ab Authentic Desktop 2007 (Typbibliothek Version 1.6) verfügbar sind:

On_BeforeShowSuggestions	= 33
On_ProjectOpened	= 34
On_Char	= 35

Events, die ab Authentic Desktop 2009 (Typbibliothek Version 2.2) verfügbar sind:

On_Initialize	= 36
On_Running	= 37
On_Shutdown	= 38

Die Namen der Events sind dieselben wie in der Skripting-Umgebung von Authentic Desktop. Bei IDE Plug-Ins spielt es keine Rolle, welche Namen verwendet werden. Die Events werden über den `ID` Wert definiert.

arrayParameters ist ein Array, das mit den Parametern des aktuell ausgelösten Event gefüllt ist. Die Reihenfolge, Art und Bedeutung der einzelnen Parameter stehen über die Skripting-Umgebung von Authentic Desktop zur Verfügung. Das Event-Modul eines Skripting-Projekts enthält vordefinierte Funktionen für alle Events vor Version 4.4. Die an die vordefinierten Funktionen übergebenen Parameter sind mit den Array-Elementen des `arrayParameters-Parameters` identisch.

Events, die über die Authentic-Ansicht von Authentic Desktop ausgelöst werden, übergeben Parameter nicht direkt. Statt dessen wird ein "event" Objekt verwendet. Das Event-Objekt kann über das Dokumentobjekt des aktiven Dokuments aufgerufen werden.

pXMLSpy enthält eine Referenz auf das Dispatch Interface des `Application`-Objekts von Authentic Desktop.

Wenn der Rückgabewert von `OnEvent()` definiert ist, so ruft weder das IDE Plug-In noch ein Event Handler innerhalb der Skripting-Umgebung dieses Event später auf. Bitte beachten Sie, dass alle IDE Plug-Ins das Event abrufen/verarbeiten, bevor dies die Skripting-Umgebung tut.

3.5.4 GetUIModifications

Siehe auch

Deklaration: `GetUIModifications()` als `String`

Beschreibung

Die `GetUIModifications()` Methode wird während der Initialisierung des Plug-In aufgerufen, um die XML-Konfigurationsdaten abzurufen, die die Änderungen an der Benutzeroberfläche von Authentic Desktop definieren. Die Methode wird beim ersten Laden des Plug-In und bei jedem Start von Authentic Desktop aufgerufen.

Eine ausführliche Anleitung zum Ändern der Benutzeroberfläche finden Sie auch unter [Konfigurations-XML](#).

Beispiel

```
Public Function IXMLSpyPlugIn_GetUIModifications() As String
    ' GetUIModifications() gets the XML file with the specified modifications
of
    ' the UI from the config.xml file in the plug-in folder
    Dim strPath As String
    strPath = App.Path

    If Len(strPath) > 0 Then
        Dim fso As New FileSystemObject
        Dim file As file

        Set file = fso.GetFile(strPath & "\config.xml")

        If (Not (file Is Nothing)) Then
            Dim stream As TextStream
            Set stream = file.OpenAsTextStream(ForReading)

            ' this replaces the token '**path**' from the XML file with
            ' the actual installation path of the plug-in to get the image
file

            Dim strMods As String
            strMods = stream.ReadAll
            strMods = Replace(strMods, "**path**", strPath)

            IXMLSpyPlugIn_GetUIModifications = strMods
        Else
            IXMLSpyPlugIn_GetUIModifications = ""
        End If
    End If
End Function
```

3.5.5 GetDescription

Siehe auch

Deklaration: `GetDescription()` als `String`

Beschreibung

`GetDescription()` dient zur Definition des Beschreibungsstring für die Plug-In-Einträge, die im Dialogfeld "Anpassen" angezeigt werden.

Beispiel

```
Public Function IXMLSpyPlugIn_GetDescription() As String
    IXMLSpyPlugIn_GetDescription = "Sample Plug-in for XMLSpy; This Plug-in
    demonstrates the implementation of a simple VisualBasic DLL as a Plug-in for
    XMLSpy."
End Function
```

4 Die XMLSpy API

Mit Hilfe der COM-basierten XMLSpy API von Authentic Desktop können andere Applikationen die Funktionalitäten von Authentic Desktop verwenden. Dadurch können nun die verschiedensten Aufgaben vom Validieren einer XML-Datei bis zum Ändern von komplexem XML-Inhalt (mit der [XMLData](#) Schnittstelle) automatisiert werden. Die XMLSpy API wird auch von der integrierten Skripting-Umgebung von Authentic Desktop verwendet, um Applikationsfunktionalitäten aufzurufen (eine ausführliche Beschreibung dazu finden Sie unter [Skripting](#)). Außerdem wird sie von [IDE Plugins](#) verwendet.

Authentic Desktop und die XMLSpy API entsprechen den von Microsoft vorgegebenen allgemeinen Spezifikationen für Automation Server. Die Methoden und Eigenschaften der XMLSpy API können mit Skripting-Sprachen wie JavaScript und VBScript und von gängigen Entwicklungsumgebungen wie z.B. solchen, die C, C++, VisualBasic und Delphi verwenden, aufgerufen werden.

Berücksichtigen Sie bitte in Ihrem Client-Code die folgenden Einschränkungen:

- Denken Sie daran, dass bei einem Crash Ihres Client-Code eventuell noch Instanzen von Authentic Desktop im System laufen.
- Behalten Sie Referenzen auf Objekte - vor allem jene von der `XMLData`-Schnittstelle - nicht länger als nötig im Arbeitsspeicher. Wenn der Benutzer zwischen zwei Aufrufen Ihres Client Eingaben macht, ist nicht gewährleistet, dass diese Referenzen noch gültig sind.
- Beenden Sie Authentic Desktop mit der Methode [Application.Quit](#).
- Vergessen Sie nicht, Dialogfelder zu deaktivieren, wenn die Benutzeroberfläche nicht sichtbar ist.
- Nähere Informationen, wie Sie lästige Fehlermeldungen vermeiden, finden Sie unter [Fehlerbehandlung](#).
- Geben Sie Referenzen explizit frei, wenn Sie C oder C++ verwenden.

Diese Dokumentation

Die Dokumentation zur XMLSpy API ist in zwei Hauptteile gegliedert.

- Der erste Teil besteht aus einer [Übersicht](#), in der das Objektmodell für die API, wichtige Konzepte und [Verwendungsbeispiele](#) beschrieben werden, um Sie mit den wichtigsten Grundlagen vertraut zu machen.
- Der zweite Teil ist ein Referenzteil, der Beschreibungen aller Schnittstellenobjekte der XMLSpy API enthält.

In den einzelnen Releases wurden Verbesserungen und Aktualisierungen vorgenommen. Die Unterschiede zwischen einzelnen Releases sind in [Anmerkungen zur Release](#) dokumentiert.

4.1 Übersicht

In dieser Übersicht über die XMLSpy API finden Sie eine Beschreibung des Objektmodells für die API sowie der wichtigsten API-Prozesse. Es werden die folgenden Themen behandelt:

- [Das Objektmodell](#)
- [Einfacher Dokumentzugriff](#)
- [Fehlerbehandlung](#)
- [Events](#)
- [Import und Export von Daten](#)
- [Ändern der Dokumentstruktur mit Hilfe von XMLData](#)
- [DOM und XMLData](#)

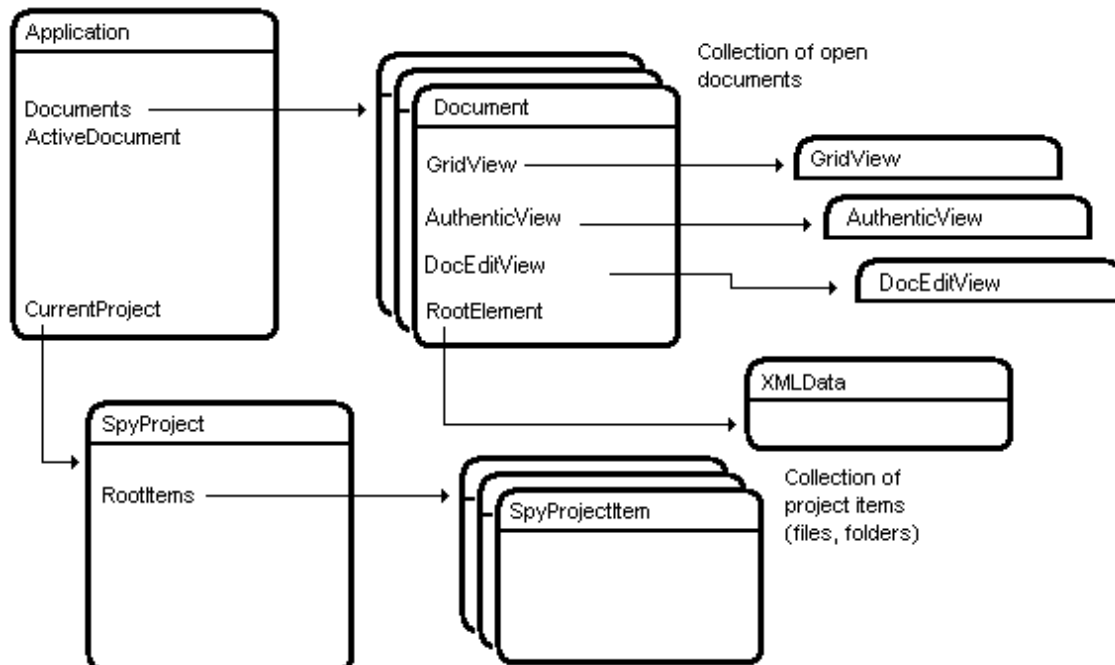
Änderungen an der API in den einzelnen Versionen der Typbibliothek sind im Abschnitt [Anmerkungen zur Release](#) aufgelistet.

4.1.1 Objektmodell

Der Ausgangspunkt für alle Applikationen, die die Authentic Desktop-API verwenden, ist das [Application](#) Objekt. Dieses Objekt enthält allgemeine Methoden wie Import/Export-Funktionen und Referenzen auf die offenen Dokumente und das offene Projekt, falls vorhanden.

Um eine Instanz des `Application` Objekts zu erstellen, rufen Sie von VisualBasic aus `CreateObject("XMLSpy.Application")` oder von der Entwicklungsumgebung Ihre Wahl eine ähnliche Funktion auf, um ein COM-Objekt zu erstellen. Es müssen keine weiteren Objekte erstellt werden, um die komplette XMLSpy API zu verwenden (dies ist auch gar nicht möglich). Alle anderen Schnittstellen werden über andere Objekte aufgerufen, wobei das `Application`-Objekt als Ausgangspunkt dient.

In der Abbildung unten sehen Sie die Verknüpfungen zwischen den Hauptobjekten der Authentic Desktop API:



Das `Application`-Objekt besteht aus den folgenden Teilen:

1. Dokumentsammlung und Referenz auf das aktive Dokument.
2. Referenz auf das aktuelle Projekt und Methoden zum Erstellen und Öffnen von Projekten.
3. Methoden zur Unterstützung des Exports und Imports aus/in Datenbanken, Textdateien und Word-Dokumente.
4. URL-Verwaltung.
5. Methoden für Makro-Menübefehle.

Nach Erstellung eines `Application`-Objekts können Sie beginnen, die Funktionalitäten von Authentic Desktop zu nutzen. In den meisten Fällen werden Sie entweder ein Projekt öffnen

und die Dokumente von dort aufrufen oder ein Dokument direkt über die [Documents](#) Schnittstelle öffnen.

4.1.2 Einfacher Dokumentzugriff

Erstellen und Öffnen von Dateien

Verwenden Sie die Methoden der [Documents](#) Schnittstelle, um Dokumente zu erstellen und zu öffnen. Die Schnittstelle wird über die [Application.Documents](#) Eigenschaft aufgerufen.

Beispiel:

```
Dim objDoc As Document
Set objDoc = objSpy.Documents.OpenFile("C:\xmlfiles\myfile.xml", False)
```

Manchmal ist es notwendig, dass dem Benutzer ein Dialogfeld zur Auswahl einer Datei angezeigt wird. Dies erfolgt normalerweise in der Skripting-Umgebung und Sie müssen nur "True" als zweiten Parameter an `OpenFile()` übergeben.

Um eine neue Datei zu erstellen, verwenden Sie [Documents.NewFile\(\)](#). Eine bereits vorhandene Datei mit demselben Pfad wird beim nächsten Aufruf von [Document.Save\(\)](#) überschrieben.

Dokumentensammlung

Alle offenen Dokumente können über die [Documents](#) Sammlung aufgerufen werden. Alle Sammlungsobjekte in der Authentic Desktop API entsprechen den Microsoft-Spezifikationen. So kann etwa in VisualBasic die For-Each-Schleife verwendet werden, um durch die offenen Dokumente zu iterieren.

Beispiel:

```
Dim objDocs As Documents
Dim objDoc As Document

Set objDocs = objSpy.Documents

For Each objDoc In objDocs
    'do something useful with your document
    objDoc.AssignXSL "C:\myXSL.xsl", False
Next
```

Eine weitere Methode, ein Dokument aufzurufen, ist die [Documents.Item](#) Methode. Die Methode nimmt einen Index als Parameter und holt das entsprechende Dokumentobjekt aus der Sammlung. Bitte beachten Sie, dass 1 der erste Indexwert ist. [Documents.Count](#) ruft die Gesamtanzahl der Dokumente ab.

Beispiel:

```
Dim objDocs As Documents
Dim objDoc As Document

Set objDoc = objDocs.Item(1) 'gets the first document
```

Validierung

Eine häufige Aufgabe ist die Validierung von Dokumenten anhand eines zugewiesenen Schemas oder einer DTD. Wenn der XML-Datei noch kein Schema und keine DTD zugewiesen wurde, verwenden Sie [Document.AssignSchema](#) oder [Document.AssignDTD](#) um die entsprechenden Referenzen zum Dokument hinzuzufügen.

Beispiele:

```
objSpy.ActiveDocument.AssignSchema "C:\mySchema.xsd", False
```

Or

```
objSpy.ActiveDocument.AssignDTD "C:\myDTD.dtd", False
```

Wenn der Benutzer aufgefordert werden soll, ein Schema oder eine DTD auszuwählen, übergeben Sie an diese Funktionen als zweiten Parameter "True", um ein Dateidialogfeld anzuzeigen. Mit dieser Methode wird nur die Referenz in das Dokument eingefügt. Es wird nicht überprüft, ob die angegebene Datei vorhanden ist. Wenn der Dateipfad nicht gültig ist, schlägt die Validierung fehl.

Nach Zuweisung eines gültigen Schemas oder einer DTD für Ihre Datei können Sie diese mit Hilfe von [Document.IsValid](#) validieren. IsValid benötigt einige Ausgabeparameter, die als VARIANTS deklariert werden müssen, damit sie von Skriptsprachen wie VBScript und JavaScript aufgerufen werden können.

Beispiel:

```
Dim bValid As Boolean
Dim strMsg As Variant
Dim nPos As Variant
Dim objBadXMLData As Variant

bValid = objSpy.ActiveDocument.IsValid(strMsg, nPos, objBadXMLData)

If bValid = False Then
    a = MsgBox("The document is not valid:" & Chr(13) &
        strMsg & Chr(13) & "position: " & nPos &
        Chr(13) & "XMLData name: " & objBadXMLData.Name)
Else
    a = MsgBox("The document is valid")
End If
```

4.1.3 Fehlerbehandlung

Die Authentic Desktop-API gibt Fehler auf zwei verschiedene Arten zurück. Jede API-Methode gibt ein `HRESULT` zurück. Dieser Rückgabewert enthält Informationen über etwaige Fehlfunktionen während der Ausführung der Methode. Wenn der Aufruf erfolgreich war, ist der Rückgabewert gleich `S_OK`. C/C++-Programmierer verwenden `HRESULT` im Allgemeinen, um Fehler zu erkennen.

VisualBasic, Skripting-Sprachen und andere Entwicklungsumgebungen gestatten dem Programmierer keinen Zugriff auf das zurückgegebene `HRESULT` eines COM-Aufrufs. Sie verwenden den zweiten Mechanismus zur Auslösung eines Fehlers, der auch von der Authentic Desktop-API unterstützt wird, nämlich die `Info` Schnittstelle. Bei Auftreten eines Fehlers erstellt die API ein neues Objekt, das die `Info` Schnittstelle implementiert. Die Entwicklungsumgebung nimmt diese Schnittstelle und befüllt ihre eigenen Fehlerbehandlungsmechanismen mit den hierin enthaltenen Informationen.

In den nächsten Absätzen wird beschrieben, wie Sie mit Fehlern umgehen, die in anderen Entwicklungsumgebungen von der Authentic Desktop-API ausgelöst werden.

VisualBasic

Eine gebräuchliche Methode, um Fehler in VisualBasic zu behandeln, ist einen Error Handler zu definieren. Dieser Error Handler kann mit der `On Error GoTo ErrorHandler`-Anweisung definiert werden. Normalerweise zeigt der Handler eine Fehlermeldung an und räumt den Code auf, um unnötige Referenzen und alle Arten von Ressourcenlecks zu verhindern. VisualBasic befüllt sein eigenes `Err` Objekt mit den Informationen aus der `Info` Schnittstelle.

Beispiel:

```
Sub Validate()
    'place variable declarations here

    'set error handler
    On Error GoTo ErrorHandler

    'if IsValid() fails, program execution continues
    'at ErrorHandler:
    bValid = objSpy.ActiveDocument.IsValid( strMsg, nPos, objBadXMLData)

    'additional code comes here

    'exit
Exit Sub

ErrorHandler:
Set objBadXMLData = Nothing

MsgBox("Error: " & (Err.Number - vbObjectError) & Chr(13) &
    "Description: " & Err.Description)
End Sub
```

JavaScript

Die Microsoft-Implementierung von JavaScript (JScript) bietet einen try-catch-Mechanismus zur Behandlung von Fehlern, die durch COM-Aufrufe ausgelöst wurden. Diese Methode ist der VisualBasic-Methode sehr ähnlich, da auch hier ein Fehlerobjekt deklariert wird, das die erforderlichen Informationen enthält.

Beispiel:

```
Function Validate()
{
    // please insert variable declarations here
```

```

try{
    bValid = objSpy.ActiveDocument.IsValid( sMsg, nPos, objBad );
}
catch( Error ) {
    objBad = Null;
    sError = Error.description;
    nErrorCode = Error.number;
    Return False;
}

Return bValid;
}

```

C/C++

In C/C++ haben Sie ganz einfachen Zugriff auf das HRESULT des COM-Aufrufs und das IErrorInterface.

```

HRESULT hr;

// Call IsValid() from the XMLSpy API
If( FAILED( hr = ipDoc->IsValid( &varMsg, &varPos, &varBadData ) ) ) {
    IErrorInfo *ipErrorInfo = Null;

    If( SUCCEEDED( ::GetErrorInfo( 0, &ipErrorInfo ) ) ) {
        BSTR bstrDescr;
        ipErrorInfo->GetDescription( &bstrDescr );

        // handle Error information
        wprintf( L"Error message: %t%s\n", bstrDescr );
        ::SysFreeString( bstrDescr );

        // release Error info
        ipErrorInfo->Release();
    }
}

```

4.1.4 Events

Die Automation-Schnittstelle von XMLSpy bietet eine große Gruppe von Events, um eine enge Integration von XMLSpy und seinen Clients zu ermöglichen. Die Art, wie Events von Clients verwendet werden können, ist abhängig von der Technologie, die verwendet wird, um eine Verbindung zu XMLSpy herzustellen, und von der Programmiersprache des Client.

XMLSPY Skripting-Umgebung - mit dem FormEditor erstellte Makros, Formulare und Event Handler

Die XMLSPY Skripting-Umgebung ruft automatisch den Event Handler mit dem entsprechenden Namen auf. Mit Hilfe des FormEditor können Sie Code zu denjenigen Events hinzufügen, die Sie behandeln möchten. Bei der Erstellung eines neuen Projekts definiert der FormEditor im voraus die korrekten Signaturen für alle Event Handler. Nähere Informationen dazu finden Sie unter [Skripting-Umgebung](#).

VBScript

Alle Event Handler-Funktionen beginnen mit 'On_'. Damit sie kompatibel mit vorhandenen Events bleiben, kann sich ihr Name geringfügig von dem im Connection Point Interface verwendeten unterscheiden. Die Angabe von Event-Parameters ist optional, sodass alte Event Handler, die keine Parameter erwartet haben, weiterhin funktionsfähig bleiben.

JScript

Alle Event Handler-Funktionen beginnen mit 'On_'. Damit sie kompatibel mit vorhandenen Events bleiben, kann sich ihr Name geringfügig von dem im Connection Point Interface verwendeten unterscheiden. Die Angabe von Event-Parameters ist optional, sodass alte Event Handler, die keine Parameter erwartet haben, weiterhin funktionsfähig bleiben.

XMLSPY IDE Plugin

XMLSPY Plugins werden durch einen Aufruf der Methode [IXMLSpyPlugin.OnEvent](#) über ausstehende Events informiert. Unterschiedliche Events werden durch eindeutige Nummern identifiziert. Parameter werden in Form eines Save-Array an den Event Handler übergeben. Alternativ dazu können Plugins mit Hilfe des Connection Point-Mechanismus Events auf bestimmten Objekten wie z.B. einem beliebigen externen Client anfordern. Nähere Informationen dazu siehe unten.

Externe Clients

COM legt fest, dass der Connection Point-Mechanismus eine Methode definiert, wie ein Client sich selbst für Callbacks auf einem Server registrieren kann. Die Automation-Schnittstelle für XMLSpy definiert die nötigen Event-Schnittstellen. Wie eine Verbindung zu diesen Events hergestellt wird, hängt von der auf dem Client verwendeten Programmiersprache ab.

VBA - VisualBasic for Applications

Verwenden Sie die Dim WithEvents-Anweisung, um Events von einer dafür vorgesehenen Objektinstanz zu empfangen.

```
' tell VBA to connect to events once this objects gets created
Dim WithEvents objView As XMLSpyLib.AuthenticView
' initialize the object somewhere in your code
...
Set objView = objSpy.ActiveDocument.AuthenticView
...
' sample event handler for the OnMouseEvent of the object in objView
Private Function objView_OnMouseEvent (ByVal i_nXPos As Long, ByVal
i_nYPos As Long,
                                     ByVal i_eMouseEvent As
XMLSpyLib.SPYMouseEvent,
                                     ByVal i_pRange As
XMLSpyLib.IAuthenticRange ) As Boolean
```



```

    If (i_eMouseEvent = (XMLSpyLib.spyLeftButtonDownMask Or
XMLSpyLib.spyCtrlKeyDownMask)) Then
        On Error Resume Next
        i_pRange.Select
        objView_OnMouseEvent = True
    Else
        objView_OnMouseEvent = False
    End If
End Function

```

Windows Skripting-Technologien - VBScript

Verwenden Sie die Methode WScript.ConnectObject um Events zu empfangen.

```

' the event handler function
Function DocEvent_OnBeforeCloseDocument(objDocument)
    Call WScript.Echo("received event - before closing document")
End Function

' create or connect to XMLSPY
Set objWshShell = WScript.CreateObject("WScript.Shell")
Set objFSO = WScript.CreateObject("Scripting.FileSystemObject")
Set objSpy = WScript.GetObject("", "XMLSpy.Application")

' create document object and connect to its events
objSpy.Visible = True
Set objDoc = objSpy.Documents.OpenFile ("C:\\Program
Files\\Altova\\XMLSPY2004\\Examples\\OrgChart.xml", False)
Call WScript.ConnectObject(objDoc, "DocEvent_")

' keep running while waiting on the event
' in the meantime close the document in XMLSPY manually
Call WScript.Echo ("sleeping for 10 seconds ...")
Call WScript.Sleep (10000)

Set objDoc = Nothing
Call WScript.Echo ("stopped listening for event")
Call objSpy.Quit

```

Windows Skripting-Technologien - JScript

Verwenden Sie die Methode WScript.ConnectObject um Events zu empfangen.

```

// the event handler function
function DocEvent_OnBeforeCloseDocument(objDocument)
{
    WScript.Echo("received event - before closing document");
}

// create or connect to XMLSPY
try
{
    // create the environment and XMLSPY
    objWshShell = WScript.CreateObject("WScript.Shell");
    objFSO = WScript.CreateObject("Scripting.FileSystemObject");
    objSpy = WScript.GetObject("", "XMLSpy.Application");
}
catch(err)
{ WScript.Echo ("Can't create WScript.Shell object or XMLSPY"); }

// create document object and connect to its events
objSpy.Visible = true;
objDoc = objSpy.Documents.OpenFile ("C:\\Program
Files\\Altova\\XMLSPY2004\\Examples\\OrgChart.xml", false);
WScript.ConnectObject(objDoc, "DocEvent_");

```

```

// keep running while waiting on the event
// in the meantime close this document in XMLSPY manually
WScript.Echo ("sleeping for 10 seconds ...");
WScript.Sleep (10000);

objDoc = null;
WScript.Echo ("stopped listening for event");
objSpy.Quit();

```

C++

Im Folgenden sehen Sie einen Auszug eines C++ Client, der Connection Points über ATL verwendet

```

#import "XMLSpy.tlb"
using namespace XMLSpyLib;

static const int XMLSPYSE_APPLICATION_SOURCE_ID = 0;
class XMLSpySEApplicationEventPump;

// nested class to receive application events
typedef IDispatchImpl< XMLSPYSE_APPLICATION_SOURCE_ID,
                      XMLSpySEApplicationEventPump,
                      &DIID__IApplicationEvents,
                      &LIBID_XMLSpyLib, TYPELIB_MAJ, TYPELIB_MIN >
                      XMLSpySEApplicationEventImpl;

// the class receiving the application events using a sink map
class XMLSpySEApplicationEventPump :
public XMLSpySEApplicationEventImpl
{
private:
    static _ATL_FUNC_INFO Info_OnBeforeOpenProject;

public:
    XMLSpySEApplicationEventPump ();
    ~XMLSpySEApplicationEventPump();

    // connection point events on XMLSpySEApplicationEventImpl
public:
    void __stdcall OnBeforeOpenProject (IFileSelectionDlg
*io_ipDlg);

    BEGIN_SINK_MAP( XMLSpySEApplicationEventPump)
        // OnBeforeOpenProject
        SINK_ENTRY_INFO( XMLSPYSE_APPLICATION_SOURCE_ID,
        DIID__IApplicationEvents,
        1, OnBeforeOpenProject,
        &Info_OnBeforeOpenProject)
    END_SINK_MAP()
};

// we use here explicit declaration of event signature to avoid loading
the typelib
_ATL_FUNC_INFO XMLSpySEApplicationEventPump::Info_OnBeforeOpenProject =
{CC_STDCALL, VT_EMPTY, 1, {VT_DISPATCH}};

XMLSpySEApplicationEventPump::XMLSpySEApplicationEventPump ()
{
    ...
    // get an IUnknown pointer to XMLSpy application from somewhere and
    store it in ipXMLSPY
    HRESULT hRes = XMLSpySEApplicationEventImpl::DispEventAdvise( ipXMLSPY,
    &DIID__IApplicationEvents);
    ...
}

```

```
}

XMLSpySEApplicationEventPump::~XMLSpySEApplicationEventPump()
{
    ...
    // disconnect from connection point
    HRESULT hRes =
XMLSpySEApplicationEventImpl::DispEventUnadvise(ipXMLSPY,
&DIID__IApplicationEvents);
    ...
}

// the handler for the OnBeforeOpenProject event
void __stdcall XMLSpySEApplicationEventPump::OnBeforeOpenProject
(IFileSelectionDlg *io_ipDlg)
{
    // do something with io_ipDlg here
}
```

4.1.5 Import und Export von Daten

Bevor Sie Ihre Import- und Exportaufgaben mit der Authentic Desktop-API implementieren, ist es sinnvoll, die Verbindungen, Parameter und SQL-Abfragen usw. in Authentic Desktop zu testen. Auf diese Art können Sie die Ergebnisse überprüfen und eventuell Anpassungen an allen Import- und Export-Parametern vornehmen.

Die meisten der Methoden für das Importieren und Exportieren von Daten werden in das [Application](#) Objekt platziert. Die restlichen Funktionen können über die [Document](#) Schnittstelle aufgerufen werden.

Bevor Sie mit dem eigentlichen Import und Export beginnen können, müssen zuerst einige Vorbereitungen getroffen werden. Jede Import/Export-Aufgabe besteht aus zwei Teilen. Sie müssen eine Verbindung zu Ihren Daten und das entsprechende Verhalten für den Import/Export-Vorgang definieren

Bei einem Import wird eine Verbindung zu einer Datenbank, einer Textdatei oder einem Word-Dokument hergestellt. Definiert wird im Grunde, welche Daten (Spalten) in Authentic Desktop importiert werden sollen.

Bei einem Export wird eine Verbindung zu einer Datenbank oder einer Textdatei hergestellt. Geben Sie an, welche Daten (Elemente der XML-Datei) und welche zusätzlichen Parameter (z.B. automatische Schlüsselgenerierung oder Anzahl der untergeordneten Ebenen) aus der XML-Struktur verwendet werden sollen.

Die Eigenschaften in den Schnittstellen [DatabaseConnection](#), [TextImportExportSettings](#) und [ExportSettings](#) haben Standardwerte. Nähere Informationen dazu finden Sie in den entsprechenden Beschreibungen im Kapitel [Schnittstellen](#).

Import aus einer Datenbank

Dies sind die Schritte, um für den Import eine Verbindung zu einer vorhandenen Datenbank herzustellen:

1. Verwenden Sie ein [DatabaseConnection](#) Objekt und definieren Sie die Eigenschaften:
Die Methode [Application.GetDatabaseSettings](#) gibt ein neues Objekt für eine Datenbankverbindung zurück:

```
Dim objImpSettings As DatabaseConnection
Set objImpSettings = objSpy.GetDatabaseSettings
```

Sie müssen entweder einen **ADO Connection String** definieren,

- `objImpSettings.ADOConnection = strADOConnection`

oder den **Pfad** zu einer bestehenden Datenbankdatei:

- `objImpSettings.File = "C:\myDatabase.mdb"`

Abschließend erstellen Sie eine **SQL select-Anweisung**, um festzulegen, welche Daten abgefragt werden sollen:

- `objImpSettings.SQLSelect = "SELECT * FROM myTable"`

2. Rufen Sie [Application.GetDatabaseImportElementList](#) auf, um eine Sammlung der Ergebnisspalten der SQL-Abfrage zu erhalten:

```
Dim objElementList As ElementList
```

```
Set objElementList =  
objSpy.GetDatabaseImportElementList(objImpSettings)
```

Mit Hilfe dieser Sammlung können Sie festlegen, welche Spalten importiert werden sollen und welchen Typ die neuen Elemente haben sollen. Jedes Element der Sammlung steht für eine zu importierende Spalte. Wenn Sie ein Element entfernen, wird die entsprechende Spalte nicht importiert. Zusätzlich dazu können Sie die Eigenschaft [ElementListItem.ElementKind](#) ändern, um den Typ der erstellten XML-Elemente für die einzelnen Spalten zu definieren.

Bitte beachten Sie, dass `GetDatabaseImportElementList()` die SQL-Abfrage ausführt und einen zeitaufwändigen Aufruf initiieren könnte. Um dies zu vermeiden, können Sie einen null-pointer (in VisualBasic Nothing) als zweiten Parameter an `ImportFromDatabase()` übergeben, um alle Spalten als einfache XML-Elemente zu importieren.

3. Starten Sie den Import mit [Application.ImportFromDatabase](#):

```
Dim objImpDoc As Document  
Set objImpDoc =  
objSpy.ImportFromDatabase(objImpSettings, objElementList)
```

Import aus einer Textdatei

Der Import von Daten aus einer Textdatei erfolgt ähnlich wie der Import aus einer Datenbank. Sie müssen andere Schnittstellen (Beschreibung siehe Schritt 1-3 unten) mit anderen Methoden und Eigenschaften verwenden:

1. Verwenden Sie ein [TextImportExportSettings](#) Objekt und definieren Sie die Eigenschaften:
Die Methode [Application.GetTextImportExportSettings](#) gibt ein neues Objekt zurück, das eine Textdatei für den Import definiert.

```
Dim objImpSettings As TextImportExportSettings  
Set objImpSettings = objSpy.GetTextImportExportSettings
```

Sie müssen zumindestens für die Eigenschaft `ImportFile` den Pfad zu der zu importierenden Datei definieren. Eine weitere wichtige Eigenschaft ist `HeaderRow`. Setzen Sie sie auf "False", wenn die Textdatei keine Überschrift als Kopfzeile enthält.

```
objImpSettings.ImportFile = "C:\myFile.txt"  
objImpSettings.HeaderRow = False
```

Rufen Sie [Application.GetTextImportElementList](#) auf, um eine Sammlung aller Spalten in der Textdatei zu erhalten:

```
Dim objElementList As ElementList  
Set objElementList = objSpy.GetTextImportElementList(objImpSettings)
```

3. Starten Sie den Import mit [Application.ImportFromText](#):

```
Dim objImpDoc As Document  
Set objImpDoc = objSpy.ImportFromText(objImpSettings, objElementList)
```

Export in eine Datenbank

1. Verwenden Sie ein [DatabaseConnection](#) Objekt und definieren Sie die erforderlichen Eigenschaften.
Alle Eigenschaften mit Ausnahme von `SQLSelect` sind für den Export relevant.

`ADOConnection` oder `File` definiert das Ziel für die Ausgabe. Sie müssen nur eine davon definieren.

2. Füllen Sie ein [ExportSettings](#) Objekt mit den erforderlichen Werten. Diese Eigenschaften sind dieselben Optionen wie die im Exportdialogfeld von Authentic Desktop. Wählen Sie die Menüoption **Konvertieren | Exportieren in Textdatei/Datenbank** um die Optionen zu sehen und eine Kombination von Exporteinstellungen auszuprobieren. Anschließend können Sie diese Einstellungen jederzeit auf die Eigenschaften der Schnittstelle übertragen.

Rufen Sie "[Application.GetExportSettings](#)" auf, um ein `ExportSettings` Objekt aufzurufen:

```
Dim objExpSettings As ExportSettings
Set objExpSettings = objSpy.GetExportSettings

objExpSettings.CreateKeys = False
objExpSettings.ExportAllElements = False
objExpSettings.SubLevelLimit = 2
```

3. Erstellen Sie mit [Document.GetExportElementList](#) eine Elementliste. Mit Hilfe der Elementliste können Sie XML-Elemente aus dem Exportprozess ausnehmen. Außerdem enthält diese Liste in den Eigenschaften `RecordCount` und `FieldCount` Informationen über die Anzahl der Datensätze und Felder. Setzen Sie die [ExportSettings.ElementList](#) Eigenschaft auf diese Sammlung. Sie können die Elementliste auf null/Nothing (Standardeinstellung) setzen, um alle Elemente zu exportieren.
4. Rufen Sie [Document.ExportToDatabase](#) auf, um den Export auszuführen. Die Beschreibung der `ExportToDatabase` Methode enthält auch ein Codebeispiel für einen Datenbankexport.

Export in eine Textdatei

1. Verwenden Sie ein [TextImportExportSettings](#) Objekt und definieren Sie die erforderlichen Eigenschaften.
2. Füllen Sie ein [ExportSettings](#) Objekt mit den erforderlichen Werten. Siehe Punkt 2 aus "Export in eine Datenbank" weiter oben auf dieser Seite.
3. Erstellen Sie eine Elementliste mit [Document.GetExportElementList](#). Siehe Punkt 3 aus "Export in eine Datenbank" weiter oben auf dieser Seite.
4. Rufen Sie [Document.ExportToText](#) auf, um den Export auszuführen. Die Beschreibung der `ExportToText` Methode enthält auch ein Codebeispiel für einen Datenbankexport.

4.1.6 Ändern der Dokumentstruktur mit Hilfe von XMLData

Mit Hilfe von XMLData haben Sie Zugriff auf die Elemente einer gerade offenen XML-Datei. Sie können damit alle nötigen Änderungen an den Elementen der XML-Struktur vornehmen. Die Hauptfunktionalität von XMLData ist:

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

Struktur von XMLData

Bevor Sie die XMLData Schnittstelle verwenden können, müssen Sie wissen, wie eine bestehende XML-Datei in eine XMLData Struktur gemappt wird. Wichtig ist dabei, dass XMLData keine eigene Sorte von Objekten für Attribute hat.

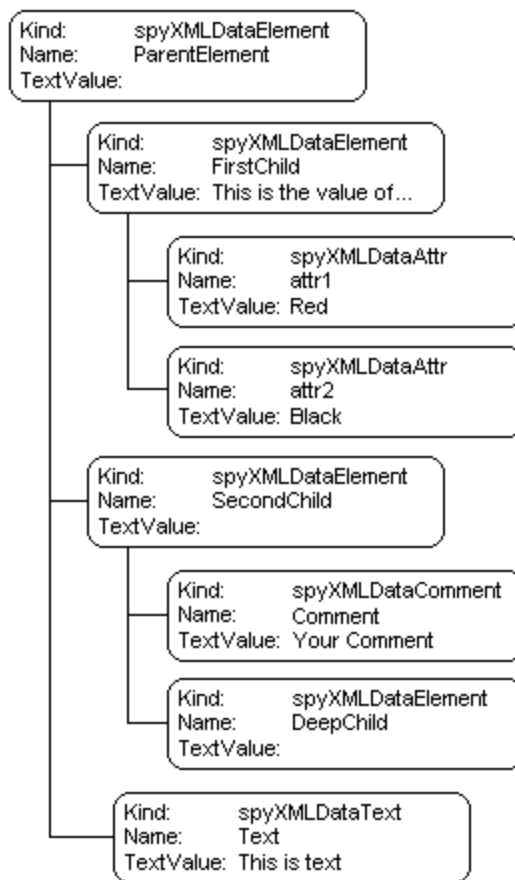
Die Attribute eines Elements sind ebenfalls Child-Elemente des Elements. Mit Hilfe der Eigenschaft [XMLData.Kind](#) können Sie zwischen den verschiedenen Arten von Child-Elementen eines Elements unterscheiden.

Beispiel:

Dieser XML-Code

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

wird auf die folgende XMLData-Objektstruktur gemappt:



Das übergeordnete Element aller XML-Elemente in einem Dokument ist die Eigenschaft [Document.RootElement](#). Das erste Element des Inhalts eines Dokuments - ohne den XML-Prolog - wird über die Eigenschaft [Document.DataRoot](#) aufgerufen. Verwenden Sie eines dieser `XMLData` Objekte, um Referenzen auf alle anderen XML-Elemente in der Struktur zu erhalten.

Name und Wert von Elementen

Um den Namen und Wert der verschiedenen Arten von XML-Elementen zu erhalten und zu ändern, verwenden Sie die Eigenschaften [XMLData.Name](#) und [XMLData.TextValue](#). Es kann vorkommen, dass einigen Arten von `XMLData` Objekten und leeren Elementen kein Textwert zugewiesen ist.

Erstellen und Einfügen neuer XMLData Objekte

Um eine neue Einheit in der XML-Sprache zu erstellen, gehen Sie folgendermaßen vor:

1. Erstellen Sie das neue `XMLData` Objekt:
Verwenden Sie die [Document.CreateChild](#) Methode, um ein neues `XMLData` Objekt zu erstellen. Definieren Sie nach dem Einfügen des neuen XML-Elements seinen Namen und Wert (siehe Punkt 3).
2. Suchen Sie die richtige Einfügeposition für das neue `XMLData` Objekte:
Um ein neues `XMLData` Objekt einzufügen, müssen Sie zuerst eine Referenz auf das übergeordnete Element abrufen. Wenn das neue Child-Element das letzte Child-Element des übergeordneten Elements sein soll, fügen Sie das `XMLData` Objekt mit Hilfe der [XMLData.AppendChild](#) Methode ein. Wenn das neue Child-Element in der Reihenfolge der Child-Elemente an einer anderen Stelle eingefügt werden soll, verwenden Sie [XMLData.GetFirstChild](#) und [XMLData.GetNextChild](#), um den

Iterator zum Child-Element zu verschieben, vor dem das neue Child-Element eingefügt werden soll.

3. Fügen Sie das neue Child-Element mit Hilfe von [XMLData.InsertChild](#) ein. Das neue Child-Element wird unmittelbar vor dem aktuellen Child-Element eingefügt.

Im folgenden Beispiel wird zwischen den Elementen <FirstChild> und <SecondChild> ein drittes Child-Element eingefügt:

```
Dim objParent As XMLData
Dim objChild As XMLData
Dim objNewChild As XMLData

Set objNewChild = objSpy.ActiveDocument.CreateChild(spyXMLDataElement)

'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
objNewChild.Name = "OneAndAHalf"
Set objNewChild = Nothing
```

Child-Elemente sollten in einer bestimmten Reihenfolge eingefügt werden. Fügen Sie bitte keine Attribute nach weiteren Child-Elemente in einer Sequenz ein. Stellen Sie also sicher, dass Attribute am Anfang stehen.

Kopieren eines bestehenden XMLData-Objekts

Wenn Sie bestehende XMLData Objekte an einer anderen Stelle im selben Dokument oder in einem anderen Dokument einfügen möchten, können Sie die Methoden [XMLData.InsertChild](#) und [XMLData.AppendChild](#) dazu nicht verwenden. Diese Methoden funktionieren nur bei neuen XMLData Objekten.

Anstatt InsertChild oder AppendChild zu verwenden, müssen Sie die Hierarchie manuell kopieren. Die folgende in JavaScript geschriebene Funktion ist ein Beispiel für das rekursive Kopieren von XMLData:

```
// -----
// XMLSpy scripting environment - GlobalDeclarations - JScript
// return a deep copy of the XMLData Object.
// the new object is owned by the document sepcified in objDoc
// -----
function XMLDataDeepCopy(objXMLData, objDoc)
{
    // create new element of same kind
    var objNew = objDoc.CreateChild(objXMLData.Kind);

    // copy name, some element have default names and do not
    // allow to modifying these names.
    try
    { objNew.Name = objXMLData.Name; }
    catch (e)
    {
        if ((e.number & 0xffff) != 1513) // modification operation not allowed
            throw(e);
    }

    // copy the text value
    objNew.TextValue = objXMLData.TextValue;

    // copy and insert all children
```

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

    while( objChild)
    {
        try
        {
            objNew.AppendChild( XMLDataDeepCopy( objChild, objDoc));
            objChild = objXMLData.GetNextChild();
        }
        catch( e)
        {
            if ((e.number & 0xffff) == 1503)    // end of list
                objChild = null;
            else
                throw( e);
        }
    }
}

return objNew;
}
```

Löschen von XMLData-Objekten

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

Um XMLData Objekte zu löschen, müssen Sie das übergeordnete Element der zu löschenden Elemente aufrufen. Rufen Sie mit Hilfe von [XMLData.GetFirstChild](#) und [XMLData.GetNextChild](#) eine Referenz auf das übergeordnete XMLData Objekt ab.

Beispiele, wie man XML-Elemente löscht, finden Sie unter den Beschreibungen zu den Methoden [EraseAllChildren](#) und [EraseCurrentChild](#).

4.1.7 DOM und XMLData

Über die `XMLData` Schnittstelle haben Sie Zugriff auf die gesamte XML-Struktur, die dem aktuellen Dokument zugrunde liegt. Sie benötigen dafür weniger Methoden als DOM und der Zugriff ist einfacher. Die `XMLData` Schnittstelle ist eine möglichst einfache Methode, um vorhandene oder neu erstellte XML-Daten zu lesen und zu bearbeiten. Eventuell sollten Sie jedoch eine DOM-Struktur verwenden, weil Sie auf diese von einer externen Quelle aus zugreifen können oder weil Sie die MSXML DOM-Implementierung vorziehen.

Die unten gezeigten Funktionen `ProcessDOMNode()` und `ProcessXMLDataNode()` konvertieren alle Segmente einer XML-Struktur zwischen `XMLData` und DOM.

Um die `ProcessDOMNode()` Funktion zu verwenden:

- übergeben Sie das Root-Element des zu konvertierenden DOM-Segments in `objNode` und
- übergeben Sie das Plugin-Objekt mit der `CreateChild()` Methode in `objCreator`

Um die `ProcessXMLDataNode()` Funktion zu verwenden:

- übergeben Sie das Root-Element des `XMLData`-Segments in `objXMLData` und
- übergeben Sie das mit MSXML in `xmlDoc` erstellte `DOMDocument`

```

////////////////////////////////////
// DOM To XMLData conversion
Function ProcessDOMNode( objNode, objCreator)
{
    var objRoot;
    objRoot = CreateXMLDataFromDOMNode( objNode, objCreator);

    If( objRoot ) {
        If( ( objNode.nodeValue != Null ) && ( objNode.nodeValue.length > 0 ) )
            objRoot.TextValue = objNode.nodeValue;
        // add attributes
        If( objNode.attributes ) {
            var Attribute;
            var oNodeList = objNode.attributes;

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

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

                objRoot.AppendChild( newNode );
            }
        }
        If( objNode.hasChildNodes ) {
            try {
                // add children
                var Item;
                oNodeList = objNode.childNodes;

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

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

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

```

```

    }
  }
  Return objRoot;
}

Function CreateXMLDataFromDOMNode( objNode, objCreator)
{
  var bSetName = True;
  var bSetValue = True;

  var nKind = 4;

  switch( objNode.nodeType) {
    Case 2: nKind = 5; break;
    Case 3: nKind = 6; bSetName = False; break;
    Case 4: nKind = 7; bSetName = False; break;
    Case 8: nKind = 8; bSetName = False; break;
    Case 7: nKind = 9; break;
  }
  var objNew = Null;
  objNew = objCreator.CreateChild( nKind );

  If( bSetName)
    objNew.Name = objNode.nodeName;

  If( bSetValue && ( objNode.nodeValue != Null))
    objNew.TextValue = objNode.nodeValue;

  Return objNew;
}
////////////////////////////////////
// XMLData To DOM conversion

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

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

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

        While( True ) {
          If( objChild ) {
            var newNode;
            newNode = ProcessXMLDataNode( objChild, xmlDoc );

            If( newNode.nodeType == 2 ) {
              // child node is an attribute
              objRoot.attributes.setNamedItem( newNode );
            }
            Else
              objRoot.appendChild( newNode );
          }
          objChild = objXMLData.GetNextChild();
        }
      }
      catch( err ) {
      }
    }
  }
}

```

```
    Return objRoot;
}

Function CreateDOMNodeFromXMLData( objXMLData, xmlDoc)
{
    switch( objXMLData. Kind) {
        Case 4: Return xmlDoc.createElement( objXMLData. Name);
        Case 5: Return xmlDoc.createAttribute( objXMLData. Name);
        Case 6: Return xmlDoc.createTextNode( objXMLData. TextValue);
        Case 7: Return xmlDoc.createCDATASection( objXMLData. TextValue);
        Case 8: Return xmlDoc.createComment( objXMLData. TextValue);
        Case 9: Return
xmlDoc.createProcessingInstruction( objXMLData. Name, objXMLData. TextValue);
    }

    Return xmlDoc.createElement( objXMLData. Name);
}

Function IsTextNodeEnabled( objNode)
{
    switch( objNode. nodeType) {
        Case 1:
        Case 2:
        Case 5:
        Case 6:
        Case 11: Return True;
    }
    Return False;
}
```

4.1.8 Nicht mehr verwendete Zeilenoperationen in der Authentic-Ansicht

Wenn im Schema, auf dem das XML-Dokument basiert, festgelegt ist, dass ein Element wiederholbar ist, kann eine derartige Struktur in der Authentic-Ansicht als Tabelle dargestellt werden. Bei Darstellung in einer Tabelle können Zeilen und ihr Inhalt einzeln bearbeitet werden, wodurch Sie jedes der wiederholbaren Elemente einzeln bearbeiten können. Solche Zeilenoperationen würden von einem externen Skript durchgeführt werden.

Wenn ein externes Skript Zeilenoperationen durchführen soll, sind zwei Schritte erforderlich:

- Im ersten Schritt wird überprüft, ob sich der Cursor gerade in einer Zeile befindet, in der eine Eigenschaft verwendet wird. Eine solche Überprüfung könnte z.B. sein `IsRowInsertEnabled`. Diese Methode gibt das Ergebnis TRUE oder FALSE zurück.
- Wenn der Rückgabewert TRUE ist, kann eine Zeilenmethode wie z.B. `RowAppend` aufgerufen werden. (`RowAppend` hat keine Parameter und gibt keinen Wert zurück.)

Im Folgenden sehen Sie eine Liste von Eigenschaften und Methoden, die für Tabellenoperationen zur Verfügung stehen. Jede Eigenschaft gibt einen booleschen Wert zurück und die Methoden haben keine Parameter.

Eigenschaft	Methode	Tabellenoperationen
<code>IsRowInsertEnabled</code>	RowInsert , wird nun ersetzt durch AuthenticRange.InsertRow	Operation "Zeile einfügen"
<code>IsRowAppendEnabled</code>	RowAppend , wird nun ersetzt durch AuthenticRange.AppendRow	Operation "Zeile anhängen"
<code>IsRowDeleteEnabled</code>	RowDelete , wird nun ersetzt durch AuthenticRange.DeleteRow	Operation "Zeile löschen"
<code>IsRowMoveUpEnabled</code>	RowMoveUp , wird nun ersetzt durch AuthenticRange.MoveRowUp	XML-Daten eine Zeile nach oben verschieben
<code>IsRowMoveDownEnabled</code>	RowMoveDown , wird nun ersetzt durch AuthenticRange.MoveRowDown	XML-Daten eine Zeile nach unten verschieben
<code>IsRowDuplicateEnabled</code>	RowDuplicate , wird nun ersetzt durch AuthenticRange.DuplicateRow	Aktuell ausgewählte Zeile duplizieren

4.1.9 Nicht mehr verwendete Bearbeitungsoptionen in der Authentic-Ansicht

Wenn XML-Daten als Daten in der Authentic-Ansicht angezeigt werden, können einzelne Elemente mit Hilfe von Standardbearbeitungsoperationen wie Ausschneiden, Kopieren und Einfügen bearbeitet werden. Es können jedoch nicht alle XML-Daten-Nodes bearbeitet werden. Um daher eine Bearbeitungsoperation durchzuführen, muss zuerst mit Hilfe einer Eigenschaft überprüft werden, ob eine Bearbeitung möglich ist. Anschließend wird eine Methode aufgerufen, um die Bearbeitungsoperation durchzuführen.

Die einzige Methode, für die es keinen Test gibt, ist die Methode `EditSelectAll`, die automatisch alle im Dokument angezeigten Elemente auswählt.

Im Folgenden sehen Sie eine Liste der Eigenschaften und Methoden zur Durchführung von Bearbeitungsoperationen. Jede Methode gibt einen boolschen Wert zurück und die Methoden haben keine Parameter.

Eigenschaft	Methode	Bearbeitungsoperation
<code>IsEditUndoEnabled</code>	<u>EditUndo</u> , wird nun ersetzt durch <u><code>AuthenticView.Undo</code></u>	Bearbeitungsoperation rückgängig machen
<code>IsEditRedoEnabled</code>	<u>EditRedo</u> , wird nun ersetzt durch <u><code>AuthenticView.Redo</code></u>	Bearbeitungsoperation wiederholen
<code>IsEditCopyEnabled</code>	<u>EditCopy</u> , wird nun ersetzt durch <u><code>AuthenticRange.Copy</code></u>	Ausgewählten Text in die Zwischenablage kopieren
<code>IsEditCutEnabled</code>	<u>EditCut</u> , wird nun ersetzt durch <u><code>AuthenticRange.Cut</code></u>	Ausgewählten Text ausschneiden und in der Zwischenablage ablegen
<code>IsEditPasteEnabled</code>	<u>EditPaste</u> , wird nun ersetzt durch <u><code>AuthenticRange.Paste</code></u>	Aus der Zwischenablage an der aktuellen Cursorposition einfügen
<code>IsEditClearEnabled</code>	<u>EditClear</u> , wird nun ersetzt durch <u><code>AuthenticRange.Delete</code></u>	Ausgewählten Text aus dem XML-Dokument löschen

4.2 Beispiele für die Verwendung

Die Automation-Schnittstelle von XMLSpy kann auf viele verschiedene Arten verwendet und von den verschiedensten Programmiersprachen aufgerufen werden. In diesem Abschnitt finden Sie Beispielcode, in dem einige der Funktionen der neuen [AuthenticView](#) und [AuthenticRange](#) Schnittstelle veranschaulicht werden. Diese Beispiele sollten es Ihnen erleichtern, Ihr eigenes Projekt aufzubauen. Zwar sind die Funktionen komplexer als die zu den verschiedenen Methoden und Eigenschaften beschriebenen, doch handelt es sich hierbei um Fragmente, die eventuell durch zusätzlichen Code ergänzt oder angepasst werden müssen, um in Ihrer Umgebung korrekt ausgeführt werden zu können.

Die aufgelisteten Codebeispiele sind:

- [JScript: Bubble Sort Dynamic Tables](#)
- [VBScript: Using object-level events](#)

In diesem Abschnitt werden außerdem einige Operationen kurz beschrieben, die in der Authentic-Ansicht ausgeführt werden können. Dies soll Ihnen einen Eindruck über die Verwendungsmöglichkeiten der XMLSpy API vermitteln.

4.2.1 JScript: Bubble-Sortierung dynamischer Tabellen

Mit dem folgenden JScript-Fragment wird jede dynamische Tabelle nach der Tabellenspalte sortiert, in der sich der Cursor gerade befindet. Die Sortierung erfolgt auf dem Bildschirm. Für alle ausgeführten Operationen steht der Undo-Puffer zur Verfügung.

Wenn Sie JScript auf Ihrem Computer ausführen können, was höchstwahrscheinlich der Fall ist, kopieren Sie den folgenden Code in eine Datei mit der Erweiterung ".js". Führen Sie das Skript aus, indem Sie im Windows Explorer darauf doppelklicken oder führen Sie es über die Befehlszeile aus.

```
// some useful XMLSpy enum constants
var spyAuthenticTag = 6;
var spyAuthenticTable = 9;
var spyAuthenticTableRow = 10;
var spyAuthenticTableColumn = 11;

var spyAuthenticRangeBegin = 2;

// example call for the sort table function
try
{
    var objSpy = objSpy = WScript.GetObject("", "XMLSpy.Application");
    // use current selection to indicate which column to sort
    SortCurrentTable (objSpy.ActiveDocument.AuthenticView.Selection);
}
catch (err)
{
    WScript.Echo ("Please open a document in authentic view, and select a
    table column\n" +
    "Error : (" + (err.number & 0xffff) + ") " +
    err.description);
}

// we assume that XMLSpy is running, a document with a dynamic table
// is open, and the cursor is in a table column that will
// be used for sorting.
function SortCurrentTable (objCursor)
{
    if (objCursor.IsInDynamicTable())
    {
        // calculate current column index
        var nColIndex = 0;
        while (true)
        {
            // go left column-by-column
            try { objCursor.GotoPrevious(spyAuthenticTableColumn); }
            catch (err) { break; }
            nColIndex++;
        }

        // count number of table rows, so the bubble loops become simpler.
        // goto begin of table
        var objTableStart =
        objCursor.ExpandTo(spyAuthenticTable).CollapsToBegin().Clone();
        var nRows = 1;
        while (true)
        {
            // go down row-by-row
            try { objTableStart.GotoNext(spyAuthenticTableRow); }
            catch (err) { break; }
            nRows++;
        }

        // bubble sort through table
        for (var i = 0; i < nRows - 1; i++)
```

```

{
    // select correct column in first table row
    var objBubble =
    objCursor.ExpandTo(spyAuthenticTable).CollapsToBegin().Clone();
    objBubble.Goto (spyAuthenticTableColumn, nColIndex,
    spyAuthenticRangeBegin).ExpandTo(spyAuthenticTag);

    // bubble this row down as far as necessary
    for (var j = 0; j < nRows - i - 1; j++)
    {
        var strField1 = objBubble.Text;
        // now look for the comparison table cell: start of next row and right
        // of the correct column
        var strField2 = objBubble.GotoNext(spyAuthenticTableRow).
            Goto (spyAuthenticTableColumn, nColIndex,
            spyAuthenticRangeBegin).
            ExpandTo(spyAuthenticTag).Text;
        if (strField1 > strField2)
        {
            objBubble.MoveRowUp(); // swap the rows
            // and re-calculate objBubble to select the cell to bubble
            objBubble.GotoNext(spyAuthenticTableRow).
                Goto (spyAuthenticTableColumn, nColIndex,
                spyAuthenticRangeBegin).
                ExpandTo(spyAuthenticTag);
        }
    }
}
else
    WScript.Echo ("please, select a table cell first");
}

```

4.2.2 VBScript: Verwendung von Events auf Objektebene

Die Authentic-Ansicht unterstützt nun die Verbindung zu Events pro Objekt. Die Implementierung dieser Funktion basiert auf COM Connection Points und steht in Umgebungen zur Verfügung, die diesen Mechanismus unterstützen.

Das folgende Beispiel ist ein VB-Codefragment, in dem gezeigt wird, wie man von einem VB-Projekt aus eine Verbindung zu Events auf Objektebene herstellt.

```
'
'-----
' VB code snippet - connecting to object level events
'-----

Dim objSpy As XMLSpyLib.Application
' use VBA keyword WithEvents to automatically connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView

' this is the event callback routine that will be connected to the
' OnSelectionChanged event of object objView
Private Sub objView_OnSelectionChanged (ByVal i_ipNewRange As
XMLSpyLib.IAuthenticRange)
    MsgBox ("new selection: " & i_ipNewRange.Text)
End Sub

' this is the event callback routine that will be connected to the
' OnSelectionChanged event of object objView
Private Sub objView_OnSelectionChanged (ByVal i_ipNewRange As
XMLSpyLib.IAuthenticRange)
    MsgBox ("new selection: " & i_ipNewRange.Text)
End Sub

' this is the event callback routine that will be connected to the
' OnMouseEvent event of object objView.
' If you click with the left mouse button while pressing a control key,
' the current selection will be set to the tag below the current
' mouse cursor position
Private Function objView_OnMouseEvent (ByVal i_nXPos As Long, ByVal i_nYPos
As Long, ByVal i_eMouseEvent As XMLSpyLib.SPYMouseEvent, ByVal i_pRange As
XMLSpyLib.IAuthenticRange) As Boolean
    If (i_eMouseEvent = (XMLSpyLib.spyLeftButtonDownMask Or
XMLSpyLib.spyCtrlKeyDownMask)) Then
        On Error Resume Next
        i_pRange.Select
        objView_OnMouseEvent = True
    Else
        objView_OnMouseEvent = False
    End If
End Function

' connect to running XMLSpy application
' this code will most likely be in some Form_Load() subroutine
Set objSpy = GetObject("", "XMLSpy.Application")
objSpy.ShowApplication (True)
If (objSpy.ActiveDocument Is Nothing) Then
    Dim objDoc As XMLSpyLib.Document
    Rem replace path below With a valid absolute path on your machine
    Set objDoc = objSpy.Documents.OpenFile("InputData\orgchart.xml", False)
End If

If (objSpy.ActiveDocument.CurrentViewMode <> spyViewAuthentic) Then
    objSpy.ActiveDocument.SwitchViewMode (spyViewAuthentic)
End If
Set objView = objSpy.ActiveDocument.AuthenticView
' now the object objView is set and selection change events are received
```

```
' continue here with something useful ...  
' and serve the windows message loop  
  
' if you want to stop receiving events from object objView  
' add the following line somewhere appropriate  
Set objView = Nothing
```

4.3 Schnittstellen

Objekthierarchie

[Application](#)

[SpyProject](#)

[SpyProjectItems](#)

[SpyProjectItem](#)

[Documents](#)

[Document](#)

[GridView](#)

[AuthenticView](#)

[AuthenticRange](#)

[AuthenticDataTransfer](#) (vormals DocEditDataTransfer)

[OldAuthenticView](#) (vormals DocEditView, **wird nicht mehr verwendet**, jetzt ersetzt durch [AuthenticView](#) und [AuthenticRange](#))

[AuthenticSelection](#) (vormals DocEditSelection, **wird nicht mehr verwendet**, jetzt ersetzt durch [AuthenticRange](#))

[AuthenticEvent](#) (vormals DocEditEvent, **wird nicht mehr verwendet**)

[AuthenticDataTransfer](#) (vormals DocEditDataTransfer)

[XMLData](#)

[Dialogs](#)

[CodeGeneratorDlg](#)

[FileSelectionDlg](#)

[SchemaDocumentationDlg](#)

[DatabaseConnection](#)

[ExportSettings](#)

[TextImportExportSettings](#)

[ElementList](#)

[ElementListItem](#)

[Enumerations](#)

Beschreibung

Dieses Kapitel enthält die Referenz der Authentic Desktop 1.5 Typlibibliothek.

Die meisten der hier beschriebenen Beispiele sind in VisualBasic verfasst. Bei diesen Codefragmenten wird davon ausgegangen, dass eine Variable namens **objSpy vom Typ Application** definiert und gesetzt ist. Einige der Codebeispiele wurden auch in JavaScript geschrieben.

4.3.1 Application

See also

Methoden

[GetDatabaseImportElementList](#)

[GetDatabaseSettings](#)

[GetDatabaseTables](#)

[ImportFromDatabase](#)

[GetTextImportElementList](#)

[GetTextImportExportSettings](#)

[ImportFromText](#)

[ImportFromWord](#)

[ImportFromSchema](#)

[GetExportSettings](#)

[NewProject](#)

[OpenProject](#)

[AddMacroMenuItem](#)

[ClearMacroMenu](#)

[ShowForm](#)

[ShowApplication](#)

[URLDelete](#)

[URLMakeDirectory](#)

[FindFiles](#)

[Quit](#)

Eigenschaften

[Application](#)

[Parent](#)

[ActiveDocument](#)

[Documents](#)

[CurrentProject](#)

[Dialogs](#)

[WarningNumber](#)

[WarningText](#)

Beschreibung

Application ist die Root für alle anderen Objekte. Es ist das einzige Objekt, das Sie mit `CreateObject` (VisualBasic) oder anderen ähnlichen COM-Funktionen erstellen können.

Beispiel

```
Dim objSpy As Application
Set objSpy = CreateObject("XMLSpy.Application")
```

Events

OnBeforeOpenDocument

Siehe auch

Event: *OnBeforeOpenDocument*(*objDialog* als [FileSelectionDlg](#))

Beschreibung

Dieses Event wird immer dann ausgelöst, wenn ein Dokument mit dem Menübefehl OpenFile oder OpenURL geöffnet wird. Es wird nach dem Auswählen einer Dokumentdatei aber vor dem Öffnen des Dokuments gesendet. Das Dialogfeldobjekt zur Auswahl der Datei wird mit dem Namen der ausgewählten Dokumentdatei initialisiert. Sie können diese Auswahl ändern. Um mit dem Öffnen des Dokuments fortzufahren, belassen Sie die Eigenschaft [FileSelectionDlg.DialogAction](#) von *io_objDialog* auf dem Standardwert [spyDialogOK](#). Um das Öffnen des Dokuments abzuberechnen, setzen Sie die Eigenschaft auf [spyDialogCancel](#).

Beispiele

Im Folgenden finden Sie Beispiele dafür, wie ein Skript für dieses Event erstellt werden kann.

XMLSpy scripting environment - VBScript:

```
Function On_BeforeOpenDocument(objDialog)
EndFunction
```

XMLSpy scripting environment - JScript:

```
function On_BeforeOpenDocument(objDialog)
{
}
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugin.OnEvent (26, ...) //nEventId=26
```

OnBeforeOpenProject

Siehe auch

Event: *OnBeforeOpenProject*(*objDialog* als [FileSelectionDlg](#))

Beschreibung

Dieses Event wird ausgelöst, wenn eine Projektdatei ausgewählt, aber bevor sie geöffnet wird. Das Dialogfeldobjekt zur Auswahl der Datei wird mit dem Namen der ausgewählten Projektdatei initialisiert. Sie können diese Auswahl ändern. Um mit dem Öffnen des Projekts fortzufahren, belassen Sie die Eigenschaft [FileSelectionDlg.DialogAction](#) von *io_objDialog* auf dem Standardwert [spyDialogOK](#). Um das Öffnen des Projekts abzuberechnen, setzen Sie die Eigenschaft auf [spyDialogCancel](#).

Beispiele

Im Folgenden sehen Sie einige Beispiele, wie ein Skript für dieses Event erstellt werden kann.

XMLSpy scripting environment - VBScript:

```
Function On_BeforeOpenProject(objDialog)
EndFunction
```

XMLSpy scripting environment - JScript:

```
function On_BeforeOpenProject(objDialog)
{
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (25, ...) //nEventId=25
```

OnDocumentOpened

Siehe auch

Event: OnDocumentOpened(objDocument als [Document](#))

Beschreibung

Dieses Event wird ausgelöst, wenn ein Dokument in Authentic Desktop geöffnet wird. Voraussetzung ist, dass eine Datei über das Dialogfeld OpenFile oder OpenURL geöffnet oder eine neue Datei erstellt oder eine Datei in Authentic Desktop gezogen wird. Das neue Dokument wird als Parameter übergeben. Die Operation kann nicht abgebrochen werden.

Beispiele

Im Folgenden finden Sie Beispiele dafür, wie ein Skript für dieses Event erstellt werden kann.

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_OpenDocument(objDocument)
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_OpenDocument(objDocument)
{
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (7, ...) //nEventId=7
```

OnProjectOpened

Siehe auch

Event: OnProjectOpened(objProject als [SpyProject](#))

Beschreibung

Dieses Event wird ausgelöst, wenn ein Projekt in Authentic Desktop geöffnet wird. Das neue Projekt wird als Parameter übergeben.

Beispiele

Im Folgenden finden Sie Beispiele dafür, wie ein Skript für dieses Event erstellt werden kann.

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_OpenProject(objProject)
EndFunction
```


XMLSpy Skripting-Umgebung - JScript:

```
function On_OpenProject(objProject)
{
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (6, ...) //nEvent=6
```

ActiveDocument**Siehe auch****Eigenschaft:** [ActiveDocument](#) als [Document](#)**Beschreibung**

Referenz auf das aktive Dokument. Wenn kein Dokument geöffnet ist, ist `ActiveDocument` null (nothing).

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

AddMacroMenuItem**Siehe auch****Methode:** [AddMacroMenuItem](#)([strMacro](#) als String, [strDisplayText](#) als String)**Beschreibung**

Fügt einen Menübefehl zum Menü **Extras** hinzu. Dieser neue Menübefehl ruft das von `strMacro` definierte Makro auf. Siehe auch [Aufrufen von Makros](#) von Authentic Desktop".

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
- 1108 Die Anzahl der Makroelemente ist auf 16 beschränkt.

Application**Siehe auch****Eigenschaft:** [Application](#) als [Application](#) (schreibgeschützt)**Beschreibung**

Ruft das Authentic Desktop application-Objekt auf.

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ClearMacroMenu

Siehe auch

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

Rückgabewert

Keiner

Beschreibung

Entfernt alle Menübefehle aus dem Menü **Extras**. Siehe auch Aufrufen von Makros von Authentic Desktop".

Fehler

1111 Das application-Objekt ist nicht mehr gültig.

CurrentProject

Siehe auch

Eigenschaft: [CurrentProject](#) als [SpyProject](#)

Beschreibung

Referenz auf das aktive Dokument. Wenn kein Projekt geöffnet ist, ist [CurrentProject](#) gleich Null (nothing).

Fehler

1111 Das application-Objekt ist nicht mehr gültig.

1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Dialogs

Siehe auch

Eigenschaft: [Dialogs](#) als [Dialogs](#) (schreibgeschützt)

Beschreibung

Ruft die integrierten Dialogfelder von Authentic Desktop auf.

Fehler

1111 Das application-Objekt ist nicht mehr gültig.

1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Documents

Siehe auch

Eigenschaft: [Documents](#) als [Documents](#)

Beschreibung

Sammlung aller offenen Dokumente. Siehe auch [Einfacher Dokumentzugriff](#).

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

FindInFiles

Siehe auch

Method: [FindInFiles](#)(*pSettings* als [FindInFilesDlg](#)) als [FindInFilesResults](#)

Beschreibung

Gibt ein [FindInFilesResults](#) Objekt zurück, das Informationen über die Dateien enthält, die den angegebenen Kriterien entsprechen.

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

GetDatabaseImportElementList

Siehe auch

Method: [GetDatabaseImportElementList](#)(*pImportSettings* als [DatabaseConnection](#)) als [ElementList](#)

Beschreibung

Die Funktion gibt eine Sammlung von [ElementListItems](#) zurück, wobei die [ElementListItem.Name](#) Eigenschaften die Namen der Felder enthalten, die für den Import ausgewählt werden können und die [ElementListItem.ElementKind](#) Eigenschaften, je nachdem, welcher Wert in [DatabaseConnection.AsAttributes](#) übergeben wird, entweder auf [spyXMLDataAttr](#) oder [spyXMLDataElement](#) initialisiert werden. Diese Liste dient als Filter dafür, was schließlich durch einen zukünftigen Aufruf von [ImportFromDatabase](#) importiert wird. Verwenden Sie [ElementList.RemoveElement](#) um Felder vom Export auszuschließen.

Für die Herstellung der Verbindung zur Datenbank muss eines der Objekte [DatabaseConnection.File](#), [DatabaseConnection.ADOConnection](#) oder [DatabaseConnection.ODBCConnection](#) sowie [DatabaseConnection.SQLSelect](#) ausgefüllt werden. Verwenden Sie die Eigenschaft [DatabaseConnection.AsAttributes](#) um [ElementListItem.ElementKind](#) der erzeugten Elementliste entweder auf [spyXMLDataAttr](#) oder [spyXMLDataElement](#) zu initialisieren.

Beispiel

Siehe Beispiel unter [ImportFromDatabase](#).

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
- 1107 Import aus Datenbank ist fehlgeschlagen.
- 1112 Es wurde eine ungültige Datenbank angegeben.
- 1114 Select-Anweisung fehlt.
- 1119 Die Liste der Datenbankelemente konnte nicht importiert werden.

GetDatabaseSettings

Siehe auch

Methode: `GetDatabaseSettings()` als [DatabaseConnection](#)

Beschreibung

`GetDatabaseSettings` erstellt ein neues Objekt für Datenbankeinstellungen. Mit Hilfe des Objekts werden die Datenbankverbindungsparameter für die Methoden [GetDatabaseTables](#), [GetDatabaseImportElementList](#), [ImportFromDatabase](#), [ImportFromSchema](#) und [ExportToDatabase](#) definiert.

Beispiel

Siehe Beispiel von [ImportFromDatabase](#).

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

GetDatabaseTables

Siehe auch

Methode: `GetDatabaseTables(pImportSettings als DatabaseConnection)` als [ElementList](#)

Beschreibung

`GetDatabaseTables` liest die Tabellennamen aus der in *pImportSettings* definierten Datenbank. Für die Datenbankverbindung muss eine der folgenden Eigenschaften ausgefüllt werden [DatabaseConnection.File](#), [DatabaseConnection.ADOConnection](#) oder [DatabaseConnection.ODBCConnection](#). Alle anderen Eigenschaften werden ignoriert. Die Funktion gibt eine Sammlung von `ElementListItem`s zurück, wobei die [ElementListItem.Name](#) Eigenschaften die Namen der in der angegebenen Datenbank gespeicherten Tabellen enthält. Die restlichen Eigenschaften von [ElementListItem](#) werden nicht verwendet.

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
- 1112 Es wurde eine ungültige Datenbank angegeben
- 1113 Fehler beim Lesen der Datenbanktabellendaten.
- 1118 Die Abfrage der Datenbanktabelle ist fehlgeschlagen.

Beispiel

```
Dim objImpSettings As DatabaseConnection
Set objImpSettings = objSpy.GetDatabaseSettings
objImpSettings.ADOConnection = TxtADO.Text

'store table names in list box
ListTables.Clear

Dim objList As ElementList
Dim objItem As ElementListItem
On Error GoTo ErrorHandler
Set objList = objSpy.GetDatabaseTables(objImpSettings)
```

```

For Each objItem In objList
    ListTables.AddItem objItem.Name
Next

```

GetExportSettings

Siehe auch

Methode: `GetExportSettings()` als [ExportSettings](#) (schreibgeschützt)

Beschreibung

`GetExportSettings` erstellt ein neues Objekt allgemeiner Exporteinstellungen. Mit Hilfe dieses Objekts werden die Parameter an die Exportfunktionen übergeben und das Verhalten der Exportaufrufe wird definiert. Siehe auch die Exportfunktionen aus [Document](#) und die Beispiele unter [Import und Export](#).

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

GetTextImportElementList

Siehe auch

Methode: `GetTextImportElementList(pImportSettings` als [TextImportExportSettings](#)) als [ElementList](#)

Beschreibung

`GetTextImportElementList` ruft Importinformationen über die in `pImportSettings` definierte Textdatei ab. Die Funktion gibt eine Sammlung von `ElementListItems` zurück, wobei die [ElementListItem.Name](#) Eigenschaften die Namen der in der Datei gefundenen Felder enthält. Die Werte der restlichen Eigenschaften werden nicht definiert.

Wenn die Textdatei keine Spaltenüberschriften enthält, setzen Sie `pImportSettings` [HeaderRow](#) auf `True`. Die erzeugte Elementliste enthält allgemeine Spaltennamen wie 'Field1' usw.

Siehe auch [Import und Export von Daten](#).

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
- 1107 Import aus Datenbank ist fehlgeschlagen.
- 1115 Fehler beim Import der Textelementliste. Für die Importdatei kann kein Parser erstellt werden.
- 1116 Fehler beim Import der Textelementliste.

Beispiel

```

' -----
' VBA client code fragment - import selected fields from text file
' -----
Dim objImpSettings As TextImportExportSettings
Set objImpSettings = objSpy.GetTextImportExportSettings

objImpSettings.ImportFile = "C:\ImportMe.txt"
objImpSettings.HeaderRow = False

```

```

Dim objList As ElementList
Set objList = objSpy.GetTextImportElementList(objImpSettings)

'exclude first column
objList.RemoveItem 1

Dim objImpDoc As Document
On Error Resume Next
Set objImpDoc = objSpy.ImportFromText(objImpSettings, objList)
CheckForError

```

GetTextImportExportSettings

Siehe auch

Methode: `GetTextImportExportSettings()` als [TextImportExportSettings](#) (schreibgeschützt)

Beschreibung

`GetTextImportExportSettings` erstellt ein neues Objekt der allgemeinen Import- und Exporteinstellungen für Textdateien. Siehe auch Beispiel zu [Application.GetTextImportElementList](#) und [Import and Export](#).

Siehe auch [Import und Export von Daten](#).

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ImportFromDatabase

Siehe auch

Methode: `ImportFromDatabase(pImportSettings als DatabaseConnection, pElementList als ElementList) als Document`

Rückgabewert

Erstellt ein neues Dokument, das die aus der Datenbank importierten Daten enthält.

Beschreibung

`ImportFromDatabase` importiert Daten aus einer Datenbank, wie in `pImportSettings` definiert und erstellt ein neues Dokument, das diese Daten enthält. Eine der folgenden Eigenschaften muss dazu ausgefüllt werden: [DatabaseConnection.File](#), [DatabaseConnection.ADOConnection](#) oder [DatabaseConnection.ODBCConnection](#) und [DatabaseConnection.SQLSelect](#). Zusätzlich dazu können Sie mittels [DatabaseConnection.AsAttributes](#), [DatabaseConnection.ExcludeKeys](#), [DatabaseConnection.IncludeEmptyElements](#) und [NumberDateTimeFormat](#) weitere Parameter für den Import eingeben.

Der Parameter `pElementList` definiert, ob Felder der ausgewählten Daten in das neu erstellte Dokument geschrieben werden sollen und welche davon als Elemente und welche als Attribute erstellt werden sollen. Dieser Parameter kann NULL sein, d.h. alle ausgewählten Felder werden als XML-Elemente importiert.

Erforderliche Schritte, die vor dem Import von Daten aus einer Datenbank durchzuführen sind, werden unter [GetDatabaseSettings](#) und [GetDatabaseImportElementList](#)

beschrieben.

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
- 1107 Import aus Datenbank ist fehlgeschlagen.
- 1112 Es wurde eine ungültige Datenbank angegeben.
- 1114 Select-Anweisung fehlt.
- 1117 Transformation in XML ist fehlgeschlagen.
- 1120 Datenbankimport ist fehlgeschlagen.

Beispiel

```
Dim objImpSettings As DatabaseConnection
Set objImpSettings = objSpy.GetDatabaseSettings

objImpSettings.ADOConnection = strADOConnection
objImpSettings.SQLSelect = "SELECT * FROM MyTable"

Dim objDoc As Document
On Error Resume Next
Set objDoc = objSpy.ImportFromDatabase(objImpSettings,
objSpy.GetDatabaseImportElementList(objImpSettings))
' CheckForError here
```

ImportFromSchema

Siehe auch

Methode: `ImportFromSchema(pImportSettings als DatabaseConnection, strTable als String, pSchemaDoc als Document)` als [Document](#)

Rückgabewert

Erstellt ein neues Dokument, das die aus der Datenbank gemäß der Schemadefinition in *pSchemaDoc* importierten Daten enthält.

Beschreibung

`ImportFromSchema` importiert Daten aus einer in `pImportSettings` definierten Datenbank. Dazu muss eine der folgenden Eigenschaften ausgefüllt werden [DatabaseConnection.File](#), [DatabaseConnection.ADOConnection](#) oder [DatabaseConnection.ODBCConnection](#). Zusätzlich dazu können Sie mittels [DatabaseConnection.AsAttributes](#), [DatabaseConnection.ExcludeKeys](#) und [NumberDateTimeFormat](#) weitere Parameter für den Import eingeben. Alle anderen Parameter werden ignoriert.

`ImportFromSchema` verwendet zum Auswählen der Daten keine explizite SQL-Anweisung. Statt dessen muss eine Strukturdefinition des Dokuments in Form eines XML-Schema-Dokuments in *pSchemaDoc* erstellt werden. Anhand dieser Definition wird automatisch die Datenbank-Select-Anweisung abgeleitet. Definieren Sie in *strTable* den Tabellennamen der Import-Root, die der Root-Node im neuen Dokument werden soll.

Erforderliche Schritte, die vor dem Import von Daten aus einer Datenbank anhand einer Schemadefinition durchzuführen sind, werden unter [GetDatabaseSettings](#) und [GetDatabaseTables](#) beschrieben. Verwenden Sie zum Erstellen einer Schemadefinitionsdatei den Befehl "Datenbankschema erstellen" aus dem Authentic Desktop -Menü "Konvertieren".

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
- 1107 Import aus Datenbank ist fehlgeschlagen.
- 1112 Es wurde eine ungültige Datenbank angegeben.
- 1120 Datenbankimport ist fehlgeschlagen.
- 1121 Es konnte kein Validator für das angegebene Schema erstellt werden.
- 1122 Fehler beim Parsen des Schemas für den Datenbankimport.

ImportFromText**Siehe auch**

Methode: `ImportFromText`(*pImportSettings* als [TextImportExportSettings](#), *pElementList* als [ElementList](#)) als [Document](#)

Beschreibung

`ImportFromText` importiert die Textdatei, wie in `pImportSettings` definiert. Der Parameter `pElementList` kann als Importfilter verwendet werden. Übergeben Sie entweder die durch einen vorherigen Aufruf von [GetTextImportElementList](#) zurückgegebene Liste oder den Wert `nil` um alle Spalten zu importieren. Um den Import unnötiger Spalten zu vermeiden, entfernen Sie die entsprechenden Feldnamen mit Hilfe von [ElementList.RemoveElement](#) aus `pElementList` bevor Sie `ImportFromText` aufrufen.

Die Methode gibt das neu erstellte Dokument mit den importierten Daten zurück. Dieses Dokument ist dasselbe wie das aktive Dokument von Authentic Desktop.

Siehe auch [Import und Export von Daten](#)

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
- 1107 Import aus Textdatei ist fehlgeschlagen
- 1117 Transformation in XML ist fehlgeschlagen.

Beispiel

```
' -----
' VBA client code fragment - import from text file
' -----

Dim objImpSettings As TextImportExportSettings
Set objImpSettings = objSpy.GetTextImportExportSettings

objImpSettings.ImportFile = strFileName
objImpSettings.HeaderRow = False

Dim objImpDoc As Document
On Error Resume Next
Set objImpDoc = objSpy.ImportFromText(objImpSettings,
    objSpy.GetTextImportElementList(objImpSettings))

CheckForError
```

ImportFromWord**Siehe auch**

Methode: `ImportFromWord`(*strFile* als String) als [Document](#)

Beschreibung

`ImportFromWord` importiert das MS Word-Dokument `strFile` in ein neues XML-Dokument.

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
Import aus Dokument ist fehlgeschlagen.

NewProject**Siehe auch**

Methode: `NewProject`(`strPath` als String, `bDiscardCurrent` als Boolean)

Beschreibung

`NewProject` erstellt ein neues Projekt.

Wenn es bereits ein offenes Projekt gibt, das geändert wurde und `bDiscardCurrent` "false" ist, schlägt `NewProject()` fehl.

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1102 Es ist bereits ein Projekt geöffnet, aber `bDiscardCurrent` ist `true`.
- 1103 Das neue Projekt konnte nicht erstellt werden.

OpenProject**Siehe auch**

Methode: `OpenProject`(`strPath` als String, `bDiscardCurrent` als Boolean, `bDialog` als Boolean)

Parameter

`strPath`

Der Pfad und Dateiname des zu öffnenden Projekts. Kann leer sein, wenn `bDialog` "true" ist.

`bDiscardCurrent`

Aktuell geöffnetes Projekt verwerfen. Änderungen daran gehen verloren.

`bDialog`

Dialogfelder für Benutzereingabe anzeigen.

Rückgabewert

Keiner

Beschreibung

`OpenProject` öffnet ein vorhandenes Projekt. Wenn bereits ein Projekt geöffnet ist, das geändert wurde und `bDiscardCurrent` "false" ist, schlägt `OpenProject()` fehl.

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.

- 1101 Das angegebene Projekt kann nicht geöffnet werden.
- 1102 Es ist bereits ein Projekt geöffnet, aber *bDiscardCurrent* ist *true*.

Parent

Siehe auch

Eigenschaft: *Parent* als [Application](#) (schreibgeschützt)

Beschreibung

Ruft das Authentic Desktop application-Objekt auf.

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Quit

Siehe auch

Method: [Quit\(\)](#)

Rückgabewert

Keiner

Beschreibung

Diese Methode beendet Authentic Desktop. Alle geänderten Dokumente werden ohne Speicherung der Änderungen geschlossen. Dies gilt auch für ein offenes Projekt.

Wenn Authentic Desktop automatisch von einem Client-Programm als Automation Server gestartet wird, wird die Applikation bei Schließen des Client-Programms nicht automatisch geschlossen, falls noch ein Projekt oder Dokument geöffnet ist. Verwenden Sie die Quit-Methode, um sicherzustellen, dass das Programm automatisch beendet wird.

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.

ReloadSettings

Siehe auch

Methode: [ReloadSettings](#)

Rückgabewert

Beschreibung

Die Applikationseinstellungen werden aus der Registrierungsdatei neu geladen.

Verfügbar mit Typbibliothek Version 1.5

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.

RunMacro

Siehe auch

Methode: `RunMacro`(`strMacro` als String)

Rückgabewert

Beschreibung

Ruft das angegebene Makro entweder von den Projektskripts (falls vorhanden) oder den globalen Skripts aus auf.

Verfügbar mit Typbibliothek Version 1.5

Fehler

1111 Das application-Objekt ist nicht mehr gültig.

ScriptingEnvironment

Siehe auch

Eigenschaft: `ScriptingEnvironment` als IUnknown (schreibgeschützt)

Beschreibung

Referenz auf jede aktive Skripting-Umgebung. Mit Hilfe dieser Eigenschaft kann die TypeLibrary der XMLSpyFormEditor.exe Applikation, die als aktuelle Skripting-Umgebung verwendet wird, aufgerufen werden.

Verfügbar mit Typbibliothek Version 1.5

Fehler

1111 Das application-Objekt ist nicht mehr gültig.

1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowApplication

Siehe auch

Methode: `ShowApplication`(`bShow` als Boolean)

Rückgabewert

Keiner

Beschreibung

Die Methode blendet Authentic Desktop ein(`bShow = True`) oder aus (`bShow = False`).

Fehler

1110 Das application-Objekt ist nicht mehr gültig.

ShowFindInFiles

Siehe auch

Methode: `ShowFindInFiles`(`pSettings` als `FindFilesDlg`) als Boolean

Rückgabewert

Gibt "false" zurück, wenn der Benutzer die Schaltfläche "Abbrechen" gedrückt hat, andernfalls "true".

Beschreibung

Zeigt das Dialogfeld FindInFiles mit den vordefinierten Einstellungen an. Die benutzerdefinierten Änderungen an den Einstellungen werden im übergebenen Dialogfeldobjekt gespeichert.

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.

ShowForm**Siehe auch**

Methode: `ShowForm(strFormName als String) als Long`

Rückgabewert

Gibt Null zurück, wenn der Benutzer eine Abbrechen-Schaltfläche betätigt hat, oder die Formularaufrufe `TheView.Cancel()` .

Beschreibung

Zeigt das Formular `strFormName` an.

Formulare, Events und Event Handler können mit der Skripting Umgebung erstellt werden. Wählen Sie im Menü **Extras** den Befehl "Skripting-Umgebung...", um die Skripting-Umgebung aufzurufen..

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben..

URLDelete**Siehe auch**

Methode: `URLDelete( strURL als String, strUser als String, strPassword als String)`

Rückgabewert

Keiner

Beschreibung

Die Methode löscht die Datei an der URL `strURL` .

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1109 Fehler beim Löschen der Datei an der angegebenen URL.

URLMakeDirectory

Siehe auch

Methode: `URLMakeDirectory`(*strURL* als String, *strUser* als String, *strPassword* als String)

Rückgabewert

Keiner

Beschreibung

Die Methode erstellt ein neues Verzeichnis an der URL *strURL* .

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Es wurde ein ungültiger Parameter angegeben.

Visible

Siehe auch

Eigenschaft: `Visible` als Boolean

Beschreibung

Setzt oder ruft das Sichtbarkeitsattribut von Authentic Desktop ab. Diese Standard-Automation-Eigenschaft macht die Verwendung von [ShowApplication](#) überflüssig.

Fehler

- 1110 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

WarningNumber

Siehe auch

Eigenschaft: `WarningNumber` als integer

Beschreibung

Einige Methoden füllen die Eigenschaft `WarningNumber` mit zusätzlichen Informationen, wenn ein Fehler auftritt.

Derzeit wird diese Eigenschaft nur mit [Documents.OpenFile](#) befüllt.

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

WarningText

Siehe auch

Eigenschaft: `WarningText` als String

Beschreibung

Einge Methoden befüllen die Eigenschaft `WarningText` mit zusätzlichen Informationen, wenn ein Fehler auftritt.

Derzeit wird diese Eigenschaft nur mit [Documents.OpenFile](#) befüllt.

Fehler

- 1111 Das application-Objekt ist nicht mehr gültig.
- 1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

4.3.2 AuthenticDataTransfer

Umbenannt von DocEditDataTransfer in AuthenticDataTransfer

Das `DocEditView` Objekt wird in `AuthenticView` umbenannt.
`DocEditSelection` wird in `AuthenticSelection` umbenannt.
`DocEditEvent` wird in `AuthenticEvent` umbenannt.
`DocEditDataTransfer` wird in `AuthenticDataTransfer` umbenannt.

Es wird empfohlen, diese Objekte - mit Ausnahme von `AuthenticDataTransfer` - nicht mehr zu verwenden. Wir werden die bestehenden Funktionalitäten vorerst auf unbestimmte Zeit weiter unterstützen, doch werden keine neuen Funktionen zu diesen Schnittstellen hinzugefügt. Alle bisher in [DocEditView](#), [DocEditSelection](#), [DocEditEvent](#) und [DocEditDataTransfer](#) verfügbaren Funktionalitäten stehen nun über [AuthenticView](#), [AuthenticRange](#) und [AuthenticDataTransfer](#) zur Verfügung. Es wurden viele neue Funktionen hinzugefügt.

Beispiele zum Migrieren von DocEdit in Authentic finden Sie unter der Beschreibung der verschiedenen Methoden und Eigenschaften der einzelnen DocEdit-Objekte.

Siehe auch

Methoden

[getData](#)

Eigenschaften

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

Beschreibung

Die Events `OnDragOver` und `OnBeforeDrop` liefern mit einer Instanz des Typs `AuthenticDataTransfer` Informationen über das gezogene Objekt. Sie enthält eine Beschreibung des gezogenen Objekts und seines Inhalts. Letzterer steht entweder als String oder als Pointer auf ein COM-Objekt, das die `IUnknown`-Schnittstelle unterstützt, zur Verfügung.

dropEffect

Siehe auch

Eigenschaft: `dropEffect` als long

Beschreibung

Die Eigenschaft speichert den Drop-Effekt aus dem Standard-Event Handler. Sie können den Drop-Effekt definieren, wenn Sie diesen Wert ändern und für den Event Handler TRUE zurückgeben. (Oder wenn Sie [AuthenticEvent.cancelBubble](#) auf TRUE setzen, wenn Sie noch die veraltete `AuthenticEvent` Schnittstelle verwenden).

Fehler

2101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

getData

Siehe auch

Methode: [getData\(\)](#) als Variant

Beschreibung

Ruft die Daten im Zusammenhang mit dem gezogenen Objekt ab. Abhängig von [AuthenticDataTransfer.type](#) handelt es sich hierbei entweder um einen String oder um einen COM Interface Pointer vom Typ `Unknown`.

Fehler

2101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ownDrag

Siehe auch

Eigenschaft: [ownDrag](#) als Boolean (schreibgeschützt)

Beschreibung

Die Eigenschaft ist `TRUE`, wenn die aktuelle Quelle für das Ziehen des Objekts von innerhalb der Authentic-Ansicht kommt.

Fehler

2101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

type

Siehe auch

Eigenschaft: [type](#) als String (schreibgeschützt)

Beschreibung

Enthält den Datentyp, den Sie mit der [DocEditDataTransfer.getData](#) erhalten.

Derzeit werden die folgenden Datentypen unterstützt:

OWN	Daten aus der Authentic-Ansicht selbst
TEXT	Unformatierter Text
UNICODETEXT	Unformatierter Text als UNICODE

Fehler

2101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

4.3.3 AuthenticRange

Siehe auch

Die erste Tabelle enthält eine Liste der Eigenschaften und Methoden von `AuthenticRange`, die zum Navigieren durch das Dokument und zum Auswählen bestimmter Abschnitte verwendet werden.

Eigenschaften

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

[Collapse](#)
[CollapseToBegin](#)
[CollapseToEnd](#)
[ExpandTo](#)
[Goto](#)
[GotoNext](#)
[GotoPrevious](#)
[IsEmpty](#)
[IsEqual](#)

Methoden

[MoveBegin](#)
[MoveEnd](#)
[NextCursorPosition](#)
[PreviousCursorPosition](#)
[Select](#)
[SelectNext](#)
[SelectPrevious](#)
[SetFromRange](#)

Die folgende Tabelle enthält eine Liste der Methoden zur Bearbeitung von Inhalt. Die meisten dieser Methoden können über Kontextmenüs aufgerufen werden.

Eigenschaften

[Text](#)

Edit operations

[Copy](#)
[Cut](#)
[Delete](#)
[IsCopyEnabled](#)
[IsCutEnabled](#)
[IsDeleteEnabled](#)
[IsPasteEnabled](#)
[Paste](#)

Dynamic table operations

[AppendRow](#)
[DeleteRow](#)
[DuplicateRow](#)
[InsertRow](#)
[IsFirstRow](#)
[IsDynamicTable](#)
[IsLastRow](#)
[MoveRowDown](#)
[MoveRowUp](#)

Die folgenden Methoden liefern die Funktionen der Authentic-Eingabehilfenfenster für Bereichsobjekte.

Operations of the entry helper windows

Elements

[CanPerformActionWithin](#)
[CanPerformAction](#)
[PerformAction](#)

Attributes

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

Entities

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

Beschreibung

`AuthenticRange` Objekte sind die Cursor-Auswahl der Automation-Schnittstelle. Sie können damit auf jede Cursorposition in der Authentic-Ansicht zeigen oder einen Dokumentabschnitt auswählen. Die für `AuthenticRange` Objekte verfügbaren Operationen funktionieren anschließend an dieser Auswahl auf dieselbe Art wie die entsprechenden Operationen der Benutzeroberfläche an der aktuellen Auswahl des Benutzers. Der wichtigste Unterschied besteht darin, dass Sie beliebig viele `AuthenticRange` Objekte gleichzeitig verwenden können, während es auf der Benutzeroberfläche immer nur genau eine Cursorauswahl gibt.

Um zu Anfang ein Bereichsobjekt abzurufen, verwenden Sie [AuthenticView.Selection](#).

um einen Bereich zu erhalten, der der aktuellen Cursorauswahl auf der Benutzeroberfläche entspricht. Alternativ dazu stehen einige einfache Bereiche über die schreibgeschützten Eigenschaften [AuthenticView.DocumentBegin](#), [AuthenticView.DocumentEnd](#) und [AuthenticView.WholeDocument](#) zur Verfügung. Die flexibelste Methode ist [AuthenticView.Goto](#). Sie können damit in einem Aufruf zu einem bestimmten Bereich des Dokuments navigieren. Um eine komplexere Auswahl zu treffen, kombinieren Sie die obigen Methoden mit den verschiedenen Navigationsmethoden zu den in der ersten Tabelle auf dieser Seite aufgelisteten Bereichsobjekten.

Eine weitere Methode, einen Dokumentabschnitt auszuwählen, ist die Positioneigenschaften des Bereichsobjekts zu verwenden. Es stehen zwei Systeme zur Positionierung zur Verfügung, die beliebig miteinander kombiniert werden können:

- Für den Beginn und das Ende eines Bereichs können **absolute** Textcursorpositionen, beginnend mit der Position 0 am Dokumentanfang, gesetzt und abgerufen werden. Nähere Informationen dazu finden Sie unter [FirstTextPosition](#) und [LastTextPosition](#). Diese Methode erfordert komplexe interne Berechnungen und sollte nur mit Vorsicht eingesetzt werden.
- Für den Beginn und das Ende eines Bereichs können das **XMLData**-Element und eine Textposition innerhalb dieses Elements 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 Dokumentstruktur. Sie kann zum Suchen von XMLData-Objekten und zum Durchführen von Operationen, die über die Benutzeroberfläche nicht aufgerufen werden können, verwendet werden.

Änderungen am Dokumentinhalt können durch verschiedene Methoden vorgenommen werden:

- Mit Hilfe der [Text](#)-Eigenschaft können Sie den durch das Bereichsobjekt ausgewählten Dokumenttext abrufen. Wenn die Eigenschaft gesetzt ist, wird der ausgewählte Dokumenttext durch den neuen Text ersetzt.
- Die Standardbearbeitungsfunktionen [Cut](#), [Copy](#), [Paste](#) und [Delete](#).
- Tabellenoperationen für Tabellen, die dynamisch wachsen können.
- Methoden, die den Funktionalitäten der Authentic-Eingabehilfenfenster zugeordnet sind.
- Aufrufe der [XMLData](#) Objekte des zugrunde liegenden Dokuments, um diese direkt zu bearbeiten.

AppendRow

Siehe auch

Methode: [AppendRow\(\)](#) als Boolean

Beschreibung

Wenn sich der Anfang eines Bereichs innerhalb einer dynamischen Tabelle befindet, fügt diese Methode eine neue Zeile am Ende der ausgewählten Tabelle hinzu. Die Bereichsauswahl wird geändert, so dass sie auf den Beginn der neuen Zeile zeigt. Die Funktion gibt *true* zurück, wenn die Anhängoperation erfolgreich war. Anderfalls wird *false* zurückgegeben.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```
1 -----
```

```

' XMLSpy scripting environment - VBScript
' Append row at end of current dynamically growable table
' -----
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' check if we can insert something
If objRange.IsInDynamicTable Then
    objRange.AppendRow
    ' objRange points to beginning of new row
    objRange.Select
End If

```

Application

Siehe auch

Eigenschaft: [Application](#) als [Application](#) (schreibgeschützt)

Beschreibung

Ruft das Authentic Desktop application-Objekt auf.

Fehler

- 2001 The authentic range object or its related view object is no longer valid.
Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

CanPerformAction

Siehe auch

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

Beschreibung

[CanPerformAction](#) und die damit in Zusammenhang stehenden Methoden ermöglichen den Zugriff auf die Eingabehilfenfunktionen von Authentic. Diese Funktion gestattet die mühelose und einheitliche Bearbeitung des Dokumentinhalts, ohne dass Sie wissen müssen, wo genau die Änderung durchgeführt wird. Der Beginn des Bereichsobjekts dient dazu, die nächste gültige Stelle zu finden, an der die angegebene Aktion durchgeführt werden kann. Sobald diese Stelle gefunden wurde, gibt die Methode *True* zurück, andernfalls *False*.

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

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 2007 Es wurde eine ungültige Aktion definiert.

Beispiele

Siehe [PerformAction](#).

CanPerformActionWith

Siehe auch

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

Beschreibung

`PerformActionWith` und die damit in Zusammenhang stehenden Methoden ermöglichen den Zugriff auf die Eingabehilfenfunktionen von Authentic. Diese Funktion gestattet die mühelose und einheitliche Bearbeitung des Dokumentinhalts, ohne dass Sie wissen müssen, wo genau die Änderung durchgeführt wird.

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

Tipp: Verwenden Sie zur Anwendung der Aktion [CanPerformActionWith](#).

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 2007 Es wurde eine ungültige Aktion definiert.

Beispiele

Siehe [PerformAction](#).

Clone

Siehe auch

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

Beschreibung

Gibt eine Kopie des Bereichsobjekts zurück.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

CollapsToBegin

Siehe auch

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

Beschreibung

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

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

CollapsToEnd

Siehe auch

Methode: `CollapsToEnd()` as [AuthenticRange](#)

Beschreibung

Setzt den Beginn des Bereichsobjekts auf sein Ende. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Copy

Siehe auch

Methode: `Copy()` als Boolean

Beschreibung

Gibt *False* zurück, wenn der Bereich keine Abschnitte des Dokuments enthält, die kopiert werden können.

Gibt *True* zurück, wenn Text - und im Fall vollständig ausgewählter XML-Elemente auch die Elemente - in den Kopieren/Einfügen-Puffer kopiert wurde.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Cut

Siehe auch

Methode: `Cut()` als Boolean

Beschreibung

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

Gibt *True* zurück, nachdem Text - und im Fall von vollständig ausgewählten XML-Elementen auch die Elemente aus dem Dokument gelöscht und im Kopieren/Einfügen-Puffer gespeichert wurden.-

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Delete

Siehe auch

Methode: `Delete()` als Boolean

Beschreibung

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

Gibt *True* zurück, nachdem Text und im Fall von vollständig ausgewählten XML-Elementen auch die Elemente aus dem Dokument gelöscht wurden.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

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 Bereichsauswahl wird geändert, so dass sie auf das nächste Element nach der gelöschten Zeile zeigt. Die Funktion gibt *true* zurück, wenn die Operation erfolgreich war, andernfalls ist der Rückgabewert *false*.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```
' -----  
' XMLSpy scripting environment - VBScript  
' Delete selected row from dynamically growing table  
' -----  
Dim objRange  
' we assume that the active document is open in authentic view mode  
Set objRange = Application.ActiveDocument.AuthenticView.Selection  
  
' check if we are in a table  
If objRange.IsInDynamicTable Then  
    objRange.DeleteRow  
End If
```

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 nach der ausgewählten ein. Die Bereichsauswahl wird geändert, so dass sie auf den Anfang der nächsten Zeile zeigt. Die Funktion gibt *true* zurück, wenn die Operation erfolgreich war, andernfalls ist der Rückgabewert *false*.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```
'-----
' XMLSpy scripting environment - VBScript
' duplicate row in current dynamically growable table
'-----
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' check if we can insert something
If objRange.IsInDynamicTable Then
    objRange.DuplicateRow
    ' objRange points to beginning of new row
    objRange.Select
End If
```

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 oder diese enthält. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2003 Die Erweiterung des Bereichs würde über das Ende des Dokuments hinausgehen.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

FirstTextPosition

Siehe auch

Eigenschaft: [FirstTextPosition](#) als Long

Beschreibung

Setzt oder ruft den am weitesten links gelegenen Positionsindex des Bereichsobjekts ab. Dieser Index ist immer kleiner oder gleich der [LastTextPosition](#). Die Indizierung beginnt am Dokumentanfang mit 0 und steigt dann mit jeder neuen Position, die der Textcursor einnehmen kann, schrittweise an. Die schrittweise Erhöhung der Textposition um 1 hat denselben Effekt, wie die nach rechts-Taste. Die schrittweise Verringerung der Textposition um 1 hat denselben Effekt, wie die nach links-Taste.

Wenn Sie [FirstTextPosition](#) auf einen höheren Wert als den der aktuellen [LastTextPosition](#) setzen, wird [LastTextPosition](#) auf den neuen Wert von [FirstTextPosition](#) gesetzt.

Tipp: Seien Sie beim Setzen der Textcursorposition vorsichtig, da dies im Vergleich zur Cursorpositionierung mittels `XMLData` eine kostspielige Operation ist.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 2006 Es wurde eine Textposition außerhalb des Dokuments definiert.

Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' -----

Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

nDocStartPosition = objAuthenticView.DocumentBegin.FirstTextPosition
nDocEndPosition = objAuthenticView.DocumentEnd.FirstTextPosition

' let's create a range that selects the whole document
' in an inefficient way
Dim objRange
' we need to get a (any) range object first
Set objRange = objAuthenticView.DocumentBegin
objRange.FirstTextPosition = nDocStartPosition
objRange.LastTextPosition = nDocEndPosition

' let's check if we got it right
If objRange.IsEqual(objAuthenticView.WholeDocument) Then
    MsgBox "Test using direct text cursor positioning was ok"
Else
    MsgBox "Ooops! "
End If
```

FirstXMLData

Siehe auch

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

Beschreibung

Setzt oder ruft das erste `XMLData` Element im zugrunde liegenden Dokument ab, das durch die Bereichsauswahl teilweise oder zur Gänze ausgewählt ist. Der genaue Anfang der Auswahl wird durch das [FirstXMLDataOffset](#) Attribut definiert.

Immer wenn Sie `FirstXMLData` auf ein neues Datenobjekt setzen, wird [FirstXMLDataOffset](#) auf die erste Cursorposition innerhalb dieses Elements gesetzt. Nur `XMLData`-Objekte, die eine Cursorposition haben, können verwendet werden. Wenn Sie `FirstXMLData` / [FirstXMLDataOffset](#) auf eine Position setzen, die größer ist als der aktuelle Wert von [LastXMLData](#) / [LastXMLDataOffset](#) wird weiteres auf die neue Anfangsposition verschoben.

Tipp: Sie können das zugrunde liegende XML-Dokument in Fällen, in denen die mit dem [AuthenticRange](#)-Objekt verfügbaren Methoden nicht ausreichen, mit Hilfe der Eigenschaften [FirstXMLData](#) und [LastXMLData](#) direkt aufrufen und bearbeiten

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 2008 Interner Fehler
- 2009 Das XMLData-Objekt kann nicht aufgerufen werden.

Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' show name of currently selected XMLData element
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objXMLData
Set objXMLData = objAuthenticView.Selection.FirstXMLData
' authentic view adds a 'text' child element to elements
' of the document which have content. So we have to go one
' element up.
Set objXMLData = objXMLData.Parent
MsgBox "Current selection selects element " & objXMLData.Name
```

FirstXMLDataOffset

Siehe auch

Eigenschaft: [FirstXMLDataOffset](#) as Long

Beschreibung

Setzt oder ruft den Versatz der Cursorposition innerhalb des Elements [FirstXMLData](#) für den Anfang des Bereichs ab. Die Versatzpositionen basieren auf den von der Eigenschaft [Text](#) zurückgegebenen Zeichen und beginnen mit 0. Verwenden Sie beim Definieren eines neuen Versatzes -1, um den Versatz an die letzte mögliche Position im Element zu setzen. Beachten Sie bitte besonders die folgenden Fälle:

- Die Textform von Einträgen in Auswahllisten, Kontrollkästchen und ähnlichen Steuerelementen kann anders aussehen, als angezeigt. Zwar basiert der Datenversatz auf diesem Text, doch gibt es nur zwei gültige Positionen für den Versatz, eine am Anfang und eine am Ende des Eintrags. Wenn Sie versuchen, den Versatz irgendwo mitten im Eintrag zu positionieren, hat dies zur Folge, dass der Versatz an das Ende des Eintrags gesetzt wird.
- Die Textform von XML-Entities kann sich in ihrer Länge von der Darstellung auf dem Bildschirm unterscheiden. Der Versatz basiert auf der Textform.

Wenn [FirstXMLData](#) / [FirstXMLDataOffset](#) eine Position nach dem aktuellen [LastXMLData](#) / [LastXMLDataOffset](#) wählt, wird zweiteres an die neue Anfangsposition verschoben.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht gültig.
- 2005 Es wurde ein ungültiger Versatz definiert.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' Select the complete text of an XMLData element
```

```

' using XMLData based selection and ExpandTo
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

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

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

' were we successful?
If objRange.IsEqual(objRange2) Then
    objRange.Select()
Else
    MsgBox "Oops"
End If

```

GetElementAttributeNames

Siehe auch

Methode: `GetElementAttributeNames` (*strElementName* als String,
out_arrAttributeNames als Variant)

Beschreibung

Ruft die Namen aller Attribute für das einschließende Element mit dem angegebenen Namen auf. Verwenden Sie die Element/Attribut-Paare, um den Attributwert mit den Methoden [GetElementAttributeValue](#) und [SetElementAttributeValue](#) zu setzen oder abzurufen.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2005 Invalid element name was specified.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

Siehe [SetElementAttributeValue](#).

GetElementAttributeValue

Siehe auch

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

Beschreibung

Ruft für das in *strElementName* angegebene Element den Wert des in *strAttributeName* definierten Attributs ab. Wenn das Attribut unterstützt wird, aber keinen Wert zugewiesen hat, wird der leere String zurückgegeben. Um die Namen der von einem Element unterstützten

Attribute herauszufinden, verwenden Sie [GetElementAttributeNames](#), oder [HasElementAttribute](#).

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsojekt ist nicht mehr gültig.
- 2005 Es wurde ein ungültiger Elementname angegeben.
Es wurde ein ungültiger Attributname angegeben.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

Siehe [SetElementAttributeValue](#).

GetElementHierarchy

Siehe auch

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

Beschreibung

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

Die Namen der Elementhierarchie identifizieren zusammen mit dem Bereichsobjekt `XMLData` Elemente im Dokument. Die Attribute dieser Elemente können direkt von [GetElementAttributeNames](#) und damit in Zusammenhang stehenden Methoden aufgerufen werden.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsojekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

Siehe [SetElementAttributeValue](#).

GetEntityNames

Siehe auch

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

Beschreibung

Ruft die Namen alle definierten Entities ab. Die Liste der abgerufenen Entities hängt nicht von der aktuellen Auswahl oder der aktuellen Position ab. Verwenden Sie einen dieser Namen mit der Funktion [InsertEntity](#).

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsojekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

Siehe [InsertEntity](#).

Goto

Siehe auch

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

Beschreibung

Definiert den Bereich, der auf den Anfang des *nCount* Elements vom Typ *eKind* zeigt. Die Anfangsposition wird durch den Parameter *eFrom* definiert.

Mit Hilfe positiver Werte für *nCount* können Sie zum Ende des Dokuments navigieren. Mit Hilfe negativer Werte können Sie zum Beginn des Dokuments navigieren. die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2003 Das Ziel liegt hinter dem Ende des Dokuments.
- 2004 Das Ziel liegt vor dem Anfang des Dokuments.
- 2005 Es wurde eine ungültige Elementart definiert.
Es wurde eine ungültige Anfangsposition definiert.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

GotoNext

Siehe auch

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

Beschreibung

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

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2003 Das Ziel liegt hinter dem Ende des Dokuments.
- 2005 Es wurde eine ungültige Elementart definiert.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' Scan through the whole document word-by-word
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentBegin
Dim bEndOfDocument
bEndOfDocument = False

On Error Resume Next
```

```

While Not bEndOfDocument
    objRange.GotoNext(spyAuthenticWord).Select
    If ((Err.number - vbObjectError) = 2003) Then
        bEndOfDocument = True
        Err.Clear
    ElseIf (Err.number <> 0) Then
        Err.Raise ' forward error
    End If
Wend

```

GotoNextCursorPosition

Siehe auch

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

Beschreibung

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

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2003 Das Ziel liegt hinter dem Ende des Dokuments.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

GotoPrevious

Siehe auch

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

Beschreibung

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

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2004 Das Ziel liegt vor dem Anfang des Dokuments.
- 2005 Es wurde eine ungültige Elementart definiert.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```

' -----
' XMLSpy scripting environment - VBScript
' Scan through the whole document tag-by-tag
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentEnd
Dim 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

```

GotoPreviousCursorPosition

Siehe auch

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

Beschreibung

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

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2004 Das Ziel liegt vor dem Anfang des Dokuments.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

HasElementAttribute

Siehe auch

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

Beschreibung

Überprüft, ob das einschließende Element mit dem Namen `strElementName` das in `strAttributeName` definierte Attribut unterstützt.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Es wurde ein ungültiger Elementname angegeben.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

InsertEntity

Siehe auch

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

Beschreibung

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

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Es wurde ein unbekannter Name für den Eintrag angegeben.

Beispiele

```

' -----
' XMLSpy scripting environment - VBScript
' Insert the first entity in the list of available entities
' -----
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' first we get the names of all available entities as they
' are shown in the entry helper of XMLSpy
Dim arrEntities
objRange.GetEntityNames arrEntities

' we insert the first one of the list
If UBound(arrEntities) >= 0 Then
    objRange.InsertEntity arrEntities(0)
Else
    MsgBox "Sorry, no entities are available for this document"
End If

```

InsertRow**Siehe auch**

Methode: `InsertRow()` als Boolean

Beschreibung

Wenn sich der Beginn des Bereichs innerhalb einer dynamischen Tabelle befindet, fügt diese Methode eine neue Zeile vor der aktuellen ein. Der Auswahlbereich wird geändert, sodass er auf den Beginn der neu eingefügten Zeile zeigt. Die Funktion gibt *true* zurück, wenn die Einfügeoperation erfolgreich war, andernfalls ist das Ergebnis *false*.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbereichsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```

' -----
' XMLSpy scripting environment - VBScript
' Insert row at beginning of current dynamically growing table
' -----
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' check if we can insert something
If objRange.IsInDynamicTable Then
    objRange.InsertRow
    ' objRange points to beginning of new row
    objRange.Select
End If

```

IsCopyEnabled**Siehe auch**

Eigenschaft: `IsCopyEnabled` als Boolean (schreibgeschützt)

Beschreibung

Überprüft, ob die Kopieroperation für diesen Bereich unterstützt wird.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsCutEnabled

Siehe auch

Eigenschaft: `IsCutEnabled` als Boolean (schreibgeschützt)

Beschreibung

Überprüft, ob die Ausschneideoperation für diesen Bereich unterstützt wird.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsDeleteEnabled

Siehe auch

Eigenschaft: `IsDeleteEnabled` als Boolean (schreibgeschützt)

Beschreibung

Überprüft, ob die Löschoption für diesen Bereich unterstützt wird.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsEmpty

Siehe auch

Methode: `IsEmpty()` als Boolean

Beschreibung

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

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsEqual

Siehe auch

Methode: `IsEqual` (`objCmpRange` als `AuthenticRange`) als Boolean

Beschreibung

Überprüft, ob der Beginn und das Ende beider Bereiche identisch sind.

Fehler

- 2001 Eines der beiden verglichenen Bereichsobjekte ist ungültig.
- 2005 Es wurde eine ungültige Adresse für einen Rückgabeparameter angegeben.

IsFirstRow

Siehe auch

Eigenschaft: `IsFirstRow()` als Boolean (schreibgeschützt)

Beschreibung

Überprüft, ob der Bereich in der ersten Zeile einer Tabelle ist. Welche Tabelle berücksichtigt wird, ist vom Ausmaß des Bereichs abhängig. Wenn die Auswahl über eine einzige Tabellenzeile hinausgeht, wird überprüft, ob diese Tabelle das erste Element in einer einbettenden Tabelle ist. Nähere Informationen dazu finden Sie im Benutzerhandbuch unter Eingabehilfen.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsojekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsInDynamicTable

Siehe auch

Methode: `IsInDynamicTable()` als Boolean

Beschreibung

Überprüft, ob sich der gesamte Bereich innerhalb einer Tabelle befindet, die die verschiedenen Zeilenoperationen wie Einfügen, Anhängen, Kopieren usw. unterstützt.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsojekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsLastRow

Siehe auch

Eigenschaft: `IsLastRow()` als Boolean (schreibgeschützt)

Beschreibung

Überprüft, ob der Bereich in der letzten Zeile einer Tabelle ist. Welche Tabelle berücksichtigt wird, hängt vom Ausmaß des Bereichs ab. Wenn die Auswahl über eine einzige Tabellenzeile hinausgeht, wird überprüft, ob diese Tabelle das letzte Element in einer einbettenden Tabelle ist. Nähere Informationen dazu finden Sie im Benutzerhandbuch unter Eingabehilfen.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsPasteEnabled**Siehe auch**

Eigenschaft: `IsPasteEnabled` als Boolean (schreibgeschützt)

Beschreibung

Überprüft, ob die Einfügeoperationen für diesen Bereich unterstützt wird.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsTextStateApplied**Siehe auch**

Methode: `IsTextStateApplied` (`i_strElementName` als String) als Boolean

Beschreibung

Überprüft, ob der gesamte ausgewählte Text in ein XML-Element mit dem Namen `i_strElementName` eingebettet ist. Ein typisches Beispiel für den Parameter `i_strElementName` ist "strong", "bold" oder "italic".

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

LastTextPosition**Siehe auch**

Eigenschaft: `LastTextPosition` als Long

Beschreibung

Setzt oder ruft den am weitesten rechts gelegenen Positionsindex des Bereichsobjekts ab. Dieser Index ist immer größer oder gleich `FirstTextPosition`. Die Indizierung beginnt am Dokumentanfang mit 0 und steigt dann mit jeder neuen Position, die der Textcursor einnehmen kann, schrittweise an. Die schrittweise Erhöhung der Textposition um 1 hat denselben Effekt, wie die nach rechts-Taste. Die schrittweise Verringerung der Textposition um 1 hat denselben Effekt, wie die nach links-Taste.

Wenn Sie `LastTextPosition` auf einen niedrigeren Wert als den der aktuellen `FirstTextPosition` setzen, wird `FirstTextPosition` auf den neuen Wert von `LastTextPosition` gesetzt.

Tipp: Seien Sie beim Setzen der Textcursorposition vorsichtig, da dies im Vergleich zur Cursorpositionierung mittels `XMLData` eine kostspielige Operation ist.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 2006 Es wurde eine Textposition außerhalb des Dokuments definiert.

Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' -----

Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

nDocStartPosition = objAuthenticView.DocumentBegin.FirstTextPosition
nDocEndPosition = objAuthenticView.DocumentEnd.FirstTextPosition

' let's create a range that selects the whole document
' in an inefficient way
Dim objRange
' we need to get a (any) range object first
Set objRange = objAuthenticView.DocumentBegin
objRange.FirstTextPosition = nDocStartPosition
objRange.LastTextPosition = nDocEndPosition

' let's check if we got it right
If objRange.IsEqual(objAuthenticView.WholeDocument) Then
    MsgBox "Test using direct text cursor positioning was ok"
Else
    MsgBox "Oops! "
End If
```

LastXMLData

Siehe auch

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

Beschreibung

Setzt oder ruft das letzte `XMLData` Element im zugrunde liegenden Dokument ab, das durch die Bereichsauswahl teilweise oder zur Gänze ausgewählt ist. Das genaue Ende der Auswahl wird durch das [LastXMLDataOffset](#) Attribut definiert.

Immer wenn Sie `LastXMLData` auf ein neues Datenobjekt setzen, wird [LastXMLDataOffset](#) auf die letzte Cursorposition innerhalb dieses Elements gesetzt. Nur `XMLData`-Objekte, die eine Cursorposition haben, können verwendet werden. Wenn Sie `LastXMLData` / [LastXMLDataOffset](#) auf eine Position setzen, die niedriger ist als der aktuelle Wert von [FirstXMLData](#) / [FirstXMLDataOffset](#) wird zweites auf die neue Endposition verschoben.

Tipp: Sie können das zugrunde liegende XML-Dokument in Fällen, in denen die mit dem

[AuthenticRange](#)-Objekt verfügbaren Methoden nicht ausreichen, mit Hilfe der Eigenschaften [FirstXMLData](#) und [LastXMLData](#) direkt aufrufen und bearbeiten

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 2008 Interner Fehler
- 2009 Das `XMLData` Objekt kann nicht aufgerufen werden.

LastXMLDataOffset

Siehe auch

Eigenschaft: [LastXMLDataOffset](#) als Long

Beschreibung

Setzt oder ruft den Versatz der Cursorposition innerhalb des Elements [LastXMLData](#) für das Ende des Bereichs ab.

Die Versatzpositionen basieren auf den von der Eigenschaft [Text](#) zurückgegebenen Zeichen und beginnen mit 0. Verwenden Sie beim Definieren eines neuen Versatzes -1, um den Versatz an die letzte mögliche Position im Element zu setzen. Beachten Sie bitte besonders die folgenden Fälle:

- Die Textform von Einträgen in Auswahllisten, Kontrollkästchen und ähnlichen Steuerelementen kann anders aussehen, als angezeigt. Zwar basiert der Datenversatz auf diesem Text, doch gibt es nur zwei gültige Positionen für den Versatz, eine am Anfang und eine am Ende des Eintrags. Wenn Sie versuchen, den Versatz irgendwo mitten im Eintrag zu positionieren, hat dies zur Folge, dass der Versatz an das Ende des Eintrags gesetzt wird.
- Die Textform von XML-Entities kann sich in ihrer Länge von der Darstellung auf dem Bildschirm unterscheiden. Der Versatz basiert auf der Textform.

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

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht gültig.
- 2005 Es wurde ein ungültiger Versatz definiert.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' Select the complete text of an XMLData element
' using XMLData based selection and ExpandTo
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

' first we use the XMLData based range properties
' to select all text of the first XMLData element
' in the current selection
Dim objRange
```

```

Set objRange = objAuthenticView.Selection
objRange.FirstXMLDataOffset = 0 ' start at beginning of element text
objRange.LastXMLData = objRange.FirstXMLData ' select only one element
objRange.LastXMLDataOffset = -1 ' select till its end

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

' were we successful?
If objRange.IsEqual(objRange2) Then
    objRange.Select()
Else
    MsgBox "Ooops"
End If

```

MoveBegin

Siehe auch

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

Beschreibung

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

Verwenden Sie positive Zahlen für *nCount*, um die Position zum Ende des Dokuments hin zu verschieben und negative Zahlen um die Position zum Anfang des Dokuments hin zu verschieben. Das Ende des Bereichs bleibt gleich, es sei denn, der neue Anfang hat einen höheren Wert als das Ende. In diesem Fall wird das Ende an den neuen Anfang verschoben. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtobjekt ist nicht mehr gültig.
- 2003 Das Ziel liegt hinter dem Ende des Dokuments.
- 2004 Das Ziel liegt vor dem Anfang des Dokuments.
- 2005 Es wurde eine ungültige Elementart definiert.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

MoveEnd

Siehe auch

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

Beschreibung

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

Verwenden Sie positive Zahlen für *nCount*, um die Position zum Ende des Dokuments hin zu verschieben und negative Zahlen um die Position zum Anfang des Dokuments hin zu verschieben. Der Anfang des Bereichs bleibt gleich, es sei denn, das neue Ende hat einen niedrigeren Wert als der Anfang. In diesem Fall wird der Anfang an das neue Ende verschoben. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2003 Das Ziel liegt hinter dem Ende des Dokuments.
- 2004 Das Ziel liegt vor dem Anfang des Dokuments.
- 2005 Es wurde eine ungültige Elementart definiert.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

MoveRowDown**Siehe auch**

Methode: [MoveRowDown\(\)](#) als Boolean

Beschreibung

Wenn sich der Anfang des Bereichs innerhalb einer dynamischen Tabelle befindet und eine Zeile auswählt, die nicht die letzte Zeile in dieser Tabelle ist, tauscht diese Methode diese Zeile gegen die unmittelbar darunter gelegene Zeile aus. Die Auswahl des Bereichs verschiebt sich mit der Zeile, ändert sich aber ansonsten nicht. Die Funktion gibt *true* zurück, wenn der Verschiebevorgang erfolgreich war, andernfalls wird *false* zurückgegeben.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

MoveRowUp**Siehe auch**

Method: [MoveRowUp\(\)](#) as Boolean

Beschreibung

Wenn sich der Anfang des Bereichs innerhalb einer dynamischen Tabelle befindet und eine Zeile auswählt, die nicht die erste Zeile in dieser Tabelle ist, tauscht diese Methode diese Zeile gegen die unmittelbar darüber gelegene Zeile aus. Die Auswahl des Bereichs verschiebt sich mit der Zeile, ändert sich aber ansonsten nicht. Die Funktion gibt *true* zurück, wenn der Verschiebevorgang erfolgreich war, andernfalls wird *false* zurückgegeben

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

Siehe [JScript - Bubble-Sortierung dynamischer Tabellen](#).

Parent**Siehe auch**

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

Beschreibung

Ruft die Ansicht auf, die dieses Bereichsobjekt enthält.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsojekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Paste**Siehe auch**

Methode: `Paste()` als Boolean

Beschreibung

Gibt *False* zurück, wenn der Kopieren/Einfügen-Puffer leer ist oder sein Inhalt die aktuelle Auswahl nicht ersetzen kann.

Löscht andernfalls die aktuelle Auswahl, fügt den Inhalt des Kopieren/Einfügen-Puffers ein und gibt *True* zurück.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsojekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

PerformAction**Siehe auch**

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

Beschreibung

`PerformAction` und die dazugehörigen Methoden verschaffen Ihnen Zugriff auf die Eingabehilfenfunktionen von Authentic. Diese Funktion gestattet die einfach und einheitliche Bearbeitung des Dokumentinhalts, ohne dass Sie wissen müssen, wo genau die Änderung stattfinden soll. Der Anfang des Bereichsobjekts dient dazu, die nächste gültige Stelle zu finden, an der die angegebene Aktion durchgeführt werden kann. Wenn keine derartige Stelle gefunden wird, gibt die Methode *False* zurück. Andernfalls wird das Dokument geändert und der Bereich zeigt auf den Anfang der Änderung.

Tipp: Um Elementnamen zu finden, die als der zweite Parameter übergeben werden können, verwenden Sie `CanPerformActionWith`.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsojekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 2007 Es wurde eine ungültige Aktion definiert.

Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' Insert the innermost element
' -----
Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection
```

```

' we determine the elements that can be inserted at the current position
Dim arrElements()
objRange.CanPerformActionWith spyAuthenticInsertBefore, arrElements

' we insert the first (innermost) element
If UBound(arrElements) >= 0 Then
    objRange.PerformAction spyAuthenticInsertBefore, arrElements(0)
    ' objRange now points to the beginning of the inserted element
    ' we set a default value and position at its end
    objRange.Text = "Hello"
    objRange.ExpandTo(spyAuthenticTag).CollapsToEnd().Select
Else
    MsgBox "Can't insert any elements at current position"
End If

```

Select

Siehe auch

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

Beschreibung

Macht diesen Bereich zur aktuellen Auswahl der Benutzeroberfläche. Dasselbe Ergebnis können Sie mit: '[objRange.Parent.Selection](#) = [objRange](#)' erzielen.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.

Beispiele

```

' -----
' XMLSpy scripting environment - VBScript
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

' set current selection to end of document
objAuthenticView.DocumentEnd.Select()

```

SelectNext

Siehe auch

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

Beschreibung

Wählt das Element vom Typ [eKind](#) nach dem aktuellen Ende des Bereichs aus. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
- 2003 Das Ziel liegt hinter dem Ende des Dokuments.
- 2005 Es wurde eine ungültige Elementart definiert.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele


```

' -----
' XMLSpy scripting environment - VBScript
' Scan through the whole document word-by-word
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentBegin
Dim bEndOfDocument
bEndOfDocument = False

On Error Resume Next
While Not bEndOfDocument
    objRange.SelectNext(spyAuthenticWord).Select
    If ((Err.number - vbObjectError) = 2003) Then
        bEndOfDocument = True
        Err.Clear
    ElseIf (Err.number <> 0) Then
        Err.Raise ' forward error
    End If
Wend

```

SelectPrevious

Siehe auch

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

Beschreibung

Wählt das Element vom Typ *eKind* vor dem aktuellen Anfang des Bereichs aus. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsbjekt ist nicht mehr gültig.
- 2004 Das Ziel liegt vor dem Anfang des Dokuments
- 2005 Es wurde eine ungültige Elementart definiert.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```

' -----
' XMLSpy scripting environment - VBScript
' Scan through the whole document tag-by-tag
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentEnd
Dim bEndOfDocument
bBeginOfDocument = False

On Error Resume Next
While Not bBeginOfDocument
    objRange.SelectPrevious(spyAuthenticTag).Select
    If ((Err.number - vbObjectError) = 2004) Then
        bBeginOfDocument = True
        Err.Clear
    End If
Wend

```

```

    ElseIf (Err.number <> 0) Then
        Err.Raise ' forward error
    End If
Wend

```

SetElementAttributeValue

Siehe auch

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

Beschreibung

Ruft den Wert des in *strAttributeName* definierten Attributs für das mit *strElementName* identifizierte Element auf. Wenn das Attribut unterstützt wird, aber keinen Wert zugewiesen hat, wird der leere String zurückgegeben. Um die Namen der von einem Element unterstützten Attribute zu ermitteln, verwenden Sie [GetElementAttributeNames](#) oder [HasElementAttribute](#).

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtobjekt ist nicht mehr gültig.
- 2005 Es wurde ein ungültiger Elementname angegeben.
Es wurde ein ungültiger Attributname definiert.
Es wurde ein ungültiger Attributwert definiert.

Beispiele

```

' -----
' XMLSpy scripting environment - VBScript
' Get and set element attributes
' -----

Dim objRange
' we assume that the active document is open in authentic view mode
Set objRange = Application.ActiveDocument.AuthenticView.Selection

' first we find out all the elements below the beginning of the range
Dim arrElements
objRange.GetElementHierarchy arrElements

If IsArray(arrElements) Then
    If UBound(arrElements) >= 0 Then
        ' we use the top level element and find out its valid attributes
        Dim arrAttrs()
        objRange.GetElementAttributeNames arrElements(0), arrAttrs

        If UBound(arrAttrs) >= 0 Then
            ' we retrieve the current value of the first valid
            attribute

            Dim strAttrVal
            strAttrVal = objRange.GetElementAttributeValue
            (arrElements(0), arrAttrs(0))
            MsgBox "current value of " & arrElements(0) & "/" &
            arrAttrs(0) & " is: " & strAttrVal

            ' we change this value and read it again
            strAttrVal = "Hello"
            objRange.SetElementAttributeValue arrElements(0),
            arrAttrs(0), strAttrVal
            strAttrVal = objRange.GetElementAttributeValue
            (arrElements(0), arrAttrs(0))
            MsgBox "new value of " & arrElements(0) & "/" &

```

```
arrAttrs(0) & " is: " & strAttrVal
    End If
End If
End If
```

SetFromRange

Siehe auch

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

Beschreibung

Setzt den Bereich des Objekts auf dieselbe Anfangs- und Endposition wie *objSrcRange* .

Fehler

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

Text

Siehe auch

Eigenschaft: `Text` als String

Beschreibung

Setzt oder ruft den durch ein Bereichsobjekt ausgewählten Textinhalt ab.

Die Anzahl der abgerufenen Zeichen muss nicht unbedingt identisch sein, da es Textcursorpositionen zwischen dem Beginn und dem Ende des ausgewählten Bereichs gibt. Die meisten Dokumentelemente unterstützen eine Cursor-Endposition, die sich von der Cursor-Beginnposition des folgenden Elements unterscheidet. Dropdown-Listen haben nur eine Cursorposition, können jedoch Strings jeder Länge auswählen. Im Fall von Optionsfeldern und Kontrollkästchen enthält die Text-eigenschaft den String des entsprechenden XML-Elements.

Wenn mit dem Bereich mehr als ein Element ausgewählt wird, ist der Text die Verkettung der Einzeltexte. XML Entities werden erweitert, sodass '&' als '&' erwartet wird.

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

Fehler

- 2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtobjekt ist nicht mehr gültig.
- 2005 Es wurde eine ungültige Adresse für einen Rückgabeparameter angegeben.

4.3.4 AuthenticView

Siehe auch

Eigenschaften

[Application](#)
[AsXMLString](#)
[DocumentBegin](#)
[DocumentEnd](#)
[Event](#)
[MakeVisible](#)
[Parent](#)
[Selection](#)
[XMLDataRoot](#)
[WholeDocument](#)

Methoden

[Goto](#)
[IsRedoEnabled](#)
[IsUndoEnabled](#)
[Print](#)
[Redo](#)
[Undo](#)
[UpdateXMLInstanceEntities](#)

Events

[OnBeforeCopy](#)
[OnBeforeCut](#)
[OnBeforeDelete](#)
[OnBeforeDrop](#)
[OnBeforePaste](#)
[OnDragOver](#)
[OnKeyboardEvent](#)
[OnMouseEvent](#)
[OnSelectionChanged](#)

Beschreibung

Über AuthenticView und seine untergeordneten Objekte [AuthenticRange](#) und [AuthenticDataTransfer](#) steht Ihnen eine Schnittstelle für die Authentic-Ansicht zur Verfügung, über die Sie Dokumentinhalt einfach und konsistent ändern können. Diese Schnittstellen treten nun anstelle der folgenden nun als **nicht mehr verwendet** markierten Schnittstellen:

[OldAuthenticView](#) (der alte Name war DocEditView)
[AuthenticSelection](#) (der alte Name war DocEditSelection, wird nun ersetzt durch [AuthenticRange](#))
[AuthenticEvent](#) (der alte Name war DocEditEvent) [Interfaces](#)

Mit AuthenticView erhalten Sie einfachen Zugriff auf spezielle Funktionen wie Drucken, den mehrstufigen Rückgängig-Puffer und die aktuelle Cursorauswahl oder -position.

AuthenticView verwendet zur einfachen Navigation im Dokument und zur flexiblen Auswahl logischer Textelemente Objekte vom Typ [AuthenticRange](#). Verwenden Sie für einfache Auswahldefinitionen die Eigenschaften [DocumentBegin](#), [DocumentEnd](#) oder [WholeDocument](#) und die [Goto](#) Methode für komplexere Auswahldefinitionen. Informationen zum Navigieren relativ zu einem bestimmten Dokumentbereich finden Sie unter den Methoden und Eigenschaften des [AuthenticRange](#) Objekts.

Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' secure access to authentic view object
' -----
Dim objDocument
Set objDocument = Application.ActiveDocument
If (Not objDocument Is Nothing) Then
    ' we have an active document, now check for view mode
    If (objDocument.CurrentViewMode <> spyViewAuthentic) Then
        If (Not objDocument.SwitchViewMode (spyViewAuthentic)) Then
            MsgBox "Active document does not support authentic view
mode"
        Else
            ' now it is safe to access the authentic view object
            Dim objAuthenticView
            Set objAuthenticView = objDocument.AuthenticView
            ' now use the authentic view object

        End If
    End If
End If
```

```
Else
    MsgBox "No document is open"
End If
```

Events

OnBeforeCopy

Siehe auch

Event: `OnBeforeCopy()` als Boolean

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_DocEditCopy()
    'On_DocEditCopy=False 'to disable operation
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_DocEditCopy()
{
    //return false; //to disable operation
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugin.OnEvent (21, ...) //nEventId=21
```

Beschreibung

Dieses Event wird ausgelöst, bevor eine Kopieroperation an einem Dokument durchgeführt wird. Bei Rückgabe von *True* (oder nichts) wird die Kopieroperation gestattet, bei Rückgabe von *False* wird das Kopieren deaktiviert.

OnBeforeCut

Siehe auch

Event: `OnBeforeCut()` als Boolean

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_DocEditCut()
    'On_DocEditCut=False 'to disable operation
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_DocEditCut()
{
    //return false; //to disable operation
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugin.OnEvent (20, ...) //nEventId=20
```

Beschreibung

Dieses Event wird ausgelöst, bevor eine Ausschneideoperation an einem Dokument durchgeführt wird. Geben Sie *True* (oder nichts) zurück, um den Vorgang zu gestatten und

False um den Vorgang zu deaktivieren.

OnBeforeDelete

Siehe auch

Event: `OnBeforeDelete()` als Boolean

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_DocEditClear()
    'On_DocEditClear=False 'to disable operation
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_DocEditClear()
{
    //to unfile/*to disable operation*/
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugin.OnEvent (22, ...) //nEventId=22
```

Beschreibung

Dieses Event wird ausgelöst, bevor eine Löschoperation an einem Dokument durchgeführt wird. Geben Sie *True* (oder nichts) zurück, um den Vorgang zu gestatten und *False* um den Vorgang zu deaktivieren.

OnBeforeDrop

Siehe auch

Event: `OnBeforeDrop ( i_nXPos als Long, i_nYPos als Long, i_ipRange als AuthenticRange, i_ipData als cancel Boolean`

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_DocEditDrop(nXPos, nYPos, objRange, objData)
    'On_DocEditDrop=False 'to disable operation
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_DocEditDrop(nXPos, nYPos, objRange, objData)
{
    //to unfile/*to disable operation*/
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugin.OnEvent (11, ...) //nEventId=11
```

Beschreibung

Dieses Event wird ausgelöst, nachdem ein Objekt mit der Maus in das Applikationsfenster gezogen wurde. Alle mit dem Event in Zusammenhang stehenden Informationen werden als Parameter übergeben.

Die ersten beiden Parameter definieren die Mausposition zum Zeitpunkt des Auftretens des Event. Der Parameter *objRange* übergibt ein Bereichsobjekt, das das XML-Element unterhalb

der Mausposition auswählt. Der Wert dieses Parameters kann *NULL* sein. Überprüfen Sie dies, bevor Sie das Bereichsobjekt aufrufen. Mit Hilfe des Parameters *objData* können Sie Informationen über das gezogene Objekt aufrufen.

Geben Sie *False* zurück, um den Drop-Vorgang abubrechen. Geben Sie *True* (oder nichts) zurück, um mit einem normalen Vorgang fortzusetzen.

Beispiele

```
'
'-----
' VB code snippet - connecting to object level events
'-----
' access XMLSpy (without checking for any errors)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' this is the event callback routine connected to the OnBeforeDrop
' event of object objView
Private Function objView_OnBeforeDrop( ByVal i_nXPos As Long, ByVal i_nYPos
As Long,
                                     ByVal i_ipRange As IAuthenticRange,
                                     ByVal i_ipData As
IAuthenticDataTransfer) As Boolean

    If (Not i_ipRange Is Nothing) Then
        MsgBox ("Dropping on content is prohibited");
        Return False;
    Else
        Return True;
    End If
End Function

' use VBA keyword WithEvents to connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continue here with something useful ...
' and serve the windows message loop
```

OnBeforePaste

Siehe auch

Event: *OnBeforePaste* (*objData* als Variant, *strType* als String) als Boolean

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_DocEditPaste(objData, strType)
    'On_DocEditPaste=False 'to disable operation
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_DocEditPaste(objData, strType)
{
    //On_DocEditPaste=False 'to disable operation
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (19, ...) //nEventId=19
```

Beschreibung

Dieses Event wird ausgelöst, bevor eine Einfügeoperation am Dokument ausgeführt wird. Der Parameter *strType* ist entweder "TEXT", "UNICODETEXT" oder "IUNKNOWN". In den ersten beiden Fällen enthält *objData* eine String-Darstellung des Objekts, das eingefügt wird. Im letzteren Fall enthält *objData* einen Pointer auf eine IUnknown COM-Schnittstelle.

Geben Sie *True* (oder nichts) zurück, um die Einfügeoperation zu gestatten. Geben Sie *False* zurück, um die Operation zu deaktivieren.

OnDragOver**Siehe auch**

Event: *OnDragOver* (*nXPos* als Long, *nYPos* als Long, *eMouseEvent* als [SPYMouseEvent](#), *objRange* als [AuthenticRange](#), *objData* als [AuthenticDataTransfer](#)) als Boolean

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_DocEditDragOver(nXPos, nYPos, eMouseEvent, objRange, objData)
    'On_DocEditDragOver=False 'to disable operation
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_DocEditDragOver(nXPos, nYPos, eMouseEvent, objRange, objData)
{
    //return false; //to disable operation
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugin.OnEvent (10, ...) //nEventId=10
```

Beschreibung

Dieses Event wird ausgelöst, wenn ein Objekt, von innerhalb oder außerhalb der Authentic-Ansicht mit der Maus über das Applikationsfenster gezogen wird. Alle mit dem Event in Zusammenhang stehenden Informationen werden als Parameter übergeben.

Die ersten drei Parameter definieren die Mausposition, den Status der Maustaste und den Status virtueller Schaltflächen zum Zeitpunkt, zu dem das Event auftritt. Der Parameter *objRange* übergibt ein Bereichsobjekt, das das XML-Element unterhalb der Mausposition auswählt. Der Wert dieses Parameters kann *NULL* sein. Überprüfen Sie dies, bevor Sie das Bereichsobjekt aufrufen. Mit Hilfe des Parameters *objData* können Sie Informationen über das gezogene Objekt aufrufen.

Geben Sie *False* zurück, um den Drag-Vorgang abzuberechnen. Geben Sie *True* (oder nichts) zurück, um mit einem normalen Vorgang fortzusetzen.

Beispiele

```
'
'-----
' VB code snippet - connecting to object level events
'
'-----
' access XMLSpy (without checking for any errors)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' this is the event callback routine connected to the OnDragOver
' event of object objView
Private Function objView_OnDragOver(ByVal i_nXPos As Long, ByVal i_nYPos As
```



```

Long,
                                ByVal i_eMouseEvent As SPYMouseEvent,
                                ByVal i_ipRange As IAuthenticRange,
                                ByVal i_ipData As
IAuthenticDataTransfer) As Boolean

    If (((i_eMouseEvent And spyShiftKeyDownMask) <> 0) And
        (Not i_ipRange Is Nothing)) Then
        MsgBox ("Floating over element " &
i_ipRange.FirstXMLData.Parent.Name);
    End If

    Return True;
End Function

' use VBA keyword WithEvents to connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continue here with something useful ...
' and serve the windows message loop

```

OnKeyboardEvent

Siehe auch

Event: *OnKeyboardEvent* (*eKeyEvent* als [SPYKeyEvent](#), *nKeyCode* als Long, *nVirtualKeyStatus* als Long) als Boolean

XMLSpy Skripting-Umgebung - VBScript:

```

Function On_DocEditKeyboardEvent(eKeyEvent, nKeyCode, nVirtualKeyStatus
)
    'On_DocEditKeyboardEvent=True 'to cancel bubbling of event
EndFunction

```

XMLSpy Skripting-Umgebung - JScript:

```

function On_DocEditKeyboardEvent(eKeyEvent, nKeyCode, nVirtualKeyStatus)
{
    //true to cancel bubbling of event
}

```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (30, ...) //nEventId=30
```

Beschreibung

Dieses Event wird für *WM_KEYDOWN*, *WM_KEYUP* und *WM_CHAR* Windows-Meldungen ausgelöst.

Der eigentliche Meldungstyp steht im *eKeyEvent* Parameter zur Verfügung. Der Status virtueller Tasten ist im Parameter *nVirtualKeyStatus* kombiniert. Verwenden Sie die im Enumerations-Datentyp [SPYVirtualKeyMask](#) definierten Bit-Masken, um eine Überprüfung der unterschiedlichen Tasten und ihrer Kombinationen durchzuführen.

Anmerkung: Die folgenden Events aus der Skripting-Umgebung und dem IDE-Plugin von XMLSpy werden weiterhin unterstützt, werden aber mit diesem Event nicht mehr verwendet:

```

On_DocEditKeyUp( )                IXMLSpyPlugIn.OnEvent (13, ...)    //nEventId=
13
On_DocEditKeyDown( )              IXMLSpyPlugIn.OnEvent (12, ...)    //nEventId=
12

```

```
On_DocEditKeyPressed( )      IXMLSpyPlugIn. OnEvent (14, ...) //nEventId=
14
```

Beispiele

```
'
'-----
' VB code snippet - connecting to object level events
'-----
' access XMLSpy (without checking for any errors)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' this is the event callback routine connected to the OnKeyboard
' event of object objView
Private Function objView_OnKeyboardEvent(ByVal i_keyEvent As Long, ByVal
io_pnKeyCode As Long, ByVal i_nVirtualKeyStatus As Long) As Boolean
    If ((i_keyEvent = XMLSpyLib.spyKeyUp) And ((i_nVirtualKeyStatus And
XMLSpyLib.spyCtrlKeyMask) <> 0)) Then
        MsgBox ("Ctrl " & io_pnKeyCode & " pressed")
        objView_OnKeyboardEvent = True
    Else
        objView_OnKeyboardEvent = False
    End If
End Function

' use VBA keyword WithEvents to connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continue here with something useful ...
' and serve the windows message loop
```

OnMouseEvent

Siehe auch

Event: `OnMouseEvent` (*nXPos* als Long, *nYPos* als Long, *eMouseEvent* als `SPYMouseEvent`, *objRange* als `AuthenticRange`) als Boolean

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_DocEditMouseEvent(nXPos, nYPos, eMouseEvent, objRange)
    'On_DocEditMouseEvent=True 'to cancel bubbling of event
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_DocEditMouseEvent(nXPos, nYPos, eMouseEvent, objRange)
{
    //true, to cancel bubbling of event
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn. OnEvent (31, ...) //nEventId=31
```

Beschreibung

Dieses Event wird für jede Mausbewegung und jede Mausschaltflächen-Windows-Meldung ausgelöst.

Der eigentliche Meldungstyp und der Status der Maustaste stehen im *eMouseEvent* Parameter

zur Verfügung. Verwenden Sie die im Enumerations-Datentyp [SPYMouseEvent](#) definierten Bit-Masken, um eine Überprüfung auf die unterschiedlichen Meldungen, Tastenstatus und ihre Kombinationen durchzuführen.

Der Parameter *objRange* definiert den Teil des Dokuments, der sich an der aktuellen Mauscursorposition befindet. Das Bereichsobjekt wählt immer einen vollständigen Tag des Dokuments aus. (Dies kann sich in zukünftigen Versionen ändern, wenn ein genauerer Positionierungsmechanismus zur Verfügung steht). Wenn sich an der aktuellen Position kein auswählbarer Teil des Dokuments befindet, ist das Bereichsobjekt *Null*.

Anmerkung: Die folgenden Events aus der Skripting-Umgebung und dem IDE-Plugin von XMLSpy werden weiterhin unterstützt, werden aber mit diesem Event nicht mehr verwendet:

<code>On_DocEditMouseMove()</code>	<code>IXMLSpyPlugIn.OnEvent (15, ...)</code>	<code>/</code>
<code>nEventId=15</code>		
<code>On_DocEditButtonUp()</code>	<code>IXMLSpyPlugIn.OnEvent (16, ...)</code>	<code>/</code>
<code>nEventId=16</code>		
<code>On_DocEditButtonDown()</code>	<code>IXMLSpyPlugIn.OnEvent (17, ...)</code>	<code>/</code>
<code>nEventId=17</code>		
<code>On_DocEditButtonDoubleClick()</code>	<code>IXMLSpyPlugIn.OnEvent (24, ...)</code>	<code>/</code>
<code>nEventId=24</code>		

Beispiele

```
'
'-----
' VB code snippet - connecting to object level events
'-----
' access XMLSpy (without checking for any errors)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' this is the event callback routine connected to the OnMouseEvent
' event of object objView. If you click with the left mouse button
' while pressing a control key, the current selection will be set
' to the tag below the current mouse cursor position
Private Function objView_OnMouseEvent(ByVal i_nXPos As Long, ByVal i_nYPos
As Long, ByVal i_eMouseEvent As XMLSpyLib.SPYMouseEvent, ByVal i_pRange As
XMLSpyLib.IAuthenticRange) As Boolean
    If (i_eMouseEvent = (XMLSpyLib.spyLeftButtonDownMask Or
XMLSpyLib.spyCtrlKeyDownMask)) Then
        On Error Resume Next
        i_pRange.Select
        objView_OnMouseEvent = True
    Else
        objView_OnMouseEvent = False
    End If
End Function

' use VBA keyword WithEvents to connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continue here with something useful ...
' and serve the windows message loop
```

OnSelectionChanged

Siehe auch

Event: `OnSelectionChanged` (*objNewSelection* als [AuthenticRange](#))

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_DocEditSelectionChanged(objNewSelection)
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_DocEditSelectionChanged(objNewSelection)
{
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (23, ...) //nEventId=23
```

Beschreibung

Dieses Event wird ausgelöst, wenn die sich die Auswahl auf der Benutzeroberfläche ändert.

Beispiele

```
'
'-----
' VB code snippet - connecting to object level events
'-----
' access XMLSpy (without checking for any errors)
Dim objSpy As XMLSpyLib.Application
Set objSpy = GetObject("", "XMLSpy.Application")

' this is the event callback routine connected to the OnSelectionChanged
' event of object objView
Private Sub objView_OnSelectionChanged (ByVal i_ipNewRange As
XMLSpyLib.IAuthenticRange)
    MsgBox ("new selection: " & i_ipNewRange.Text)
End Sub

' use VBA keyword WithEvents to connect to object-level event
Dim WithEvents objView As XMLSpyLib.AuthenticView
Set objView = objSpy.ActiveDocument.AuthenticView

' continue here with something useful ...
' and serve the windows message loop
```

Application**Siehe auch**

Eigenschaft: [Application](#) als [Application](#) (schreibgeschützt)

Beschreibung

Ruft das Authentic Desktop application-Objekt auf.

Fehler

- 2000 The authentic view object is no longer valid.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

AsXMLString**Siehe auch**

Eigenschaft: [AsXMLString](#) als String

Beschreibung

Gibt den Dokumentinhalt als XML-String zurück oder definiert ihn als solchen. Wenn der Inhalt auf einen neuen Wert gesetzt wird, wird die gerade verwendete Schema- oder sps-Datei nicht geändert. Wenn der neue `XMLString` der aktuellen Schemadatei nicht entspricht, wird Fehler 2011 zurückgegeben.

Fehler

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2011 `AsXMLString` wurde auf einen Wert gesetzt, der gemäß der aktuellen Schemadatei kein gültiger XML-Code ist.

DocumentBegin

Siehe auch

Eigenschaft: `DocumentBegin` als [AuthenticRange](#) (schreibgeschützt)

Beschreibung

Ruft ein Bereichsobjekt auf, das auf den Anfang des Dokuments zeigt.

Fehler

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

DocumentEnd

Siehe auch

Eigenschaft: `DocumentEnd` als [AuthenticRange](#) (schreibgeschützt)

Beschreibung

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

Fehler

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Event

Siehe auch

Eigenschaft: `Event` als [AuthenticEvent](#) (schreibgeschützt)

Beschreibung

This property gives access to parameters of the last event in the same way as [ObjAuthenticView.event](#) does. Since all events for the scripting environment and external clients are now available with parameters this `Event` property should only be used from within IDE-Plugins.

Fehler

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

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 in Richtung des Dokumentendes zu navigieren. Verwenden Sie negative Wert um zum Anfang des Dokuments hin zu navigieren.

Fehler

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2003 Das Ziel liegt hinter dem Ende des Dokuments.
- 2004 Das Ziel liegt vor dem Anfang des Dokuments.
- 2005 Es wurde eine ungültige Elementart definiert.
Die Ausgangsposition des Dokuments ist keine *spyAuthenticDocumentBegin* oder *spyAuthenticDocumentEnd* Position.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' -----

Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

On Error Resume Next
Dim objRange
' goto beginning of first table in document
Set objRange = objAuthenticView.Goto (spyAuthenticTable, 1,
spyAuthenticDocumentBegin)
If (Err.number = 0) Then
    objRange.Select()
Else
    MsgBox "No table found in document"
End If
```

IsRedoEnabled

Siehe auch

Eigenschaft: `IsRedoEnabled` als Boolean (schreibgeschützt)

Beschreibung

True, wenn Wiederhol-Schritte verfügbar sind und [Redo](#) möglich ist.

Fehler

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsUndoEnabled

Siehe auch

Eigenschaft: `IsUndoEnabled` als Boolean (schreibgeschützt)

Beschreibung

True, wenn Rückgängig-Schritte verfügbar sind und [Undo](#) möglich ist.

Fehler

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

MarkupVisibility

Siehe auch

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

Beschreibung

Setzt oder ruft die aktuelle Sichtbarkeit des Markup-Codes ab.

Fehler

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2005 Es wurde ein ungültiger Enumerationswert angegeben.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Parent

Siehe auch

Eigenschaft: `Parent` als [Document](#) (schreibgeschützt)

Beschreibung

Ruft das in dieser Ansicht angezeigte Dokument auf.

Fehler

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

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 "Druckvorschau" angezeigt. Wenn `bPromptUser` auf `True` gesetzt ist, wird das Dialogfeld "Drucken" angezeigt. Wenn beide Parameter auf `False` gesetzt sind, wird das

Dokument ohne weitere Eingabe durch den Benutzer gedruckt.

Fehler

2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.

Redo

Siehe auch

Methode: `Redo()` als Boolean

Beschreibung

Stellt die durch den letzten Rückgängig-Befehl rückgängig gemachte Änderung wieder her.

Fehler

2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.

2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Selection

Siehe auch

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

Beschreibung

Setzt oder ruft die aktuelle Textauswahl auf der Benutzeroberfläche ab.

Fehler

2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.

2002 Es ist keine Cursorauswahl aktiv.

2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' -----
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

' if we are the end of the document, re-start at the beginning
If (objAuthenticView.Selection.IsEqual(objAuthenticView.DocumentEnd)) Then
    objAuthenticView.Selection = objAuthenticView.DocumentBegin
Else
    ' objAuthenticView.Selection =
objAuthenticView.Selection.GotoNextCursorPosition()
    ' or shorter:
    objAuthenticView.Selection.GotoNextCursorPosition().Select
End If
```

Undo

Siehe auch

Methode: `Undo()` als Boolean

Beschreibung

Macht die letzte Änderung des Dokuments innerhalb dieser Ansicht rückgängig.

Fehler

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

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, sodass die XML-Datei korrekt validiert werden kann. Beachten Sie bitte, dass dadurch auch Schemadateien neu geladen werden können.

Fehler

Die Methode gibt niemals einen Fehler zurück.

Beispiel

```
// -----
// XMLSpy scripting environment - JavaScript
// -----
if( Application.ActiveDocument &&
( Application.ActiveDocument.CurrentViewMode == 4) )
{
    var objDocType;
    objDocType =
Application.ActiveDocument.DocEditView.XMLRoot.GetFirstChild(10);

    if( objDocType)
    {
        var objEntity = Application.ActiveDocument.CreateChild(14);
        objEntity.Name = "child";
        objEntity.TextValue = "SYSTEM \"child.xml\"";
        objDocType.AppendChild( objEntity);

        Application.ActiveDocument.AuthenticView.UpdateXMLInstanceEntities();
    }
}
```

WholeDocument

Siehe auch

Eigenschaft: `WholeDocument` als [AuthenticRange](#) (schreibgeschützt)

Beschreibung

Ruft ein Bereichsobjekt ab, das das gesamte Dokument auswählt.

Fehler

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

XMLDataRoot

Siehe auch

Eigenschaft: `XMLDataRoot` als [XMLData](#) (schreibgeschützt)

Beschreibung

Gibt das am höchsten gestufte XMLData-Element des aktuellen Dokuments zurück oder definiert dieses. Dieses Element beschreibt normalerweise die Dokumentstruktur und wäre von der Art `spyXMLDataXMLDocStruct`, `spyXMLDataXMLEntityDocStruct` oder `spyXMLDataDTDDocStruct`.

Fehler

- 2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
- 2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

4.3.5 CodeGeneratorDlg

Siehe auch

Nur verfügbar/aktiv in der Enterprise Edition. Bei Aufruf von einer anderen Version aus wird ein Fehler zurückgegeben.

Eigenschaften und Methoden

Standard-Automation-Eigenschaften

[Application](#)

[Parent](#)

Eigenschaften zur Auswahl der Programmiersprache

[ProgrammingLanguage](#)

[TemplateFileName](#)

[CompatibilityMode](#)

Einstellungen für C++-Code

[CPPSettings_DOMType](#)

[CPPSettings_LibraryType](#)

[CPPSettings_UseMFC](#)

[CPPSettings_GenerateVC6ProjectFile](#)

[CPPSettings_GenerateVSPProjectFile](#)

Einstellungen für C#-Code

[CSharpSettings_ProjectType](#)

Behandlung von Dialogfeldern für die oben genannten Codegenerierungseigenschaften

[PropertySheetDlgAction](#)

Eigenschaften zur Auswahl des Ausgabepfads

[OutputPath](#)

[OutputPathDlgAction](#)

Darstellung des Ergebnisses

[OutputResultDlgAction](#)

Beschreibung

Mit Hilfe dieses Objekts können Sie die Generierung von Programmcode für Schemadateien konfigurieren. Die Methode [GenerateProgramCode](#) erwartet einen CodeGeneratorDlg als Parameter, um die Codegenerierung sowie die damit in Zusammenhang stehenden Interaktionen des Benutzers zu konfigurieren.

Application**Siehe auch**

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Eigenschaft: [Application](#) als [Application](#) (schreibgeschützt)

Beschreibung

Ruft das Authentic Desktop application-Objekt auf.

Fehler

2200 Das Objekt ist nicht mehr gültig.

2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

CompatibilityMode

Eigenschaft: [CompatibilityMode](#) als Boolean

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung

Wird auf `true` gesetzt, um Code zu generieren, der mit XMLSpy 2005R3 kompatibel ist. Wird auf `false` gesetzt, um neu hinzugefügte Codegenerierungsfeatures zu verwenden.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

CPPSettings_DOMType

Eigenschaft: [CPPSettings_DOMType](#) als [SPYDOMType](#)

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung

Definiert eine der Einstellungen für die Konfiguration der Generierung von C++-Code.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

CPPSettings_GenerateVC6ProjectFile

Eigenschaft: [CPPSettings_GenerateVC6ProjectFile](#) als Boolean

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung

Definiert eine der Einstellungen für die Konfiguration der Generierung von C++-Code.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

CPPSettings_GenerateVSProjectFile

Eigenschaft: [CSharpSettings_GenerateVSProjectFile](#) als [SPYProjectType](#)

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung

Definiert eine der Einstellungen für die Konfiguration der Generierung von C++-Code. Nur `spyVisualStudio2003Project` (=1) und `spyVisualStudio2005Project` (=4) sind gültige Projekttypen.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

CPPSettings_LibraryType

Eigenschaft: [CPPSettings_LibraryType](#) als [SPYLibType](#)

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung

Definiert eine der Einstellungen für die Konfiguration der Generierung von C++-Code.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

CPPSettings_UseMFC

Eigenschaft: [CPPSettings_UseMFC](#) als Boolean

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung

Definiert eine der Einstellungen für die Konfiguration der Generierung von C++-Code.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

CSharpSettings_ProjectType

Eigenschaft: [CSharpSettings_ProjectType](#) als [SPYProjectType](#)

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung

Definiert die einzige Einstellung für die Konfiguration der Generierung von C#-Code.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

OutputPath

Eigenschaft: [OutputPath](#) als String

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung

Wählt das Basisverzeichnis für den gesamten generierten Code aus.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

OutputPathDialogAction

Eigenschaft: [OutputPathDialogAction](#) als [SPYDialogAction](#)

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung

Definiert, wie das untergeordnete Dialogfeld zur Auswahl des Ausgabepfads für die Codegenerierung behandelt wird. Setzen Sie diesen Wert auf *spyDialogUserInput(2)*, damit das Dialogfeld standardmäßig mit dem aktuellen Wert der Eigenschaft [OutputPath](#) angezeigt wird. Verwenden Sie *spyDialogOK(0)*, damit das Dialogfeld nicht angezeigt wird.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

OutputResultDialogAction

Eigenschaft: [OutputResultDialogAction](#) als [SPYDialogAction](#)

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung

Definiert, wie das untergeordnete Dialogfeld behandelt wird, das fragt, ob das Ergebnis der Codegenerierung angezeigt werden soll. Setzen Sie diesen Wert auf *spyDialogUserInput(2)*, damit das Dialogfeld angezeigt wird. Verwenden Sie *spyDialogOK(0)*, damit das Dialogfeld dem Benutzer nicht angezeigt wird.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

Parent

Siehe auch

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

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

Beschreibung

Ruft das übergeordnete Element des Objekts auf.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ProgrammingLanguage

Eigenschaft: [ProgrammingLanguage](#) als [ProgrammingLanguage](#)

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung

Wählt die Ausgabesprache aus, in der der Code generiert werden soll.

Achtung: Wenn Sie diese Eigenschaft auf C++, C# oder Java setzen, wird auch die Eigenschaft [TemplateFileName](#) auf die entsprechende mit Authentic Desktop installierte Vorlagendatei gesetzt. Wenn Sie C++, C# oder Java-Code basierend auf Ihren eigenen Vorlagen generieren möchten, definieren Sie zuerst die Programmiersprache und wählen Sie dann Ihre Vorlagendatei aus.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

PropertySheetDialogAction

Eigenschaft: [PropertySheetDialogAction](#) als [SPYDialogAction](#)

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung

Definiert, wie das untergeordnete Dialogfeld, mit dem die Codegenerierung konfiguriert wird, behandelt wird. Setzen Sie diesen Wert auf *spyDialogUserInput(2)* um das Dialogfeld mit den aktuellen Werten als Standardwerten anzuzeigen. Verwenden Sie *spyDialogOK(0)*, damit das Dialogfeld nicht angezeigt wird.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

TemplateFileName

Eigenschaft: [TemplateFileName](#) als String

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein

Fehler zurückgegeben.

Beschreibung

Wählt die Vorlagendatei für die Codegenerierung aus. Im Ordner "SPL" Ihres Authentic Desktop Installationsverzeichnis sind Vorlagendateien für C++, C# oder Java enthalten.

Wenn Sie diese Eigenschaft auf eine der Codegenerierungsvorlagendateien Ihrer Authentic Desktop Installation setzen, wird die Eigenschaft [ProgrammingLanguage](#) automatisch auf den entsprechenden Wert gesetzt.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

4.3.6 DatabaseConnection

Siehe auch

Eigenschaften für den Import und Export

[File](#) oder
[ADOConnection](#) oder
[ODBCConnection](#)

Eigenschaften nur für den Import

[DatabaseKind](#)
[SQLSelect](#)
[AsAttributes](#)
[ExcludeKeys](#)
[IncludeEmptyElements](#)
[NumberDateTimeFormat](#)
[NullReplacement](#)
[CommentIncluded](#)

Eigenschaften nur für den Export

[CreateMissingTables](#)
[CreateNew](#)
[TextFieldLen](#)
[DatabaseSchema](#)

Beschreibung

`DatabaseConnection` definiert die Parameter für die Datenbankverbindung.

Bitte beachten Sie, dass sich die Eigenschaften der `DatabaseConnection` Schnittstelle auf die Einstellungen des Import- und Exportdialogfelds von Authentic Desktop beziehen.

ADOConnection

Siehe auch

Eigenschaft: `ADOConnection` als `String`

Beschreibung

Die Eigenschaft `ADOConnection` enthält einen Connection String. Verwenden Sie entweder diese Eigenschaft oder [ODBCConnection](#) oder [File](#), um eine Datenbank zu referenzieren.

Fehler

Es werden keine Fehlercodes zurückgegeben.

Beispiel

```
Dim objSpyConn As DatabaseConnection
Set objSpyConn = objSpy.GetDatabaseSettings

Dim objADO As DataLinks
Set objADO = CreateObject("DataLinks")

If Not (objADO Is Nothing) Then
    Dim objConn As Connection
    Set objConn = objADO.PromptNew
    objSpyConn.ADOConnection = objConn.ConnectionString
End If
```

AsAttributes

Siehe auch

Eigenschaft: `AsAttributes` als `Boolean`

Beschreibung

Setzen Sie `AsAttributes` auf "true", wenn alle Importfelder initialisiert werden sollen, um als Attribute importiert zu werden. Der Standardwert ist "false" und initialisiert alle Felder, um als Elemente importiert zu werden. Diese Eigenschaft wird nur in Aufrufen von [Application.GetDatabaseImportElementList](#) verwendet.

Fehler

Es werden keine Fehlercodes zurückgegeben.

CommentIncluded

Siehe auch

Eigenschaft: `CommentIncluded` als `Boolean`

Beschreibung

Diese Eigenschaft legt fest, ob zusätzliche Kommentare zum generierten XML-Code hinzugefügt werden. Der Standardwert ist true. Diese Eigenschaft wird nur beim Import aus einer Datenbank verwendet.

Fehler

Es werden keine Fehlercodes zurückgegeben.

CreateMissingTables

Siehe auch

Eigenschaft: `CreateMissingTables` als `Boolean`

Beschreibung

Wenn `CreateMissingTables` "true" ist, werden Tabellen, die nicht bereits in der Exportdatenbank definiert sind, beim Export erstellt. Die Standardeinstellung ist true. Diese Eigenschaft wird nur beim Export in Datenbanken verwendet.

Fehler

Es werden keine Fehlercodes zurückgegeben.

CreateNew

Siehe auch

Eigenschaft: `CreateNew` als `Boolean`

Beschreibung

Setzen Sie `CreateNew` auf "true", wenn Sie beim Export eine neue Datenbank erstellen möchten. Eine möglicherweise bereits vorhandene Datenbank wird dabei überschrieben. Siehe auch [DatabaseConnection.File](#). Der Standardwert ist "false". Diese Eigenschaft wird nur beim Export in eine Datenbank verwendet.

Fehler

Es werden keine Fehlercodes zurückgegeben.

DatabaseKind

Siehe auch

Eigenschaft: `DatabaseKind` als [SPYDatabaseKind](#)

Beschreibung

Wählen Sie aus, welche Art von Datenbank Zugriff erhält. Der Standardwert ist `spyDB_Unspecified(7)` und ist in den meisten Fällen ausreichend. Diese Eigenschaft wird nur beim Import von Datenbanken verwendet.

Fehler

Es werden keine Fehlercodes zurückgegeben.

DatabaseSchema

Siehe auch

Eigenschaft: `DatabaseSchema` als String

Beschreibung

Diese Eigenschaft definiert das Schema, das für den Export in schemafähige Datenbanken verwendet wird. Der Standardwert ist `""`. Diese Eigenschaft wird nur beim Export in eine Datenbank verwendet.

Fehler

Es werden keine Fehlercodes zurückgegeben.

ExcludeKeys

Siehe auch

Eigenschaft: `ExcludeKeys` als Boolean

Beschreibung

Setzen Sie `ExcludeKeys` auf `"true"`, wenn Sie alle Schlüsselspalten vom Import ausnehmen möchten. Der Standardwert ist `"false"`. Diese Eigenschaft wird nur beim Import von Datenbanken verwendet.

Fehler

Es werden keine Fehlercodes zurückgegeben.

File

Siehe auch

Eigenschaft: `File` als String

Beschreibung

Die Eigenschaft `File` definiert den Pfad für die Datenbank beim Export oder Import. Diese Eigenschaft kann nur in Verbindung mit einer Microsoft Access-Datenbank verwendet werden. Verwenden Sie entweder diese Eigenschaft oder [ODBCConnection](#) oder [ADOConnection](#) um die Datenbank zu referenzieren.

Siehe auch [Import und Export](#).

Fehler

Es werden keine Fehlercodes zurückgegeben.

IncludeEmptyElements

Siehe auch

Eigenschaft: [IncludeEmptyElements](#) als `Boolean`

Beschreibung

Setzen Sie `IncludeEmptyElements` auf "false" wenn alle leeren Elemente ausgenommen werden sollen. Der Standardwert ist "true". Diese Eigenschaft wird nur beim Import von Datenbanken verwendet.

Fehler

Es werden keine Fehlercodes zurückgegeben.

NullReplacement

Siehe auch

Eigenschaft: [NullReplacement](#) als `String`

Beschreibung

Diese Eigenschaft enthält den Textwert, der beim Import von leeren Elementen (Nullwerten) verwendet wird. Der Standardwert ist "". Diese Eigenschaft wird nur beim Import von Datenbanken verwendet.

Fehler

Es werden keine Fehlercodes zurückgegeben.

NumberDateTimeFormat

Siehe auch

Eigenschaft: [NumberDateTimeFormat](#) als [SPYNumberDateTimeFormat](#)

Beschreibung

Die Eigenschaft `NumberDateTimeFormat` definiert das Format von Datums- und Uhrzeitwerten. Die Standardeinstellung ist [spySystemLocale](#). Diese Eigenschaft wird nur beim Import von Datenbanken verwendet.

Fehler

Es werden keine Fehlercodes zurückgegeben.

ODBCConnection

Siehe auch

Eigenschaft: `ODBCConnection` als `String`

Beschreibung

Die Eigenschaft `ODBCConnection` enthält einen ODBC Connection String. Verwenden Sie entweder diese Eigenschaft oder [ADOConnection](#) oder [File](#) um eine Datenbank zu referenzieren.

Fehler

Es werden keine Fehlercodes zurückgegeben.

SQLSelect

Siehe auch

Eigenschaft: `SQLSelect` als `String`

Beschreibung

Die SQL-Abfrage für den Import ist in der Eigenschaft `SQLSelect` gespeichert. Diese Eigenschaft wird nur beim Import von Datenbanken verwendet. Siehe auch [Import und Export](#).

Fehler

Es werden keine Fehlercodes zurückgegeben.

TextFieldLen

Siehe auch

Eigenschaft: `TextFieldLen` als `Integer`

Beschreibung

Die Eigenschaft `TextFieldLen` definiert die Länge der beim Export erstellten Textfelder. Standardwert ist 255. Diese Eigenschaft wird nur beim Export in eine Datenbank verwendet.

Fehler

Es werden keine Fehlercodes zurückgegeben.

4.3.7 Dialogs

Siehe auch

Eigenschaften und Methoden

Standard-Automation-Eigenschaften

[Application](#)

[Parent](#)

Verschiedene Dialog-Objekte

[CodeGeneratorDlg](#)

[FileSelectionDlg](#)

[SchemaDocumentationDlg](#)

[GenerateSampleXMLDlg](#)

[DTDSchemaGeneratorDlg](#)

[FindInFilesDlg](#)

[WSDLDocumentationDlg](#)

Beschreibung

Über das Objekt "Dialogs" haben Sie Zugriff auf die verschiedenen integrierten Dialogfelder von Authentic Desktop. Mit Hilfe dieser Dialog-Objekte können die Felder von Benutzerdialogfeldern initialisiert werden, bevor sie dem Benutzer angezeigt werden oder es kann von Ihrem Programm eine vollständige Eingabe durch den Benutzer simuliert werden.

Application

Siehe auch

Eigenschaft: [Application](#) als [Application](#) (schreibgeschützt)

Beschreibung

Ruft das Authentic Desktop application-Objekt auf.

Fehler

- 2300 Das Objekt ist nicht mehr gültig.
- 2301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

CodeGeneratorDlg

Siehe auch

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Eigenschaft: [CodeGeneratorDlg](#) als [CodeGeneratorDlg](#) (schreibgeschützt)

Beschreibung

Ruft eine neue Instanz eines Dialogfeldobjekts für die Codegenerierung ab. Sie benötigen dieses Objekt, um die nötigen Parameter an die Codegenerierungsmethoden zu übergeben. Als Anfangswerte werden die zuletzt bei diesem Dialogfeld verwendeten genommen.

Fehler

- 2300 Das Dialogs-Objekt oder eines seiner übergeordneten Objekte ist nicht mehr gültig.

2301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

FileSelectionDlg

Siehe auch

Eigenschaft: `FileSelectionDlg` als [FileSelectionDlg](#) (schreibgeschützt)

Beschreibung

Ruft eine neue Instanz eines Dialogfeldobjekts zur Dateiauswahl auf.

Dialogfeldobjekte zur Dateiauswahl werden mit einigen Events an Sie übergeben, die das Öffnen oder Speichern von Dokumenten und Projekten anzeigen.

Fehler

- 2300 Das Dialogs-Objekt oder eines seiner übergeordneten Objekte ist nicht mehr gültig.
- 2301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Parent

Siehe auch

Eigenschaft: `Parent` als [Application](#) (schreibgeschützt)

Beschreibung

Ruft das Authentic Desktop application-Objekt auf.

Fehler

- 2300 Das Objekt ist nicht mehr gültig.
- 2301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

SchemaDocumentationDlg

Siehe auch

Eigenschaft: `SchemaDocumentationDlg` als [SchemaDocumentationDlg](#) (schreibgeschützt)

Beschreibung

Ruft eine neue Instanz eines Dialogfeldobjekts ab, das die Generierung von Schemadokumentation parametrisiert. Informationen zur Verwendung finden Sie unter [Document.GenerateSchemaDocumentation](#).

Fehler

- 2300 Das Dialogs-Objekt oder eines seiner übergeordneten Objekte ist nicht mehr gültig.
- 2301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

GenerateSampleXMLDlg

Siehe auch

Eigenschaft: `GenerateSampleXMLDlg` als [GenerateSampleXMLDlg](#) (schreibgeschützt)

Beschreibung

Ruft eine neue Instanz eines Dialogfeldobjekts ab, das die Generierung einer XML-Beispieldatei anhand eines W3C-Schemas oder einer DTD parametrisiert. Informationen zur Verwendung finden Sie unter [GenerateSampleXML](#).

Fehler

- 2300 Das Dialogs-Objekt oder eines seiner übergeordneten Objekte ist nicht mehr gültig.
- 2301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

DTDSchemaGeneratorDlg

Siehe auch

Eigenschaft: `DTDSchemaGeneratorDlg` als [DTDSchemaGeneratorDlg](#) (schreibgeschützt)

Beschreibung

Ruft eine neue Instanz eines Dialogfeldobjekts ab, das die Generierung eines Schemas oder einer DTD parametrisiert. Informationen zur Verwendung finden Sie unter [Document.GenerateDTDOrSchemaEx](#).

Fehler

- 2300 Das Dialogs-Objekt oder eines seiner übergeordneten Objekte ist nicht mehr gültig.
- 2301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

FindInFilesDlg

Siehe auch

Eigenschaft: `FindInFilesDlg` als [FindInFilesDlg](#) (schreibgeschützt)

Beschreibung

Ruft eine neue Instanz eines Dialogfeldobjekts ab, das die Suche oder Ersetzung von Strings in Dateien parametrisiert. Informationen zur Verwendung finden Sie unter [Application.FindInFiles](#).

Fehler

- 2300 Das Dialogs-Objekt oder eines seiner übergeordneten Objekte ist nicht mehr gültig.
- 2301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

WSDLDocumentationDlg

Siehe auch

Eigenschaft: `WSDLDocumentationDlg` als [WSDLDocumentationDlg](#) (schreibgeschützt)

Beschreibung

Ruft eine neue Instanz eines Dialogfeldobjekts ab, das die Generierung von WSDL-Dokumentation parametrisiert. Informationen zur Verwendung finden Sie unter [Document.GenerateWSDLDocumentation](#).

Fehler

- 2300 Das Dialogs-Objekt oder eines seiner übergeordneten Objekte ist nicht mehr gültig.
- 2301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

4.3.8 Document

Siehe auch

Eigenschaften und Methoden

Standard-Automation-Eigenschaften

[Application](#)

[Parent](#)

Verschiedene Dokumenteigenschaften und Methoden

[SetActiveDocument](#)

[Encoding](#)

[SetEncoding \(obsolete\)](#)

[Suggestions](#)

XML-Validierung

[IsWellFormed](#)

[IsValid](#)

[SetExternalIsValid](#)

Dokumentkonvertierung und -transformation

[AssignDTD](#)

[AssignSchema](#)

[AssignXSL](#)

[AssignXSLEO](#)

[ConvertDTDOrSchema](#)

[ConvertDTDOrSchemaEx](#)

[GenerateDTDOrSchema](#)

[GenerateDTDOrSchemaEx](#)

[CreateSchemaDiagram](#)

[ExecuteXQuery](#)

[TransformXSL](#)

[TransformXSLEx](#)

[TransformXSLEO](#)

[GenerateProgramCode](#) (nur Enterprise Edition)

[GenerateSchemaDocumentation](#)

[GenerateSampleXML](#)

[GenerateWSDLDocumentation](#)

Dokumentexport

[GetExportElementList](#)

[ExportToText](#)

[ExportToDatabase](#)

Dateispeicherung und -benennung

[FullName](#)

[Name](#)

[Path](#)

[GetPathName \(obsolete\)](#)

[SetPathName \(obsolete\)](#)

[Title](#)

[IsModified](#)

[Saved](#)

[SaveAs](#)

[Save](#)

[SaveInString](#)

[SaveToURL](#)

[Close](#)

Aufruf einer Ansicht

[CurrentViewMode](#)
[SwitchViewMode](#)
[AuthenticView](#)
[GridView](#)
[DocEditView \(obsolete\)](#)

Zugriff auf XMLData

[RootElement](#)
[DataRoot](#)
[CreateChild](#)
[UpdateViews](#)
[StartChanges](#)
[EndChanges](#)
[UpdateXMLData](#)

Beschreibung

Document-Objekte repräsentieren in Authentic Desktop geöffnete XML-Dokumente.

Verwenden Sie eine der folgenden Eigenschaften, um Dokumente aufzurufen, die bereits in Authentic Desktop geöffnet sind:

[Application.ActiveDocument](#)
[Application.Documents](#)

Verwenden Sie eine der folgenden Methoden, um ein neues Dokument in Authentic Desktop zu öffnen:

[Documents.OpenFile](#)
[Documents.OpenURL](#)
[Documents.OpenURLDialog](#)
[Documents.NewFile](#)
[Documents.NewFileFromText](#)
[SpyProjectItem.Open](#)
[Application.ImportFromDatabase](#)
[Application.ImportFromSchema](#)
[Application.ImportFromText](#)
[Application.ImportFromWord](#)
[Document.ConvertDTDOrSchema](#)
[Document.GenerateDTDOrSchema](#)

Events***OnBeforeSaveDocument***

Siehe auch

Event: *OnBeforeSaveDocument*(*objDocument* als [Document](#), *objDialog* als [FileSelectionDlg](#))

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_BeforeSaveDocument(objDocument, objDialog)  
EndFunction  
  
' old handler - now obsolete  
' return string to save to new file name  
' return empty string to cancel save operation
```

```
' return nothing to save to original name
Function On_SaveDocument(objDocument, strFilePath)
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_BeforeSaveDocument(objDocument, objDialog)
{

// old handler - now obsolete
// return string to save to new file name
// return empty string to cancel save operation
// return nothing to save to original name
function On_SaveDocument(objDocument, strFilePath)
{
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (27, ...) //nEventId=27
```

Beschreibung

Dieses Event wird bei jedem Versuch, ein Dokument zu speichern ausgelöst. Das Dialogfeldobjekt zur Dateiauswahl wird mit dem für die Dokumentdatei gewählten Namen initialisiert. Sie können diese Auswahl ändern. Um mit dem Speichern des Dokuments fortzufahren, belassen Sie die Eigenschaft [FileSelectDialogAction](#) von *io_objDialog* auf ihrem Standardwert [spyDialogOK](#). Um das Speichern des Dokuments abubrechen, setzen Sie diese Eigenschaft auf [spyDialogCancel](#).

OnBeforeCloseDocument

Siehe auch

Event: [OnBeforeCloseDocument\(objDocument als Document\)](#) als Boolean

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_BeforeCloseDocument(objDocument)
'On_BeforeCloseDocument=False 'to prohibit closing of document
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_BeforeCloseDocument(objDocument)
{
//return false to prohibit closing of document
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (28, ...) //nEventId=28
```

Beschreibung

Dieses Event wird bei jedem Versuch ein Dokument zu schließen ausgelöst. Um das Schließen des Dokuments zu verhindern, muss der Wert "false" zurückgegeben werden.

OnBeforeValidate

Siehe auch

Event: `OnBeforeValidate(objDocument` als [Document](#), `bOnLoading` als Boolean, `bOnCommand` als Boolean) als Boolean

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_BeforeValidate(objDocument, bOnLoading, bOnCommand)
    On_BeforeValidate=bCancelDefaultValidation      'set by the script if
    necessary
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_BeforeValidate(objDocument, eViewMode, bActivated)
{
    return bCancelDefaultValidation                //set by the script if necessary
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugin.OnEvent (32, ...) //nEventId=32
```

Beschreibung

Dieses Event wird ausgelöst, bevor das Dokument validiert wird. Die Standardvalidierung kann durch Rückgabe von "false" vom Event Handler unterdrückt werden. In diesem Fall sollte das Script auch mit Hilfe der Methode [SetExternallsValid](#) das Validierungsergebnis definieren.

`bOnLoading` ist "true", wenn das Event bei der Anfangsvalidierung beim Laden des Dokuments ausgelöst wird.

`bOnCommand` ist "true", wenn der Benutzer in der Symbolleiste oder im Menü den Befehl "Validieren" ausgewählt hat.

Verfügbar mit Typbibliothek Version 1.5

OnCloseDocument

Siehe auch

Event: `OnCloseDocument(objDocument` als [Document](#))

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_Close Document(objDocument)
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_Close Document(objDocument)
{
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugin.OnEvent (8, ...) //nEventId=8
```

Beschreibung

Dieses Event wird ausgelöst, wenn ein Dokument geschlossen wird. Ändern Sie das Dokument

nicht von diesem Event aus.

OnViewActivation

Siehe auch

Event: `OnViewActivation(objDocument als Document, eViewMode als SPYViewModes, bActivated als Boolean)`

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_ViewActivation(objDocument, eViewMode, bActivated)
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_ViewActivation(objDocument, eViewMode, bActivated)
{
}
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (29, ...) //nEventId=29
```

Beschreibung

Dieses Event wird immer dann ausgelöst, wenn eine Ansicht eines Dokuments eingeblendet (d.h. zur aktiven Ansicht wird) oder ausgeblendet wird (d.h. eine andere Ansicht wird aktiv oder das Dokument wird geschlossen). Das erste Ansichtsaktivierungsevent nach dem Öffnen eines Dokuments kann jedoch nicht empfangen werden, da es kein document-Objekt gibt, von dem das Event abgerufen werden kann. Verwenden Sie stattdessen das Event [Application.OnDocumentOpened](#).

Application

Siehe auch

Eigenschaft: `Application` als [Application](#) (schreibgeschützt)

Beschreibung

Ruft das Authentic Desktop application-Objekt auf.

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

AssignDTD

Siehe auch

Methode: `AssignDTD(strDTDFile als String, bDialog als Boolean)`

Beschreibung

Die Methode platziert eine Referenz auf die DTD-Datei "strDTDFile" in das Dokument. Beachten Sie, dass kein Fehler auftritt, wenn die Datei nicht existiert oder kein Zugriff darauf besteht. Wenn `bDialog` "true" ist, zeigt Authentic Desktop ein Dialogfeld zum Definieren der Datei an.

Siehe auch [Simple document access](#).

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1409 Dem Dokument darf keine DTD zugewiesen werden.

AssignSchema

Siehe auch

Methode: `AssignSchema` (`strSchemaFile` als String, `bDialog` als Boolean)

Beschreibung

Die Methode platziert eine Referenz auf die Schemadatei "strSchemaFile" in das Dokument. Beachten Sie, dass kein Fehler auftritt, wenn die Datei nicht existiert oder kein Zugriff darauf besteht. Wenn `bDialog` "true" ist, zeigt Authentic Desktop ein Dialogfeld zum Definieren der Datei an.

Siehe auch [Simple document access](#).

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1409 Dem Dokument darf keine Schemadatei zugewiesen werden.

AssignXSL

Siehe auch

Methode: `AssignXSL` (`strXSLFile` als String, `bDialog` als Boolean)

Beschreibung

Die Methode platziert eine Referenz auf die XLS-Datei "strXSLFile" in das Dokument. Beachten Sie, dass kein Fehler auftritt, wenn die Datei nicht existiert oder kein Zugriff darauf besteht. Wenn `bDialog` "true" ist, zeigt Authentic Desktop ein Dialogfeld zum Definieren der Datei an.

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1409 Dem Dokument darf keine XSL-Datei zugewiesen werden

AssignXSLFO

Siehe auch

Methode: `AssignXSLFO` (`strXSLFOFile` als String, `bDialog` als Boolean)

Beschreibung

Die Methode platziert eine Referenz auf die XSLFO-Datei "strXSLFile" in das Dokument. Beachten Sie, dass kein Fehler auftritt, wenn die Datei nicht existiert oder kein Zugriff darauf besteht. Wenn `bDialog` "true" ist, zeigt Authentic Desktop ein Dialogfeld zum Definieren der Datei an.

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1409 Dem Dokument darf keine XSL-Datei zugewiesen werden

AsXMLString

Siehe auch

Eigenschaft: [AsXMLString](#) als String

Beschreibung

Diese Eigenschaft dient zum Abrufen oder Definieren des Inhalts des Dokuments.

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1404 Das XMLData-Objekt kann nicht erstellt werden.
- 1407 Der Ansichtsmodus kann nicht gewechselt werden.

AuthenticView

Siehe auch

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

Beschreibung

Gibt ein Objekt zurück, über das Sie Zugriff auf die Eigenschaften und Methoden haben, die für die Authentic-Ansicht spezifisch sind. Das zurückgegebene Objekt ist nur gültig, wenn das aktuelle Dokument im Authentic-Ansichtsmodus geöffnet wird. Die Lebenszeit eines Objekts endet mit dem nächsten Ansichtswechsel. Jeder Versuch, Objekte oder eines ihrer Children danach aufrufen, hat einen Fehler zur Folge, der angibt, dass das Objekt ungültig ist.

[AuthenticView](#) und [DocEditView](#) bieten beide Automationszugriff auf den Authentic-Ansichtsmodus von XMLSpy. Dass sich die beiden Funktionen überlappen, ist beabsichtigt. Eine zukünftige Version von [AuthenticView](#) wird alle Funktionalitäten von [DocEditView](#) und seinen Unterobjekten enthalten, sodass [DocEditView](#) nicht mehr benötigt wird.

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1417 Das Dokument muss im Authentic-Ansichtsmodus geöffnet sein.

Beispiele

```
' -----
' XMLSpy scripting environment - VBScript
' secure access to authentic view object
' -----
Dim objDocument
Set objDocument = Application.ActiveDocument
If (Not objDocument Is Nothing) Then
    ' we have an active document, now check for view mode
    If (objDocument.CurrentViewMode <> spyViewAuthentic) Then
        If (Not objDocument.SwitchViewMode (spyViewAuthentic)) Then
            MsgBox "Active document does not support authentic view
mode"
        Else
            ' now it is safe to access the authentic view object
            Dim objAuthenticView
            Set objAuthenticView = objDocument.AuthenticView
            ' now use the authentic view object

        End If
    End If
End If
```



```
Else
    MsgBox "No document is open"
End If
```

Close

Siehe auch

Methode: `Close` (*bDiscardChanges* als Boolean)

Beschreibung

Rufen Sie diese Methode auf, um ein Dokument zu schließen. Wenn `bDiscardChanges` "true" ist und das Dokument geändert wurde, wird das Dokument geschlossen aber nicht gespeichert.

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1401 Das Dokument muss zuerst gespeichert werden.

ConvertDTDOrSchema

Siehe auch

Methode: `ConvertDTDOrSchema` (*nFormat* als [SPYDTDSchemaFormat](#), *nFrequentElements* als [SPYFrequentElements](#))

Parameter

nFormat

Setzt das Schemaausgabeformat auf DTD oder W3C.

nFrequentElements

Erstellt komplexe Elemente als Elemente oder complexTypes.

Beschreibung

`ConvertDTDOrSchema` nimmt das vorhandene Schemaformat und konvertiert es in ein anderes Format.

Zur Feinabtimung einer DTD / Schemakonvertierung verwenden Sie

[ConvertDTDOrSchemaEx](#).

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1412 Fehler bei der Konvertierung.

ConvertDTDOrSchemaEx

Siehe auch

Methode: `ConvertDTDOrSchemaEx` (*nFormat* als [SPYDTDSchemaFormat](#), *nFrequentElements* als [SPYFrequentElements](#), *sOutputPath* als String, *nOutputPathDialogAction* als [SPYDialogAction](#))

Parameters

nFormat

Setzt das Schemaausgabeformat auf DTD oder W3C.

nFrequentElements

Erstellt komplexe Elemente als Elemente oder complexTypes.

sOutputPath

Der Dateipfad für die neu generierte Datei.

nOutputPathDialogAction

Definiert die Dialogfeldinteraktion für diesen Aufruf.

Beschreibung

ConvertDTDOrSchemaEx nimmt ein vorhandenes Schemaformat und konvertiert es in ein anderes Format.

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1412 Fehler bei der Konvertierung.

CreateChild

Siehe auch

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

Rückgabewert

Die Methode gibt das neue XMLData Objekt zurück.

Beschreibung

Um ein neues XMLData Objekt zu erstellen, verwenden Sie die Methode CreateChild(). Siehe auch [Verwendung von XMLData](#).

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1404 Das XMLData-Objekt kann nicht erstellt werden.
- 1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

CreateSchemaDiagram

Siehe auch

Method: CreateSchemaDiagram (*nKind* als [SPYSchemaDefKind](#), *strName* as String, *strFile* as String)

Rückgabewert

Keiner.

Beschreibung

Die Methode erstellt ein Diagramm des Schematyps *strName* von der Art *nKind* und speichert die Ausgabedatei in *strFile*.

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1414 Das Diagramm konnte nicht gespeichert werden.
- 1415 Es wurde ein ungültiger Schemadefinitionstyp angegeben.

CurrentViewMode

Siehe auch

Methode: [CurrentViewMode](#) als [SPYViewModes](#)

Beschreibung

Die Eigenschaft enthält den aktuellen Ansichtsmodus des Dokuments. Siehe auch [Document.SwitchViewMode](#).

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

DataRoot

Siehe auch

Eigenschaft: [DataRoot](#) als [XMLData](#) (schreibgeschützt)

Beschreibung

Über diese Methode haben Sie Zugriff auf das erste XMLData-Objekt des Dokuments vom Typ *spyXMLDataElement*. Dies ist normalerweise das Root-Element für alle Dokumentinhaltsdaten. Eine Beschreibung, wie Sie das Root-Element des gesamten Dokuments inkl. XML-Prologdaten abrufen, finden Sie unter [XMLSpyDocument.RootElement](#). Wenn [CurrentViewMode](#) nicht *spyViewGrid* oder *spyViewAuthentic* ist, ist möglicherweise ein [UpdateXMLData](#) erforderlich, um Zugriff auf die neuesten [XMLData](#) zu erhalten.

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

DocEditView

Siehe auch

Methode: [DocEditView](#) als [DocEditView](#)

Beschreibung

Enthält eine Referenz auf das aktuelle Authentic Objekt.

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 1417 Das Dokument muss im Authentic-Ansichtsmodus geöffnet sein.

Encoding

Siehe auch

Eigenschaft: [Encoding](#) als String

Beschreibung

Über diese Eigenschaft haben Sie Zugriff auf den Encoding-Wert des Dokuments. Diese

Eigenschaft kann nur aufgerufen werden, wenn das Dokument in *spyViewGrid*, *spyViewText* oder *spyViewAuthentic* geöffnet ist. Um den aktuellen Ansichtsmodus eines Dokuments abzurufen, siehe Beschreibung [CurrentViewMode](#).

Mit dieser Eigenschaft wird die Methode [SetEncoding](#) nicht mehr benötigt.

Mögliche Werte sind zum Beispiel:

```
8859-1,
8859-2,
ASCII, ISO-646,
850,
1252,
1255,
SHIFT-JIS, MS-KANJI,
BIG5, FIVE,
UTF-7,
UTF-8,
UTF-16
```

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 1416 Die Operation wird im aktuellen Ansichtsmodus nicht unterstützt.

EndChanges

Siehe auch

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

Beschreibung

Verwenden Sie die Methode [EndChanges](#) um alle Änderungen seit dem letzten Aufruf von [Document.StartChanges](#) anzuzeigen.

Fehler

- 1400 Das Objekt ist nicht mehr gültig.

ExecuteXQuery

Siehe auch

Methode: [ExecuteXQuery](#) (*strXMLFileName* als String)

Beschreibung

Führt die im Dokument enthaltene XQuery-Anweisung aus. Verwendet die XML-Datei, die als XML-Quelle für die Transformation definiert wurde. Wenn Ihr XQuery-Script keine XML-Quelle verwendet, setzen Sie den Parameter *strXMLFileName* auf einen leeren String.

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1423 XQuery-Transformationsfehler.
- 1424 Es konnten nicht alle für die Operation benötigten Dateien geladen werden. Höchstwahrscheinlich existiert die in *strXMLFileName* definierte Datei nicht oder ist ungültig.

ExportToDatabase

Siehe auch

Methode: `ExportToDatabase` (*pFromChild* als [XMLData](#), *pExportSettings* als [ExportSettings](#), *pDatabase* als [DatabaseConnection](#))

Beschreibung

`ExportToDatabase` exportiert das XML-Dokument beginnend mit dem Element `pFromChild`. Der Parameter `pExportSettings` definiert das Exportverhalten (siehe [Application.GetExportSettings](#)). Der Parameter `pDatabase` definiert das Ziel des Exports (siehe [Application.GetDatabaseSettings](#)).

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1407 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
- 1416 Fehler beim Export.
- 1429 Datenbankauswahl fehlt.
- 1430 Dokumentexport fehlgeschlagen.

Beispiel

```
Dim objDoc As Document
Set objDoc = objSpy.ActiveDocument

'set the behaviour of the export with ExportSettings
Dim objExpSettings As ExportSettings
Set objExpSettings = objSpy.GetExportSettings

'set the destination with DatabaseConnection
Dim objDB As DatabaseConnection
Set objDB = objSpy.GetDatabaseSettings

objDB.CreateMissingTables = True
objDB.CreateNew = True
objDB.File = "C:\Export.mdb"

objDoc.ExportToDatabase objDoc.RootElement, objExpSettings, objDB
If Err.Number <> 0 Then
    a = MsgBox("Error: " & (Err.Number - vbObjectError) & Chr(13) &
        "Description: " & Err.Description)
End If
```

ExportToText

Siehe auch

Methode: `ExportToText` (*pFromChild* als [XMLData](#), *pExportSettings* als [ExportSettings](#), *pTextSettings* als [TextImportExportSettings](#))

Beschreibung

`ExportToText` exportiert Tabellendaten aus dem Dokument beginnend mit `pFromChild` in eine oder mehrere Textdateien. Die Spalten der erzeugten Tabellen werden in alphabetischer Reihenfolge nach den Namen der Tabellenspalten generiert. Verwenden Sie [GetExportElementList](#), um zu sehen, welche Daten exportiert werden. Der Parameter `pExportSettings` definiert die näheren Einzelheiten für den Export. Setzen Sie die Eigenschaft [ExportSettings.ElementList](#) auf die - möglicherweise geänderte - Liste, die von

[GetExportElementList](#) zurückgegeben wird, damit nicht alle enthaltenen Tabellen exportiert werden. Der Parameter `pTextSettings` definiert die Optionen für den Textexport und -import. Sie müssen die Eigenschaft [TextImportExportSettings.DestinationFolder](#) setzen, bevor Sie `ExportToText` aufrufen.

Siehe auch [Import and export of data](#).

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1407 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
- 1416 Fehler beim Export..
- 1430 Dokumentexport fehlgeschlagen.

Beispiel

```
' -----
' VBA client code fragment - export document to text files
' -----

Dim objDoc As Document
Set objDoc = objSpy.ActiveDocument

Dim objExpSettings As ExportSettings
Set objExpSettings = objSpy.GetExportSettings
objExpSettings.ElementList = objDoc.GetExportElementList(
    objDoc.RootElement,
    objExpSettings)

Dim objTextExp As TextImportExportSettings
Set objTextExp = objSpy.GetTextExportSettings
objTextExp.HeaderRow = True
objTextExp.DestinationFolder = "C:\Exports"

On Error Resume Next
objDoc.ExportToText objDoc.RootElement, objExpSettings, objTextExp

If Err.Number <> 0 Then
    a = MsgBox("Error: " & (Err.Number - vbObjectError) & Chr(13) &
        "Description: " & Err.Description)
End If
```

FullName

Siehe auch

Eigenschaft: [FullName](#) als String

Beschreibung

Mit Hilfe dieser Eigenschaft kann der vollständige Dateiname - einschließlich des Pfads - unter dem das Dokument gespeichert wird, abgerufen oder definiert werden. Die Gültigkeit des Namens wird vor dem nächsten Speichervorgang nicht überprüft.

Diese Eigenschaft macht die Methoden [GetPathName](#) und [SetPathName](#) überflüssig.

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1402 Es wurde ein leerer String als vollständiger Dateiname definiert.

GenerateDTDOrSchema

Siehe auch

Methode: `GenerateDTDOrSchema` (*nFormat* als [SPYDTDSchemaFormat](#), *nValuesList* als integer, *nDetection* als [SPYTypeDetection](#), *nFrequentElements* als [SPYFrequentElements](#))

Parameter

nFormat

Setzt das Schemaausgabeformat auf DTD oder W3C.

nValuesList

Generiert nur diese Anzahl an enumerations-facets pro Typ. Setzen Sie den Wert auf -1 um die Anzahl nicht zu begrenzen.

nDetection

Definiert die Genauigkeit der Ermittlung eines simpleType.

nFrequentElements

Sollen die Typen für alle Elemente als global definiert werden? Verwenden Sie diesen Wert *spyGlobalComplexType* um sie als globale Elemente zu definiert. Verwenden Sie andernfalls den Wert *spyGlobalElements*.

Beschreibung

Mit Hilfe dieser Methode können Sie automatisch eine DTD oder ein Schema für das aktuelle XML-Dokument generieren.

Informationen zur Feinabstimmung der DTD / Schema-Generierung finden Sie unter [GenerateDTDOrSchemaEx](#).

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1407 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.

GenerateDTDOrSchemaEx

Siehe auch

Methode: `GenerateDTDOrSchemaEx` (*objDlg* als [DTDSchemaGeneratorDlg](#)) als [Document](#)

Beschreibung

Verwenden Sie diese Methode, um automatisch eine DTD oder ein Schema für das aktuelle XML-Dokument zu generieren. Mit Hilfe eines [DTDSchemaGeneratorDlg](#) Objekts werden die Informationen an den Schema-/DTD-Generator übergeben. Der Generierungsvorgang kann so konfiguriert werden, dass der Benutzer entweder Eingaben machen kann oder dass der Prozess ohne weitere Eingaben durch den Benutzer ausgeführt wird.

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1407 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.

GenerateProgramCode

Methode: [GenerateProgramCode](#) (*objDlg* als [CodeGeneratorDlg](#))

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung

Generiert Java, C++ oder C#-Klassendateien anhand der XML-Schemadefinitionen in Ihrem Dokument. Die Daten werden mit Hilfe eines [CodeGeneratorDlg](#) Objekts an den Code Generator übergeben. Der Generierungsvorgang kann so konfiguriert werden, dass der Benutzer entweder Eingaben machen kann oder dass der Prozess ohne weitere Eingaben durch den Benutzer ausgeführt wird.

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1407 Es wurde ein leerer Dateiname angegeben.
- 1421 Diese Funktion steht in dieser Edition nicht zur Verfügung.

GenerateSampleXML

Methode: [GenerateSampleXML](#) (*objDlg* als [GenerateSampleXMLDlg](#)) als [Document](#)

Beschreibung

Generiert eine XML-Beispieldatei, wenn das Dokument ein Schema oder eine DTD ist. Verwenden Sie [Dialogs.GenerateSampleXMLDlg](#) um eine initialisierte Gruppe von Optionen aufzurufen.

Verfügbar mit Typbibliothek Version 1.5

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.

GenerateSchemaDocumentation

Methode: [GenerateSchemaDocumentation](#) (*objDlg* als [SchemaDocumentationDlg](#))

Beschreibung

Generiert Dokumentation für eine Schemadefinitionsdatei im HTML-, MS-Word- oder RTF-Format. Der Parameter "objDlg" dient zum Parametrisieren des Generierungsvorgangs. Verwenden Sie [Dialogs.SchemaDocumentationDlg](#) um eine initialisierte Gruppe von Optionen aufzurufen. Sie müssen zumindest die Eigenschaft [SchemaDocumentationDlg.OutputFile](#) definieren, bevor Sie mit der Generierung beginnen können.

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1407 Es wurden entweder ungültige Parameter übergeben oder es wurde ein leere Dateiname für die Ausgabe definiert.
- 1417 Das Dokument ist nicht in der Schemaansicht geöffnet. Vielleicht handelt es sich um keine '.xsd' Datei.
- 1421 Die Funktion steht in dieser Edition nicht zur Verfügung.
- 1422 Fehler bei der Generierung

GenerateWSDLDocumentation

Methode: `GenerateWSDLDocumentation` (*objDlg* als [WSDLDocumentationDlg](#))

Beschreibung

Generiert Dokumentation für eine WSDL-Definitionsdatei im HTML-, MS-Word- oder RTF-Format. Der Parameter "objDlg" dient zum Parametrisieren des Generierungsvorgangs. Verwenden Sie [D:\bgs\SchemaDocumentationDlg](#) um eine initialisierte Gruppe von Optionen aufzurufen. Sie müssen zumindest die Eigenschaft [SchemaDocumentationDlg.OutputFile](#) definieren, bevor Sie mit der Generierung beginnen können.

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1407 Es wurden entweder ungültige Parameter übergeben oder es wurde ein leerer Dateiname für die Ausgabe definiert.
- 1417 Das Dokument ist nicht in der Schemaansicht geöffnet. Vielleicht handelt es sich um keine '.xsd' Datei.
- 1421 Die Funktion steht in dieser Edition nicht zur Verfügung.
- 1422 Fehler bei der Generierung

GetExportElementList

Siehe auch

Methode: `GetExportElementList` (*pFromChild* als [XMLData](#), *pExportSettings* als [ExportSettings](#)) als [ElementList](#)

Beschreibung

`GetExportElementList` erstellt eine Elementsammlung für den Export aus dem Dokument. Der Export wird durch die Einstellungen in `pExportSettings` definiert und beginnt beim Element `pFromChild`. Die Funktion gibt eine Sammlung von `ElementListItems` zurück, wobei die Eigenschaften [ElementListItem.Name](#) die Namen der Tabellen enthalten, die aus dem Dokument exportiert werden können. Die Eigenschaft [ElementListItem.FieldCount](#) enthält die Anzahl der Spalten in der Tabelle. Die Eigenschaft [ElementListItem.RecordCount](#) enthält die Anzahl der Datensätze in der Tabelle. Die Eigenschaft [ElementListItem.ElementKind](#) wird nicht verwendet.

Siehe auch [Import und Export von Daten](#).

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1407 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
- 1427 Es konnte kein Parser für die angegebene XML-Datei erstellt werden.
- 1428 Der Export der Elementliste ist fehlgeschlagen.

GetPathName (nicht mehr verwendet)

Wird ersetzt durch [Document.FullName](#)

```
// ----- javascript sample -----  
// instead of:  
// strPathName = Application.ActiveDocument.GetPathName();  
// use now:  
strPathName = Application.ActiveDocument.FullName;
```

Siehe auch

Methode: [GetPathName\(\)](#) als String

Beschreibung

Die Methode `GetPathName` ruft den Pfad des aktiven Dokuments auf.

Siehe auch [Document.SetPathName](#) (nicht mehr verwendet).

GridView

Siehe auch

Eigenschaft: [GridView](#) als [GridView](#)

Beschreibung

Über diese Eigenschaft erhalten Sie Zugriff auf die Funktionalitäten der Grid-Ansicht für das Dokument.

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 1417 Das Dokument muss in der Grid-Ansicht geöffnet sein.

IsModified

Siehe auch

Eigenschaft: [IsModified](#) als Boolean

Beschreibung

True, wenn das Dokument geändert wurde.

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IsValid

Siehe auch

Methode: [IsValid](#) ([strError](#) als Variant, [nErrorPos](#) als Variant, [pBadData](#) als Variant) als Boolean

Rückgabewert

True, wenn das Dokument gültig ist, false, wenn es nicht gültig ist.

Beschreibung

IsValid() validiert das Dokument anhand des damit verknüpften Schemas bzw. der DTD. strError gibt dieselbe Fehlermeldung zurück wie Authentic Desktop, wenn Sie die Datei im Editor validieren.

nErrorPos wird nicht mehr verwendet und steht nur noch aus Kompatibilitätsgründen zur Verfügung. pBadData ist eine Referenz auf das [XMLData](#) Objekt, das den Fehler ausgelöst hat. Es steht nur zur Verfügung, wenn das Dokument gerade in der Textansicht, der Grid-Ansicht oder der Authentic-Ansicht angezeigt wird.

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1407 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
- 1408 Datei kann nicht validiert werden.

Beispiele

For an example written in VBA see [Overview - Simple document access](#)

```
' -----
' XMLSpy scripting environment - VBScript
' -----
Dim errorText
Dim errorPos
Dim badData

' validate the active document
Set doc = Application.ActiveDocument
valid = doc.IsValid(errorText, errorPos, badData)

If (Not valid) Then
    ' create the validation error message text
    Text = errorText

    If ( Not IsEmpty( badData) ) Then
        Text = Text & "(" & badData.Name & "/" & badData.TextValue & ")"
    End If

    Call MsgBox("validation error[" & errorPos & "]: " & Text)
End If

// -----
// XMLSpy scripting environment - JScript
// -----
// define as arrays to support their usage as return parameters
var errorText = new Array(1);
var errorPos = new Array(1);
var badData = new Array(1);

// validate the active document
var doc = Application.ActiveDocument;
var valid = doc.IsValid(errorText, errorPos, badData);

if (! valid)
{
```

```

    // compose the error description
    var text = errorText;

    // access that XMLData object only if filled in
    if ( badData[ 0] != null)
        text += "(" + badData[ 0].Name + "/" + badData[ 0].TextValue + ")"
;

    MsgBox("validation error[" + errorPos + "]: " + text);
}

```

IsWellFormed

Siehe auch

Methode: `IsWellFormed` (*pData* als [XMLData](#), *bWithChildren* als [Boolean](#), *strError* als Variant, *nErrorPos* als Variant, *pBadXMLData* als Variant) als Boolean

Rückgabewert

True, wenn das Dokument wohlgeformt ist.

Beschreibung

`IsWellFormed` überprüft das Dokument beginnend beim Element *pData* auf Wohlgeformtheit.

Wenn das Dokument nicht wohlgeformt ist, enthält *strError* eine Fehlermeldung, *nErrorPos* die Position der Datei und *pBadXMLData* enthält eine Referenz auf das Element, das die Wohlgeformtheit verletzt. Diese Out-Parameter sind als VARIANTS definiert, um Scripting-Sprachen wie VBScript zu unterstützen.

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1407 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben

Beispiel

Siehe [IsValid](#).

Name

Siehe auch

Eigenschaft: `Name` als String (schreibgeschützt)

Beschreibung

Verwenden Sie diese Eigenschaft, um den Namen - ohne Pfad - der Dokumentdatei abzurufen. Um den Dateinamen für ein Dokument zu ändern, verwenden Sie die Eigenschaft [FullName](#).

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Parent

Siehe auch

Eigenschaft: `Parent` als [Documents](#) (schreibgeschützt)

Beschreibung

Ruft das übergeordnete Objekt des Document-Objekts auf.

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Path

Siehe auch

Eigenschaft: `Path` als String (schreibgeschützt)

Beschreibung

Verwenden Sie diese Eigenschaft, um den Pfad der Dokumentdatei - ohne Dateinamen - abzurufen. Um den Dateinamen und Pfad für ein Dokument zu ändern, verwenden Sie die Eigenschaft [FullName](#).

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

RootElement

Siehe auch

Eigenschaft: `RootElement` als [XMLData](#) (schreibgeschützt)

Beschreibung

Über die Eigenschaft `RootElement` haben Sie Zugriff auf das Root-Element der XML-Struktur des Dokuments einschließlich der XML-Prologdaten. Um das erste Element des Inhalts eines Dokuments aufzurufen, navigieren Sie zum ersten Child der Art `spyXMLDataElement` oder verwenden Sie die Eigenschaft [Document.DataRoot](#). Wenn [CurrentViewMode](#) nicht `spyViewGrid` oder `spyViewAuthentic` ist, ist möglicherweise ein [UpdateXMLData](#) erforderlich, um Zugriff auf die neuesten [XMLData](#) zu erhalten.

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Save

Siehe auch

Methode: `Save()`

Beschreibung

Die Methode schreibt alle Änderungen des Dokuments in die dazugehörige Datei. Siehe auch

[Document.FullName.](#)

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1407 Es wurde ein leerer Dateiname angegeben.
- 1403 Fehler beim Speichern der Datei. Wahrscheinlich ist der Dateiname ungültig.

SaveAs

Siehe auch

Methode: [SaveAs](#) ([strFileName](#) als String)

Beschreibung

Speichert das Dokument in der angegebenen Datei. Wenn das Dokument erfolgreich gespeichert werden konnte, wird die Eigenschaft [FullName](#) auf den angegebenen Dateinamen gesetzt.

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1407 Es wurde ein leerer Dateiname angegeben.
- 1403 Fehler beim Speichern der Datei. Wahrscheinlich ist der Dateiname ungültig.

Saved

Siehe auch

Eigenschaft: [Saved](#) als Boolean (schreibgeschützt)

Beschreibung

Mit Hilfe dieser Eigenschaft können Sie überprüfen, ob das Dokument seit der letzten Änderung gespeichert wurde. Sie gibt die Negation von [IsModified](#) zurück.

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

SaveInString

Siehe auch

Methode: [SaveInString](#) ([pData](#) als [XMLData](#), [bMarked](#) als Boolean) als String

Parameter

[pData](#)

[XMLData](#) Element als Start. Setzen Sie [pData](#) auf [Document.RootElement](#), wenn die gesamte Datei kopiert werden soll.

[bMarked](#)

Wenn [bMarked](#) "true" ist, werden nur die in der Grid-Ansicht ausgewählten Elemente kopiert.

Rückgabewert

Gibt einen String mit den XML-Daten zurück.

Beschreibung

`SaveInString` beginnt am Element `pData` und konvertiert die `XMLData` Objekte in eine String-Darstellung.

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1407 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.

SaveToURL

Siehe auch

Methode: `SaveToURL` (*strURL* als String, *strUser* als String, *strPassword* als String)

Rückgabewert

Beschreibung

`SaveToURL()` schreibt das Dokument in die URL *strURL*. Diese Methode definiert den permanenten Dateipfad des Dokuments nicht.

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1402 Es wurde eine ungültige URL angegeben.
- 1403 Fehler beim Speichern unter der URL.

SetActiveDocument

Siehe auch

Methode: `SetActiveDocument()`

Beschreibung

Die Methode definiert das Dokument als das aktive und bringt es in den Vordergrund.

Fehler

- 1400 Das Objekt ist nicht mehr gültig.

Document.SetEncoding (nicht mehr verwendet)

Wird ersetzt durch [Document.Encoding](#)

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.SetEncoding("UTF-16");  
// use now:  
Application.ActiveDocument.Encoding = "UTF-16";
```

Siehe auch

Methode: `SetEncoding` (*strEncoding* als String)

Beschreibung

`SetEncoding` definiert die Kodierung des Dokuments wie der Menübefehl "Datei/Kodierung..." in Authentic Desktop. Mögliche Werte für `strEncoding` sind z.B.:

8859-1,
8859-2,
ASCII, ISO-646,
850,
1252,
1255,
SHIFT-JIS, MS-KANJI,
BIG5, FIVE,
UTF-7,
UTF-8,
UTF-16

SetExternallsValid

Siehe auch

Methode: `SetExternalIsValid` (*bValid* als Boolean)

Parameter

bValid

Definiert das Ergebnis eines externen Validierungsvorgangs.

Beschreibung

Die mit dieser Methode definierte interne Information wird nur beim Abbruch der Standardvalidierung in jedem [OnBeforeValidate](#) Handler abgefragt.

Verfügbar mit Typbibliothek Version 1.5

Fehler

1400 Das Objekt ist nicht mehr gültig.

Document.SetPathName (nicht mehr verwendet)

Wird ersetzt durch [Document.FullName](#)

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.SetPathName("C:\\myXMLFiles\\test.xml");  
// use now:  
Application.ActiveDocument.FullName = "C:\\myXMLFiles\\test.xml";
```

Siehe auch

Methode: `SetPathName` (*strPath* als String)

Beschreibung

Die Methode `SetPathName` definiert den Pfad des aktiven Dokuments. `SetPathName` kopiert den String nur und überprüft nicht, ob der Pfad gültig ist. Alle nachfolgenden Speicheroperationen erfolgen in dieser Datei.

StartChanges

Siehe auch

Methode: `StartChanges()`

Beschreibung

Nach Ausführung von `StartChanges` aktualisiert Authentic Desktop seine Editor-Fenster erst, wenn [Document.EndChanges](#) aufgerufen wird. Dadurch wird die Geschwindigkeit bei der Ausführung komplexer Operationen an der XML-Struktur verbessert.

Fehler

1400 Das Objekt ist nicht mehr gültig.

Suggestions

Eigenschaft: [Suggestions](#) als Array

Beschreibung

Diese Eigenschaft enthält die letzten gültigen Vorschläge des Benutzers für dieses Dokument. Die von XMLSpy generierten Vorschläge können bearbeitet werden, bevor sie dem Benutzer im [OnBeforeShowSuggestions](#) Event angezeigt werden.

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1407 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.

SwitchViewMode

Siehe auch

Methode: [SwitchViewMode](#) (*nMode* als [SPYViewModes](#)) als Boolean

Rückgabewert

Gibt "true" zurück, wenn der Ansichtsmodus gewechselt wird.

Beschreibung

Die Methode definiert den aktuellen Ansichtsmodus des Dokuments in Authentic Desktop. Siehe auch [Document.CurrentViewMode](#).

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 1417 Es wurde ein ungültiger Ansichtsmodus definiert.

TextView

Siehe auch

Eigenschaft: [TextView](#) als [TextView](#)

Beschreibung

Über diese Eigenschaft haben Sie Zugriff auf die Textansichtsfunktionalität des Dokuments.

Fehler

- 1400 Das Objekt ist nicht mehr gültig.
- 1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Title

Siehe auch

Eigenschaft: [Title](#) als String (schreibgeschützt)

Beschreibung

[Title](#) enthält den Dateinamen des Dokuments. Um den Pfad und Dateinamen der Datei

abzurufen, verwenden Sie [FullName](#).

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

TransformXSL

Siehe auch

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

Beschreibung

[TransformXSL](#) verarbeitet das XML-Dokument anhand der damit verknüpften XSL-Datei. Informationen, wie man in das Dokument eine Referenz auf eine XSL-Datei platziert, finden Sie unter [Document.AssignXSL](#).

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1411 Fehler bei der Transformation.

TransformXSLEx

Siehe auch

Method: [TransformXSL\(*nAction* as \[SPYDialogAction\]\(#\)\)](#)

Beschreibung

[TransformXSL](#) verarbeitet das XML-Dokument über die damit verknüpfte XSL-Datei. Informationen, wie man in das Dokument eine Referenz auf eine XSL-Datei platziert, finden Sie unter [Document.AssignXSL](#).

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1411 Fehler bei der Transformation.

TransformXSLFO

Siehe auch

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

Beschreibung

[TransformXSLFO](#) verarbeitet das XML-Dokument anhand der damit verknüpften XSLFO-Datei. Informationen, wie man in das Dokument eine Referenz auf eine XSLFO-Datei platziert, finden Sie unter [AssignXSLFO](#). Bevor Sie diese Methode verwenden, müssen Sie Authentic Desktop einen FOP-Prozessor zuweisen.

Fehler

- 1400 Das document-Objekt ist nicht mehr gültig.
- 1411 Fehler bei der Transformation.

UpdateViews

Siehe auch

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

Beschreibung

Rufen Sie zum Neuzeichnen der Grid-Ansicht und der Strukturansicht `UpdateViews` auf. Dies ist manchmal von Bedeutung, nachdem Sie die `XMLData`-Struktur eines Dokuments geändert haben. Diese Methode zeichnet die Textansicht von Authentic Desktop nicht neu.

Fehler

1400 Das document-Objekt ist nicht mehr gültig.

UpdateXMLData

Siehe auch

Methode: [UpdateXMLData\(\)](#) als Boolean

Beschreibung

Die [XMLData](#)-Struktur wird anhand der aktuellen Ansicht aktualisiert. Beachten Sie bitte, dass dieser Vorgang in der Textansicht fehlschlagen kann, wenn der aktuelle XML-Text nicht wohlgeformt ist. Dies ist nicht notwendig, wenn [CurrentViewMode](#) `spyViewGrid` oder `spyViewAuthentic` ist, weil die [XMLData](#) in diesen Ansichten automatisch aktualisiert werden.

Verfügbar mit Typbibliothek Version 1.5

Fehler

1400 Das document-Objekt ist nicht mehr gültig.

4.3.9 Documents

Siehe auch

Eigenschaften

[Count](#)

[Item](#)

Methoden

[NewFile](#)

[OpenFile](#)

[OpenURL](#)

[OpenURLDialog](#)

[NewFileFromText](#)

Beschreibung

Diese Objekt repräsentiert die gerade in Authentic Desktop geöffneten Dokumente. Verwenden Sie dieses Objekt, um weitere Dokumente zu öffnen oder durch die bereits offenen Dokumente zu iterieren.

Beispiele

```
' -----  
' XMLSpy scripting environment - VBScript  
' iterate through open documents  
' -----  
  
Dim objDocuments  
Set objDocuments = Application.Documents  
  
For Each objDoc In objDocuments  
    'do something useful with your document  
    objDoc.SetActiveDocument()  
Next  
  
// -----  
// XMLSpy scripting environment - JScript  
// close all open documents  
// -----  
for ( var iter = new Enumerator ( Application.Documents );  
      ! iter.atEnd();  
      iter.moveNext() )  
{  
    // MsgBox ("Closing file " + iter.item().Name);  
    iter.item().Close ( true );  
}
```

Count

Siehe auch

Eigenschaft: [Count](#) als long

Beschreibung

Anzahl der offenen Dokumente.

Item

Siehe auch

Methode: `Item` (`n` als long) als [Document](#)

Beschreibung

Ruft das Dokument mit dem Index `n` in dieser Sammlung ab. Der Index ist 1-basiert.

NewFile

Siehe auch

Methode: `NewFile` (`strFile` als String, `strType` als String) als [Document](#)

Parameter

`strFile`

Der vollständige Pfad der neuen Datei.

`strType`

Der Typ der neuen Datei als String (z.B. "xml", "xsd", ...)

Rückgabewert

Gibt die neue Datei zurück.

Beschreibung

`NewFile` erstellt eine neue Datei vom Typ `strType` (d.h. "xml"). Die neu erstellte Datei ist auch das ActiveDocument.

NewFileFromText

Siehe auch

Methode: `NewFileFromText` (`strText` als String, `strType` als String) als [Document](#)

Parameter

`strText`

Der Inhalt des neuen Dokuments ist unformatierter Text.

`strType`

Der zu erstellende Dokumenttyp (d.h. "xml").

Rückgabewert

Die Methode gibt das neue Dokument zurück.

Beschreibung

`NewFileFromText` erstellt ein neues Dokument mit `strText` als seinem Inhalt.

OpenFile

Siehe auch

Methode: `OpenFile` (`strPath` als String, `bDialog` als Boolean) als [Document](#)

Parameter

`strPath`

Pfad und Dateiname der zu öffnenden Datei.

`bDialog`

Zeigt die Dialogfelder für die Benutzereingaben an..

Rückgabewert

Gibt die geöffnete Datei zurück, wenn die Methode erfolgreich war.

Beschreibung

`OpenFile` öffnet die Datei `strPath` . Wenn `bDebug` `TRUE` ist, wird ein Dateidialogfeld angezeigt.

Beispiel

```
Dim objDoc As Document
Set objDoc = objSpy.Documents.OpenFile(strFile, False)
```

OpenURL

Siehe auch

Methode: `OpenURL` (`strURL` als String, `nURLType` als [SPYURLTypes](#), `nLoading` als [SPYLoading](#), `strUser` als String, `strPassword` als String) als [Document](#)

Parameter

`strURL`

URL zum Öffnen eines Dokuments.

`nURLType`

Typ des zu öffnenden Dokuments. Setzen Sie den Wert auf -1, um den Typ automatisch zu ermitteln.

`nLoading`

Setzen Sie `nLoading` auf 0 (Null), wenn das Dokument aus dem Cache oder vom Proxy-Server geladen werden soll. Setzen Sie `nLoading` andernfalls auf 1.

`strUser`

Name des Benutzers, falls erforderlich. Kann leer bleiben.

`strPassword`

Passwort für die Authentifizierung. Kann leer bleiben.

Rückgabewert

Die Methode gibt das geöffnete Dokument zurück.

Beschreibung

`OpenURL` öffnet die URL `strURL`.

OpenURLDialog

Siehe auch

Methode: `OpenURLDialog` (`strURL` als String, `nURLType` als [SPYURLTypes](#), `nLoading` als [SPYLoading](#), `strUser` als String, `strPassword` als String) als [Document](#)

Parameter

`strURL`

URL zum Öffnen eines Dokuments.

`nURLType`

Typ des zu öffnenden Dokuments. Setzen Sie den Wert auf -1, um den Typ automatisch zu ermitteln.

`nLoading`

Setzen Sie `nLoading` auf 0 (Null), wenn das Dokument aus dem Cache oder vom Proxy-Server geladen werden soll. Setzen Sie `nLoading` andernfalls auf 1.

`strUser`

Name des Benutzers, falls erforderlich. Kann leer bleiben.

`strPassword`

Passwort für die Authentifizierung. Kann leer bleiben.

Rückgabewert

Die Methode gibt das geöffnete Dokument zurück.

Beschreibung

`OpenURLDialog` zeigt dem Benutzer das Dialogfeld "URL öffnen" an und setzt in die Eingabefelder die angegebenen Parameter ein.

4.3.10 DTDSchemaGeneratorDlg

Siehe auch

Eigenschaften und Methoden

Standard-Automation-Eigenschaften

[Application](#)

[Parent](#)

[DTDSchemaFormat](#)

[ValueList](#)

[TypeDetection](#)

[FrequentElements](#)

[MergeAllEqualNamed](#)

[ResolveEntities](#)

[AttributeTypeDefinition](#)

[GlobalAttributes](#)

[OnlyStringEnums](#)

[MaxEnumLength](#)

[OutputPath](#)

[OutputPathDebugAction](#)

Beschreibung

Mit Hilfe dieses Objekts können Sie die Generierung eines Schemas oder einer DTD konfigurieren. Die Methode [GenerateDTDOrSchemaEx](#) erwartet einen [DTDSchemaGeneratorDlg](#) als Parameter, um die Generierung sowie die damit in Zusammenhang stehenden Interaktionen mit dem Benutzer zu konfigurieren.

Application

Eigenschaft: [Application](#) als [Application](#) (schreibgeschützt)

Beschreibung

Ruft das Authentic Desktop application-Objekt auf.

Fehler

- 3000 Das Objekt ist nicht mehr gültig.
- 3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

AttributeTypeDefinition

Eigenschaft: [AttributeTypeDefinition](#) als [SPYAttributeTypeDefinition](#)

Beschreibung

Definiert, wie Attributdefinitionen zusammengeführt werden.

Fehler

- 3000 Das Objekt ist nicht mehr gültig.
- 3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

DTDSchemaFormat

Eigenschaft: `DTDSchemaFormat` als [SPYDTDSchemaFormat](#)

Beschreibung

Setzt das Schemaausgabeformat auf DTD oder W3C.

Fehler

- 3000 Das Objekt ist nicht mehr gültig.
- 3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

FrequentElements

Eigenschaft: `FrequentElements` als [SPYFrequentElements](#)

Beschreibung

Sollen die Typen für alle Elemente als global definiert werden? Verwenden Sie diesen Wert `spyGlobalComplexType` um sie global zu definieren. Verwenden Sie andernfalls den Wert `spyGlobalElements`.

Fehler

- 3000 Das Objekt ist nicht mehr gültig.
- 3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

GlobalAttributes

Eigenschaft: `GlobalAttributes` als Boolean

Beschreibung

Sollen Attribute mit demselben Namen und Typ global aufgelöst werden?

Fehler

- 3000 Das Objekt ist nicht mehr gültig.
- 3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

MaxEnumLength

Eigenschaft: `MaxEnumLength` als Integer

Beschreibung

Definiert die maximale Anzahl an Zeichen, die bei Enumerationsnamen zulässig ist. Wenn auch nur ein Wert länger als dieser ist, wird keine Enumeration generiert.

Fehler

- 3000 Das Objekt ist nicht mehr gültig.
- 3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

MergeAllEqualNamed

Eigenschaft: [MergeAllEqualNamed](#) als Boolean

Beschreibung

Sollen Typen aller Elemente mit demselben Namen in einem Typ zusammengeführt werden?

Fehler

- 3000 Das Objekt ist nicht mehr gültig.
- 3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

OnlyStringEnums

Eigenschaft: [OnlyStringEnums](#) als Boolean

Beschreibung

Definiert, ob Enumerationen nur für unformatierte Strings oder für alle Arten von Werten erstellt werden sollen.

Fehler

- 3000 Das Objekt ist nicht mehr gültig.
- 3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

OutputPath

Eigenschaft: [OutputPath](#) als String

Beschreibung

Wählt den Dateinamen für das generierte Schema/die generierte DTD aus.

Fehler

- 3000 Das Objekt ist nicht mehr gültig.
- 3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

OutputPathDialogAction

Eigenschaft: [OutputPathDialogAction](#) als [SPYDialogAction](#)

Beschreibung

Definiert wie das untergeordnete Dialogfeld zur Auswahl des Ausgabepfads für das Schema/die DTD behandelt werden soll. Setzen Sie diesen Wert auf *spyDialogUserInput(2)* damit standardmäßig das Dialogfeld mit dem aktuellen Wert der Eigenschaft [OutputPath](#) angezeigt wird. Verwenden Sie *spyDialogOK(0)* um dem Benutzer das Dialogfeld nicht anzuzeigen.

Fehler

- 3000 Das Objekt ist nicht mehr gültig.
- 3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Parent

Eigenschaft: `Parent` als [Dialogs](#) (schreibgeschützt)

Beschreibung

Ruft das dem Objekt übergeordnete Objekt auf.

Fehler

- 3000 Das Objekt ist nicht mehr gültig.
- 3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ResolveEntities

Eigenschaft: `ResolveEntities` als Boolean

Beschreibung

Sollen alle Entities vor Beginn der Generierung aufgelöst werden? Wenn ja, wird ein Infoset erstellt.

Fehler

- 3000 Das Objekt ist nicht mehr gültig.
- 3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

TypeDetection

Eigenschaft: `TypeDetection` als [SPYTypeDetection](#)

Beschreibung

Definiert die Genauigkeit, mit der ein simpleTyp ermittelt wird.

Fehler

- 3000 Das Objekt ist nicht mehr gültig.
- 3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ValueList

Eigenschaft: `ValueList` als Integer

Beschreibung

Generiert nicht mehr als diese Anzahl von Enumeration-Facets pro Typ. Setzen Sie den Wert auf -1, um keine Einschränkung zu setzen.

Fehler

- 3000 Das Objekt ist nicht mehr gültig.
- 3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

4.3.11 ElementList

Siehe auch

Eigenschaften

[Count](#)

[Item](#)

Methoden

[RemoveElement](#)

Beschreibung

Elementlisten werden beim Export und Import von Daten für verschiedene Zwecke verwendet. Je nach Verwendungszweck werden unterschiedliche Eigenschaften von [ElementListItem](#) verwendet.

Die Methode kann die folgenden Eigenschaften enthalten

- eine Liste von Tabellennamen, die durch einen Aufruf von [Application.GetDatabaseTables](#) zurückgegeben werden.
- eine Liste von Feldnamen, die durch einen Aufruf von [Application.GetDatabaseImportElementList](#) oder [Application.GetTextImportElementList](#) zurückgegeben werden,
- eine Feldnamenfilterliste, die in [Application.ImportFromDatabase](#) und [Application.ImportFromText](#) verwendet wird,
- eine Liste von Tabellennamen und der Anzahl ihrer Zeilen und Spalten, die durch Aufrufen von [GetExportElementList](#) zurückgegeben werden oder
- eine in [Document.ExportToDatabase](#) und [Document.ExportToText](#) verwendete Feldnamenfilterliste.

Count

Siehe auch

Eigenschaft: [Count](#) als long (schreibgeschützt)

Beschreibung

Anzahl der Elemente in dieser Sammlung.

Item

Siehe auch

Methode: [Item](#)([n](#) als long) als [ElementListItem](#)

Beschreibung

Ruft das Element mit dem Index [n](#) aus dieser Sammlung ab. Das erste Element hat den Index 1.

RemoveElement

Siehe auch

Method: [RemoveElement](#)([Index](#) als long)

Beschreibung

`RemoveElement` entfernt den Elementindex aus der Sammlung. Das erste Element hat den Index 1.

4.3.12 ElementListItem

Siehe auch

Eigenschaften

[Name](#)

[ElementKind](#)

[FieldCount](#)

[RecordCount](#)

Beschreibung

Ein Element in einer [ElementList](#). Die Verwendung seiner Eigenschaften ist vom Verwendungszweck der Elementliste abhängig. Nähere Informationen dazu siehe [ElementList](#).

ElementKind

Siehe auch

Eigenschaft: [ElementKind](#) als [SPYXMLDataKind](#)

Beschreibung

Definiert, ob ein Feld als XML-Element (Datenwert von `spyXMLDataElement`) oder als Attribut (Datenwert von `spyXMLDataAttr`) importiert werden soll.

FieldCount

Siehe auch

Eigenschaft: [FieldCount](#) als long (schreibgeschützt)

Beschreibung

Anzahl der Felder (d.h. Spalten) in der durch dieses Element beschriebenen Tabelle. Diese Eigenschaft ist nur nach einem Aufruf von [DocumentGetExportElementList](#) gültig.

Name

Siehe auch

Eigenschaft: [Name](#) als String (schreibgeschützt)

Beschreibung

Name des Elements. Dies ist je nach Verwendungszweck der Elementliste entweder der Name einer Tabelle oder eines Felds.

RecordCount

Siehe auch

Eigenschaft: [RecordCount](#) als long (schreibgeschützt)

Beschreibung

Anzahl der Datensätze (d.h. Zeilen) in der durch dieses Element beschriebenen Tabelle. Diese Eigenschaft ist nur nach einem Aufruf von [DocumentGetExportElementList](#) gültig.

4.3.13 ExportSettings

Siehe auch

Eigenschaften

[ElementList](#)

[EntitiesToText](#)

[ExportAllElements](#)

[SubLevelLimit](#)

[FromAttributes](#)

[FromSingleSubElements](#)

[FromTextValues](#)

[CreateKeys](#)

[IndependentPrimaryKey](#)

[Namespace](#)

[ExportCompleteXML](#)

[StartFromElement](#)

Beschreibung

`ExportSettings` enthält Optionen, die beim Export von XML-Daten in eine Datenbank oder Textdatei verwendet werden. Eine allgemeine Übersicht dazu finden Sie unter [Import und Export von Daten](#).

CreateKeys

Siehe auch

Eigenschaft: `CreateKeys` als Boolean

Beschreibung

Diese Eigenschaft aktiviert bzw. deaktiviert die Erstellung von Schlüsseln (d.h. Primär- und Sekundärschlüssel). Die Standardeinstellung ist "True".

ElementList

Siehe auch

Eigenschaft: `ElementList` als [ElementList](#)

Beschreibung

Standardeinstellung ist eine leere Liste. In dieser Elementliste wird definiert, welche Felder exportiert werden. Um die Liste der verfügbaren Felder abzurufen, verwenden Sie [Document.GetExportElementList](#). Sie können Spalten vom Export ausnehmen, indem Sie Elemente mit [ElementList.RemoveElement](#) aus dieser Liste entfernen, bevor Sie sie an [Document.ExportToDatabase](#) oder [Document.ExportToText](#) übergeben.

EntitiesToText

Siehe auch

Eigenschaft: [EntitiesToText](#) als Boolean

Beschreibung

Definiert, ob XML-Entities in Text konvertiert oder unverändert belassen werden sollen. Die Standardeinstellung ist True.

ExportAllElements

Siehe auch

Eigenschaft: [ExportAllElements](#) als Boolean

Beschreibung

Wenn diese Eigenschaft auf `true` gesetzt ist, werden alle Elemente im Dokument exportiert. Wenn sie auf `false` gesetzt ist, wird die Anzahl der zu exportierenden Unterebenen mittels [ExportSettings.SubLevelLimit](#) eingeschränkt. Die Standardeinstellung ist `true`.

ExportCompleteXML

Siehe auch

Eigenschaft: [ExportCompleteXML](#) als Boolean

Beschreibung

Definiert, ob die komplette XML-Datei oder nur das durch [StartFromElement](#) definierte Element und seine untergeordneten Elemente exportiert werden sollen. Standardeinstellung ist True.

FromAttributes

Siehe auch

Eigenschaft: [FromAttributes](#) als Boolean

Beschreibung

Setzen Sie [FromAttributes](#) auf "false", wenn keine Exportdaten anhand von Attributen erstellt werden sollen. Die Standardeinstellung ist True.

FromSingleSubElements

Siehe auch

Eigenschaft: [FromSingleSubElements](#) als Boolean

Beschreibung

Setzen Sie [FromSingleSubElements](#) auf "false", wenn keine Exportdaten von Elementen erstellt werden sollen. Die Standardeinstellung ist True.

FromTextValues

Siehe auch

Eigenschaft: [FromTextValues](#) als Boolean

Beschreibung

Setzen Sie [FromTextValues](#) auf "false", wenn keine Exportdaten anhand von Textwerten erstellt werden sollen. Die Standardeinstellung ist "True".

IndependentPrimaryKey

Siehe auch

Eigenschaft: [IndependentPrimaryKey](#) als Boolean

Beschreibung

Aktiviert bzw. deaktiviert die Erstellung unabhängiger Primärschlüsselzähler für alle Elemente. Wenn [ExportSettings.CreateKeys](#) "False" ist, wird diese Eigenschaft ignoriert. Die Standardeinstellung ist "True".

Namespace

Siehe auch

Eigenschaft: [Namespace](#) als [SPYExportNamespace](#)

Beschreibung

Mit der Standardeinstellung werden alle Namespace-Präfixe aus den Elementnamen entfernt. In einigen Datenbankformaten ist der Doppelpunkt kein zulässiges Zeichen. Die Standardeinstellung ist `spyNoNamespace`.

StartFromElement

Siehe auch

Eigenschaft: [StartFromElement](#) als String

Beschreibung

Definiert das Startelement für den Export. Diese Eigenschaft wird nur berücksichtigt, wenn [ExportCompleteXML](#) "false" ist.

SubLevelLimit

Siehe auch

Eigenschaft: [SubLevelLimit](#) als Integer

Beschreibung

Definiert die Anzahl der Unterebenen, die exportiert werden soll. Diese Eigenschaft wird ignoriert, wenn [ExportSettings.ExportAllElements](#) `true` ist.

4.3.14 FileSelectionDlg

Siehe auch

Eigenschaften und Methoden

Standard-Automation-Eigenschaften

[Accepted](#)

[Parent](#)

Dialogfeldeigenschaften

[FullName](#)

Annahme oder Abbruch der Aktion, die das Event verursacht hat.

[DialogAction](#)

Beschreibung

Mit Hilfe dieses Dialogfeldobjekts können Sie Informationen über ein Event empfangen und Daten an den Event Handler auf dieselbe Art zurückgeben, wie über ein Benutzer-Dialogfeld. Mit Hilfe von [FileSelectionDlg.FullName](#) können Sie den Dateipfad auswählen oder ändern und mit Hilfe der Eigenschaft [FileSelectionDlg.DialogAction](#) können Sie die Aktion, die das Event verursacht hat, abbrechen oder akzeptieren.

Application

Siehe auch

Eigenschaft: [Application](#) als [Application](#) (schreibgeschützt)

Beschreibung

Ruft das Authentic Desktop application-Objekt auf.

Fehler

- 2400 Das Objekt ist nicht mehr gültig.
- 2401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

DialogAction

Eigenschaft: [DialogAction](#) als [SPYDialogAction](#)

Beschreibung

Wenn Sie möchten, dass Ihr Script die Dateiauswahl ohne Interaktion durch den Benutzer durchführt, simulieren Sie die Interaktion durch den Benutzer, indem Sie die Eigenschaft entweder auf `spyDialogOK(0)` oder `spyDialogCancel(1)` setzen. Damit Ihr Script die Standardwerte einsetzt, der Benutzer das Dialogfeld aber sehen und darauf reagieren kann, verwenden Sie den Wert `spyDialogUserInput(2)`. Wenn Sie in einem Event Handler ein `FileSelectionDlg` Objekt empfangen, wird `spyDialogUserInput(2)` nicht unterstützt und als `spyDialogOK(0)` interpretiert.

Fehler

- 2400 Das Objekt ist nicht mehr gültig.
- 2401 Ungültiger Wert für die Dialogfeldaktion oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.

FullName

Eigenschaft: `FullName` als String

Beschreibung

Ruft den vollständigen Pfad der Datei auf, die durch das Dialogfeld ausgewählt wird. Bei den meisten Events, die ein `FileSelectionDlg` Objekt an Sie übergeben, können Sie diesen Wert ändern und dadurch die Aktion, die das Event verursacht hat (also Laden oder Speichern unter einem anderen Pfad), beeinflussen.

Fehler

- 2400 Das Objekt ist nicht mehr gültig.
- 2401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Parent

Siehe auch

Eigenschaft: `Parent` als [Dialogs](#) (schreibgeschützt)

Beschreibung

Ruft das übergeordnete Objekt des Objekts auf.

Fehler

- 2400 Das Objekt ist nicht mehr gültig.
- 2401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

4.3.15 FindInFilesDlg

Siehe auch

Eigenschaften und Methoden

Standard-Automation-Eigenschaften

[Application](#)

[Parent](#)

[Find](#)

[RegularExpression](#)

[Replace](#)

[DoReplace](#)

[ReplaceOnDisk](#)

[MatchWholeWord](#)

[MatchCase](#)

[SearchLocation](#)

[StartFolder](#)

[IncludeSubfolders](#)

[SearchInProjectFilesDoExternal](#)

[FileExtension](#)

[AdvancedXMLSearch](#)

[XMLElementNames](#)

[XMLElementContents](#)

[XMLAttributeNames](#)

[XMLAttributeContents](#)

[XMLComments](#)

[XMLCDATA](#)

[XMLPI](#)

[XMLRest](#)

[ShowResult](#)

Beschreibung

Mit Hilfe dieses Objekts können Sie die Suche (oder die Ersetzung) von Strings in Dateien konfigurieren. Die Methode [FindInFiles](#) erwartet einen [FindInFilesDlg](#) als Parameter.

AdvancedXMLSearch

Eigenschaft: [AdvancedXMLSearch](#) als Boolean

Beschreibung

Legt fest, ob die XML-Sucheigenschaften ([XMLElementNames](#), [XMLElementContents](#), [XMLAttributeNames](#), [XMLAttributeContents](#), [XMLComments](#), [XMLCDATA](#), [XMLPI](#) und [XMLRest](#)) berücksichtigt werden. Die Standardeinstellung ist "false".

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Application

Eigenschaft: [Application](#) als [Application](#) (schreibgeschützt)

Beschreibung

Ruft das Authentic Desktop application-Objekt auf.

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

DoReplace

Eigenschaft: [DoReplace](#) als Boolean

Beschreibung

Legt fest, ob der gefundene String durch den in [Replace](#) definierten String ersetzt werden soll. Die Standardeinstellung ist "false"

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

FileExtension

Eigenschaft: [FileExtension](#) als String

Beschreibung

Definiert den Dateifilter der Dateien, die bei der Suche berücksichtigt werden sollen. Mehrere Dateifilter müssen durch ein Semikolon voneinander getrennt werden (z.B. *.xml;*.dtd;a*.xsd). Verwenden Sie die Platzhalterzeichen * und ? um den Dateifilter zu definieren.

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Find

Eigenschaft: [Find](#) als String

Beschreibung

Definiert den zu suchenden String.

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IncludeSubfolders

Eigenschaft: [IncludeSubfolders](#) als Boolean

Beschreibung

Legt fest, ob auch Unterordner durchsucht werden sollen. Die Standardeinstellung ist "True".

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

MatchCase

Eigenschaft: [MatchCase](#) als Boolean

Beschreibung

Legt fest, ob die Groß- und Kleinschreibung bei der Suche beachtet werden soll. Die Standardeinstellung ist "True".

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

MatchWholeWord

Eigenschaft: [MatchWholeWord](#) als Boolean

Beschreibung

Legt fest, ob bei der Suche nur ganze Wörter oder Teile davon gesucht werden sollen. Die Standardeinstellung ist "false".

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Parent

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

Beschreibung

Ruft das übergeordnete Objekt des Objekts auf.

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

RegularExpression

Eigenschaft: [RegularExpression](#) als Boolean

Beschreibung

Definiert, ob [Find](#) eine Regular Expression enthält. Die Standardeinstellung ist "false".

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Replace

Eigenschaft: [Replace](#) als String

Beschreibung

Definiert den Ersetzungsstring. Der gefundene String wird nur ersetzt, wenn [DoReplace](#) auf "true" gesetzt ist.

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ReplaceOnDisk

Eigenschaft: [ReplaceOnDisk](#) als Boolean

Beschreibung

Legt fest, ob die Ersetzung direkt auf der Festplatte vorgenommen wird. Die geänderte Datei wird nicht geöffnet. Die Standardeinstellung ist "false".

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

SearchInProjectFilesDoExternal

Eigenschaft: [SearchInProjectFilesDoExternal](#) als Boolean

Beschreibung

Definiert, ob bei einer Projektsuche die externen Ordner im offenen Projektdurchsucht werden sollen. Die Standardeinstellung ist "false".

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

SearchLocation

Eigenschaft: [SearchLocation](#) als [SPYFindInFilesSearchLocation](#)

Beschreibung

Definiert den Ordner für die Suche. Die Standardeinstellung ist spyFindInFiles_Documents.

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowResult

Eigenschaft: [ShowResult](#) als Boolean

Beschreibung

Definiert, ob das Ergebnis im Ausgabefenster "In Dateien suchen" angezeigt werden soll. Die Standardeinstellung ist "false".

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

StartFolder

Eigenschaft: [StartFolder](#) als String

Beschreibung

Definiert den Ordner, in dem die Suche auf der Festplatte beginnen soll.

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

XMLAttributeContents

Eigenschaft: [XMLAttributeContents](#) als Boolean

Beschreibung

Legt fest ob Attributinhalt durchsucht werden soll, wenn [AdvancedXMLSearch](#) true ist. Die Standardeinstellung ist "True".

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

XMLAttributeNames

Eigenschaft: [XMLAttributeNames](#) als Boolean

Beschreibung

Definiert, ob Attributnamen durchsucht werden sollen, wenn [AdvancedXMLSearch](#) true ist. Die Standardeinstellung ist "True".

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

XMLCDATA

Eigenschaft: [XMLCDATA](#) als Boolean

Beschreibung

Legt fest, ob CDATA Tags durchsucht werden sollen, wenn [AdvancedXMLSearch](#) true ist. Die Standardeinstellung ist "True".

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

XMLComments

Eigenschaft: [XMLComments](#) als Boolean

Beschreibung

Legt fest, ob Kommentare durchsucht werden sollen, wenn [AdvancedXMLSearch](#) true ist. Die Standardeinstellung ist "True".

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

XMLElementContents

Eigenschaft: [XMLElementContents](#) als Boolean

Beschreibung

Legt fest, ob der Elementinhalt durchsucht werden soll, wenn [AdvancedXMLSearch](#) true ist. Die Standardeinstellung ist "True".

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

XMLElementNames

Eigenschaft: [XMLElementNames](#) als Boolean

Beschreibung

Legt fest, ob Elementnamen durchsucht werden sollen, wenn [AdvancedXMLSearch](#) true ist. Die Standardeinstellung ist "True".

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

XMLPI

Eigenschaft: [XMLPI](#) als Boolean

Beschreibung

Legt fest, ob XML-Verarbeitungsanweisungen durchsucht werden sollen, wenn [AdvancedXMLSearch](#) true ist. Die Standardeinstellung ist "True".

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

XMLRest

Eigenschaft: [XMLRest](#) als Boolean

Beschreibung

Legt fest, ob der Rest der XML-Datei (der nicht durch die anderen XML-Sucheigenschaften berücksichtigt wird) durchsucht werden soll, wenn [AdvancedXMLSearch](#) true ist. Die Standardeinstellung ist "True".

Fehler

- 3500 Das Objekt ist nicht mehr gültig.
- 3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

4.3.16 FindInFilesResult

Siehe auch

Eigenschaften und Methoden

Standard-Automation-Eigenschaften

[Application](#)

[Parent](#)

[Count](#)

[Item](#)

[Path](#)

[Document](#)

Beschreibung

Dieses Objekt repräsentiert eine Datei, die den Suchkriterien entsprochen hat. Es enthält eine Liste von [FindInFilesResultMatch](#) Objekten, die die Suchposition beschreiben.

Application

Eigenschaft: [Application](#) als [Application](#) (schreibgeschützt)

Beschreibung

Ruft das Authentic Desktop application-Objekt auf.

Fehler

- 3700 Das Objekt ist nicht mehr gültig.
- 3701 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Count

Eigenschaft: [Count](#) als long (schreibgeschützt)

Beschreibung

Anzahl der Elemente in dieser Sammlung.

Document

Eigenschaft: [Path](#) als [Document](#) (schreibgeschützt)

Beschreibung

Diese Eigenschaft gibt das [Document](#)-Objekt zurück, wenn die gefundene Datei bereits in XMLSpy geöffnet ist.

Fehler

- 3700 Das Objekt ist nicht mehr gültig.
- 3701 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Item

Methode: `Item(n` als long) als [FindInFilesResultMatch](#)

Beschreibung

Ruft das Element mit dem Index `n` aus dieser Sammlung ab. Das erste Element hat den Index 1.

Parent

Eigenschaft: `Parent` als [FindInFilesResults](#) (schreibgeschützt)

Beschreibung

Ruft das dem Objekt übergeordnete Objekt auf.

Fehler

- 3700 Das Objekt ist nicht mehr gültig.
- 3701 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Path

Eigenschaft: `Path` als String (schreibgeschützt)

Beschreibung

Gibt den Pfad der Datei zurück, die den Suchkriterien entsprochen hat.

Fehler

- 3700 Das Objekt ist nicht mehr gültig.
- 3701 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

4.3.17 FindInFilesResultMatch

Siehe auch

Eigenschaften und Methoden

Standard-Automation-Eigenschaften

[Application](#)

[Parent](#)

[Line](#)

[Position](#)

[Length](#)

[LineText](#)

[Replaced](#)

Beschreibung

Enthält die genaue Position des gesuchten String in der Datei.

Application

Eigenschaft: [Application](#) als [Application](#) (schreibgeschützt)

Beschreibung

Ruft das Authentic Desktop application-Objekt auf.

Fehler

- 3800 Das Objekt ist nicht mehr gültig.
- 3801 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Length

Eigenschaft: [Length](#) als Long (schreibgeschützt)

Beschreibung

Gibt die Länge des gefundenen String zurück.

Fehler

- 3800 Das Objekt ist nicht mehr gültig.
- 3801 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Line

Eigenschaft: [Line](#) als Long (schreibgeschützt)

Beschreibung

Gibt die Zeilennummer des gefundenen Suchbegriffs zurück. Die Zeilennummerierung beginnt mit 0.

Fehler

- 3800 Das Objekt ist nicht mehr gültig.
- 3801 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

LineText

Eigenschaft: [LineText](#) als String (schreibgeschützt)

Beschreibung

Gibt den Text der Zeile zurück.

Fehler

- 3800 Das Objekt ist nicht mehr gültig.
- 3801 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Parent

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

Beschreibung

Ruft das dem Objekt übergeordnete Objekt auf.

Fehler

- 3800 Das Objekt ist nicht mehr gültig.
- 3801 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Position

Eigenschaft: [Position](#) als Long (schreibgeschützt)

Beschreibung

Gibt die Anfangsposition des gesuchten Begriffs in der Zeile zurück. Die Positionsnummerierung beginnt mit 0.

Fehler

- 3800 Das Objekt ist nicht mehr gültig.
- 3801 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Replaced

Eigenschaft: [Replaced](#) als Boolean (schreibgeschützt)

Beschreibung

True, wenn der gefundene String ersetzt wurde.

Fehler

- 3800 Das Objekt ist nicht mehr gültig.
- 3801 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

4.3.18 FindInFilesResults

Siehe auch

Eigenschaften und Methoden

Standard-Automation-Eigenschaften

[Application](#)

[Parent](#)

[Count](#)

[Item](#)

Beschreibung

Dies ist das Ergebnis der [FindInFiles](#) Methode. Es handelt sich hierbei um eine Liste von [FindInFilesResult](#) Objekten.

Application

Eigenschaft: [Application](#) als [Application](#) (schreibgeschützt)

Beschreibung

Ruft das Authentic Desktop application-Objekt auf.

Fehler

- 3600 Das Objekt ist nicht mehr gültig.
- 3601 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Count

Eigenschaft: [Count](#) als long (schreibgeschützt)

Beschreibung

Anzahl der Elemente in dieser Sammlung.

Item

Methode: [Item](#)([n](#) als long) als [FindInFilesResult](#)

Beschreibung

Ruft das Element mit dem Index [n](#) aus dieser Sammlung ab. Das erste Element hat den Index 1.

Parent

Eigenschaft: [Parent](#) as [Application](#) (schreibgeschützt)

Beschreibung

Ruft das dem Objekt übergeordnete Objekt auf.

Fehler

- 3600 Das Objekt ist nicht mehr gültig.
- 3601 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

4.3.19 GenerateSampleXMLDlg

Siehe auch

Eigenschaften und Methoden

Standard-Automation-Eigenschaften

[Application](#)

[Parent](#)

[NonMandatoryAttributes](#)

[RepeatCount](#)

[FillAttributesWithSampleData](#)

[FillElementsWithSampleData](#)

[ContentOfNillableElementsIsNonMandatory](#)

[TryToUseNonAbstractTypes](#)

[Optimization](#)

[SchemaOrDTDAssignment](#)

[LocalNameOfRootElement](#)

[NamespaceURIOfRootElement](#)

[OptionsDialogAction](#)

Nicht mehr unterstützte Eigenschaften

[NonMandatoryElements](#) - nicht mehr verwendet

[TakeFirstChoice](#) - nicht mehr verwendet

[FillWithSampleData](#) - nicht mehr verwendet

Beschreibung

Dient zum Setzen der Parameter für die Generierung von XML-Beispielinstanzen auf Basis eines W3C-Schemas oder einer DTD.

Application

Eigenschaft: [Application](#) als [Application](#) (schreibgeschützt)

Beschreibung

Ruft das Authentic Desktop application-Objekt auf.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Parent

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

Beschreibung

Ruft das dem Objekt übergeordnete Objekt auf.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

NonMandatoryAttributes

Eigenschaft: [NonMandatoryAttributes](#) als Boolean

Beschreibung

Wenn der Wert "true" ist, werden in der XML-Beispielinstanzdatei auch nicht obligatorische Attribute erstellt.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

NonMandatoryElements - obsolete

Eigenschaft: [NonMandatoryElements](#) als Boolean

Beschreibung

Diese Eigenschaft wird nicht mehr verwendet. Verwenden Sie stattdessen [Optimization](#).

Fehler

- 0001 Die Eigenschaft steht nicht mehr zur Verfügung.

TakeFirstChoice - obsolete

Eigenschaft: [TakeFirstChoice](#) als Boolean

Beschreibung

Diese Eigenschaft wird nicht mehr verwendet.

Fehler

- 0001 Die Eigenschaft steht nicht mehr zur Verfügung.

RepeatCount

Eigenschaft: [RepeatCount](#) als long

Beschreibung

Anzahl der Elemente, die für wiederholt vorkommende Typen erstellt werden soll.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

FillWithSampleData - obsolete

Eigenschaft: [FillWithSampleData](#) als Boolean

Beschreibung

Diese Eigenschaft wird nicht mehr verwendet. Verwenden Sie stattdessen [FillAttributesWithSampleData](#) und [FillElementsWithSampleData](#), instead.

Fehler

- 0001 Die Eigenschaft steht nicht mehr zur Verfügung.

FillElementsWithSampleData

Eigenschaft: [FillElementsWithSampleData](#) als Boolean

Beschreibung

Wenn der Wert "true" ist, haben Elemente Beispieldaten.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

FillAttributesWithSampleData

Eigenschaft: [FillAttributesWithSampleData](#) als Boolean

Beschreibung

Wenn der Wert "true" ist, haben Attribute Beispieldaten.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ContentOfNillableElementsIsNonMandatory

Eigenschaft: [ContentOfNillableElementsIsNonMandatory](#) als Boolean

Beschreibung

Wenn der Wert "true" ist, wird der Inhalt von Elementen die "nillable" sind, nicht als obligatorisch behandelt.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

TryToUseNonAbstractTypes

Eigenschaft: [TryToUseNonAbstractTypes](#) als Boolean

Beschreibung

Versucht bei "true" einen nicht abstrakten Typen für xsi:type zu verwenden, wenn das Element einen abstrakten Typen hat.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Optimization

Eigenschaft: [Optimization](#) als [SPYSampleXMLGenerationOptimization](#)

Beschreibung

Definiert, welche Elemente generiert werden.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

SchemaOrDTDAssignment

Eigenschaft: [SchemaOrDTDAssignment](#) als [SPYSampleXMLGenerationSchemaOrDTDAssignment](#)

Beschreibung

Definiert, wie eine Referenz auf das damit in Zusammenhang stehende Schema oder die DTD, welche sich im Dokument befindet, in eine XML-Beispieldatei umgewandelt wird.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

LocalNameOfRootElement

Eigenschaft: [LocalNameOfRootElement](#) als String

Beschreibung

Definiert den lokalen Namen des Root-Elements für die generierte XML-Beispieldatei.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

NamespaceURIOfRootElement

Eigenschaft: [NamespaceURIOfRootElement](#) als String

Beschreibung

Definiert die Namespace URI des Root-Elements für die generierte XML-Beispieldatei.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

OptionsDialogAction

Eigenschaft: [OptionsDialogAction](#) als [SPYDialogAction](#)

Beschreibung

Damit das Script die Standardwerte einsetzt und der Benutzer das Dialogfeld sieht, und dort Eingaben machen kann, setzen Sie die Eigenschaft auf den Wert `spyDialogUserInput(2)`. Wenn Ihr Script alle Optionen im Dialogfeld "Schemadokumentation" definieren soll, ohne dass eine Interaktion durch den Benutzer erforderlich ist, verwenden Sie `spyDialogOK(0)`. Der Standardwert ist `spyDialogOK`.

Fehler

- 2200 Das Objekt ist nicht mehr gültig.
- 2201 Zur Definition der Eigenschaft wurde ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

4.3.20 GridView

Siehe auch

Methoden

[Deselect](#)

[Select](#)

[SetFocus](#)

Eigenschaften

[CurrentFocus](#)

[IsVisible](#)

Beschreibung

GridView Class

Events

OnBeforeDrag

Siehe auch

Event: [OnBeforeDrag\(\)](#) als Boolean

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_BeforeDrag()  
    'On_BeforeStartEditing=False 'prohibit dragging  
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_BeforeDrag()  
{  
    //return false; 'prohibit dragging  
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugin.OnEvent (4, ...) //nEvent=4
```

Beschreibung

Dieses Event wird beim Versuch ausgelöst, ein XMLData-Element in der Grid-Ansicht mit der Maus zu verschieben. Geben Sie *false* zurück, um zu verhindern, dass das Datenelement an eine andere Position gezogen werden kann.

OnBeforeDrop

Siehe auch

Event: [OnBeforeDrop\(objXMLData](#) als [XMLData](#)) als Boolean

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_BeforeDrop(objXMLData)  
    'On_BeforeStartEditing=False 'prohibit dragging
```

```
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_BeforeDrop(objXMLData)
{
    //unfalse/*optional logging*/
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugin.OnEvent (5, ...) //nEvent=5
```

Beschreibung

This event gets fired on an attempt to drop a previously dragged XMLData element on the grid view. Return *false* to prevent the data element to be moved from its original position to the drop destination position.

Dieses Event wird beim Versuch ausgelöst, ein zuvor mit der Maus gezogenes XMLData-Element in der Grid-Ansicht an dieser Stelle abzulegen. Geben Sie *false* zurück, um zu verhindern, dass das Datenelement von seiner ursprünglichen Position an eine andere Position gezogen werden kann.

OnBeforeStartEditing

Siehe auch

Event: OnBeforeStartEditing(objXMLData als [XMLData](#), bEditingName als Boolean) als Boolean

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_BeforeStartEditing(objXMLData, bEditingName)
    'On_BeforeStartEditing=False/*optional logging*/
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_BeforeStartEditing(objXMLData, bEditingName)
{
    //unfalse/*optional logging*/
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugin.OnEvent (1, ...) //nEvent=1
```

Beschreibung

Dieses Event wird ausgelöst, bevor der Bearbeitungsmodus für eine Grid-Zelle aktiviert wird. Wenn der Parameter *bEditingName* "true" ist, wird der Namensteil des Elements bearbeitet, wenn der Wert "false" ist, wird der Werteteil bearbeitet.

OnEditingFinished

Siehe auch

Event: OnEditingFinished(objXMLData als [XMLData](#), bEditingName als Boolean)

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_EditingFinished(objXMLData, bEditingName)
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_EditingFinished(objXMLData, bEditingName)
{
}
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugin.OnEvent (2, ...) //nEventId=2
```

Beschreibung

Dieses Event wird ausgelöst, wenn der Bearbeitungsmodus einer Grid-Zelle verlassen wird. Der Parameter *bEditingName* definiert, ob der Namensteil des Elements bearbeitet wurde.

OnFocusChanged

Siehe auch

Event: *OnFocusChanged*(*objXMLData* als [XMLData](#), *bSetFocus* als Boolean, *bEditingName* als Boolean)

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_FocusChanged(objXMLData, bSetFocus, bEditingName)
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_FocusChanged(objXMLData, bSetFocus, bEditingName)
{
}
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugin.OnEvent (3, ...) //nEventId=3
```

Beschreibung

Dieses Event wird ausgelöst, wenn der Cursorfokus auf eine Grid-Zelle verschoben wird oder diese nicht mehr im Fokus ist. Wenn der Parameter *bEditingName* "true" ist, hat sich der Fokus auf dem Namensteil des Grid-Elements geändert. Anderfalls hat sich der Fokus auf dem Werteteil geändert.

CurrentFocus

Siehe auch

Eigenschaft: *CurrentFocus* als [XMLData](#)

Beschreibung

Enthält das XML-Element mit dem aktuellen Fokus. Diese Eigenschaft ist schreibgeschützt.

Deselect

Siehe auch

Methode: `Deselect(pData als XMLData)`

Beschreibung

Hebt die Auswahl des Elements `pData` in der Grid-Ansicht auf.

IsVisible

Siehe auch

Eigenschaft: `IsVisible` als Boolean

Beschreibung

True, wenn die Grid-Ansicht die aktive Ansicht des Dokuments ist. Diese Eigenschaft ist schreibgeschützt.

Select

Siehe auch

Methode: `Select (pData als XMLData)`

Beschreibung

Wählt das XML-Element `pData` in der Grid-Ansicht aus.

SetFocus

Siehe auch

Methode: `SetFocus (pFocusData als XMLData)`

Beschreibung

Setzt den Fokus auf das Element `pFocusData` in der Grid-Ansicht.

4.3.21 SchemaDocumentationDlg

Siehe auch

Eigenschaften und Methoden

Standard-Automation-Eigenschaften

[Application](#)

[Parent](#)

Eigenschaften bzgl. Interaktion und Sichtbarkeit

[OutputFile](#)

[OutputFileDialogAction](#)

[OptionsDialogAction](#)

[ShowProgressBar](#)

[ShowResult](#)

Optionen und Methoden für die Dokumentgenerierung

[OutputFormat](#)

[EmbedDiagrams](#)

[DiagramFormat](#)

[MultipleOutputFiles](#)

[IncludeAll](#)

[IncludeIndex](#)

[IncludeGlobalAttributes](#)

[IncludeGlobalElements](#)

[IncludeLocalAttributes](#)

[IncludeLocalElements](#)

[IncludeGroups](#)

[IncludeComplexTypes](#)

[IncludeSimpleTypes](#)

[IncludeAttributeGroups](#)

[IncludeRedefines](#)

[IncludeReferencedSchemas](#)

[AllDetails](#)

[ShowDiagram](#)

[ShowNamespace](#)

[ShowType](#)

[ShowChildren](#)

[ShowUsedBy](#)

[ShowProperties](#)

[ShowSingleFacets](#)

[ShowPatterns](#)

[ShowEnumerations](#)

[ShowAttributes](#)

[ShowIdentityConstraints](#)

[ShowAnnotations](#)

[ShowSourceCode](#)

Beschreibung

Dieses Objekt kombiniert alle Optionen für die Schemadokumentgenerierung, die auch über die Dialogfelder der Benutzeroberfläche von Authentic Desktop verfügbar sind. Die Dokumentgenerierungsoptionen werden mit den Werten initialisiert, die bei der letzten Generierung von Schemadokumentation verwendet wurden. Vor Verwendung des Objekts müssen Sie jedoch die Eigenschaft [OutputFile](#) auf einen gültigen Dateipfad setzen.

Verwenden Sie [OptionsDialogAction](#), [OutputFileDialogAction](#) und

[ShowProgressBar](#) um den Grad der gewünschten Benutzerinteraktion festzulegen. Mit Hilfe von [i_bDetails](#) und [i_bDetailsOn](#) können Sie ganze Optionsgruppen auf einmal setzen. Durch Definition der einzelnen Eigenschaften können Sie die Optionen einzeln definieren.

AllDetails

Siehe auch

Methode: `AllDetails` (`i_bDetailsOn` als `Boolean`)

Beschreibung

Mit Hilfe dieser Methode können Sie alle Detailoptionen aktivieren bzw. deaktivieren.

Fehler

2900 Das Objekt ist nicht mehr gültig.

Application

Siehe auch

Eigenschaft: `Application` als `Application` (schreibgeschützt)

Beschreibung

Ruft das Authentic Desktop application-Objekt auf.

Fehler

2900 Das Objekt ist nicht mehr gültig.

2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

DiagramFormat

Siehe auch

Eigenschaft: `DiagramFormat` als `SPYImageKind`

Beschreibung

Setzen Sie diese Eigenschaft auf `ie` , um den Typ des Diagrammbilds zu definieren. Diese Eigenschaft steht für HTML-Dokumentation nicht zur Verfügung. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Verwendung der Eigenschaft ist PNG.

Fehler

2900 Das Objekt ist nicht mehr gültig.

2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

EmbedDiagrams

Siehe auch

Eigenschaft: `EmbedDiagrams` als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um das Diagramm in das generierte Dokument einzubetten. Diese Eigenschaft steht für HTML-Dokumentation nicht zur Verfügung. Die Eigenschaft wird mit den Werten initialisiert, die beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurden. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IncludeAll

Siehe auch

Methode: `IncludeAll` (`i_bInclude` als `Boolean`)

Beschreibung

Verwenden Sie diese Methode um alle Include-Optionen auszuwählen, bzw. die Auswahl für alle aufzuheben.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.

IncludeAttributeGroups

Siehe auch

Eigenschaft: `IncludeAttributeGroups` als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um Attributgruppen in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IncludeComplexTypes

Siehe auch

Eigenschaft: `IncludeComplexTypes` als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um complexTypes in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IncludeGlobalAttributes**Siehe auch**

Eigenschaft: [IncludeGlobalAttributes](#) als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um globale Attribute in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IncludeGlobalElements**Siehe auch**

Eigenschaft: [IncludeGlobalElements](#) als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um globale Elemente in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IncludeGroups**Siehe auch**

Eigenschaft: [IncludeGroups](#) als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um Gruppen in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IncludeIndex**Siehe auch**

Eigenschaft: `IncludeIndex` als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um einen Index in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IncludeLocalAttributes**Siehe auch**

Eigenschaft: `IncludeLocalAttributes` als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um lokale Attribute in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IncludeLocalElements**Siehe auch**

Eigenschaft: `IncludeLocalElements` als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um lokale Elemente in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IncludeRedefines

Siehe auch

Eigenschaft: `IncludeRedefines` als Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um Redefines in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IncludeReferencedSchemas

Siehe auch

Eigenschaft: `IncludeReferencedSchemas` als Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um referenzierte Schemas in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IncludeSimpleTypes

Siehe auch

Eigenschaft: `IncludeSimpleTypes` als Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um simpleTypes in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

MultipleOutputFiles

Siehe auch

Eigenschaft: `MultipleOutputFiles` als Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um die Dokumentationsdateien aufzuteilen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `false`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Es wurde zum Definieren der Eigenschaft ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

OptionsDialogAction

Siehe auch

Eigenschaft: [OptionsDialogAction](#) als [SPYDialogAction](#)

Beschreibung

Damit Ihr Script die Standardwerte einsetzt und der Benutzer das Dialogfeld sieht, bzw. dort Einstellungen vornehmen kann, setzen Sie diese Eigenschaft auf den Wert `spyDialogUserInput(2)`. Wenn Ihr Script alle Optionen im Dialogfeld "Schemadokumentation" definieren soll, ohne dass eine Interaktion mit dem Benutzer stattfindet, verwenden Sie `spyDialogOK(0)`. Die Standardeinstellung ist `spyDialogOK`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Es wurde zum Definieren der Eigenschaft ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

OutputFile

Siehe auch

Eigenschaft: [OutputFile](#) als `String`

Beschreibung

Der vollständige Pfad und Name der Datei, die die generierte Dokumentation enthalten soll. Bei einer HTML-Ausgabe werden zusätzliche '.png' Dateien auf Basis dieses Dateinamens generiert. Der Standardwert für diese Eigenschaft ist ein leerer String, der ersetzt werden muss, bevor Sie dieses Objekt in einem Aufruf von [DocumentGenerateSchemaDocumentation](#) verwenden.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

OutputFileDialogAction

Siehe auch

Eigenschaft: [OutputFileDialogAction](#) als [SPYDialogAction](#)

Beschreibung

Damit der Benutzer die Ausgabedatei in einem Dialogfeld zur Dateiauswahl auswählen kann, setzen Sie diese Eigenschaft auf `spyDialogUserInput(2)`. Wenn der in [OutputFile](#) gespeicherte Wert verwendet werden soll und keine Interaktion mit dem Benutzer vorgesehen ist, verwenden Sie `spyDialogOK(0)`. Der Standardwert ist `spyDialogOK`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Es wurde zum Definieren der Eigenschaft ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

OutputFormat

Siehe auch

Eigenschaft: `OutputFormat` als [SPYSchemaDocumentationFormat](#)

Beschreibung

Definiert, welche Art von Dokumentation generiert werden soll: HTML (Wert=0), MS-Word (Wert=1) oder RTF (Wert=2). Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist HTML.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Es wurde zum Definieren der Eigenschaft ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Parent

Siehe auch

Eigenschaft: `Parent` als [Dialogs](#) (schreibgeschützt)

Beschreibung

Ruft das dem Objekt übergeordnete Objekt auf.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowAnnotations

Siehe auch

Eigenschaft: `ShowAnnotations` als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um die Annotationen zu einer Typdefinition in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowAttributes

Siehe auch

Eigenschaft: `ShowAttributes` als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um die Typdefinitionsattribute in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowChildren

Siehe auch

Eigenschaft: `ShowChildren` als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um die untergeordneten Elemente einer Typdefinition als Links in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowDiagram

Siehe auch

Eigenschaft: `ShowDiagram` als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um die Typdefinitionen als Diagramme in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowEnumerations

Siehe auch

Eigenschaft: `ShowEnumerations` als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um die in einer Typdefinition enthaltenen Enumerationen in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowIdentityConstraints

Siehe auch

Eigenschaft: `ShowIdentityConstraints` als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um die Identity Constraints einer Typdefinition in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowNamespace

Siehe auch

Eigenschaft: `ShowNamespace` als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um den Namespace der Typdefinitionen in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowPatterns

Siehe auch

Eigenschaft: [ShowPatterns](#) als Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um die Patterns einer Typdefinition der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowProgressBar

Siehe auch

Eigenschaft: [ShowProgressBar](#) als Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um das Fenster für die Dokumentgenerierung sichtbar zu machen. Verwenden Sie `false`, um es auszublenden. Der Standardwert ist `false`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowProperties

Siehe auch

Eigenschaft: [ShowProperties](#) als Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um die Eigenschaften der Typdefinition in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowResult

Siehe auch

Eigenschaft: [ShowResult](#) als Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um das erzeugte Dokument automatisch zu öffnen, wenn die Generierung erfolgreich abgeschlossen wurde. HTML-Dokumentation wird in Authentic Desktop geöffnet. Um Word-Dokumentation anzuzeigen, wird MS-Word gestartet. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowSingleFacets**Siehe auch**

Eigenschaft: [ShowSingleFacets](#) als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um die Facets einer Typdefinition in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowSourceCode**Siehe auch**

Eigenschaft: [ShowSourceCode](#) als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um den XML-Quellcode für die Typdefinitionen in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowType**Siehe auch**

Eigenschaft: [ShowType](#) als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um den Typ von Typdefinitionen in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim

letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowUsedBy**Siehe auch**

Eigenschaft: [ShowUsedBy](#) als Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um die used-by-Beziehung für Typdefinitionen in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateSchemaDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

- 2900 Das Objekt ist nicht mehr gültig.
- 2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

4.3.22 SpyProject

Siehe auch

Methoden

[CloseProject](#)

[SaveProject](#)

[SaveProjectAs](#)

Eigenschaften

[RootItems](#)

[ProjectFile](#)

Beschreibung

`SpyProject` Class

CloseProject

Siehe auch

Deklaration: `CloseProject`(*bDiscardChanges* als Boolean, *bCloseFiles* als Boolean, *bDialog* als Boolean)

Parameter

bDiscardChanges

Setzen Sie *bDiscardChanges* auf FALSE, wenn die Änderungen in den offenen Projektdateien und im Projekt gespeichert werden sollen.

bCloseFiles

Setzen Sie *bCloseFiles* auf TRUE, um alle offenen Projektdateien zu schließen.

bDialog

Zeigt die Dialogfelder für die Benutzereingabe an.

Beschreibung

`CloseProject` schließt das aktuelle Projekt.

ProjectFile

Siehe auch

Deklaration: `ProjectFile` als String

Beschreibung

Pfad und Dateiname des Projekts.

RootItems

Siehe auch

Deklaration: `RootItems` als [SpyProjectItems](#)

Beschreibung

Root-Ebene der Sammlung der Projektelemente.

SaveProject

Siehe auch

Deklaration: `SaveProject`

Beschreibung

`SaveProject` speichert das aktuelle Projekt.

SaveProjectAs

Siehe auch

Deklaration: `SaveProjectAs` (`strPath` als String, `bDialog` als Boolean)

Parameter

`strPath`

Der vollständige Pfad mit dem Dateinamen der neuen Projektdatei.

`bDialog`

Wenn `bDialog` TRUE ist, wird ein Datei-Dialogfeld angezeigt.

Beschreibung

`SaveProjectAs` speichert die Projektdaten unter einem neuen Pfad.

4.3.23 SpyProjectItem

Siehe auch

Methoden

[Open](#)

Eigenschaften

[ChildItems](#)

[ParentItem](#)

[FileExtensions](#)

[ItemType](#)

[Name](#)

[Path](#)

[ValidateWith](#)

[XMLForXSLTransformation](#)

[XSLForXMLTransformation](#)

[XSLTransformationFileExtension](#)

[XSLTransformationFolder](#)

Beschreibung

[SpyProjectItem](#) Class

ChildItems

Siehe auch

Deklaration: [ChildItems](#) als [SpyProjectItems](#)

Beschreibung

Wenn es sich bei dem Element um einen Ordner handelt, ist [ChildItems](#) die Sammlung des Ordnerinhalts.

FileExtensions

Siehe auch

Deklaration: [FileExtensions](#) als String

Beschreibung

Dient zum Definieren der Dateierweiterungen, wenn das Projektelement ein Ordner ist.

ItemType

Siehe auch

Deklaration: [ItemType](#) als [SPYProjectItemTypes](#)

Beschreibung

Die Eigenschaft ist schreibgeschützt.

Name

Siehe auch

Deklaration: [Name](#) als String

Beschreibung

Name des Projektelements. Die Eigenschaft ist schreibgeschützt.

Open

Siehe auch

Deklaration: [Open](#) als [Document](#)

Rückgabewert

Das Projektelement wird als Dokument geöffnet.

Beschreibung

Öffnet das Projektelement.

ParentItem

Siehe auch

Deklaration: [ParentItem](#) als [SpyProjectItem](#)

Beschreibung

Das dem aktuellen Projektelement übergeordnete Element. Kann NULL (nichts) sein, wenn das Projektelement ein Element der obersten Ebene ist.

Path

Siehe auch

Deklaration: [Path](#) als String

Beschreibung

Pfad des Projektelements. Diese Eigenschaft ist schreibgeschützt.

ValidateWith

Siehe auch

Deklaration: [ValidateWith](#) als String

Beschreibung

Dient zum Definieren des Schemas/der DTD für die Validierung.

XMLForXSLTransformation

Siehe auch

Deklaration: `XMLForXSLTransformation` als String

Beschreibung

Dient zum Definieren der XML-Datei für die XSL-Transformation.

XSLForXMLTransformation

Siehe auch

Deklaration: `XSLForXMLTransformation` als String

Beschreibung

Dient zum Definieren der XSL-Datei für die XML-Transformation.

XSLTransformationFileExtension

Siehe auch

Deklaration: `XSLTransformationFileExtension` als String

Beschreibung

Dient zum Definieren der Dateierweiterung für die Ausgabedateien der XSL-Transformation.

XSLTransformationFolder

Siehe auch

Deklaration: `XSLTransformationFolder` als String

Beschreibung

Dient zum Definieren des Zielordners für die Ausgabedateien der XSL-Transformation.

4.3.24 SpyProjectItems

Siehe auch

Methoden

[AddFile](#)
[AddFolder](#)
[AddURL](#)
[RemoveItem](#)

Eigenschaften

[Count](#)
[Item](#)

Beschreibung

SpyProjectItems Class

AddFile

Siehe auch

Deklaration: [AddFile](#) (*strPath* als String)

Parameters

strPath

Der vollständige Pfadname eines neuen Projektelements

Beschreibung

Die Methode fügt eine neue Datei zur Sammlung der Projektelemente hinzu.

AddFolder

Siehe auch

Deklaration: [AddFolder](#) (*strName* als String)

Parameter

strName

Name des neuen Ordners

Beschreibung

Die Methode [AddFolder](#) fügt einen Ordner mit dem Namen *strName* zur Sammlung der Projektelemente hinzu.

AddURL

Siehe auch

Deklaration: [AddURL](#) (*strURL* als String, *nURLType* als [SPYURLTypes](#), *strUser* als String, *strPassword* als String, *bSave* als Boolean)

Beschreibung

`stURL`

URL zum Öffnen als Dokument.

`nURLType`

Typ des zu öffnenden Dokuments. Setzen Sie die Wert zur automatischen Ermittlung auf -1.

`stUser`

Name des Benutzers, falls erforderlich. Kann leer bleiben.

`stPassword`

Passwort für die Authentifizierung. Kann leer bleiben.

`bSave`

Speichert die Benutzer- und Passwortinformationen.

Beschreibung

Die Methode fügt einen URL-Eintrag zur Projektsammlung hinzu.

Count

Siehe auch

Deklaration: `Count` als long

Beschreibung

Diese Eigenschaft ruft die Anzahl der Projektelemente in der Sammlung ab. Die Eigenschaft ist schreibgeschützt.

Item

Siehe auch

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

Beschreibung

Ruft das nte Element der Sammlung der Projektelemente auf. Das erste Element hat den Index 1.

RemoveItem

Siehe auch

Deklaration: `RemoveItem` (`pItem` als [SpyProjectItem](#))

Beschreibung

`RemoveItem` löscht das Element `pItem` aus der Sammlung der Projektelemente.

4.3.25 TextImportExportSettings

Siehe auch

Eigenschaften nur für den Import

[ImportFile](#)

Eigenschaften nur für den Export

[DestinationFolder](#)

[FileExtension](#)

[CommentIncluded](#)

[RemoveDelimiter](#)

[RemoveNewline](#)

Eigenschaften für Import und Export

[HeaderRow](#)

[FieldDelimiter](#)

[EnclosingCharacter](#)

[Encoding](#)

[EncodingByteOrder](#)

Beschreibung

`TextImportExportSettings`
haben.

enthält Optionen, die Textimport- und -exportfunktionen gemeinsam

CommentIncluded

Siehe auch

Eigenschaft: [CommentIncluded](#) als Boolean

Beschreibung

Diese Eigenschaft legt fest, ob zusätzliche Kommentare zur generierten Textdatei hinzugefügt werden sollen. Die Standardeinstellung ist true. Diese Eigenschaft wird nur beim Export von Textdateien verwendet.

DestinationFolder

Siehe auch

Eigenschaft: [DestinationFolder](#) als String

Beschreibung

Die Eigenschaft `DestinationFolder` legt den Ordner fest, in dem die gespeicherten Dateien beim Textexport abgelegt werden.

EnclosingCharacter

Siehe auch

Eigenschaft: [EnclosingCharacter](#) als [SPYTextEnclosing](#)

Beschreibung

Diese Eigenschaft definiert das Zeichen, das alle Feldwerte für den Import und Export umschließt. Der Standardwert ist [spyNoEnclosing](#).

Encoding

Siehe auch

Eigenschaft: [Encoding](#) als String

Beschreibung

Die Eigenschaft [Encoding](#) definiert die Zeichensatzcodierung für die zu importierenden und exportierenden Textdateien.

EncodingByteOrder

Siehe auch

Eigenschaft: [EncodingByteOrder](#) als [SPYEncodingByteOrder](#)

Beschreibung

Die Eigenschaft [EncodingByteOrder](#) definiert die Bytefolge für Unicode-Zeichen. Standardeinstellung ist [spyNONE](#).

FieldDelimiter

Siehe auch

Eigenschaft: [FieldDelimiter](#) als [SPYTextDelimiters](#)

Beschreibung

Die Eigenschaft [FieldDelimiter](#) definiert die Trennzeichen zwischen den Feldern beim Import und Export. Standardeinstellung ist [spyTabulator](#).

FileExtension

Siehe auch

Eigenschaft: [FileExtension](#) als String

Beschreibung

Diese Eigenschaft definiert die Dateierweiterung für beim Textexport erstellte Dateien.

HeaderRow

Siehe auch

Eigenschaft: [HeaderRow](#) als Boolean

Beschreibung

Die Eigenschaft [HeaderRow](#) wird beim Export und Import verwendet. Setzen Sie beim Import [HeaderRow](#) auf true, wenn die erste Zeile der Textdatei die Namen der Spalten enthält. Setzen Sie [HeaderRow](#) beim Export auf true, wenn die erste Zeile der erstellten Textdateien die Namen der Spalten enthalten soll. Der Standardwert ist true.

ImportFile

Siehe auch

Eigenschaft: [ImportFile](#) als String

Beschreibung

Diese Eigenschaft dient dazu, die Textdatei für den Import zu definieren. Der String muss ein vollständig qualifizierter Pfad sein. Siehe auch [Import und Export](#).

RemoveDelimiter

Siehe auch

Eigenschaft: [RemoveDelimiter](#) als Boolean

Beschreibung

Die Eigenschaft [RemoveDelimiter](#) definiert, ob Zeichen im Text, die dem Trennzeichen entsprechen, entfernt werden sollen. Der Standardwert ist false. Diese Eigenschaft wird beim Export in Textdateien verwendet.

RemoveNewline

Siehe auch

Eigenschaft: [RemoveNewline](#) als Boolean

Beschreibung

Die Eigenschaft [RemoveNewline](#) definiert, ob Zeichen für den Beginn einer neuen Zeile im Text entfernt werden sollen. Die Standardeinstellung ist false. Die Eigenschaft wird nur beim Export in Textdateien verwendet.

4.3.26 TextView

Siehe auch

Eigenschaften und Methoden

[Application](#)
[Parent](#)

[LineFromPosition](#)
[PositionFromLine](#)
[LineLength](#)
[SetText](#)
[GetRangeText](#)
[ReplaceText](#)
[MoveCaret](#)
[GoToLineChar](#)
[SelectText](#)
[SelectionStart](#)
[SelectionEnd](#)
[Text](#)
[LineCount](#)
[Length](#)

Beschreibung

Events

OnBeforeShowSuggestions

Siehe auch

Event: [OnBeforeShowSuggestions\(\)](#) als `Boolean`

Beschreibung

Dieses Event wird ausgelöst, bevor das Vorschlagsfenster angezeigt wird. Die [Document](#)-Eigenschaft [Suggestions](#) enthält einen String Array, der dem Benutzer empfohlen wird. Die angezeigten Empfehlungen können vom Benutzer während des Events bearbeitet werden. Davor muss der Eigenschaft [Suggestions](#) ein leerer Array zugewiesen werden. Die beste Stelle dafür ist das [OnDocumentOpened](#) Event. Damit das Vorschlagsfenster nicht angezeigt wird, geben Sie `false` zurück. Um es weiter anzuzeigen, verwenden Sie den Wert `true`.

Beispiele

Unten finden Sie Beispiele dafür, wie dieses Event in Form eines Scripts geschrieben werden kann.

XMLSpy Skripting-Umgebung - VBScript:

```
Function On\_BeforeShowSuggestions()  
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On\_BeforeShowSuggestions()  
{  
}
```


XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (33, ...) //nEventId=33
```

OnChar**Siehe auch**

Event: **OnChar**(*nChar* als Long, *bExistSuggestions* als Boolean) als Boolean

Beschreibung

Dieses Event wird bei jedem Tastenanschlag ausgelöst. Der Parameter *nChar* ist die Taste, die gedrückt wurde und *bExistSuggestions* gibt darüber Auskunft ob ein von Authentic Desktop generiertes Vorschlagsfenster nach dem Drücken dieser Taste angezeigt wird. Die [Document](#) Eigenschaft [Suggestions](#) enthält einen String Array, der dem Benutzer empfohlen wird. Die angezeigten Empfehlungen können vom Benutzer während des Events bearbeitet werden. Davor muss der Eigenschaft [Suggestions](#) ein leerer Array zugewiesen werden. Die beste Stelle dafür ist das [OnDocumentOpened](#) Event. Damit das Vorschlagsfenster nicht angezeigt wird, geben Sie *false* zurück. Um es weiter anzuzeigen, verwenden Sie den Wert *true*. Es kann auch ein neues Vorschlagsfenster erstellt werden, wenn Authentic Desktop keines bereitstellt. Setzen Sie die [Document](#) Eigenschaft [Suggestions](#) auf einen String Array mit Ihren Empfehlungen und geben Sie *true* zurück. Dieses Event wird vor dem [OnBeforeShowSuggestions](#) Event ausgelöst. Wenn Sie festlegen, dass das Vorschlagsfenster nicht angezeigt werden soll, indem Sie den Wert auf *false* setzen, dann wird [OnBeforeShowSuggestions](#) nicht ausgelöst.

Beispiele

Unterhalb finden Sie Beispiele, wie ein Skript für dieses Event erstellt werden kann.

XMLSpy Skripting-Umgebung - VBScript:

```
Function On_Char(nChar, bExistSuggestions )
EndFunction
```

XMLSpy Skripting-Umgebung - JScript:

```
function On_Char(nChar, bExistSuggestions )
{
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent (35, ...) //nEventId=35
```

Application

Eigenschaft: [Application](#) als [Application](#) (schreibgeschützt)

Beschreibung

Ruft das Authentic Desktop application-Objekt auf.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

GetRangeText

Methode: `GetRangeText`(`nStart` als Long, `nEnd` als Long) als String

Beschreibung

Gibt den Text im angegebenen Bereich zurück.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

GoToLineChar

Methode: `GoToLineChar`(`nLine` als Long, `nChar` als Long)

Beschreibung

Verschiebt den Cursor an die angegebene Zeilen- und Zeichenposition.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Length

Eigenschaft: `Length` als Long

Beschreibung

Gibt die Anzahl der Zeichen des Dokuments zurück.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

LineCount

Eigenschaft: `LineCount` als Long

Beschreibung

Gibt die Anzahl der Zeilen im Dokument zurück.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

LineFromPosition

Methode: `LineFromPosition`(`nCharPos` als Long) als Long

Beschreibung

Gibt die Zeilennummer der Zeichenposition zurück.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

LineLength

Methode: `LineLength`(`nLine` als Long) als Long

Beschreibung

Gibt die Länge der Zeile zurück.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

MoveCaret

Methode: `MoveCaret`(`nDiff` als Long)

Beschreibung

Verschiebt den Cursor um `nDiff` Zeichen.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Parent

Eigenschaft: `Parent` als [Document](#) (schreibgeschützt)

Beschreibung

Ruft das dem Objekt übergeordnete Objekt auf.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

PositionFromLine

Methode: `PositionFromLine`(`nLine` als Long) als Long

Beschreibung

Gibt die Anfangsposition der Zeile zurück.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ReplaceText

Methode: `ReplaceText`(`nPosFrom` als Long, `nPosTill` als Long, `sText` als String)

Beschreibung

Ersetzt den Text im angegebenen Bereich.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

SelectionEnd

Eigenschaft: SelectionEnd als Long

Beschreibung

Gibt die Endposition der Textauswahl zurück bzw. definiert diese.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

SelectionStart

Eigenschaft: SelectionStart als Long

Beschreibung

Gibt die Anfangsposition der Textauswahl zurück bzw. definiert diese.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

SelectText

Methode: SelectText(nPosFrom als Long, nPosTill als Long)

Beschreibung

Wählt den Text im angegebenen Bereich aus.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

SelText

Eigenschaft: SelText als String

Beschreibung

Gibt den ausgewählten Text zurück bzw. definiert diesen.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Text

Eigenschaft: Text als String

Beschreibung

Gibt den Dokumenttext zurück bzw. definiert diesen.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

4.3.27 WSDLDocumentationDlg

Siehe auch

Eigenschaften und Methoden

Standard-Automation-Eigenschaften

[Application](#)

[Parent](#)

Eigenschaften zur Interaktion und Sichtbarkeit

[OutputFile](#)

[OutputFileDialogAction](#)

[OptionsDialogAction](#)

[ShowProgressBar](#)

[ShowResult](#)

Optionen und Methoden zur Dokumentgenerierung

[OutputFormat](#)

[EmbedDiagrams](#)

[DiagramFormat](#)

[MultipleOutputFiles](#)

[IncludeAll](#)

[IncludeBinding](#)

[IncludeImportedWSDLFiles](#)

[IncludeMessages](#)

[IncludeOverview](#)

[IncludePortType](#)

[IncludeService](#)

[IncludeTypes](#)

[AllDetails](#)

[ShowBindingDiagram](#)

[ShowExtensibility](#)

[ShowMessageParts](#)

[ShowPort](#)

[ShowPortTypeDiagram](#)

[ShowPortTypeOperations](#)

[ShowServiceDiagram](#)

[ShowSourceCode](#)

[ShowTypesDiagram](#)

[ShowUsedBy](#)

Beschreibung

Dieses Objekt kombiniert alle Optionen für die WSDL-Dokumentgenerierung, die über Dialogfelder der Benutzeroberfläche von Authentic Desktop zur Verfügung stehen. Die Dokumentgenerierungsoptionen werden mit den Werten initialisiert, die bei der letzten Generierung von WSLD-Dokumentation verwendet wurden. Bevor Sie das Objekt jedoch verwenden, müssen Sie die [OutputFile](#) Eigenschaft auf einen gültigen Dateipfad setzen. Mit Hilfe von [OptionsDialogAction](#), [OutputFileDialogAction](#) und [ShowProgressBar](#) können Sie den Grad der gewünschten Benutzerinteraktion definieren. Mit Hilfe von [IncludeAll](#) und [AllDetails](#) können Sie ganze Optionsgruppen auf einmal definieren. Oder Sie definieren die Eigenschaften einzeln, um die Feineinstellung vorzunehmen.

AllDetails

Siehe auch

Methode: `AllDetails` (`i_bDetailsOn` als `Boolean`)

Beschreibung

Mit Hilfe dieser Methode können Sie alle detaillierten Optionen aktivieren bzw. deaktivieren.

Fehler

3900 Das Objekt ist nicht mehr gültig.

Application

Siehe auch

Eigenschaft: `Application` als `Application` (schreibgeschützt)

Beschreibung

Ruft das Authentic Desktop application-Objekt auf.

Fehler

3900 Das Objekt ist nicht mehr gültig.

3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

DiagramFormat

Siehe auch

Eigenschaft: `DiagramFormat` als `SPYImageKind`

Beschreibung

Setzen Sie diese Eigenschaft auf `ice` , um den Diagrammbildtyp festzulegen. Diese Eigenschaft steht für HTML-Dokumentation nicht zur Verfügung. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von `DocumentGenerateWSDLDocumentation` verwendet wurde. Die Standardeinstellung für die erste Ausführung ist PNG.

Fehler

3900 Das Objekt ist nicht mehr gültig.

3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

EmbedDiagrams

Siehe auch

Eigenschaft: `EmbedDiagrams` als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `ice` , um das Diagramm in das generierte Dokument einzubetten. Diese Eigenschaft steht für HTML-Dokumentation nicht zur Verfügung. Die Eigenschaft wird mit den Werten initialisiert, die beim letzten Aufruf von `DocumentGenerateWSDLDocumentation` verwendet wurden. Der Standardwert für die erste

Ausführung ist true.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IncludeAll

Siehe auch

Methode: `IncludeAll (i_bInclude als Boolean )`

Beschreibung

Verwenden Sie diese Methode um alle Include-Optionen auszuwählen, bzw. die Auswahl für alle aufzuheben.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.

IncludeBinding

Siehe auch

Eigenschaft: `IncludeBinding` als Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um Bindings in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateWSDLDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IncludeImportedWSDLFiles

Siehe auch

Eigenschaft: `IncludeImportedWSDLFiles` als Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um importierte WSDL-Dateien in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateWSDLDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IncludeMessages

Siehe auch

Eigenschaft: [IncludeMessages](#) als Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um Messages in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateWSDLDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IncludeOverview

Siehe auch

Eigenschaft: [IncludeOverview](#) als Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um eine Übersicht in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateWSDLDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IncludePortType

Siehe auch

Eigenschaft: [IncludePortType](#) als Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um Port Types in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateWSDLDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IncludeService

Siehe auch

Eigenschaft: [IncludeService](#) als Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um Services in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateWSDLDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

IncludeTypes**Siehe auch**

Eigenschaft: [IncludeTypes](#) als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um Typen in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateWSDLDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

MultipleOutputFiles**Siehe auch**

Eigenschaft: [MultipleOutputFiles](#) als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um die Dokumentationsdateien aufzuteilen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateWSDLDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `false`.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Zum Definieren der Eigenschaft wurde ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

OptionsDialogAction**Siehe auch**

Eigenschaft: [OptionsDialogAction](#) als [SPYDialogAction](#)

Beschreibung

To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value `spyDialogUserInput(2)`. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use `spyDialogOK(0)`. Default is `spyDialogOK`.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Zum Definieren der Eigenschaft wurde ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

OutputFile**Siehe auch**

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

Beschreibung

Der vollständige Pfad und Name der Datei, die die generierte Dokumentation enthalten soll. Bei einer HTML-Ausgabe werden zusätzliche '.png' Dateien auf Basis dieses Dateinamens generiert. Der Standardwert für diese Eigenschaft ist ein leerer String, der ersetzt werden muss, bevor Sie diese Objekten in einem Aufruf von [DocumentGenerateWSDLDocumentation](#) verwenden.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

OutputFileDialogAction**Siehe auch**

Eigenschaft: [OutputFileDialogAction](#) als [SPYDialogAction](#)

Beschreibung

Damit der Benutzer die Ausgabedatei in einem Dialogfeld zur Dateiauswahl auswählen kann, setzen Sie diese Eigenschaft auf `spyDialogUserInput(2)`. Wenn der in [OutputFile](#) gespeicherte Wert verwendet werden soll und keine Interaktion mit dem Benutzer vorgesehen ist, verwenden Sie `spyDialogOK(0)`. Der Standardwert ist `spyDialogOK`.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Zum Definieren der Eigenschaft wurde ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

OutputFormat**Siehe auch**

Eigenschaft: [OutputFormat](#) als [SPYSchemaDocumentationFormat](#)

Beschreibung

Definiert, welche Art von Dokumentation generiert werden soll: HTML (Wert=0), MS-Word (Wert=1) oder RTF (Wert=2). Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateWSDLDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist HTML.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.

- 3901 Zum Definieren der Eigenschaft wurde ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Parent

Siehe auch

Eigenschaft: `Parent` als [Dialogs](#) (schreibgeschützt)

Beschreibung

Ruft das dem Objekt übergeordnete Objekt auf.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowBindingDiagram

Siehe auch

Eigenschaft: `ShowBindingDiagram` als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um Binding-Diagramme in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateWSDLDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowExtensibility

Siehe auch

Eigenschaft: `ShowExtensibility` als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um Service- und Binding-Erweiterungen der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateWSDLDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowMessageParts

Siehe auch

Eigenschaft: `ShowMessageParts` als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um Message-Teile von Messagesn der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateWSDLDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowPort

Siehe auch

Eigenschaft: `ShowPort` als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um Service Ports in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateWSDLDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowPortTypeDiagram

Siehe auch

Eigenschaft: `ShowPortTypeDiagram` als `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um Port Type-Diagramme in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateWSDLDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowPortTypeOperations

Siehe auch

Eigenschaft: `ShowPortTypeOperations` als `Boolean`

Beschreibung

Set this property to `true`, to show port type operations in the WSDL documentation. The property

is initialized with the value used during the last call to [DocumentGenerateWSDLDocumentation](#). Der Standardwert für die erste Ausführung ist true.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowProgressBar

Siehe auch

Eigenschaft: [ShowProgressBar](#) als Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um das Fenster, in dem die Fortschrittsanzeige für die Dokumentgeneration angezeigt wird, sichtbar zu machen. Mit `false` wird es ausgeblendet. Die Standardeinstellung ist `false`.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowResult

Siehe auch

Eigenschaft: [ShowResult](#) als Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um das erzeugte Dokument automatisch zu öffnen, wenn die Generierung erfolgreich abgeschlossen wurde. HTML-Dokumentation wird in Authentic Desktop geöffnet. Um Word-Dokumentation anzuzeigen, wird MS-Word gestartet. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateWSDLDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowServiceDiagram

Siehe auch

Eigenschaft: [ShowServiceDiagram](#) als Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um Service-Diagramme in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateWSDLDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowSourceCode

Siehe auch

Eigenschaft: `ShowSourceCode` as `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um Quellcode für die Includes in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateWSDLDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowTypesDiagram

Siehe auch

Eigenschaft: `ShowTypesDiagram` as `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um Typ-Diagramme in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateWSDLDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

ShowUsedBy

Siehe auch

Eigenschaft: `ShowUsedBy` as `Boolean`

Beschreibung

Setzen Sie diese Eigenschaft auf `true`, um used-by-Beziehungen für Types, Bindings und Message-Definitionen in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von [DocumentGenerateWSDLDocumentation](#) verwendet wurde. Der Standardwert für die erste Ausführung ist `true`.

Fehler

- 3900 Das Objekt ist nicht mehr gültig.
- 3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

4.3.28 XMLData

Siehe auch

Eigenschaften

[Kind](#)

[Name](#)

[TextValue](#)

[HasChildren](#)

[MayHaveChildren](#)

[Parent](#)

Methoden

[GetFirstChild](#)

[GetNextChild](#)

[GetCurrentChild](#)

[InsertChild](#)

[AppendChild](#)

[EraseAllChildren](#)

[EraseCurrentChild](#)

[IsSameNode](#)

[CountChildren](#)

[CountChildrenKind](#)

[GetChild](#)

[GetChildKind](#)

[HasChildrenKind](#)

Beschreibung

Über die XMLData-Schnittstelle haben Sie auf XML-Ebene direkten Zugriff auf das Dokument. Sie können die XML-Darstellung des Dokuments anzeigen und direkt bearbeiten. Beachten Sie jedoch die folgenden Einschränkungen:

- Die XMLData-Darstellung ist nur gültig, wenn das Dokument in der Grid-Ansicht oder der Authentic-Ansicht angezeigt wird.
- In der Authentic-Ansicht werden zusätzliche XMLData-Elemente automatisch als übergeordnete Elemente der einzelnen sichtbaren Dokumentelemente eingefügt. Normalerweise handelt es sich dabei um ein XMLData-Element von der Art `spyXMLDataElement`, wobei die [Name](#) Eigenschaft auf 'Text' gesetzt ist.
- Wenn Sie die XMLData-Schnittstelle von einem anderen Ansichtsmodus aus verwenden, werden zwar keine Fehler angezeigt, aber die Änderungen werden in der Ansicht nicht angezeigt und gehen u.U. beim nächsten Ansichtswechsel verloren.

Objekte dieser Klasse repräsentieren die unterschiedlichen Bestandteile eines XML-Dokuments. Informationen zu den verfügbaren Teiltypen finden Sie unter dem Enumerationstyp [SPYXMLDataKind](#). Jeder Teil kennt seine untergeordneten Elemente, wodurch eine XMLData-Struktur mit [DocumentRootElement](#) auf der obersten Ebene gebildet wird. Um das oberste Element des Dokumentinhalts (ohne XML Header) abzurufen, verwenden Sie [DocumentDataRoot](#). Beispiele, wie man die XMLData-Struktur durchquert, finden Sie unter [UsingXMLData to modify document structure](#) und [GetNextChild](#).

AppendChild

Siehe auch

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

Beschreibung

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

Fehler

- 1500 Das XMLData-Element ist nicht mehr gültig.
- 1505 Es wurde eine ungültige XMLData-Art definiert.
- 1506 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 1507 Das Element kann keine untergeordneten Elemente haben.
- 1512 Zyklische Einfügung - ein neues Datenelement ist bereits Teil des Dokuments.
- 1514 Es wurde eine ungültige XMLData-Art für diese Position definiert.
- 1900 Das Dokument darf nicht geändert werden.

Beispiel

```
Dim objCurrentParent As XMLData
Dim objNewChild As XMLData

Set objNewChild = objSpy.ActiveDocument.CreateChild(spyXMLDataElement)
Set objCurrentParent = objSpy.ActiveDocument.RootElement

objCurrentParent.AppendChild objNewChild

Set objNewChild = Nothing
```

CountChildren

Siehe auch

Deklaration: `CountChildren` als long

Beschreibung

`CountChildren` ruft die Anzahl der Child-Elemente ab.

Verfügbar mit Typbibliothek Version 1.5

Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.

CountChildrenKind

Siehe auch

Deklaration: `CountChildrenKind` (*nKind* als [SPYXMLDataKind](#)) als long

Beschreibung

`CountChildrenKind` ruft die Anzahl der Child-Elemente dieser bestimmten Art ab.

Verfügbar mit Typbibliothek Version 1.5

Fehler

1500 Das XMLData-Objekt ist nicht mehr gültig.

EraseAllChildren

Siehe auch

Deklaration: [EraseAllChildren](#)

Beschreibung

[EraseAllChildren](#) löscht alle zum XMLData Objekt gehörigen Child-Elemente.

Fehler

1500 Das XMLData-Objekt ist nicht mehr gültig.
1900 Das Dokument darf nicht geändert werden.

Beispiel

In diesem Beispiel werden alle Elemente des aktiven Dokuments gelöscht.

```
Dim objCurrentParent As XMLData

Set objCurrentParent = objSpy.ActiveDocument.RootElement
objCurrentParent.EraseAllChildren
```

EraseCurrentChild

Siehe auch

Deklaration: [EraseCurrentChild](#)

Beschreibung

[EraseCurrentChild](#) löscht das aktuelle XMLData Child-Objekt. Bevor Sie [EraseCurrentChild](#) aufrufen, müssen Sie einen internen Iterator mit [XMLData.GetFirstChild](#) initialisieren. Nachdem Sie das aktuelle Child-Element gelöscht haben, erhöht [EraseCurrentChild](#) den internen Iterator des XMLData-Elements. Es wird kein Fehler zurückgegeben, wenn das letzte Child-Element gelöscht wird und der Iterator über das Ende der Child-Liste hinaus verschoben wird. Der nächste Aufruf von [EraseCurrentChild](#) gibt allerdings den Fehler 1503 zurück.

Fehler

1500 Das XMLData-Objekt ist nicht mehr gültig.
1503 Es wurde für dieses XMLData-Objekt kein Iterator initialisiert oder der Iterator zeigt hinter das letzte Child-Element.
1900 Das Dokument darf nicht geändert werden.

Beispiele

```
// -----
// XMLSpy scripting environment - JScript
// erase all children of XMLData
// -----
// let's get an XMLData element, we assume that the
// cursor selects the parent of a list in grid view
var objList = Application.ActiveDocument.GridView.CurrentFocus;
```

```

// the following line would be shorter, of course
//objList.EraseAllChildren();

// but we want to demonstrate the usage of EraseCurrentChild
if ((objList != null) && (objList.HasChildren))
{
    try
    {
        objEle = objList.GetFirstChild(-1);
        while (objEle != null)
            objList.EraseCurrentChild();
        // no need to call GetNextChild
    }
    catch (err)
    {
        // 1503 - we reached end of child list
        if ((err.number & 0xffff) != 1503) throw (err);
    }
}

```

GetChild

Siehe auch

Deklaration: `GetChild` (*position* als long) als [XMLData](#)

Rückgabewert

Gibt ein XML-Element als `XMLData` Objekt zurück.

Beschreibung

`GetChild` gibt eine Referenz auf das Child am angegebenen Index (nullbasiert) zurück.

Verfügbar mit Typbibliothek Version 1.5

Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

GetChildKind

Siehe auch

Deklaration: `GetChildKind` (*position* als long, *nKind* als [SPYXMLDataKind](#)) als [XMLData](#)

Rückgabewert

Gibt ein XML-Element als `XMLData` Objekt zurück.

Beschreibung

`GetChildKind` gibt eine Referenz auf ein Child dieser Art am angegebenen Index (nullbasiert) zurück. Der Positionsparameter ist relativ zur Anzahl der Child-Elemente der angegebenen Art und nicht zu allen Child-Elementen des Objekts.

Verfügbar mit Typbibliothek Version 1.5

Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

GetCurrentChild

Siehe auch

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

Rückgabewert

Gibt ein XML-Element als `XMLData` Objekt zurück.

Beschreibung

`GetCurrentChild` ruft das aktuelle Child-Element ab. Bevor Sie `GetCurrentChild` aufrufen, müssen Sie einen internen Iterator mit [XMLData.GetFirstChild](#) initialisieren.

Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1503 Es wurde für dieses XMLData-Objekt kein Iterator initialisiert.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

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-Element zurück. Setzen Sie `nKind = -1`, um einen Iterator für alle Arten von Child-Elementen abzurufen.
Anmerkung: Der Iterator wird innerhalb des XMLData-Objekts gespeichert und wird zerstört, wenn das XMLData-Objekt zerstört wird. Stellen Sie sicher, dass Sie eine Referenz auf dieses Objekt beibehalten, solange Sie [GetCurrentChild](#), [GetNextChild](#) oder [EraseCurrentChild](#) verwenden möchten.

Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1501 Es wurde eine ungültige XMLData-Art definiert.
- 1504 Das Element hat keine untergeordneten Elemente der angegebenen Art.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiel

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

GetNextChild

Siehe auch

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

Rückgabewert

Gibt ein XML-Element als `XMLData` Objekt zurück.

Beschreibung

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

Überprüfen Sie das letzte Child-Element des Elements, wie im Beispiel unten gezeigt.

Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1503 Für dieses XMLData-Objekt wurde kein Iterator initialisiert.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```
' -----
' VBA code snippet - iterate XMLData children
' -----

On Error Resume Next
Set objParent = objSpy.ActiveDocument.RootElement

'get elements of all kinds
Set objCurrentChild = objParent.GetFirstChild(-1)

Do
    'do something useful with the child

    'step to next child
    Set objCurrentChild = objParent.GetNextChild
Loop Until (Err.Number - vbObjectError = 1503)

// -----
// XMLSpy scripting environment - JScript
// iterate through children of XMLData
// -----
try
{
    var objXMLData = ... // initialize somehow
    var objChild = objXMLData.GetFirstChild(-1);

    while (true)
    {
        // do something usefull with objChild

        objChild = objXMLData.GetNextChild();
    }
}
catch (err)
{
    if ((err.number & 0xffff) == 1504)
        ; // element has no children
    else if ((err.number & 0xffff) == 1503)
        ; // last child reached
    else
        throw (err);
}
```

HasChildrenKind

Siehe auch

Deklaration: `HasChildrenKind` (*nKind* als [SPYXMLDataKind](#)) als Boolean

Beschreibung

Die Methode gibt true zurück, wenn das Objekt das übergeordnete Objekt von anderen XMLData Objekten dieser Art ist.

Verfügbar mit Typbibliothek Version 1.5

Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

HasChildren**Siehe auch**

Deklaration: [HasChildren](#) als Boolean

Beschreibung

Die Eigenschaft ist true, wenn das Objekt das übergeordnete Objekt von anderen XMLData Objekten ist. Diese Eigenschaft ist schreibgeschützt.

Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

InsertChild**Siehe auch**

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

Beschreibung

[InsertChild](#) fügt das neue Child-Element vor dem aktuellen Child-Element ein (Informationen zum Setzen des aktuellen Child-Elements finden Sie auch unter [XMLData.GetFirstChild](#), [XMLData.GetNextChild](#)).

Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1503 Es wurde für dieses XMLData-Objekt kein Iterator initialisiert.
- 1505 Es wurde eine ungültige XMLData-Art definiert.
- 1506 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
- 1507 Das Element kann keine Child-Elemente haben.
- 1512 Zyklische Einfügung - ein neues Datenelement ist bereits Teil des Dokuments.
- 1514 Es wurde eine ungültige XMLData-Art für diese Position definiert.
- 1900 Das Dokument darf nicht geändert werden.

Beispiele

Siehe [Using XMLData to modify document structure](#).

IsSameNode

Siehe auch

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

Beschreibung

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

Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1506 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Kind

Siehe auch

Deklaration: `Kind` als [SPYXMLDataKind](#)

Beschreibung

Art dieses `XMLData` Objekts. Diese Eigenschaft ist schreibgeschützt.

Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

MayHaveChildren

Siehe auch

Deklaration: `MayHaveChildren` als Boolean

Beschreibung

Gibt an, ob Child-Elemente zu diesem `XMLData` Objekt hinzugefügt werden dürfen. Diese Eigenschaft ist schreibgeschützt.

Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Name

Siehe auch

Deklaration: `Name` als String

Beschreibung

Dient zum Bearbeiten und Abrufen des Namens des `XMLData` Objekts.

Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Parent

Siehe auch

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

Rückgabewert

Übergeordnetes Element als `XMLData` Objekt. Nichts (oder NULL), wenn es kein übergeordnetes Element gibt.

Beschreibung

Diesem Element übergeordnetes Element. Diese Eigenschaft ist schreibgeschützt.

Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

TextValue

Siehe auch

Deklaration: `TextValue` als String

Beschreibung

Dient zur Bearbeitung und Abrufen des Textwerts dieses `XMLData` Objekts.

Fehler

- 1500 Das XMLData-Objekt ist nicht mehr gültig.
- 1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

Siehe [„ndern der Dokumentstruktur Hilfe von XMLData“](#).

4.4 Schnittstellen (nicht mehr verwendet)

Die in diesem Teil beschriebenen Schnittstellen werden nicht mehr verwendet. Es wird empfohlen Ihre Applikationen auf die neuen Schnittstellen zu migrieren. Tipps zur Migration finden Sie unter den verschiedenen Eigenschaften und Methoden in diesem Teil.

4.4.1 AuthenticEvent (obsolete)

Wird ersetzt durch [AuthenticView](#) und [AuthenticRange](#)

Das DocEditView Objekt wird umbenannt in OldAuthenticView.
DocEditSelection wird umbenannt in AuthenticSelection.
DocEditEvent wird umbenannt in AuthenticEvent.
DocEditDataTransfer wird umbenannt in AuthenticDataTransfer.

Die Verwendung dieser Objekte wird nicht mehr empfohlen - mit Ausnahme der Verwendung für AuthenticDataTransfer. Bestehende Funktionalitäten werden auch in Zukunft auf unbestimmte Zeit unterstützt, doch es werden keine neuen Funktionen zu dieser Schnittstelle hinzugefügt. Alle bisher in [DocEditView](#), [DocEditSelection](#), [DocEditEvent](#) und [DocEditDataTransfer](#) verfügbaren Funktionalitäten stehen nun über [AuthenticView](#), [AuthenticRange](#) und [AuthenticDataTransfer](#) zur Verfügung. Es wurden viele neue Funktionen hinzugefügt.

Beispiele zur Migration von DocEdit in Authentic finden Sie unter der Beschreibung der verschiedenen Methoden und Eigenschaften der einzelnen DocEdit Objekte.

Siehe auch

Eigenschaften

[altKey](#)
[altLeft](#)
[ctrlKey](#)
[ctrlLeft](#)
[shiftKey](#)
[shiftLeft](#)

[keyCode](#)
[repeat](#)

[button](#)

[clientX](#)
[clientY](#)

[dataTransfer](#)

[srcElement](#)
[fromElement](#)

[propertyName](#)

[cancelBubble](#)
[returnValue](#)

[type](#)

Beschreibung

DocEditEvent Schnittstelle.

altKey (nicht mehr verwendet)

Wird ersetzt durch Parameter für

[AuthenticView.OnKeyboardEvent](#) (On_AuthenticView_KeyPressed)

[AuthenticView.OnMouseEvent](#) (On_AuthenticView_MouseEvent)

[AuthenticView.OnDragOver](#) (On_AuthenticView_DragOver)

Das Event-Objekt, das die Informationen des letzten Event enthält, wird nun durch Parameter für die verschiedenen Event Handler-Funktionen ersetzt, um den Datenzugriff zu erleichtern. Aus Kompatibilitätsgründen wird das Event Objekt auf unbestimmte Zeit weiter unterstützt. Es sind keine Verbesserungen geplant. Wir empfehlen Ihnen, nach Möglichkeit die neuen Event Handler Funktionen zu verwenden.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditKeyPressed ()
// {
//     if ( Application.ActiveDocument.DocEditView.event.altKey ||
//         Application.ActiveDocument.DocEditView.event.altLeft)
//         MsgBox ("alt key is down");
// }
// use now:
function On_AuthenticView_KeyPressed (SPYKeyEvent i_eKeyEvent, long
i_nKeyCode, SPYVirtualKeyMask i_nVirtualKeyStatus)
{
    if (i_nVirtualKeyStatus & spyAltKeyMask)
        MsgBox ("alt key is down");
}
```

Siehe auch

Deklaration: [altKey](#) als Boolean

Beschreibung

True, wenn die rechte ALT-Taste gedrückt wird.

altLeft (nicht mehr verwendet)

Wird ersetzt durch Parameter für

[AuthenticView.OnKeyboardEvent](#) (On_AuthenticView_KeyPressed)

[AuthenticView.OnMouseEvent](#) (On_AuthenticView_MouseEvent)

[AuthenticView.OnDragOver](#) (On_AuthenticView_DragOver)

Das Event-Objekt, das die Informationen des letzten Event enthält, wird nun durch Parameter für die verschiedenen Event Handler-Funktionen ersetzt, um den Datenzugriff zu erleichtern. Aus Kompatibilitätsgründen wird das Event Objekt auf unbestimmte Zeit weiter unterstützt. Es sind keine Verbesserungen geplant. Wir empfehlen Ihnen, nach Möglichkeit die neuen Event Handler Funktionen zu verwenden.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditKeyDown ( )
// {
//     if ( Application.ActiveDocument.DocEditView.event.altKey ||
//         Application.ActiveDocument.DocEditView.event.altLeft)
//         MsgBox ("alt key is down");
// }
// use now:
function On_AuthenticView_KeyDown ( SPYKeyEvent i_eKeyEvent, long i_nKeyCode,
SPYVirtualKeyMask i_nVirtualKeyStatus)
{
    if ( i_nVirtualKeyStatus & spyAltKeyMask)
        MsgBox ("alt key is down");
}
```

Siehe auch

Deklaration: `altLeft` als Boolean

Beschreibung

True, wenn die linke ALT-Taste gedrückt wird.

button (nicht mehr verwendet)

Wird ersetzt durch Parameter für

[AuthenticView.OnMouseEvent](#) (On_AuthenticView_MouseEvent)

[AuthenticView.OnDragOver](#) (On_AuthenticView_DragOver)

Das Event-Objekt, das die Informationen des letzten Event enthält, wird nun durch Parameter für die verschiedenen Event Handler-Funktionen ersetzt, um den Datenzugriff zu erleichtern. Aus Kompatibilitätsgründen wird das Event Objekt auf unbestimmte Zeit weiter unterstützt. Es sind keine Verbesserungen geplant. Wir empfehlen Ihnen, nach Möglichkeit die neuen Event Handler Funktionen zu verwenden.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditButtonDown ()
// {
//     if ( Application.ActiveDocument.DocEditView.event.button == 1)
//         MsgBox ("left mouse button down detected");
// }
// use now:
function On_AuthenticView_MouseEvent (long i_nXPos, long i_nYPos,
SPYMouseEvent i_eMouseEvent, IAuthenticRange *i_ipRange)
{
    if (i_eMouseEvent & spyLeftButtonDownMask)
        MsgBox ("left mouse button down detected");
}
```

Siehe auch

Deklaration: `button` als long

Beschreibung

Definiert, welche Maustaste gedrückt wird:

0	Es wird keine Taste gedrückt.
1	Die linke Taste wird gedrückt.
2	Die rechte Taste wird gedrückt.
3	Es werden die linke und die rechte Taste gedrückt.
4	Die mittlere Taste wird gedrückt.
5	Es werden die linke und die mittlere Taste gedrückt.
6	Es werden die rechte und die mittlere Taste gedrückt.
7	Alle drei Taste werden gedrückt.

cancelBubble (nicht mehr verwendet)

Wird ersetzt durch den Booleschen Rückgabewert der folgenden Event Handler-Funktionen

[AuthenticView.OnKeyboardEvent](#) (On_AuthenticView_KeyPressed)

[AuthenticView.OnMouseEvent](#) (On_AuthenticView_MouseEvent)

[AuthenticView.OnDragOver](#) (On_AuthenticView_DragOver)

Das Event-Objekt, das die Informationen des letzten Event enthält, wird nun durch Parameter für die verschiedenen Event Handler-Funktionen ersetzt, um den Datenzugriff zu erleichtern. Aus Kompatibilitätsgründen wird das Event Objekt auf unbestimmte Zeit weiter unterstützt. Es sind keine Verbesserungen geplant. Wir empfehlen Ihnen, nach Möglichkeit die neuen Event Handler Funktionen zu verwenden

Wenn von einer Event Handler-Funktion *true* zurückgegeben wird, bedeutet dies, dass das Event behandelt wurde und die normale Event-Behandlung abgebrochen werden soll.

```
// ----- XMLSpy scripting environment - javascript sample -----  
// instead of:  
// function On_DocEditKeyPressed ()  
// {  
//     if ( Application.ActiveDocument.DocEditView.event.keyCode == 0x20)  
//     {  
//         // cancel key processing, swallow spaces :-)  
//         Application.ActiveDocument.DocEditView.event.cancelBubble = true;  
//     }  
// }  
// use now:  
function On_AuthenticView_KeyPressed ( SPYKeyEvent i_eKeyEvent, long  
i_nKeyCode, SPYVirtualKeyMask i_nVirtualKeyStatus)  
{  
    if ( i_nKeyCode == 0x20)  
        return true; // cancel key processing, swallow spaces :-)  
}
```

Siehe auch

Dellaration: [cancelBubble](#) als Boolean

Beschreibung

Setzen Sie `cancelBubble` auf TRUE, wenn der Standard-Event Handler nicht aufgerufen werden soll.

clientX (nicht mehr verwendet)

Wird ersetzt durch Parameter für

[AuthenticView.OnMouseEvent](#) (On_AuthenticView_MouseEvent)

[AuthenticView.OnBeforeDrop](#) (On_AuthenticView_BeforeDrop)

[AuthenticView.OnDragOver](#) (On_AuthenticView_DragOver)

Das Event-Objekt, das die Informationen des letzten Event enthält, wird nun durch Parameter für die verschiedenen Event Handler-Funktionen ersetzt, um den Datenzugriff zu erleichtern. Aus Kompatibilitätsgründen wird das Event Objekt auf unbestimmte Zeit weiter unterstützt. Es sind keine Verbesserungen geplant. Wir empfehlen Ihnen, nach Möglichkeit die neuen Event Handler Funktionen zu verwenden

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditMouseMove ()
// {
//     MsgBox ("moving over " +
// Application.ActiveDocument.DocEditView.event.clientX +
//         "/" + Application.ActiveDocument.DocEditView.event.clientY);
// }
// use now:
function On_AuthenticView_MouseEvent (long i_nXPos, long i_nYPos,
SPYMouseEvent i_eMouseEvent, IAuthenticRange *i_ipRange)
{
    if (i_eMouseEvent & spyMouseMoveMask)
        MsgBox ("moving over " + i_nXPos + "/" + i_nYPos);
}
```

Siehe auch

Deklaration: `clientX` als long

Beschreibung

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

clientY (nicht mehr verwendet)

Wird ersetzt durch Parameter für

[AuthenticView.OnMouseEvent](#) (On_AuthenticView_MouseEvent)

[AuthenticView.OnBeforeDrop](#) (On_AuthenticView_BeforeDrop)

[AuthenticView.OnDragOver](#) (On_AuthenticView_DragOver)

Das Event-Objekt, das die Informationen des letzten Event enthält, wird nun durch Parameter für die verschiedenen Event Handler-Funktionen ersetzt, um den Datenzugriff zu erleichtern. Aus Kompatibilitätsgründen wird das Event Objekt auf unbestimmte Zeit weiter unterstützt. Es sind keine Verbesserungen geplant. Wir empfehlen Ihnen, nach Möglichkeit die neuen Event Handler Funktionen zu verwenden

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditMouseMove ()
// {
//     MsgBox ("moving over " +
// Application.ActiveDocument.DocEditView.event.clientX +
//         "/" + Application.ActiveDocument.DocEditView.event.clientY);
// }
// use now:
function On_AuthenticView_MouseEvent (long i_nXPos, long i_nYPos,
SPYMouseEvent i_eMouseEvent, IAuthenticRange *i_ipRange)
{
    if (i_eMouseEvent & spyMouseMoveMask)
        MsgBox ("moving over " + i_nXPos + "/" + i_nYPos);
}
```

Siehe auch

Deklaration: `clientY` als long

Beschreibung

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

ctrlKey (nicht mehr verwendet)**Wird ersetzt durch Parameter für**[AuthenticView.OnKeyboardEvent](#) (On_AuthenticView_KeyPressed)[AuthenticView.OnMouseEvent](#) (On_AuthenticView_MouseEvent)[AuthenticView.OnDragOver](#) (On_AuthenticView_DragOver)

Das Event-Objekt, das die Informationen des letzten Event enthält, wird nun durch Parameter für die verschiedenen Event Handler-Funktionen ersetzt, um den Datenzugriff zu erleichtern. Aus Kompatibilitätsgründen wird das Event Objekt auf unbestimmte Zeit weiter unterstützt. Es sind keine Verbesserungen geplant. Wir empfehlen Ihnen, nach Möglichkeit die neuen Event Handler Funktionen zu verwenden.

```
// ----- XMLSpy scripting environment - javascript sample -----  
// instead of:  
// function On_DocEditMouseMove ()  
// {  
//     if ( Application.ActiveDocument.DocEditView.event.ctrlKey ||  
//         Application.ActiveDocument.DocEditView.event.altLeft )  
//         MsgBox ("control key is down");  
// }  
// use now:  
function On_AuthenticView_MouseEvent (long i_nXPos, long i_nYPos,  
SPYMouseEvent i_eMouseEvent, IAuthenticRange *i_ipRange)  
{  
    if (i_eMouseEvent & spyCtrlKeyMask)  
        MsgBox ("control key is down");  
}
```

Siehe auch

Deklaration: [ctrlKey](#) als Boolean

Beschreibung

True, wenn die rechte Strg-Taste gedrückt wird.

ctrlLeft (nicht mehr verwendet)

Wird ersetzt durch parameters to

[AuthenticView.OnKeyboardEvent](#) (On_AuthenticView_KeyPressed)

[AuthenticView.OnMouseEvent](#) (On_AuthenticView_MouseEvent)

[AuthenticView.OnDragOver](#) (On_AuthenticView_DragOver)

Das Event-Objekt, das die Informationen des letzten Event enthält, wird nun durch Parameter für die verschiedenen Event Handler-Funktionen ersetzt, um den Datenzugriff zu erleichtern. Aus Kompatibilitätsgründen wird das Event Objekt auf unbestimmte Zeit weiter unterstützt. Es sind keine Verbesserungen geplant. Wir empfehlen Ihnen, nach Möglichkeit die neuen Event Handler Funktionen zu verwenden.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditMouseMove ()
// {
//     if ( Application.ActiveDocument.DocEditView.event.ctrlKey ||
//         Application.ActiveDocument.DocEditView.event.altLeft)
//         MsgBox ("control key is down");
// }
// use now:
function On_AuthenticView_MouseEvent (long i_nXPos, long i_nYPos,
SPYMouseEvent i_eMouseEvent, IAuthenticRange *i_ipRange)
{
    if (i_eMouseEvent & spyCtrlKeyMask)
        MsgBox ("control key is down");
}
```

Siehe auch

Deklaration: `ctrlLeft` als Boolean

Beschreibung

True, wenn die linke Strg-Taste gedrückt wird.

dataTransfer (nicht mehr verwendet)**Wird ersetzt durch Parameter für****[AuthenticView.OnBeforeDrop](#)** (On_AuthenticView_BeforeDrop)**[AuthenticView.OnDragOver](#)** (On_AuthenticView_DragOver)

Das Event-Objekt, das die Informationen des letzten Event enthält, wird nun durch Parameter für die verschiedenen Event Handler-Funktionen ersetzt, um den Datenzugriff zu erleichtern. Aus Kompatibilitätsgründen wird das Event Objekt auf unbestimmte Zeit weiter unterstützt. Es sind keine Verbesserungen geplant. Wir empfehlen Ihnen, nach Möglichkeit die neuen Event Handler Funktionen zu verwenden.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditDrop ()
// {
//     if ( Application.ActiveDocument.DocEditView.event.dataTransfer !=
// null)
//         if (!
Application.ActiveDocument.DocEditView.event.dataTransfer.ownDrag)
//         {
//             // cancel key processing, don't drop foreign objects :-)
//             Application.ActiveDocument.DocEditView.event.cancelBubble =
true;
//         }
// }
// use now:
function On_AuthenticView_BeforeDrop ( long i_nXPos, long i_nYPos,
IAuthenticRange *i_ipRange,
IAuthenticDataTransfer *i_ipData)
{
    if (i_ipRange != null)
        if (! i_ipRange.ownDrag)
            return true; // cancel key processing, don't drop foreign
objects :-)

    return false;
}
```

Siehe auch**Deklaration:** [dataTransfer](#) als Variant**Beschreibung**Eigenschaft [dataTransfer](#)

fromElement (nicht mehr verwendet)

Nicht unterstützt

Siehe auch

Deklaration: `fromElement` als Variant (nicht unterstützt)

Beschreibung

Derzeit wird diese Eigenschaft von keinem Event definiert.

keyCode (nicht mehr verwendet)

Wird ersetzt durch einen Parameter für
[AuthenticView.OnKeyboardEvent](#) (On_AuthenticView_KeyPressed)

Das Event-Objekt, das die Informationen des letzten Event enthält, wird nun durch Parameter für die verschiedenen Event Handler-Funktionen ersetzt, um den Datenzugriff zu erleichtern. Aus Kompatibilitätsgründen wird das Event Objekt auf unbestimmte Zeit weiter unterstützt. Es sind keine Verbesserungen geplant. Wir empfehlen Ihnen, nach Möglichkeit die neuen Event Handler Funktionen zu verwenden.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditKeyPressed ()
// {
//     if ( Application.ActiveDocument.DocEditView.event.keyCode == 0x20)
//     {
//         // cancel key processing, swallow spaces :-)
//         Application.ActiveDocument.DocEditView.event.cancelBubble = true;
//     }
// }
// use now:
function On_AuthenticView_KeyPressed ( SPYKeyEvent i_eKeyEvent, long
i_nKeyCode, SPYVirtualKeyMask i_nVirtualKeyStatus)
{
    if ( i_nKeyCode == 0x20)
        return true; // cancel key processing, swallow spaces :-)
}
```

Siehe auch

Deklaration: `keyCode` als long

Beschreibung

Tastaturcode der aktuell gedrückten Taste. Diese Eigenschaft ist read-write.

propertyName (nicht mehr verwendet)

Nicht unterstützt

Siehe auch

Deklaration: `propertyName` als String (nicht unterstützt)

Beschreibung

Derzeit wird diese Eigenschaft von keinem Event definiert.

repeat (nicht mehr verwendet)

Nicht unterstützt

Siehe auch

Deklaration: `repeat` als Boolean (nicht unterstützt)

Beschreibung

True, wenn das `onkeydown` Event wiederholt wird.

returnValue (nicht mehr verwendet)

Nicht mehr unterstützt

Siehe auch

Deklaration: `returnValue` als Variant

Beschreibung

Mit Hilfe von `returnValue` können Sie einen Rückgabewert für Ihren Event Handler setzen.

shiftKey (nicht mehr verwendet)

Wird ersetzt durch Parameter für

[AuthenticView.OnKeyboardEvent](#) (On_AuthenticView_KeyPressed)

[AuthenticView.OnMouseEvent](#) (On_AuthenticView_MouseEvent)

[AuthenticView.OnDragOver](#) (On_AuthenticView_DragOver)

Das Event-Objekt, das die Informationen des letzten Event enthält, wird nun durch Parameter für die verschiedenen Event Handler-Funktionen ersetzt, um den Datenzugriff zu erleichtern. Aus Kompatibilitätsgründen wird das Event Objekt auf unbestimmte Zeit weiter unterstützt. Es sind keine Verbesserungen geplant. Wir empfehlen Ihnen, nach Möglichkeit die neuen Event Handler Funktionen zu verwenden.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditDragOver ()
// {
//     if ( Application.ActiveDocument.DocEditView.event.shiftKey ||
//         Application.ActiveDocument.DocEditView.event.shiftLeft)
//         MsgBox ("shift key is down");
// }
// use now:
function On_AuthenticView_DragOver (long i_nXPos, long i_nYPos,
                                   SPYMouseEvent i_eMouseEvent,
                                   IAuthenticRange *i_ipRange,
                                   IAuthenticDataTransfer *i_ipData)
{
    if (i_eMouseEvent & spyShiftKeyMask)
        MsgBox ("shift key is down");
}
```

Siehe auch

Deklaration: [shiftKey](#) als Boolean

Beschreibung

True, wenn die rechte Umschalttaste gedrückt wird.

shiftLeft (nicht mehr verwendet)

Wird ersetzt durch Parameter für

[AuthenticView.OnKeyboardEvent](#) (On_AuthenticView_KeyPressed)

[AuthenticView.OnMouseEvent](#) (On_AuthenticView_MouseEvent)

[AuthenticView.OnDragOver](#) (On_AuthenticView_DragOver)

Das Event-Objekt, das die Informationen des letzten Event enthält, wird nun durch Parameter für die verschiedenen Event Handler-Funktionen ersetzt, um den Datenzugriff zu erleichtern. Aus Kompatibilitätsgründen wird das Event Objekt auf unbestimmte Zeit weiter unterstützt. Es sind keine Verbesserungen geplant. Wir empfehlen Ihnen, nach Möglichkeit die neuen Event Handler Funktionen zu verwenden.

```
// ----- XMLSpy scripting environment - javascript sample -----
// instead of:
// function On_DocEditDragOver ()
// {
//     if ( Application.ActiveDocument.DocEditView.event.shiftKey ||
//         Application.ActiveDocument.DocEditView.event.shiftLeft)
//         MsgBox ("shift key is down");
// }
// use now:
function On_AuthenticView_DragOver (long i_nXPos, long i_nYPos,
                                   SPYMouseEvent i_eMouseEvent,
                                   IAuthenticRange *i_ipRange,
                                   IAuthenticDataTransfer *i_ipData)
{
    if (i_eMouseEvent & spyShiftKeyMask)
        MsgBox ("shift key is down");
}
```

Siehe auch

Deklaration: [shiftLeft](#) als Boolean

Beschreibung

True, wenn die linke Umschalttaste gedrückt wird.

srcElement (nicht mehr verwendet)

Wird ersetzt durch Parameter für

[AuthenticView.OnMouseEvent](#) (On_AuthenticView_MouseEvent)

[AuthenticView.OnBeforeDrop](#) (On_AuthenticView_BeforeDrop)

[AuthenticView.OnDragOver](#) (On_AuthenticView_DragOver)

Das Event-Objekt, das die Informationen des letzten Event enthält, wird nun durch Parameter für die verschiedenen Event Handler-Funktionen ersetzt, um den Datenzugriff zu erleichtern. Aus Kompatibilitätsgründen wird das Event Objekt auf unbestimmte Zeit weiter unterstützt. Es sind keine Verbesserungen geplant. Wir empfehlen Ihnen, nach Möglichkeit die neuen Event Handler Funktionen zu verwenden.

Mit der neuen Event Handler-Funktion wird anstelle des XMLData-Elements, das sich derzeit unterhalb des Mausursors befindet, ein Bereichsobjekt bereitgestellt.

```
// ----- XMLSpy scripting environment - javascript sample -----  
// instead of:  
// function On_DocEditMouseMove ()  
// {  
//     var objEvent = Application.ActiveDocument.DocEditView.event;  
//     if (objEvent.srcElement != null)  
//         MsgBox ("moving over " + objEvent.srcElement.Parent.Name);  
// }  
// use now:  
function On_AuthenticView_MouseEvent (long i_nXPos, long i_nYPos,  
SPYMouseEvent i_eMouseEvent, IAuthenticRange *i_ipRange)  
{  
    if ((i_eMouseEvent & spyMouseMoveMask) &&  
        (i_ipRange != null))  
        MsgBox ("moving over " + i_ipRange.FirstXMLData.Parent.Name);  
}
```

Siehe auch

Deklaration: [srcElement](#) als Variant

Beschreibung

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

type (nicht mehr verwendet)

Nicht unterstützt

Siehe auch

Deklaration: `type` als String (nicht unterstützt)

Beschreibung

Derzeit wird diese Eigenschaft von keinem Event definiert.

4.4.2 AuthenticSelection (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange](#)

Das DocEditView object wird umbenannt in OldAuthenticView.
DocEditSelection wird umbenannt in AuthenticSelection.
DocEditEvent wird umbenannt in AuthenticEvent.
DocEditDataTransfer wird umbenannt in AuthenticDataTransfer.

Die Verwendung dieser Objekte wird nicht mehr empfohlen - mit Ausnahme der Verwendung für AuthenticDataTransfer. Bestehende Funktionalitäten werden auch in Zukunft auf unbestimmte Zeit unterstützt, doch es werden keine neuen Funktionen zu dieser Schnittstelle hinzugefügt. Alle bisher in [DocEditView](#), [DocEditSelection](#), [DocEditEvent](#) und [DocEditDataTransfer](#) verfügbaren Funktionalitäten stehen nun über [AuthenticView](#), [AuthenticRange](#) und [AuthenticDataTransfer](#) zur Verfügung. Es wurden viele neue Funktionen hinzugefügt.

Beispiele zur Migration von DocEdit in Authentic finden Sie unter der Beschreibung der verschiedenen Methoden und Eigenschaften der einzelnen DocEdit Objekte.

Siehe auch

Eigenschaften

[Start](#)

[StartTextPosition](#)

[End](#)

[EndTextPosition](#)

End (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.LastXMLData](#)

```
// ----- javascript sample -----  
// instead of:  
// var objXMLData =  
Application.ActiveDocument.DocEditView.CurrentSelection.End;  
// use now:  
var objXMLData =  
Application.ActiveDocument.AuthenticView.Selection.LastXMLData;
```

Siehe auch

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

Beschreibung

XML-Element, an dem die aktuelle Auswahl endet.

EndTextPosition (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.LastXMLDataOffset](#)

```
// ----- javascript sample -----  
// instead of:  
// var nOffset =  
Application.ActiveDocument.DocEditView.CurrentSelection.EndTextPosition;  
// use now:  
var nOffset =  
Application.ActiveDocument.AuthenticView.Selection.LastXMLDataOffset;
```

Siehe auch

Deklaration: `EndTextPosition` als `long`

Beschreibung

Positioniert den Cursor in [DocEditSelection.End.TextValue](#), wo die Auswahl endet.

Start (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.FirstXMLData](#)

```
// ----- javascript sample -----  
// instead of:  
// var objXMLData =  
Application.ActiveDocument.DocEditView.CurrentSelection.Start;  
// use now:  
var objXMLData =  
Application.ActiveDocument.AuthenticView.Selection.FirstXMLData;
```

Siehe auch

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

Beschreibung

XML-Element, an dem die Cursorauswahl beginnt.

StartTextPosition (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.FirstXMLDataOffset](#)

```
// ----- javascript sample -----  
// instead of:  
// var nOffset =  
Application.ActiveDocument.DocEditView.CurrentSelection.StartTextPosition;  
// use now:  
var nOffset =  
Application.ActiveDocument.AuthenticView.Selection.FirstXMLDataOffset;
```

Siehe auch

Deklaration: `StartTextPosition` als long

Beschreibung

Positioniert den Cursor in [DocEditSelection.Start.TextValue](#), wo die Auswahl beginnt.

4.4.3 OldAuthenticView (nicht mehr verwendet)

Wird ersetzt durch [AuthenticView](#) und [AuthenticRange](#)

Das DocEditView object wird umbenannt in OldAuthenticView.
DocEditSelection wird umbenannt in AuthenticSelection.
DocEditEvent wird umbenannt in AuthenticEvent.
DocEditDataTransfer wird umbenannt in AuthenticDataTransfer.

Die Verwendung dieser Objekte wird nicht mehr empfohlen - mit Ausnahme der Verwendung für AuthenticDataTransfer. Bestehende Funktionalitäten werden auch in Zukunft auf unbestimmte Zeit unterstützt, doch es werden keine neuen Funktionen zu dieser Schnittstelle hinzugefügt. Alle bisher in [DocEditView](#), [DocEditSelection](#), [DocEditEvent](#) und [DocEditDataTransfer](#) verfügbaren Funktionalitäten stehen nun über [AuthenticView](#), [AuthenticRange](#) und [AuthenticDataTransfer](#) zur Verfügung. Es wurden viele neue Funktionen hinzugefügt.

Beispiele zur Migration von DocEdit in Authentic finden Sie unter der Beschreibung der verschiedenen Methoden und Eigenschaften der einzelnen DocEdit Objekte.

Siehe auch

Methoden

[LoadXML](#)

[SaveXML](#)

[EditClear](#)

[EditCopy](#)

[EditCut](#)

[EditPaste](#)

[EditRedo](#)

[EditSelectAll](#)

[EditUndo](#)

[RowAppend](#)

[RowDelete](#)

[RowDuplicate](#)

[RowInsert](#)

[RowMoveDown](#)

[RowMoveUp](#)

[ApplyTextState](#)

[IsTextStateApplied](#)

[IsTextStateEnabled](#)

[MarkupView](#)

[SelectionSet](#)

[SelectionMoveTabOrder](#)

[GetNextVisible](#)

[GetPreviousVisible](#)

[GetAllowedElements](#)

Eigenschaften[CurrentSelection](#)[event](#)[XMLRoot](#)[IsEditClearEnabled](#)[IsEditCopyEnabled](#)[IsEditCutEnabled](#)[IsEditPasteEnabled](#)[IsEditRedoEnabled](#)[IsEditUndoEnabled](#)[IsRowAppendEnabled](#)[IsRowDeleteEnabled](#)[IsRowDuplicateEnabled](#)[IsRowInsertEnabled](#)[IsRowMoveDownEnabled](#)[IsRowMoveUpEnabled](#)**Beschreibung**

Schnittstelle für die Authentic-Ansicht.

ApplyTextState (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.PerformAction](#)

Verwenden Sie spyAuthenticApply für den eAction-Parameter. Mit Hilfe der PerformAction-Methode können Sie Textstatusattribute auf jeden beliebigen Bereich des Dokuments und nicht nur auf die aktuelle Auswahl auf der Benutzeroberfläche anwenden.

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.DocEditView.ApplyTextState ("bold");  
// use now:  
if (! Application.ActiveDocument.AuthenticView.Selection.PerformAction  
(spyAuthenticApply, "bold"))  
    MsgBox ("Error: can't set current selection to bold");
```

Siehe auch

Deklaration: [ApplyTextState](#) (*elementName* als String)

Beschreibung

Wendet den durch den Parameter *elementName* definierten Textstatus an bzw. entfernt ihn. Häufige Beispiele für den Parameter *elementName* wären "strong" und "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 Wortfragment fett formatiert. Dies geschieht jedoch nur, weil der HTML-Parser weiß, dass der Tag für fett steht. Bei XML gibt es viel mehr Flexibilität. Jeder beliebige XML-Tag kann beliebig nach Ihren Wünschen definiert werden. D.h. es kann mittels XML ein Textstatus angewendet werden. Der angewendete Textstatus muss jedoch Teil des Schemas sein. So ist z.B. in den Beispielen *OrgChart.xml*, *OrgChart.sps*, *OrgChart.xsd* der Tag derselbe wie "bold". Und um "bold" anzuwenden, wird die Methode `ApplyTextState()` aufgerufen. Wie bei den Operationen "row" und "edit" muss jedoch überprüft werden, ob der Textstatus angewendet werden kann.

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

CurrentSelection (nicht mehr verwendet)

Wird ersetzt durch [AuthenticView.Selection](#)

Das zurückgegebene Objekt [AuthenticRange](#) unterstützt die Navigation mittels XMLData-Elemente sowie die Navigation über Dokumentelemente (z.B. Zeichen, Wörter, Tags) oder Textcursorpositionen.

```
// ----- javascript sample -----  
// instead of:  
// var objDocEditSel =  
Application.ActiveDocument.DocEditView.CurrentSelection;  
// use now:  
var objRange = Application.ActiveDocument.AuthenticView.Selection;
```

Siehe auch

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

Beschreibung

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

EditClear (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.Delete](#)

Mit Hilfe der Delete-Methode von [AuthenticRange](#) können Sie jeden beliebigen Bereich des Dokuments löschen, nicht nur die aktuellen Auswahl auf der Benutzeroberfläche.

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.DocEditView.EditClear();  
// use now:  
if (! Application.ActiveDocument.AuthenticView.Selection.Delete())  
    MsgBox ("Error: can't delete current selection");
```

Siehe auch

Deklaration: [EditClear](#)

Beschreibung

Löscht die aktuelle Auswahl.

EditCopy(nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.Copy](#)

Mit Hilfe der Copy-Methode von AuthenticRange können Sie jeden beliebigen Bereich des Dokuments kopieren, nicht nur die aktuellen Auswahl auf der Benutzeroberfläche.

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.DocEditView.EditCopy();  
// use now:  
if (! Application.ActiveDocument.AuthenticView.Selection.Copy())  
    MsgBox ("Error: can't copy current selection");
```

Siehe auch

Deklaration: [EditCopy](#)

Beschreibung

Kopiert die aktuelle Auswahl in die Zwischenablage.

EditCut (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.Cut](#)

Mit Hilfe der Cut-Methode von AuthenticRange können Sie jeden beliebigen Bereich des Dokuments ausschneiden, nicht nur die aktuellen Auswahl auf der Benutzeroberfläche.

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.DocEditView.EditCut();  
// use now:  
if (! Application.ActiveDocument.AuthenticView.Selection.Cut())  
    MsgBox ("Error: can't cut out current selection");
```

Siehe auch

Deklaration: [EditCut](#)

Beschreibung

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

EditPaste (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.Paste](#)

Mit Hilfe der Paste-Methode von AuthenticRange können Sie jeden beliebigen Bereich des Dokuments einfügen, nicht nur die aktuellen Auswahl auf der Benutzeroberfläche.

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.DocEditView.EditPaste();  
// use now:  
if (! Application.ActiveDocument.AuthenticView.Selection.Paste())  
    MsgBox ("Error: can't paste to current selection");
```

Siehe auch

Deklaration: [EditPaste](#)

Beschreibung

Fügt den Inhalt aus der Zwischenablage in das Dokument ein.

EditRedo (nicht mehr verwendet)

Wird ersetzt durch [AuthenticView.Redo](#)

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.DocEditView.EditRedo();  
// use now:  
if (! Application.ActiveDocument.AuthenticView.Redo())  
    MsgBox ("Error: no redo step available");
```

Siehe auch

Deklaration: [EditRedo](#)

Beschreibung

Wiederholt den zuletzt rückgängig gemachten Schritt.

EditSelectAll (nicht mehr verwendet)

Wird ersetzt durch [AuthenticView.WholeDocument](#) and [AuthenticRange.Select](#)

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.DocEditView.EditSelectAll();  
// use now:  
Application.ActiveDocument.AuthenticView.WholeDocument.Select();
```

Siehe auch

Deklaration: [EditSelectAll](#)

Beschreibung

Die Methode wählt das gesamte Dokument aus.

EditUndo (nicht mehr verwendet)

Wird ersetzt durch [AuthenticView.Undo](#)

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.DocEditView.EditUndo();  
// use now:  
if (! Application.ActiveDocument.AuthenticView.Undo())  
    MsgBox ("Error: no undo step available");
```

Siehe auch

Deklaration: [EditUndo](#)

Beschreibung

Macht die letzte Aktion rückgängig.

event (nicht mehr verwendet)

Wird ersetzt durch Parameter für [AuthenticView events](#).

Siehe auch

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

Beschreibung

Die Event-Eigenschaft enthält ein `DocEditEvent` Objekt, das Informationen über das aktuelle Event enthält.

GetAllowedElements (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.CanPerformActionWith](#)

`AuthenticRange` unterstützt nun alle Funktionalitäten der Elementeingabehilfen. Neben der Abfrage, welche Elemente eingefügt, angehängt usw. werden können, können Sie auch die Aktion selbst aufrufen. Nähere Informationen dazu finden Sie unter [AuthenticRange.PerformAction](#).

```
// ----- javascript sample -----
// instead of:
// var arrElements = New Array();
// var objDocEditView = Application.ActiveDocument.DocEditView;
// var objStartElement = objDocEditView.CurrentSelection.Start;
// var objEndElement = objDocEditView.CurrentSelection.End;
// objDocEditView.GetAllowedElements(k_ActionInsertBefore, objStartElement,
objEndElement, arrElements);
// use now:
var arrElements = New Array();
Application.ActiveDocument.AuthenticView.Selection.CanPerformActionWith
(spyAuthenticInsertBefore, arrElements);
```

Siehe auch

Deklaration: `GetAllowedElements` (*nAction* als [SpyAuthenticElementActions](#), *pStartElement* als [XMLData](#), *pEndElement* als [XMLData](#), *pElements* als Variant)

Beschreibung

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

JavaScript-Beispiel:

```
Function GetAllowed()
{
    var objView = Application.ActiveDocument.DocEditView;
```

```

var arrElements = New Array(1);

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

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

For(var i = 1;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 objView
    Set objView = Application.ActiveDocument.DocEditView

    Dim arrElements()

    Dim objStart
    Dim objEnd
    Set objStart = objView.CurrentSelection.Start
    Set objEnd = objView.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
```

GetNextVisible (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.SelectNext](#)

AuthenticRange unterstützt nun eine breite Palette an Elementnavigationsmethoden basierend auf Dokumentelementen wie Zeichen, Wörter, Tags und vieles mehr. Die Auswahl der Textpassage, die den Inhalt des nächsten XML-Elements darstellt, ist nur eine davon.

```
// ----- javascript sample -----  
// instead of:  
// var objCurrXMLData = ...  
// var objXMLData =  
Application.ActiveDocument.DocEditView.GetNextVisible(objCurrXMLData);  
// Application.ActiveDocument.DocEditView.SelectionSet (objXMLData, 0,  
objXMLData, -1);  
// use now:  
var objRange = ...  
try  
{ objRange.SelectNext (spyAuthenticTag).Select(); }  
catch (err)  
{  
    if ((err.number & 0xffff) == 2003)  
        MsgBox ("end of document reached");  
    else  
        throw (err);  
}
```

Siehe auch

Deklaration: `GetNextVisible` (*pElement* als [XMLData](#)) als [XMLData](#)

Beschreibung

Die Methode ruft das nächste sichtbare XML-Element im Dokument ab.

GetPreviousVisible (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.SelectPrevious](#)

AuthenticRange unterstützt nun eine breite Palette an Elementnavigationsmethoden basierend auf Dokumentelementen wie Zeichen, Wörter, Tags und vieles mehr. Die Auswahl der Textpassage, die den Inhalt des vorherigen XML-Elements darstellt, ist nur eine davon.

```
// ----- javascript sample -----  
// instead of:  
// var objCurrXMLData = ...  
// var objXMLData =  
Application.ActiveDocument.DocEditView.GetPreviousVisible(objCurrXMLData);  
// Application.ActiveDocument.DocEditView.SelectionSet (objXMLData, 0,  
objXMLData, -1);  
// use now:  
var objRange = ...  
try  
    { objRange.SelectPrevious (spyAuthenticTag).Select(); }  
catch (err)  
{  
    if ((err.number & 0xffff) == 2004)  
        MsgBox ("begin of document reached");  
    else  
        throw (err);  
}
```

Siehe auch

Deklaration: `GetPreviousVisible` (*pElement* als [XMLData](#)) als [XMLData](#)

Beschreibung

Die Methode ruft das vorhergehende sichtbare XML-Element im Dokument ab.

IsEditClearEnabled (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.IsDeleteEnabled](#)

Die IsDeleteEnabled-Eigenschaft wird nun für jeden Bereich des Dokuments und nicht nur für die aktuelle Auswahl auf der Benutzeroberfläche unterstützt.

```
// ----- javascript sample -----  
// instead of:  
// if ( Application.ActiveDocument.DocEditView.IsEditClearEnabled)  
//     Application.ActiveDocument.DocEditView.EditClear();  
// use now:  
var objCurrSelection = Application.ActiveDocument.AuthenticView.Selection;  
if ( objCurrSelection.IsDeleteEnabled)  
    objCurrSelection.Delete();
```

Siehe auch

Deklaration: [IsEditClearEnabled](#) als Boolean

Beschreibung

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

IsEditCopyEnabled (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.IsCopyEnabled](#)

Die IsCopyEnabled-Eigenschaft wird nun für jeden Bereich des Dokuments und nicht nur für die aktuelle Auswahl auf der Benutzeroberfläche unterstützt.

```
// ----- javascript sample -----  
// instead of:  
// if ( Application.ActiveDocument.DocEditView.IsEditCopyEnabled)  
//     Application.ActiveDocument.DocEditView.EditCopy();  
// use now:  
var objCurrSelection = Application.ActiveDocument.AuthenticView.Selection;  
if ( objCurrSelection.IsCopyEnabled)  
    objCurrSelection.Copy();
```

Siehe auch

Deklaration: [IsEditCopyEnabled](#) als Boolean

Beschreibung

True, wenn das Kopieren in die Zwischenablage möglich ist. Siehe auch [EditCopy](#) und [Bearbeitungsoperationen.](#)

IsEditCutEnabled (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.IsCutEnabled](#)

Die IsCutEnabled-Eigenschaft wird nun für jeden Bereich des Dokuments und nicht nur für die aktuelle Auswahl auf der Benutzeroberfläche unterstützt.

```
// ----- javascript sample -----  
// instead of:  
// if ( Application.ActiveDocument.DocEditView.IsEditCutEnabled)  
//     Application.ActiveDocument.DocEditView.EditCut();  
// use now:  
var objCurrSelection = Application.ActiveDocument.AuthenticView.Selection;  
if ( objCurrSelection.IsCutEnabled)  
    objCurrSelection.Cut();
```

Siehe auch

Deklaration: [IsEditCutEnabled](#) als Boolean

Beschreibung

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

IsEditPasteEnabled (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.IsPasteEnabled](#)

Die IsPasteEnabled-Eigenschaft wird nun für jeden Bereich des Dokuments und nicht nur für die aktuelle Auswahl auf der Benutzeroberfläche unterstützt.

```
// ----- javascript sample -----  
// instead of:  
// if ( Application.ActiveDocument.DocEditView.IsEditPasteEnabled)  
//     Application.ActiveDocument.DocEditView.EditPaste();  
// use now:  
var objCurrSelection = Application.ActiveDocument.AuthenticView.Selection;  
if ( objCurrSelection.IsPasteEnabled)  
    objCurrSelection.Paste();
```

Siehe auch

Deklaration: [IsEditPasteEnabled](#) als Boolean

Beschreibung

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

IsEditRedoEnabled (nicht mehr verwendet)

Wird ersetzt durch [AuthenticView.IsRedoEnabled](#)

```
// ----- javascript sample -----  
// instead of:  
// if ( Application.ActiveDocument.DocEditView.IsEditRedoEnabled)  
//     Application.ActiveDocument.DocEditView.EditRedo( );  
// use now:  
if ( Application.ActiveDocument.AuthenticView.IsRedoEnabled)  
    Application.ActiveDocument.AuthenticView.Redo( );
```

Siehe auch

Deklaration: [IsEditRedoEnabled](#) als Boolean

Beschreibung

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

IsEditUndoEnabled (nicht mehr verwendet)

Wird ersetzt durch [AuthenticView.IsUndoEnabled](#)

```
// ----- javascript sample -----  
// instead of:  
// if ( Application.ActiveDocument.DocEditView.IsEditUndoEnabled)  
//     Application.ActiveDocument.DocEditView.EditUndo( );  
// use now:  
if ( Application.ActiveDocument.AuthenticView.IsUndoEnabled)  
    Application.ActiveDocument.AuthenticView.Undo( );
```

Siehe auch

Deklaration: [IsEditUndoEnabled](#) als Boolean

Beschreibung

True, wenn [EditUndo](#) derzeit möglich ist. Siehe auch [Bearbeitungsoperationen](#).

IsRowAppendEnabled (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.IsInDynamicTable](#)

Die Operation zum Einfügen, Anhängen, Löschen und Kopieren einer Zeile ('insert', 'append', 'delete' und 'duplicate' row) stehen zur Verfügung, wenn sich die Auswahl innerhalb einer dynamischen Tabelle befindet.

```
// ----- javascript sample -----  
// instead of:  
// if ( Application.ActiveDocument.DocEditView.IsRowAppendEnabled)  
//     Application.ActiveDocument.DocEditView.RowAppend( );  
// use now:  
if ( Application.ActiveDocument.AuthenticView.Selection.IsInDynamicTable( ) )  
    Application.ActiveDocument.AuthenticView.Selection.AppendRow( );
```

Siehe auch

Deklaration: [IsRowAppendEnabled](#) als Boolean

Beschreibung

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

IsRowDeleteEnabled (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.IsInDynamicTable](#)

Die Operation zum Einfügen, Anhängen, Löschen und Kopieren einer Zeile ('insert', 'append', 'delete' und 'duplicate' row) stehen zur Verfügung, wenn sich die Auswahl innerhalb einer dynamischen Tabelle befindet.

```
// ----- javascript sample -----  
// instead of:  
// if ( Application.ActiveDocument.DocEditView.IsRowDeleteEnabled)  
//     Application.ActiveDocument.DocEditView.Rowdelete( );  
// use now:  
if ( Application.ActiveDocument.AuthenticView.Selection.IsInDynamicTable( ) )  
    Application.ActiveDocument.AuthenticView.Selection.DeleteRow( );
```

Siehe auch

Deklaration: [IsRowDeleteEnabled](#) als Boolean

Beschreibung

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

IsRowDuplicateEnabled (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.IsInDynamicTable](#)

Die Operation zum Einfügen, Anhängen, Löschen und Kopieren einer Zeile ('insert', 'append', 'delete' und 'duplicate' row) stehen zur Verfügung, wenn sich die Auswahl innerhalb einer dynamischen Tabelle befindet.

```
// ----- javascript sample -----  
// instead of:  
// if ( Application.ActiveDocument.DocEditView.IsRowDuplicateEnabled)  
//     Application.ActiveDocument.DocEditView.RowDuplicate();  
// use now:  
if ( Application.ActiveDocument.AuthenticView.Selection.IsInDynamicTable() )  
    Application.ActiveDocument.AuthenticView.Selection.DuplicateRow();
```

Siehe auch

Deklaration: [IsRowDuplicateEnabled](#) als Boolean

Beschreibung

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

IsRowInsertEnabled (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.IsInDynamicTable](#)

Die Operation zum Einfügen, Anhängen, Löschen und Kopieren einer Zeile ('insert', 'append', 'delete' und 'duplicate' row) stehen zur Verfügung, wenn sich die Auswahl innerhalb einer dynamischen Tabelle befindet.

```
// ----- javascript sample -----  
// instead of:  
// if ( Application.ActiveDocument.DocEditView.IsRowInsertEnabled)  
//     Application.ActiveDocument.DocEditView.RowInsert();  
// use now:  
if ( Application.ActiveDocument.AuthenticView.Selection.IsInDynamicTable() )  
    Application.ActiveDocument.AuthenticView.Selection.InsertRow();
```

Siehe auch

Deklaration: [IsRowInsertEnabled](#) als Boolean

Beschreibung

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

IsRowMoveDownEnabled (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.IsLastRow](#)

```
// ----- javascript sample -----  
// instead of:  
// if ( Application.ActiveDocument.OldAuthenticView.IsRowMoveDownEnabled)  
//     Application.ActiveDocument.DocEditView.RowMoveDown( );  
// use now:  
if ( ! Application.ActiveDocument.AuthenticView.Selection.IsLastRow)  
    Application.ActiveDocument.AuthenticView.Selection.MoveRowDown( );
```

Siehe auch

Deklaration: [IsRowMoveDownEnabled](#) als Boolean

Beschreibung

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

IsRowMoveUpEnabled (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.IsFirstRow](#)

```
// ----- javascript sample -----  
// instead of:  
// if ( Application.ActiveDocument.DocEditView.IsRowMoveUpEnabled)  
//     Application.ActiveDocument.DocEditView.RowMoveUp( );  
// use now:  
if ( ! Application.ActiveDocument.AuthenticView.Selection.IsFirstRow)  
    Application.ActiveDocument.AuthenticView.Selection.MoveRowUp( );
```

Siehe auch

Deklaration: [IsRowMoveUpEnabled](#) als Boolean

Beschreibung

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

IsTextStateApplied (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.IsTextStateApplied](#)

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.DocEditView.IsTextStateApplied ("bold");  
// use now:  
if ( Application.ActiveDocument.AuthenticView.Selection.IsTextStateApplied (  
"bold"))  
    MsgBox ("bold on");  
else  
    MsgBox ("bold off");
```

Siehe auch

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

Beschreibung

Überprüft, ob der Textstatus bereits angewendet wurde. Typische Beispiele für den Parameter *elementName* sind "strong" und "italic".

IsTextStateEnabled (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.CanPerformAction](#)

Verwenden Sie `spyAuthenticApply` für den *eAction*-Parameter. Mit Hilfe der `CanPerformAction` Methode können Sie an jedem Bereich des Dokuments arbeiten und nicht nur an dem aktuell auf der Benutzeroberfläche ausgewählt.

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.DocEditView.IsTextStateEnabled ("bold");  
// use now:  
if ( Application.ActiveDocument.AuthenticView.Selection.CanPerformAction  
(spyAuthenticApply, "bold"))  
    ... // e.g. enable 'bold' button
```

Siehe auch

Deklaration: `IsTextStateEnabled` (*i_strElementName* als String) als Boolean

Beschreibung

Überprüft, ob ein Textstatus angewendet werden kann. Typische Beispiele für den Parameter *elementName* sind "strong" und "italic".

LoadXML (nicht mehr verwendet)

Wird ersetzt durch [AuthenticView.AsXMLString](#)

AuthenticView unterstützt nun die Eigenschaft AsXMLString, über die Sie den Dokumentinhalt direkt als XMLString aufrufen und ersetzen können.

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.DocEditView.LoadXML ( strDocAsXMLString );  
// use now:  
try  
{ Application.ActiveDocument.AuthenticView.AsXMLString =  
strDocAsXMLString; }  
catch ( err )  
{ MsgBox ( "Error: invalid XML string" ); }
```

Siehe auch

Deklaration: `LoadXML` (*xmlString* als String)

Beschreibung

Lädt das aktuelle XML-Dokument, wobei der XML-String angewendet wird. Der neue Inhalt wird sofort angezeigt.

Der Parameter *xmlString* muss mit der XML-Deklaration beginnen, z.B.,
`objPlugIn.LoadXML("<?xml version='1.0' encoding='UTF-8'?><root></root>");`

MarkupView (nicht mehr verwendet)

Wird ersetzt durch [AuthenticView.MarkupVisibility](#)

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.DocEditView.MarkuPView = 2;  
// use now:  
Application.ActiveDocument.AuthenticView.MarkupVisibility =  
spyAuthenticMarkupLarge;
```

Siehe auch

Deklaration: `MarkupView` (*kind* als long)

Beschreibung

Standardmäßig wird das Dokument mittels HTML-Verfahren angezeigt. Manchmal ist es aber sinnvoll, wenn die Bearbeitungstags angezeigt werden. Mit dieser Methode können drei unterschiedliche Arten von Markup Tags angezeigt werden:

- 0 Markup Tags ausblenden
- 2 Große Markup Tags einblenden

- 3 Gemischte Markup Tags
 einblenden

RowAppend (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.AppendRow](#)

Mit Hilfe der Tabellenoperationen von AuthenticRange kann nun jede Tabelle im aktuellen Dokument unabhängig von der aktuellen Auswahl auf der Benutzeroberfläche bearbeitet werden.

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.DocEditView.RowAppend();  
// use now:  
if (! Application.ActiveDocument.AuthenticView.Selection.AppendRow())  
    MsgBox ("Error: can't append row");
```

Siehe auch

Deklaration: [RowAppend](#)

Beschreibung

Hängt eine Zeile an der aktuellen Position an.

Siehe auch [Zeilenoperationen](#).

RowDelete (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.DeleteRow](#)

Mit Hilfe der Tabellenoperationen von AuthenticRange kann nun jede Tabelle im aktuellen Dokument unabhängig von der aktuellen Auswahl auf der Benutzeroberfläche bearbeitet werden.

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.DocEditView.RowDelete();  
// use now:  
if (! Application.ActiveDocument.AuthenticView.Selection.DeleteRow())  
    MsgBox ("Error: can't delete row");
```

Siehe auch

Deklaration: [RowDelete](#)

Beschreibung

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

Siehe auch [Zeileoperationen](#).

RowDuplicate (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.DuplicateRow](#)

Mit Hilfe der Tabellenoperationen von AuthenticRange kann nun jede Tabelle im aktuellen Dokument unabhängig von der aktuellen Auswahl auf der Benutzeroberfläche bearbeitet werden.

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.DocEditView.RowDuplicate();  
// use now:  
if (! Application.ActiveDocument.AuthenticView.Selection.DuplicateRow())  
    MsgBox ("Error: can't duplicate row");
```

Siehe auch

Deklaration: [RowDuplicate](#)

Beschreibung

Die Methode kopiert die aktuell ausgewählten Zeilen.

Siehe auch [Zeilenoperationen](#).

RowInsert (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.InsertRow](#)

Mit Hilfe der Tabellenoperationen von AuthenticRange kann nun jede Tabelle im aktuellen Dokument unabhängig von der aktuellen Auswahl auf der Benutzeroberfläche bearbeitet werden.

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.DocEditView.RowInsert();  
// use now:  
if (! Application.ActiveDocument.AuthenticView.Selection.InsertRow())  
    MsgBox ("Error: can't insert row");
```

Siehe auch

Deklaration: [RowInsert](#)

Beschreibung

Fügt eine neue Zeile unmittelbar oberhalb der aktuellen Auswahl ein.

Siehe auch [Zeilenoperationen](#).

RowMoveDown (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.MoveRowDown](#)

Mit Hilfe der Tabellenoperationen von AuthenticRange kann nun jede Tabelle im aktuellen Dokument unabhängig von der aktuellen Auswahl auf der Benutzeroberfläche bearbeitet werden.

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.DocEditView.RowMoveDown();  
// use now:  
if (! Application.ActiveDocument.AuthenticView.Selection.MoveRowDown())  
    MsgBox ("Error: can't move row down");
```

Siehe auch

Deklaration: [RowMoveDown](#)

Beschreibung

Verschiebt die aktuelle Zeile um eine Position nach unten.

Siehe auch [Zeilenoperationen](#).

RowMoveUp (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.MoveRowUp](#)

Mit Hilfe der Tabellenoperationen von AuthenticRange kann nun jede Tabelle im aktuellen Dokument unabhängig von der aktuellen Auswahl auf der Benutzeroberfläche bearbeitet werden.

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.DocEditView.RowAppend();  
// use now:  
if (! Application.ActiveDocument.AuthenticView.Selection.MoveRowUp())  
    MsgBox ("Error: can't move row up");
```

Siehe auch

Deklaration: [RowMoveUp](#)

Beschreibung

Verschiebt die aktuelle Zeile um eine Position nach oben.

Siehe auch [Zeilenoperationen](#).

SaveXML (nicht mehr verwendet)

Wird ersetzt durch [AuthenticView.AsXMLString](#)

AuthenticView unterstützt nun die Eigenschaft XMLString, mit deren Hilfe Sie den Dokumentinhalt direkt aufrufen und als XMLString ersetzen können.

```
// ----- javascript sample -----  
// instead of:  
// var strDocAsXMLString = Application.ActiveDocument.DocEditView.SaveXML( );  
// use now:  
try  
{  
    var strDocAsXMLString =  
Application.ActiveDocument.AuthenticView.AsXMLString;  
    ... // do something here  
}  
catch (err)  
{ MsgBox ("Error: invalid XML string"); }
```

Siehe auch

Deklaration: [SaveXML](#) als String

Rückgabewert

XML-Struktur als String.

Beschreibung

Speichert die aktuellen XML-Daten als String, der an den Aufrufenden zurückgegeben wird.

SelectionModeTabOrder (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.SelectNext](#)

AuthenticRange unterstützt nun eine breite Palette an Elementnavigationsmethoden basierend auf Dokumentelementen wie Zeichen, Wörter, Tags und vieles mehr. Das Auswählen des nächsten Absatzes ist nur eine davon und die Navigation muss nicht notwendigerweise auf die aktuelle Auswahl auf der Benutzeroberfläche beschränkt sein.

```
// ----- javascript sample -----  
// instead of:  
// Application.ActiveDocument.DocEditView.SelectionMoveTabOrder(true, true);  
// use now:  
Application.ActiveDocument.AuthenticView.Selection.SelectNext  
(spyAuthenticParagraph).Select();  
// to append a row to a table use AuthenticRange.AppendRow
```

Siehe auch

Dkclaration: `SelectionModeTabOrder` (*bForward* als Boolean, *bTag* als Boolean)

Beschreibung

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

Wenn *bTag* "false" ist und sich die aktuelle Auswahl in der letzten Zelle einer Tabelle befindet, wird eine neue Zeile hinzugefügt.

SelectionSet (nicht mehr verwendet)

Wird ersetzt durch [AuthenticRange.FirstXMLData](#) und damit in Zusammenhang stehenden Eigenschaften

AuthenticRange unterstützt sowohl die Navigation über XMLData-Elemente als auch die Navigation nach Dokumentelementen (z.B. Zeichen, Wörter, Tags) oder Textcursorpositionen.

```
// ----- javascript sample -----  
// instead of:  
// if (! Application.ActiveDocument.DocEditView.SelectionSet(varXMLData1, 0,  
varXMLData2, -1))  
//     MsgBox ("Error: invalid data position");  
// use now:  
try  
{  
    var objSelection = Application.ActiveDocument.AuthenticView.Selection;  
    objSelection.FirstXMLData = varXMLData1;  
    objSelection.FirstXMLdataOffset = 0;  
    objSelection.LastXMLData = varXMLData2;  
    objSelection.LastXMLdataOffset = -1;  
    objSelection.Select();  
}  
catch (err)  
{ MsgBox ("Error: invalid data position"); }  
// to select all text between varXMLData1 and varXMLdata2, inclusive
```

Siehe auch

Deklaration: `SelectionSet` (*pStartElement* als [XMLData](#), *nStartPos* als long, *pEndElement* als [XMLData](#), *nEndPos* als long) als Boolean

Beschreibung

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

XMLRoot (nicht mehr verwendet)

Wird ersetzt durch [AuthenticView.XMLDataRoot](#)

```
// ----- javascript sample -----  
// instead of:  
// var objXMLData = Application.ActiveDocument.DocEditView.XMLRoot;  
// use now:  
var objXMLData = Application.ActiveDocument.AuthenticView.XMLDataRoot;
```

Siehe auch

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

Beschreibung

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

Nähere Informationen dazu siehe auch [Verwenden von XMLData](#).

4.5 Enumerations

Dies ist eine Liste aller von der Authentic Desktop API verwendeten Enumerations. Wenn Ihre Skripting-Umgebung Enumerations nicht unterstützt, verwenden Sie stattdessen die Zahlenwerte.

4.5.1 SPYAttributeTypeDefinition

Beschreibung

Attributtypdefinition, die zur Generierung einer XML-Beispieldatei ausgewählt werden kann. Dieser Typ wird mit den Methoden [GenerateDTDOOrSchema](#) und [GenerateDTDOOrSchemaEx](#) verwendet.

Mögliche Werte:

spyMergedGlobal	= 0
spyDistinctGlobal	= 1
spyLocal	= 2

4.5.2 SPYAuthenticActions

Beschreibung

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

Mögliche Werte:

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

4.5.3 SPYAuthenticDocumentPosition

Beschreibung

Relative und absolute Position für die Navigation mit [AuthenticRange](#) Objekten.

Mögliche Werte:

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

4.5.4 SPYAuthenticElementActions

Beschreibung

Aktionen die mit [GetAllowedElements](#) verwendet werden können (Wird ersetzt durch [AuthenticRange.CanPerformActionWith](#)).

Mögliche Werte:

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

4.5.5 SPYAuthenticElementKind

Beschreibung

Enumeration der verschiedenen Arten von Elementen, die für die Navigation und Auswahl innerhalb der [AuthenticRange](#) und [AuthenticView](#) Objekte verwendet werden.

Mögliche Werte:

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

4.5.6 SPYAuthenticMarkupVisibility

Beschreibung

Enumerationswerte zum Anpassen der Sichtbarkeit von Markup-Tags mit [MarkupVisibility](#).

Mögliche Werte:

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

4.5.7 SPYDatabaseKind

Beschreibung

Werte zur Auswahl verschiedener Datenbankarten für den Import. Informationen zur Verwendung finden Sie unter [DatabaseConnection.DatabaseKind](#).

Mögliche Werte:

spyDB_Access	= 0
spyDB_SQLServer	= 1
spyDB_Oracle	= 2
spyDB_Sybase	= 3
spyDB_MySQL	= 4
spyDB_DB2	= 5
spyDB_Other	= 6
spyDB_Unspecified	= 7

4.5.8 SPYDialogAction

Beschreibung

Werte zur Simulation verschiedener Interaktionen in Dialogfeldern. Informationen dazu, welche Dialogfelder zur Verfügung stehen, finden Sie unter [Dialogs](#).

Mögliche Werte:

spyDialogOK	= 0	//simulate click on OK button
spyDialogCancel	= 1	//simulate click on Cancel button
spyDialogUserInput	= 2	// show dialog and allow user interaction

4.5.9 SPYDOMType

Beschreibung

Enumerationswerte zum Parametrisieren der Generierung von C++ Code anhand von Schemadefinitionen.

Mögliche Werte:

spyDOMType_msxml4	= 0
spyDOMType_xerces	= 1

4.5.10 SPYDTDSchemaFormat

Beschreibung

Enumeration zum Identifizieren der verschiedenen Schemaformate.

Mögliche Werte:

spyDTD	= 0
spyW3C	= 1

4.5.11 SPYEncodingByteOrder

Beschreibung

Enumerationswerte zur Definition der Codierungsbytefolge für den Textimport und -export.

Mögliche Werte:

spyNONE	= 0
spyLITTLE_ENDIAN	= 1
spyBIG_ENDIAN	= 2

4.5.12 SPYExportNamespace

Beschreibung

Enumerationstyp, um die Behandlung von Namespace Identifiers beim Export zu konfigurieren.

Mögliche Werte:

spyNoNamespace	= 0
spyReplaceColonWithUnderscore	= 1

4.5.13 SPYFindInFilesSearchLocation

Beschreibung

Die verschiedenen Optionen, wo eine Suche durchgeführt werden kann. Dieser Typ wird mit dem Dialogfeld [FindInFilesDlg](#) verwendet.

Mögliche Werte:

spyFindInFiles_Documents	= 0
spyFindInFiles_Project	= 1
spyFindInFiles_Folder	= 2

4.5.14 SPYFrequentElements

Beschreibung

Enumerationswert zum Parametrisieren der Schemagenerierung.

Mögliche Werte:

spyGlobalElements	= 0
spyGlobalComplexType	= 1

4.5.15 SPYImageKind

Beschreibung

Enumerationswerte zum Parametrisieren des Bildtyps der generierten Dokumentation. Diese Werte werden in [SchemaDocumentationDiagramFormat](#) und [WSDLDocumentationDiagramFormat](#) verwendet.

Mögliche Werte:

spyImageType_PNG	= 0
spyImageType_EMF	= 1

4.5.16 SPYKeyEvent

Beschreibung

Enumerationstyp zum Identifizieren der verschiedenen Tastatur-Events. Diese Events entsprechen den gleichnamigen Windows-Messages.

Mögliche Werte:

spyKeyDown	= 0
spyKeyUp	= 1
spyKeyPressed	= 2

4.5.17 SPYLibType

Beschreibung

Enumerationswerte zum Parametrisieren der Generierung von C++ Code anhand von Schemadefinitionen.

Mögliche Werte:

spyLibType_static	= 0
spyLibType_dll	= 1

4.5.18 SPYLoading

Beschreibung

Enumerationswerte zum Definieren des Ladeverhaltens von URL-Dateien.

Mögliche Werte:

spyUseCacheProxy	= 0
spyReload	= 1

4.5.19 SPYMouseEvent

Beschreibung

Enumerationstyp, der die Mausstatus während eines Maus-Events definiert. Verwenden Sie diese Enumerationswerte als Bitmasken anstatt direkte Vergleiche damit durchzuführen.

Beispiele

```
' to check for ctrl-leftbutton-down in VB
If (i_eMouseEvent = (XMLSpyLib.spyLeftButtonDownMask Or
XMLSpyLib.spyCtrlKeyDownMask)) Then
    ' react on ctrl-leftbutton-down
End If

' to check for double-click with any button in VBScript
If (((i_eMouseEvent And spyDoubleClickMask) <> 0) Then
    ' react on double-click
End If
```

Mögliche Werte:

spyNoButtonMask	= 0	
spyMouseMoveMask	= 1	
spyLeftButtonMask	= 2	
spyMiddleButtonMask	= 4	
spyRightButtonMask	= 8	
spyButtonUpMask	= 16	
spyButtonDownMask	= 32	
spyDoubleClickMask	= 64	
spyShiftKeyDownMask	= 128	
spyCtrlKeyDownMask	= 256	
spyLeftButtonDownMask	= 34	// spyLeftButtonMask spyButtonDownMask
spyMiddleButtonDownMask	= 36	// spyMiddleButtonMask spyButtonDownMask
spyRightButtonDownMask	= 40	// spyRightButtonMask spyButtonDownMask
spyLeftButtonUpMask	= 18	// spyLeftButtonMask spyButtonUpMask
spyMiddleButtonUpMask	= 20	// spyMiddleButtonMask spyButtonUpMask
spyRightButtonUpMask	= 24	// spyRightButtonMask spyButtonUpMask
spyLeftDoubleClickMask	= 66	// spyRightButtonMask spyButtonUpMask
spyMiddleDoubleClickMask	= 68	// spyMiddleButtonMask spyDoubleClickMask
spyRightDoubleClickMask	= 72	// spyRightButtonMask spyDoubleClickMask

4.5.20 SPYNumberDateTimeFormat

Beschreibung

Enumerationswert zum Konfigurieren von Datenbankverbindungen.

Mögliche Werte:

spySystemLocale = 0
spySchemaCompatible = 1

4.5.21 SPYProgrammingLanguage

Beschreibung

Enumerationswerte zur Auswahl der Programmiersprache für die Codegenerierung von Schemadefinitionen.

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Mögliche Werte:

spyUndefinedLanguage	= -1
spyJava	= 0
spyCpp	= 1
spyCSharp	= 2

4.5.22 SPYProjectItemTypes

Beschreibung

Enumerationswerte zum Identifizieren der verschiedenen Elemente in Projektelementlisten.

Siehe [SpyProjectItem.ItemType](#).

Mögliche Werte:

spyUnknownItem	= 0
spyFileItem	= 1
spyFolderItem	= 2
spyURLItem	= 3

4.5.23 SPYProjectType

Beschreibung

Enumerationswerte zum Parametrisieren der Generierung von C# Code anhand von Schemadefinitionen.

Mögliche Werte:

spyVisualStudioProject	= 0
spyVisualStudio2003Project	= 1
spyBorlandProject	= 2
spyMonoMakefile	= 3
spyVisualStudio2005Project	= 4

4.5.24 SPYSampleXMLGenerationOptimization

Beschreibung

Definiert die Elemente, die in der XML-Beispieldatei generiert werden. Diese Enumeration wird in [GenerateSampleXMLDlg](#) verwendet.

Mögliche Werte:

spySampleXMLGen_Optimized	= 0
spySampleXMLGen_NonMandatoryElements	= 1
spySampleXMLGen_Everything	= 2

4.5.25 SPYSampleXMLGenerationSchemaOrDTDAssignment

Beschreibung

Definiert, welche Art von Referenz zum Schema/zur DTD zur generierten XML-Beispieldatei hinzugefügt werden soll.

Diese Enumeration wird in [GenerateSampleXMLDlg](#) verwendet.

Mögliche Werte:

spySampleXMLGen_AssignRelatively	= 0
spySampleXMLGen_AssignAbsolutely	= 1
spySampleXMLGen_DoNotAssign	= 2

4.5.26 SPYSchemaDefKind

Beschreibung

Enumerationstyp zur Auswahl von Schemadiagrammtypen.

Mögliche Werte:

spyKindElement	= 0
spyKindComplexType	= 1
spyKindSimpleType	= 2
spyKindGroup	= 3
spyKindModel	= 4
spyKindAny	= 5
spyKindAttr	= 6
spyKindAttrGroup	= 7
spyKindAttrAny	= 8
spyKindIdentityUnique	= 9
spyKindIdentityKey	= 10
spyKindIdentityKeyRef	= 11
spyKindIdentitySelector	= 12
spyKindIdentityField	= 13
spyKindNotation	= 14
spyKindInclude	= 15
spyKindImport	= 16
spyKindRedefine	= 17
spyKindFacet	= 18
spyKindSchema	= 19
spyKindCount	= 20

4.5.27 SPYSchemaDocumentationFormat

Beschreibung

Enumerationswerte zum Parametrisieren der Generierung von Schemadokumentation. Diese Werte werden in [SchemaDocumentationDbg.OutputFormat](#) und [WSDLDocumentationDbg.OutputFormat](#) verwendet.

Mögliche Werte:

spySchemaDoc_HTML	= 0
spySchemaDoc_MSWord	= 1
spySchemaDoc_RTF	= 2

4.5.28 SPYTextDelimiters

Beschreibung

Enumerationswerte zum Definieren der Texttrennzeichen für den Textexport.

Mögliche Werte:

spyTabulator	= 0
spySemicolon	= 1
spyComma	= 2
spySpace	= 3

4.5.29 SPYTextEnclosing

Beschreibung

Enumerationswert zum Definieren der den Text einschließenden Zeichen beim Textimport und -export.

Mögliche Werte:

spyNoEnclosing	= 0
spySingleQuote	= 1
spyDoubleQuote	= 2

4.5.30 SPYTypeDetection

Beschreibung

Enumeration, um auszuwählen, wie die Typerkennung bei [GenerateDTDoSchema](#) und [GenerateDTDoSchemaEx](#) funktioniert.

Mögliche Werte:

spyBestPossible	= 0
spyNumbersOnly	= 1
spyNoDetection	= 2

4.5.31 SPYURLTypes

Beschreibung

Enumeration zum Definieren der verschiedenen URL-Typen.

Mögliche Werte:

spyURLTypeAuto = -1
spyURLTypeXML = 0
spyURLTypeDTD = 1

4.5.32 SPYViewModes

Beschreibung

Enumerationswerte, die die verschiedenen Ansichtsmodi für XML-Dokumente definieren. Der Modus *spyViewAuthentic(4)* identifiziert den Modus, der zwischenzeitlich DocEdit-Modus genannt wurde und nun Authentic-Modus heißt. Der Modus *spyViewWSDL* identifiziert einen Modus, der auf der Benutzeroberfläche der Schema-Ansicht zugeordnet ist, intern aber unterschiedlich behandelt wird.

Mögliche Werte:

<code>spyViewGrid</code>	<code>= 0</code>	
<code>spyViewText</code>	<code>= 1</code>	
<code>spyViewBrowser</code>	<code>= 2</code>	
<code>spyViewSchema</code>	<code>= 3</code>	
<code>spyViewContent</code>	<code>= 4</code>	// nicht mehr verwendet
<code>spyViewAuthentic</code>	<code>= 4</code>	
<code>spyViewWSDL</code>	<code>= 5</code>	
<code>spyViewZIP</code>	<code>= 6</code>	
<code>spyViewEditionInfo</code>	<code>= 7</code>	
<code>spyViewXBRL</code>	<code>= 8</code>	

4.5.33 SPYVirtualKeyMask

Beschreibung

Enumeratiostyp für die am häufigsten verwendeten Tastaturmasken, die den Status der virtuellen Tasten identifizieren. Verwenden Sie diese Werte als Bitmasken, anstatt direkte Vergleiche damit durchzuführen. Wenn nötig können Sie mit Hilfe des 'logical or' Operators weitere Masken erstellen.

Beispiele

```
' VBScript sample: check if ctrl-key is pressed
If ((i_nVirtualKeyStatus And spyCtrlKeyMask) <> 0) Then
    ' ctrl-key is pressed
End If

' VBScript sample: check if ONLY ctrl-key is pressed
If (i_nVirtualKeyStatus == spyCtrlKeyMask) Then
    ' exactly ctrl-key is pressed
End If

// JScript sample: check if any of the right virtual keys is pressed
if ((i_nVirtualKeyStatus & (spyRightShiftKeyMask | spyRightCtrlKeyMask |
spyRightAltKeyMask)) != 0)
{
    ; ' right virtual key is pressed
}
```

Mögliche Werte:

spyNoVirtualKeyMask	= 0	
spyLeftShiftKeyMask	= 1	
spyRightShiftKeyMask	= 2	
spyLeftCtrlKeyMask	= 4	
spyRightCtrlKeyMask	= 8	
spyLeftAltKeyMask	= 16	
spyRightAltKeyMask	= 32	
spyShiftKeyMask	= 3	// spyLeftShiftKeyMask spyRightShiftKeyMask
spyCtrlKeyMask	= 12	// spyLeftCtrlKeyMask spyRightCtrlKeyMask
spyAltKeyMask	= 48	// spyLeftAltKeyMask spyRightAltKeyMask

4.5.34 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
spyXMLDataPI	= 9
spyXMLDataDefDoctype	= 10
spyXMLDataDefExternalID	= 11
spyXMLDataDefElement	= 12
spyXMLDataDefAttlist	= 13
spyXMLDataDefEntity	= 14
spyXMLDataDefNotation	= 15
spyXMLDataKindsCount	= 16

5 Verwendung der XMLSpy API mit Java

Die XMLSpy API hat eine Benutzeroberfläche, die auf Java-Klassen aufbaut, wobei jede dieser Klassen einem Objekt in der XMLSpy API entspricht. Entwickler können mit Hilfe dieser Java-Klassen mit der COM API interagieren. Diese Klassen sind unten aufgelistet und in den Unterabschnitten näher beschrieben. Eine Beschreibung der XMLSpy API Objekte selbst finden Sie in der [XMLSpy API-Dokumentation](#). Beachten Sie, dass einige API-Funktionen nur in Skripting-Umgebungen zur Verfügung stehen; diese wurden daher nicht nach Java portiert.

Java-Klassen

[SpyApplication](#)
 [SpyProject](#)
 [SpyProjectItems](#)
 [SpyProjectItem](#)
 [SpyDocuments](#)
 [SpyDoc](#)
 [SpyAuthenticView](#)
 [SpyAuthenticRange](#)
 [SpyDocEditView](#)
 [SpyDocEditSelection](#)
 [SpyGridView](#)
 [SpyTextView](#)
 [SpyXMLData](#)
 [SpyDialogs](#)
 [SpyCodeGeneratorDlg](#)
 [SpyDTDSchemaGeneratorDlg](#)
 [SpyFileSelectionDlg](#)
 [SpyFindInFilesDlg](#)
 [SpyGenerateSampleXMLDlg](#)
 [SpySchemaDocumentationDlg](#)
 [SpyDatabaseConnection](#)
 [SpyElementList](#)
 [SpyElementListItem](#)
 [SpyExportSettings](#)
 [SpyFindInFilesResults](#)
 [SpyFindInFilesResult](#)
 [SpyFindInFilesMatch](#)
 [SpyTextImportExportSettings](#)

Implementierung von COM-Eigenschaften in Java

Die Eigenschaften in Java wurden so definiert, dass Sie sowohl eine **set** als auch eine **get** Methode enthalten (set wenn dies die COM-Implementierung zulässt). So enthält z.B. die COM-Klasse `Document` die Eigenschaft **GridView**. In Java heißt die Methode `SpyDoc` und die Eigenschaft ist als eine `GetGridView` Methode definiert.

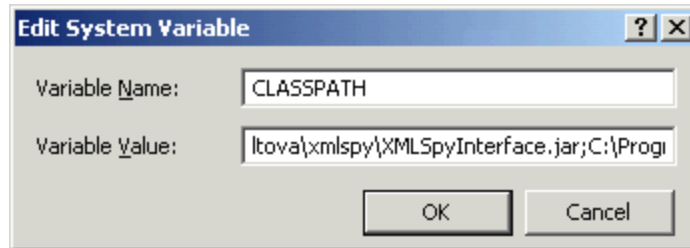
Überprüfen Sie bei Kompilierproblemen die folgenden Punkte:

- Die `xmlspylib.dll` muss in `.. \windows\system32` verfügbar sein.
- Die Datei `XMLSpyInterface.jar` muss in die ClassPath-Umgebungsvariable eingesetzt werden.

Setzen der ClassPath Variable in Windows 2000 und XP

1. Klicken Sie auf **Start | Einstellungen | Systemsteuerung | System | Erweitert | Umgebungsvariablen**. Daraufhin wird das Dialogfeld "Umgebungsvariablen" geöffnet.

2. Wenn in der Gruppe der Systemvariablen bereits ein `ClassPath` Eintrag vorhanden ist, wählen Sie den Eintrag `ClassPath` aus und klicken Sie auf die Schaltfläche **Bearbeiten**. Bearbeiten Sie den Pfad zu: "C: \Program Files\Altova\xmlspy\XMLSpyInterface.jar".



Wenn in der Gruppe der Systemvariablen noch kein Eintrag `ClassPath` vorhanden ist, klicken Sie auf die Schaltfläche **Neu**. Daraufhin erscheint das Dialogfeld "Neue Systemvariable". Geben Sie als Name der Variable `CLASSPATH` ein und "C: \Program Files\Altova\xmlspy\XMLSpyInterface.jar" als die `ClassPath` Variable (ändern Sie gegebenenfalls den Pfad zu ihrer Installation).

5.1 Sample source code

Der Befehlsparameter "SpyDoc doc = app.GetDocuments().OpenFile(...)" muss für Ihre Umgebung angepasst werden.

Das Beispiel führt folgende Schritte durch:

- Es startet eine neue XMLSpy-Instanz
- Es öffnet die Datei Datasheet.xml (ändern Sie den Pfad hier...)
- Es wechselt in die Grid-Ansicht
- Es hängt ein neues Child-Element namens "NewChild" mit dem Textwertelement "NewValuE" an das Root-Element an
- Es überprüft, ob das Dokument gültig ist und gibt eine Meldung an die Java-Konsole aus
- Es schließt die XMLSpy-Applikation und gibt diese frei

```
import XMLSpyInterface.*;

public class TestSpyInterface
{
    public TestSpyInterface() {}

    public static void main(String[] args)
    {
        SpyApplication app = null;
        SpyDoc doc = null;
        SpyXMLData oData = null;
        SpyXMLData oNewChild = null;

        try
        {
            app = new SpyApplication();
            app.ShowApplication( true );

            doc = app.GetDocuments().OpenFile("C:\\FilePath\\OrgChart.xml", true
            );

            // OrgChart.xml is in the folder C:\Documents and
            Settings\<username>My Documents\Altova\XMLSpy\. The filepath should
            be in
            // the form: C:\\Documents and
            Settings\\Username\\Folder\\Filename.xml

            if ( doc != null )
            {
                doc.SwitchViewMode( SPYViewModes.spyViewGrid );
                oData = doc.GetRootElement();
                oNewChild = doc.CreateChild( SPYXMLDataKind.spyXMLDataElement );

                oNewChild.SetName( "NewChild" );
                oNewChild.SetTextValue( "newValuE" );
                oData.AppendChild( oNewChild );

                if ( doc.IsValid() == false )
                {
                    // is to be expected after above insertion
                    System.out.println( "!!!!!!validation error: " +
                        doc.GetErrorString() );
                    System.out.println( "!!!!!!validation error: " +
                        doc.GetErrorPos() );
                    System.out.println( "!!!!!!validation error: " +
```

```
        doc.GetBadData() );
    }
}

app.Quit();
}
finally
{
    // Free any allocated resources by calling ReleaseInstance().
    if ( oNewChild != null )
        oNewChild.ReleaseInstance();

    if ( oData != null )
        oData.ReleaseInstance();

    if ( doc != null )
        oDoc.ReleaseInstance();

    if ( app != null )
        app.ReleaseInstance();
}
}
```

Wenn Sie Probleme haben, dieses Beispiel zu kompilieren, versuchen Sie bitte in der Befehlszeile (**Start | Ausführen | cmd**) die folgenden Befehle. Stellen Sie vorher sicher, dass Sie sich gerade in dem Ordner befinden, der die Java-Beispieldatei enthält.

Kompilierung

```
javac -classpath c:\IhrPfadhier\XMLSpyInterface.jar testspyinterface.java
```

Ausführung

```
java -classpath c:\IhrPfadhier\XMLSpyInterface.jar testspyinterface
```

5.2 SpyApplication

```

public class SpyApplication
{
    public void ReleaseInstance();
    public void ShowApplication( boolean bShow );
    public void Quit();
    public void AddMacroMenuItem( String sMacro, String sDisplayText );
    public void ClearMacroMenu();
    public SpyDoc GetActiveDocument();
    public SpyProject GetCurrentProject();
    public SpyDocuments GetDocuments();
    public SpyElementList GetDatabaseImportElementList( SpyDatabaseConnection
oImportSettings );
    public SpyDatabaseConnection GetDatabaseSettings();
    public SpyElementList GetDatabaseTables( SpyDatabaseConnection
oImportSettings );
    public SpyExportSettings GetExportSettings();
    public SpyElementList GetTextImportElementList( SpyTextImportExportSettings
oImportSettings );
    public SpyTextImportExportSettings GetTextImportExportSettings();
    public SpyDoc ImportFromDatabase( SpyDatabaseConnection oImportSettings,
SpyElementList oElementList );
    public SpyDoc ImportFromSchema( SpyDatabaseConnection oImportSettings,
String strTable, SpyDoc oSchemaDoc );
    public SpyDoc ImportFromText( SpyTextImportExportSettings oImportSettings,
SpyElementList oElementList );
    public SpyDoc ImportFromWord( String sFile );
    public void NewProject( String sPath, boolean bDiscardCurrent );
    public void OpenProject(String sPath , boolean bDiscardCurrent, boolean
bDialog );
    public long ShowForm( String sName );
    public void URLDelete( String sURL, String sUser, String sPassword );
    public void URLMakeDirectory( String sURL, String sUser, String sPassword );

    public int GetWarningNumber();
    public String GetWarningText();

    // since Version 2004R4
    public SpyApplication GetApplication();
    public SpyApplication GetParent();
    public SpyDialogs GetDialogs();
    public boolean GetVisible();
    public void SetVisible( boolean i_bVisibility );
    public long GetWindowHandle();

    public void ReloadSettings();
    public SpyFindInFilesResults FindInFiles( SpyFindInFilesDlg dlgSettings );
    public boolean ShowFindInFiles( SpyFindInFilesDlg dlgSettings );
    public void Selection( String sVal );
}

```

5.3 SpyCodeGeneratorDlg

Steht nur in der Enterprise Edition zur Verfügung. Bei Aufruf über eine andere Version wird eine Fehlermeldung ausgegeben.

```
// since version 2004R4
public class SpyCodeGeneratorDlg
{
    public void ReleaseInstance\(\);
    public SpyApplication GetApplication\(\);
    public SpyDialogs GetParent\(\);
    public long GetProgrammingLanguage\(\);
    public void SetProgrammingLanguage( long i_eVal );
    public String GetTemplateFileName\(\);
    public void SetTemplateFileName( String i_strVal );
    public String GetOutputPath\(\);
    public void SetOutputPath( String i_strVal );
    public long GetOutputPathDialogAction\(\);
    public void SetOutputPathDialogAction( long i_eVal );
    public long GetPropertySheetDialogAction\(\);
    public void SetPropertySheetDialogAction( long i_eVal );
    public long GetOutputResultDialogAction\(\);
    public void SetOutputResultDialogAction( long i_eVal );
    public long GetCPPSettings\_DOMType\(\);
    public void SetCPPSettings\_DOMType( long i_eVal );
    public long GetCPPSettings\_LibraryType\(\);
    public void SetCPPSettings\_LibraryType( long i_eVal );
    public boolean GetCPPSettings\_UseMFC\(\);
    public void SetCPPSettings\_UseMFC( boolean i_bVal );
    public long GetCSharpSettings\_ProjectType\(\);
    public void SetCSharpSettings\_ProjectType( long i_eVal );
}
```

5.4 SpyDatabaseConnection

```
public class SpyDatabaseConnection
{
    public void ReleaseInstance();
    public String GetADOConnection();
    public void SetADOConnection( String sValue );
    public boolean GetAsAttributes();
    public void SetAsAttributes( boolean bValue );
    public boolean GetCreateMissingTables();
    public void SetCreateMissingTables( boolean bValue );
    public boolean GetCreateNew();
    public void SetCreateNew( boolean bValue );
    public boolean GetExcludeKeys();
    public void SetExcludeKeys( boolean bValue );
    public String GetFile();
    public void SetFile( String sValue );
    public boolean GetIncludeEmptyElements();
    public void SetIncludeEmptyElements( boolean bValue );
    public long GetNumberDateTimeFormat();
    public void SetNumberDateTimeFormat( long nValue );
    public String GetODBCConnection();
    public void SetODBCConnection( String sValue );
    public String GetSQLSelect();
    public void SetSQLSelect( String sValue );
    public long GetTextFieldLen();
    public void SetTextFieldLen( long nValue );

    // since version 2004R4
    public long GetDatabaseKind();
    public void SetDatabaseKind( long nValue );

    // since version 2008R2
    public boolean GetCommentIncluded();
    public void SetCommentIncluded( boolean bValue );
    public String GetNullReplacement();
    public void SetNullReplacement( String sValue );
    public String GetDatabaseSchema();
    public void SetDatabaseSchema( String sValue );
}
```

5.5 SpyDialogs

```
// Since version 2004R4
public class SpyDialogs
{
    public SpyApplication GetApplication();
    public SpyApplication GetParent();
    public SpyCodeGeneratorDlg GetCodeGeneratorDlg();
    public SpyFileSelectionDlg GetFileSelectionDlg();
    public SpySchemaDocumentationDlg GetSchemaDocumentationDlg();
    public SpyGenerateSampleXMLDlg GetGenerateSampleXMLDlg();
    public SpyDTDSchemaGeneratorDlg GetDTDSchemaGeneratorDlg();
    public SpyFindInFilesDlg GetFindInFilesDlg();
    public SpyWSDLDocumentationDlg GetWSDLDocumentationDlg();
}
```

5.6 SpyDoc

```

public class SpyDoc
{
    public void ReleaseInstance();
    public void SetEncoding( String strEncoding );
    public void SetPathName( String strPath );
    public String GetPathName();
    public String GetTitle();
    public boolean IsModified();
    public void Save();
    public void Close( boolean bDiscardChanges );
    public void UpdateViews();
    public Long GetCurrentViewMode();
    public boolean SwitchViewMode( Long nMode );
    public SpyGridView GetGridView();
    public void SetActiveDocument();
    public void StartChanges();
    public void EndChanges();
    public void TransformXSL();
    public void AssignDTD( String sDTDFile, boolean bDialog );
    public void AssignSchema( String sSchemaFile, boolean bDialog );
    public void AssignXSL( String sXSLFile, boolean bDialog );
    public void ConvertDTDOrSchema( Long nFormat, Long nFrequentElements );
    public SpyXMLData CreateChild( Long nKind );
    public void CreateSchemaDiagram( Long nKind, String sName, String sFile );
    public SpyDocEditView GetDocEditView();
    public void ExportToDatabase( SpyXMLData oFromChild, SpyExportSettings
oExportSettings, SpyDatabaseConnection oDatabaseConnection );
    public void ExportToText( SpyXMLData oFromChild, SpyExportSettings
oExportSettings, SpyTextImportExportSettings oTextSettings );
    public void GenerateDTDOrSchema( Long nFormat, int nValuesList, Long
nDetection, Long nFrequentElements );
    public SpyElementList GetExportElementList( SpyXMLData oFromChild,
SpyExportSettings oExportSettings );
    public SpyXMLData GetRootElement();
    public String SaveInString( SpyXMLData oData, boolean bMarked );
    public void SaveToURL( String sUrl, String sUser, String sPassword );
    public String GetErrorString(); // See IsValid() or IsWellFormed()
    public int GetErrorPos(); // See IsValid() or IsWellFormed()
    public SpyXMLData GetBadData(); // See IsValid() or IsWellFormed()
    public boolean IsValid();
    public boolean IsWellFormed( SpyXMLData oData, boolean bWithChildren );

    // Since version 2004R3
    public SpyAuthenticView GetAuthenticView()

    // Since version 2004R4
    public SpyApplication GetApplication();
    public SpyDocuments GetParent();
    public String GetFullName();
    public void SetFullName( String i_strName );
    public String GetName();
    public String GetPath();
    public boolean GetSaved();
    public void SaveAs( String i_strFileNameOrPath );
    public String GetEncoding();
    public SpyXMLData GetDataRoot();
    public void GenerateProgramCode( SpyCodeGeneratorDlg i_dlg );
    public void AssignXSLFO( String i_strFile, boolean i_bUseDialog );
    public void TransformXSLFO();

```

```
public void GenerateSchemaDocumentation( SpySchemaDocumentationDlg i_dlg );

public void ExecuteXQuery( String i_strXMLSourceFile );
public void SetExternalIsValid( boolean bIsValid );
public SpyDoc GenerateSampleXML( SpyGenerateSampleXMLDlg ipGenerateXMLDlg );
public boolean UpdateXMLData();
public String GetAsXMLString();
public void SetAsXMLString( String newVal );
public SpyDoc GenerateDTDOrSchemaEx( SpyDTDSchemaGeneratorDlg
ipDTDSchemaGeneratorDlg );
public SpyDoc ConvertDTDOrSchemaEx( long nFormat, long nFrequentElements,
String sOutputPath, long nOutputPathDialogAction );
public SpyTextView GetTextView();
public String[] GetSuggestions();
public void SetSuggestions( String[] aList );
public void SetSelection( String sVal );

// Since version 2009
public void GenerateWSDLDocumentation( SpyWSDLDocumentationDlg
ipWSDLDocumenationDlg );
public void TransformXSLEx( long nDialogAction );
}
```


5.7 SpyDocuments

```
public class SpyDocuments
{
    public void ReleaseInstance();
    public long Count();
    public SpyDoc GetItem( long nNo );
    public SpyDoc NewFile( String strFile, String strType );
    public SpyDoc NewFileFromText( String nSource, String strType );
    public SpyDoc OpenFile( String sPath, boolean bDialog );
    public SpyDoc OpenURL( String sUrl, long nURLType, long nLoading, String
sUser, String sPassword );
    public SpyDoc OpenURLDialog(String sURL, long nURLType, long nLoading,
String sUser, String sPassword )
}
```

5.8 SpyDTDSchemaGeneratorDlg

```
public class SpyDTDSchemaGeneratorDlg
{
    public void ReleaseInstance();
    public SpyApplication GetApplication();
    public long GetDTDSchemaFormat();
    public void SetDTDSchemaFormat( long newVal );
    public short GetValueList();
    public void SetValueList( short newVal );
    public long GetTypeDetection();
    public void SetTypeDetection( long newVal );
    public long GetFrequentElements();
    public void SetFrequentElements( long newVal );
    public boolean GetMergeAllEqualNamed();
    public void SetMergeAllEqualNamed( boolean newVal );
    public boolean GetResolveEntities();
    public void SetResolveEntities( boolean newVal );
    public long GetAttributeTypeDefinition();
    public void SetAttributeTypeDefinition( long newVal );
    public boolean GetGlobalAttributes();
    public void SetGlobalAttributes( boolean newVal );
    public boolean GetOnlyStringEnums();
    public void SetOnlyStringEnums( boolean newVal );
    public int GetMaxEnumLength();
    public void SetMaxEnumLength( int newVal );
    public String GetOutputPath();
    public void SetOutputPath( String newVal );
    public long GetOutputPathDialogAction();
    public void SetOutputPathDialogAction( long newVal );
}
```

5.9 SpyElementList

```
public class SpyElementList
{
    public void ReleaseInstance();
    public long GetCount();
    public SpyElementListItem GetItem( long nIndex );
    public void RemoveElement( long nIndex );
}
```

5.10 SpyElementListItem

```
public class SpyElementListItem
{
    public void ReleaseInstance\(\);
    public long GetElementKind\(\);
    public void SetElementKind( long nKind );
    public long GetFieldCount\(\);
    public String GetName\(\);
    public long GetRecordCount\(\);
}
```

5.11 SpyExportSettings

```
public class SpyExportSettings
{
    public void ReleaseInstance();
    public boolean GetCreateKeys();
    public void SetCreateKeys( boolean bValue );
    public SpyElementList GetElementList();
    public void SetElementList( SpyElementList obj );
    public boolean GetEntitiesToText ();
    public void SetEntitiesToText( boolean bValue );
    public boolean GetExportAllElements();
    public void SetExportAllElements( boolean bValue );
    public boolean GetFromAttributes();
    public void SetFromAttributes( boolean bValue );
    public boolean GetFromSingleSubElements();
    public void SetFromSingleSubElements( boolean bValue );
    public boolean GetFromTextValues();
    public void SetFromTextValues( boolean bValue );
    public boolean GetIndependentPrimaryKey();
    public void SetIndependentPrimaryKey( boolean bValue );
    public long GetNamespace();
    public void SetNamespace( long nValue );
    public int GetSubLevelLimit();
    public void SetSubLevelLimit( int nValue );
}
```

5.12 SpyFileSelectionDlg

```
// Since version 2004R4
public class SpyFileSelectionDlg
{
    public void ReleaseInstance\(\);
    public SpyApplication GetApplication\(\);
    public SpyDialogs GetParent\(\);
    public String GetFullName\(\);
    public void SetFullName( String i_strName );
    public long GetDialogAction\(\);
    public void SetDialogAction( long i_eAction );
}
```

5.13 SpyFindInFilesDlg

```
public class SpyFindInFilesDlg
{
    public void ReleaseInstance();
    public SpyApplication GetApplication();
    public String GetFind();
    public void SetFind( String sNewVal );
    public boolean GetRegularExpression();
    public void SetRegularExpression( boolean bNewVal );
    public String GetReplace();
    public void SetReplace( String sNewVal );
    public boolean GetReplaceOnDisk();
    public void SetReplaceOnDisk( boolean bNewVal );
    public boolean GetDoReplace();
    public void SetDoReplace( boolean bNewVal );
    public boolean GetMatchWholeWord();
    public void SetMatchWholeWord( boolean bNewVal );
    public boolean GetMatchCase();
    public void SetMatchCase( boolean bNewVal );
    public long GetSearchLocation();
    public void SetSearchLocation( long nPosition );
    public String GetStartFolder();
    public void SetStartFolder( String sNewVal );
    public boolean GetIncludeSubfolders();
    public void SetIncludeSubfolders( boolean bNewVal );
    public boolean GetSearchInProjectFilesDoExternal();
    public void SetSearchInProjectFilesDoExternal( boolean bNewVal );
    public String GetFileExtension();
    public void SetFileExtension( String sNewVal );
    public boolean GetAdvancedXMLSearch();
    public void SetAdvancedXMLSearch( boolean bNewVal );
    public boolean GetXMLElementNames();
    public void SetXMLElementNames( boolean bNewVal );
    public boolean GetXMLElementContents();
    public void SetXMLElementContents( boolean bNewVal );
    public boolean GetXMLAttributeNames();
    public void SetXMLAttributeNames( boolean bNewVal );
    public boolean GetXMLAttributeContents();
    public void SetXMLAttributeContents( boolean bNewVal );
    public boolean GetXMLComments();
    public void SetXMLComments( boolean bNewVal );
    public boolean GetXMLCDATA();
    public void SetXMLCDATA( boolean bNewVal );
    public boolean GetXMLPI();
    public void SetXMLPI( boolean bNewVal );
    public boolean GetXMLRest();
    public void SetXMLRest( boolean bNewVal );
    public boolean GetShowResult();
    public void SetShowResult( boolean bNewVal );
}
```

5.14 SpyFindInFilesMatch

```
public class SpyFindInFilesMatch
{
    public void ReleaseInstance\(\);
    public long Line\(\);
    public long Position\(\);
    public long Length\(\);
    public String LineText\(\);
    public boolean Replaced\(\);
}
```


5.15 SpyFindInFilesResult

```
public class SpyFindInFilesResult
{
    public void ReleaseInstance\(\);
    public long Count\(\);
    public SpyFindInFilesMatch GetItem( long nNo );
    public String GetPath\(\);
    public SpyDoc GetDocument\(\);
}
```

5.16 SpyFindInFilesResults

```
public class SpyFindInFilesResults
{
    public void ReleaseInstance();
    public long Count();
    public SpyFindInFilesResult GetItem( long nNo );
}
```

5.17 SpyGenerateSampleXMLDlg

```
public class SpyGenerateSampleXMLDlg
{
    public void ReleaseInstance();
    public SpyApplication GetApplication();
    public boolean GetNonMandatoryAttributes();
    public void SetNonMandatoryAttributes( boolean newVal );
    public boolean GetNonMandatoryElements();
    public void SetNonMandatoryElements( boolean newVal );
    public boolean GetTakeFirstChoice();
    public void SetTakeFirstChoice( boolean newVal );
    public long GetRepeatCount();
    public void SetRepeatCount( long newVal );
    public boolean GetFillWithSampleData();
    public void SetFillWithSampleData( boolean newVal );
    public boolean GetFillElementsWithSampleData();
    public void SetFillElementsWithSampleData( boolean newVal );
    public boolean GetFillAttributesWithSampleData();
    public void SetFillAttributesWithSampleData( boolean newVal );
    public boolean GetContentOfNillableElementsIsNonMandatory();
    public void SetContentOfNillableElementsIsNonMandatory( boolean newVal );
    public boolean GetTryToUseNonAbstractTypes();
    public void SetTryToUseNonAbstractTypes( boolean newVal );
    public long GetOptimization();
    public void SetOptimization( long newVal );
    public long GetSchemaOrDTDAssignment();
    public void SetSchemaOrDTDAssignment( long newVal );
    public String GetLocalNameOfRootElement();
    public void SetLocalNameOfRootElement( String newVal );
    public String GetNamespaceURIOfRootElement();
    public void SetNamespaceURIOfRootElement( String newVal );
    public long GetOptionsDialogAction();
    public void SetOptionsDialogAction( long newVal );
}
```

5.18 SpyGridView

```
public class SpyGridView
{
    public void ReleaseInstance();
    public SpyXMLData GetCurrentFocus();
    public void Deselect( SpyXMLData oData );
    public boolean GetIsVisible();
    public void Select( SpyXMLData oData );
    public void SetFocus( SpyXMLData oData );
}
```

5.19 SpyProject

```
public class SpyProject
{
    public void ReleaseInstance();
    public void CloseProject( boolean bDiscardChanges, boolean bCloseFiles,
boolean bDialog );
    public String GetProjectFile();
    public void SetProjectFile( String sFile );
    public SpyProjectItems GetRootItems();
    public void SaveProject();
    public void SaveProjectAs( String sPath, boolean bDialog );
}
```

5.20 SpyProjectItem

```
public class SpyProjectItem
{
    public void ReleaseInstance();
    public SpyProjectItems GetChildItems();
    public String GetFileExtensions();
    public void SetFileExtensions( String sExtensions );
    public long GetItemType();
    public String GetName();
    public SpyDoc Open();
    public SpyProjectItem GetParentItem();
    public String GetPath();
    public String GetValidateWith();
    public void SetValidateWith( String sVal );
    public String GetXMLForXSLTransformation();
    public void SetXMLForXSLTransformation( String sVal );
    public String GetXSLForXMLTransformation();
    public void SetXSLForXMLTransformation( String sVal );
    public String GetXSLTransformationFileExtension();
    public void SetXSLTransformationFileExtension( String sVal );
    public String GetXSLTransformationFolder();
    public void SetXSLTransformationFolder( String sVal );
}
```

5.21 SpyProjectItems

```
public class SpyProjectItems
{
    public void ReleaseInstance();
    public void AddFile( String sPath );
    public void AddFolder( String sName );
    public void AddURL( String sURL, long nURLType, String sUser, String
sPassword, boolean bSave );
    public long Count();
    public SpyProjectItem GetItem( long nNumber );
    public void RemoveItem( SpyProjectItem oItemToRemove );
}
```

5.22 SpySchemaDocumentationDlg

```
// Since version 2004R4
public class SpySchemaDocumenttaionDlg
{
    public void ReleaseInstance\(\);
    public SpyApplication GetApplication\(\);
    public SpyDialogs GetParent\(\);

    public String GetOutputFile\(\);
    public void SetOutputFile( String i_strVal );
    public long GetOutputFormat\(\);
    public void SetOutputFormat( long i_eVal );

    public boolean GetShowResult\(\);
    public void SetShowResult( boolean i_bVal );
    public long GetOptionsDialogAction\(\);
    public void SetOptionsDialogAction( long i_eVal );
    public long GetOutputFileDialogAction\(\);
    public void SetOutputFileDialogAction( long i_eVal );
    public boolean GetShowProgressBar\(\);
    public void SetShowProgressBar( boolean i_bVal );

    public void IncludeAll( boolean i_bInclude );
    public boolean GetIncludeIndex\(\);
    public void SetIncludeIndex( boolean i_bVal );
    public boolean GetIncludeGlobalElements\(\);
    public void SetIncludeGlobalElements( boolean i_bVal );
    public boolean GetIncludeLocalElements\(\);
    public void SetIncludeLocalElements( boolean i_bVal );
    public boolean GetIncludeGroups\(\);
    public void SetIncludeGroups( boolean i_bVal );
    public boolean GetIncludeComplexTypes\(\);
    public void SetIncludeComplexTypes( boolean i_bVal );
    public boolean GetIncludeSimpleTypes\(\);
    public void SetIncludeSimpleTypes( boolean i_bVal );
    public boolean GetIncludeAttributeGroups\(\);
    public void SetIncludeAttributeGroups( boolean i_bVal );
    public boolean GetIncludeRedefines\(\);
    public void SetIncludeRedefines( boolean i_bVal );

    public void AllDetails( boolean i_bDetailsOn );
    public boolean GetShowDiagram\(\);
    public void SetShowDiagram( boolean i_bVal );
    public boolean GetShowNamespace\(\);
    public void SetShowNamespace( boolean i_bVal );
    public boolean GetShowType\(\);
    public void SetShowType( boolean i_bVal );
    public boolean GetShowChildren\(\);
    public void SetShowChildren( boolean i_bVal );
    public boolean GetShowUsedBy\(\);
    public void SetShowUsedBy( boolean i_bVal );
    public boolean GetShowProperties\(\);
    public void SetShowProperties( boolean i_bVal );
    public boolean GetShowSingleFacets\(\);
    public void SetShowSingleFacets( boolean i_bVal );
    public boolean GetShowPatterns\(\);
    public void SetShowPatterns( boolean i_bVal );
    public boolean GetShowEnumerations\(\);
    public void SetShowEnumerations( boolean i_bVal );
    public boolean GetShowAttributes\(\);
}
```



```
public void SetShowAttributes( boolean i_bVal );
public boolean GetShowIdentityConstraints();
public void SetShowIdentityConstraints( boolean i_bVal );
public boolean GetShowAnnotations();
public void SetShowAnnotations( boolean i_bVal );
public boolean GetShowSourceCode();
public void SetShowSourceCode( boolean i_bVal );

// Since version 2009
public boolean GetEmbedDiagrams();
public void SetEmbedDiagrams( boolean i_bVal );
public long GetDiagramFormat();
public void SetDiagramFormat( long i_nVal );
public boolean GetIncludeGlobalAttributes();
public void SetIncludeGlobalAttributes( boolean i_bVal );
public boolean GetIncludeLocalAttributes();
public void SetIncludeLocalAttributes( boolean i_bVal );
public boolean GetIncludeReferencedSchemas();
public void SetIncludeReferencedSchemas( boolean i_bVal );
public boolean GetMultipleOutputFiles();
public void GetMultipleOutputFiles( boolean i_bVal );
}
```

5.23 SpyTextImportExportSettings

```
public class SpyTextImportExportSettings
{
    public void ReleaseInstance();
    public String GetDestinationFolder();
    public void SetDestinationFolder( String sVal );
    public long GetEnclosingCharacter();
    public void SetEnclosingCharacter( long nEnclosing );
    public String GetEncoding();
    public void SetEncoding( String sVal );
    public long GetEncodingByteOrder();
    public void SetEncodingByteOrder( long nByteOrder );
    public long GetFieldDelimiter();
    public void SetFieldDelimiter( long nDelimiter );
    public String GetFileExtension ();
    public void SetFileExtension( String sVal );
    public boolean GetHeaderRow();
    public void SetHeaderRow( boolean bVal );
    public String GetImportFile();
    public void SetImportFile( String sVal );
}
```

5.24 SpyTextView

```
public class SpyTextView
{
    public void ReleaseInstance();
    public SpyApplication GetApplication();
    public SpyDoc GetParent();
    public long LineFromPosition( long nCharPos );
    public long PositionFromLine( long nLine );
    public long LineLength( long nLine );
    public String GetSelText();
    public void SetSelText( String sText );
    public String GetRangeText( long nPosFrom, long nPosTill );
    public void ReplaceText( long nPosFrom, long nPosTill, String sText );
    public void MoveCaret( long nDiff );
    public void GoToLineChar( long nLine, long nChar );
    public void SelectText( long nPosFrom, long nPosTill );
    public long GetSelectionStart();
    public void SetSelectionStart( long nNewVal );
    public long GetSelectionEnd();
    public void SetSelectionEnd( long nNewVal );
    public String GetText();
    public void SetText( String sText );
    public long LineCount();
    public long Length();
}
```

5.25 SpyWSDLDocumentationDlg

```
// Since version 2008r2spl
public class SpyWSDLDocumentationDlg
{
    public void ReleaseInstance\(\);
    public SpyApplication GetApplication\(\);

    public String GetOutputFile\(\);
    public void SetOutputFile( String sNewVal );

    public long GetOutputFileDialogAction\(\);
    public void SetOutputFileDialogAction( long nNewVal );

    public long GetOptionsDialogAction\(\);
    public void SetOptionsDialogAction( long nNewVal );

    public boolean GetShowProgressBar\(\);
    public void SetShowProgressBar( boolean bNewVal );

    public boolean GetShowResult\(\);
    public void SetShowResult( boolean bNewVal );

    public long GetOutputFormat\(\);
    public void SetOutputFormat( long nNewVal );

    public boolean GetEmbedDiagrams\(\);
    public void SetEmbedDiagrams( boolean bNewVal );

    public long GetDiagramFormat\(\);
    public void SetDiagramFormat( long nNewVal );

    public boolean GetMultipleOutputFiles\(\);
    public void SetMultipleOutputFiles( boolean bNewVal );

    public void IncludeAll( boolean bNewVal );

    public boolean GetIncludeBinding\(\);
    public void SetIncludeBinding( boolean bNewVal );

    public boolean GetIncludeImportedWSDLFiles\(\);
    public void SetIncludeImportedWSDLFiles( boolean bNewVal );

    public boolean GetIncludeMessages\(\);
    public void SetIncludeMessages( boolean bNewVal );

    public boolean GetIncludeOverview\(\);
    public void SetIncludeOverview( boolean bNewVal );

    public boolean GetIncludePortType\(\);
    public void SetIncludePortType( boolean bNewVal );

    public boolean GetIncludeService\(\);
    public void SetIncludeService( boolean bNewVal );

    public boolean GetIncludeTypes\(\);
    public void SetIncludeTypes( boolean bNewVal );

    public void AllDetails( boolean bNewVal );

    public boolean GetShowBindingDiagram\(\);
    public void SetShowBindingDiagram( boolean bNewVal );
}
```

```
public boolean GetShowExtensibility();  
public void SetShowExtensibility( boolean bNewVal );  
  
public boolean GetShowMessageParts();  
public void SetShowMessageParts( boolean bNewVal );  
  
public boolean GetShowPort();  
public void SetShowPort( boolean bNewVal );  
  
public boolean GetShowPortTypeDiagram();  
public void SetShowPortTypeDiagram( boolean bNewVal );  
  
public boolean GetShowPortTypeOperations();  
public void SetShowPortTypeOperations( boolean bNewVal );  
  
public boolean GetShowServiceDiagram();  
public void SetShowServiceDiagram( boolean bNewVal );  
  
public boolean GetShowSourceCode();  
public void SetShowSourceCode( boolean bNewVal );  
  
public boolean GetShowTypesDiagram();  
public void SetShowTypesDiagram( boolean bNewVal );  
  
public boolean GetShowUsedBy();  
public void SetShowUsedBy( boolean bNewVal );  
}
```

5.26 SpyXMLData

```
public class SpyXMLData
{
    public void ReleaseInstance();
    public void AppendChild( SpyXMLData oNewData );
    public void EraseAllChildren();
    public void EraseCurrentChild();
    public SpyXMLData GetCurrentChild();
    public SpyXMLData GetFirstChild( long nKind );
    public SpyXMLData GetNextChild();
    public boolean GetHasChildren();
    public void InsertChild( SpyXMLData oNewData );
    public boolean IsSameNode( SpyXMLData oToComp );
    public long GetKind();
    public boolean GetMayHaveChildren();
    public String GetName();
    public void SetName( String sValue );
    public SpyXMLData GetParent();
    public String GetTextValue();
    public void SetTextValue( String sValue );
}
```

5.27 Authentic

5.27.1 SpyAuthenticRange

```
// Since version 2004R3
public class SpyAuthenticRange
{
    public void ReleaseInstance();
    public SpyApplication GetApplication();
    public SpyAuthenticView GetParent();
    public SpyAuthenticRange GotoNext( long eKind );
    public SpyAuthenticRange GotoPrevious( long eKind );
    public void Select();
    public long GetFirstTextPosition();
    public void SetFirstTextPosition( long nTextPosition );
    public long GetLastTextPosition();
    public void SetLastTextPosition( long nTextPosition );
    public String GetText();
    public void SetText( String strText );
    public boolean PerformAction( long eAction, String strElementName );
    public boolean CanPerformAction( long eAction, String strElementName );
    public String[] CanPerformActionWith( long eAction );
    public SpyAuthenticRange GoTo( long eKind, long nCount, long nFrom );
    public SpyAuthenticRange SelectNext( long eKind );
    public SpyAuthenticRange SelectPrevious( long eKind );
    public SpyAuthenticRange MoveBegin( long eKind, long nCount );
    public SpyAuthenticRange MoveEnd( long eKind, long nCount );
    public SpyAuthenticRange ExpandTo( long eKind );
    public SpyAuthenticRange CollapsToBegin();
    public SpyAuthenticRange CollapsToEnd();
    public SpyAuthenticRange GotoNextCursorPosition();
    public SpyAuthenticRange GotoPreviousCursorPosition();
    public boolean IsEmpty();
    public boolean IsEqual( SpyAuthenticRange ipCmp );
    public SpyAuthenticRange Clone();
    public SpyAuthenticRange SetFromRange( SpyAuthenticRange ipSrc );
    public boolean Delete();
    public boolean Cut();
    public boolean Copy();
    public boolean Paste();
    public SpyXMLData GetFirstXMLData();
    public void SetFirstXMLData( SpyXMLData objXMLDataPtr );
    public long GetFirstXMLDataOffset();
    public void SetFirstXMLDataOffset( long nOffset );
    public SpyXMLData GetLastXMLData();
    public void SetLastXMLData( SpyXMLData objXMLDataPtr );
    public long GetLastXMLDataOffset();
    public void SetLastXMLDataOffset( long nOffset );
    public String[] GetElementHierarchy();
    public String[] GetElementAttributeNames( String strElementName );
    public boolean HasElementAttribute( String strElementName, String strAttributeName );
    public String GetElementAttributeValue( String strElementName, String strAttributeName );
    public void SetElementAttributeValue( String strElementName, String strAttributeName, String strNewValue );
    public String[] GetEntityNames();
    public void InsertEntity( String strEntityName );
    public boolean IsInDynamicTable();
    public boolean AppendRow();
    public boolean InsertRow();
}
```

```
public boolean DuplicateRow\(\) ;
public boolean DeleteRow\(\) ;
public boolean MoveRowUp\(\) ;
public boolean MoveRowDown\(\) ;

// Since version 2004R4
public boolean IsCopyEnabled\(\) ;
public boolean IsCutEnabled\(\) ;
public boolean IsPasteEnabled\(\) ;
public boolean IsDeleteEnabled\(\) ;
public boolean IsTextStateApplied\(\) (String i_strElementName ) ;
public boolean IsFirstRow\(\) ;
public boolean IsLastRow\(\) ;
}
```


5.27.2 SpyAuthenticView

```
// Since version 2004R3
public class SpyAuthenticView
{
    public void ReleaseInstance\(\);
    public SpyApplication GetApplication\(\);
    public SpyDoc GetParent\(\);
    public SpyAuthenticRange GetSelection\(\);
    public void SetSelection( SpyAuthenticRange obj );
    public SpyAuthenticRange GetDocumentBegin\(\);
    public SpyAuthenticRange GetDocumentEnd\(\);
    public SpyAuthenticRange GetWholeDocument\(\);
    public long GetMarkupVisibility\(\);
    public void SetMarkupVisibility( long eSpyAuthenticMarkupVisibility );
    public SpyAuthenticRange GoTo( long eKind, long nCount, long nFrom );
    public void Print( boolean bWithPreview, boolean bPromptUser );
    public boolean Undo();
    public boolean Redo();
    public void UpdateXMLInstanceEntities\(\);

    // Since version 2004R4
    public String GetAsXMLString\(\);
    public void SetAsXMLString( String i_strXML );
    public SpyXMLData GetXMLDataRoot\(\);
    public boolean IsUndoEnabled\(\);
    public boolean IsRedoEnabled\(\);
}
```

5.27.3 SpyDocEditSelection

```
public class SpyDocEditSelection
{
    public void ReleaseInstance\(\);
    public SpyXMLData GetEnd\(\);
    public long GetEndTextPosition\(\);
    public SpyXMLData GetStart\(\);
    public long GetStartTextPosition\(\);
}
```

5.27.4 SpyDocEditView

```
public class SpyDocEditView
{
    public void ReleaseInstance();
    public void ApplyTextState( String sElementName );
    public SpyDocEditSelection GetCurrentSelection();
    public void EditClear();
    public void EditCopy();
    public void EditCut();
    public void EditPaste();
    public void EditRedo();
    public void EditSelectAll();
    public void EditUndo();
    public SpyXMLData GetNextVisible( SpyXMLData oElement );
    public SpyXMLData GetPreviousVisible( SpyXMLData oElement );
    public boolean GetIsEditClearEnabled();
    public boolean GetIsEditCopyEnabled();
    public boolean GetIsEditCutEnabled();
    public boolean GetIsEditPasteEnabled();
    public boolean GetIsEditRedoEnabled();
    public boolean GetIsEditUndoEnabled();
    public boolean GetIsRowAppendEnabled();
    public boolean GetIsRowDeleteEnabled();
    public boolean GetIsRowDuplicateEnabled();
    public boolean GetIsRowInsertEnabled();
    public boolean GetIsRowMoveDownEnabled();
    public boolean GetIsRowMoveUpEnabled();
    public boolean IsTextStateApplied( String sElementName );
    public boolean IsTextStateEnabled( String sElementName );
    public void LoadXML( String sXML );
    public void MarkUpView( long nKind );
    public void RowAppend();
    public void RowDelete();
    public void RowDuplicate();
    public void RowInsert();
    public void RowMoveDown();
    public void RowMoveUp();
    public String SaveXML();
    public void SelectionMoveTabOrder( boolean bForward, boolean bTag );
    public boolean SelectionSet( SpyXMLData oStart, long nStartPos, SpyXMLData
oEndElement, long nEndPos );
    public SpyXMLData GetXMLRoot();
    public String[] GetAllowedElements( long nAction, SpyXMLData oStartPtr,
SpyXMLData oEndPtr );
}
```

5.28 Vordefinierte Konstanten

In diesem Abschnitt sind alle Klassen aufgelistet, die die von der Java-Schnittstelle verwendeten vordefinierten Konstanten verwenden.

5.28.1 SPYAttributeTypeDefinition

```
public class SPYAttributeTypeDefinition
{
    public final static long      = 0;
    spyMergedGlobal
    public final static long      = 1;
    spyDistinctGlobal
    public final static long spyLocal = 2;
}
```

5.28.2 SPYAuthenticActions

```
public class SPYAuthenticActions
{
    public final static long
    spyAuthenticInsertAt                = 0;
    public final static long
    spyAuthenticApply                  = 1;
    public final static long
    spyAuthenticClearSurr              = 2;
    public final static long
    spyAuthenticAppend                 = 3;
    public final static long
    spyAuthenticInsertBefore           = 4;
    public final static long
    spyAuthenticRemove                 = 5;
}
```

5.28.3 SPYAuthenticDocumentPosition

```
public class SPYAuthenticDocumentPosition
{
    public final static long
    spyAuthenticDocumentBegin           = 0;
    public final static long
    spyAuthenticDocumentEnd             = 1;
    public final static long
    spyAuthenticRangeBegin              = 2;
    public final static long
    spyAuthenticRangeEnd                = 3;
}
```

5.28.4 SPYAuthenticElementKind

```
public class SPYAuthenticElementKind
{
    public final static long      spyAuthenticChar      = 0;
    public final static long      spyAuthenticWord      = 1;
    public final static long      spyAuthenticLine      = 3;
    public final static long      spyAuthenticParagraph = 4;
    public final static long      spyAuthenticTag        = 6;
    public final static long      spyAuthenticDocument  = 8;
    public final static long      spyAuthenticTable     = 9;
    public final static long      spyAuthenticTableRow  = 10;
    public final static long      spyAuthenticTableColumn = 11;
}
```


5.28.5 SPYAuthenticMarkupVisibility

```
public class SPYAuthenticMarkupVisibility
{
    public final static long
    spyAuthenticMarkupHidden           = 0;
    public final static long
    spyAuthenticMarkupSmall           = 1;
    public final static long
    spyAuthenticMarkupLarge           = 2;
    public final static long
    spyAuthenticMarkupMixed           = 3;
}
```

5.28.6 SPYDatabaseKind

```
public class SPYLoading
{
    public final static long      spyDB_Access      = 0;
    public final static long      spyDB_SQLServer    = 1;
    public final static long      spyDB_Oracle       = 2;
    public final static long      spyDB_Sybase       = 3;
    public final static long      spyDB_MySQL        = 4;
    public final static long      spyDB_DB2          = 5;
    public final static long      spyDB_Other        = 6;
    public final static long      spyDB_Unspecified  = 7;
}
```

5.28.7 SPYDialogAction

```
public class SPYDialogAction
{
    public final static long    spyDialogOK           = 0;
    public final static long    spyDialogCancel       = 1;
    public final static long    spyDialogUserInput    = 2;
}
```

5.28.8 SPYDOMType

```
public class SPYDOMType
{
    public final static long    = 0;
    spyDOMType_msxml4
    public final static long    = 1;
    spyDOMType_xerces
}
```

5.28.9 SPYDTDSchemaFormat

```
public class SPYDTDSchemaFormat
{
    public final static long spyDTD      = 0;
    public final static long spyDCD      = 1;
    public final static long spyXMLData = 2;
    public final static long spyBizTalk = 3;
    public final static long spyW3C     = 4;
}
```

5.28.10 SPYEncodingByteOrder

```
public class SPYEncodingByteOrder
{
    public final static long spyNONE           = 0;
    public final static long spyLITTLE_ENDIAN = 1;
    public final static long spyBIG_ENDIAN     = 2;
}
```

5.28.11 SPYExportNamespace

```
public class SPYExportNamespace
{
    public final static long spyNoNamespace = 0;
    public final static long spyReplaceColonWithUnderscore = 1;
}
```

5.28.12 SPYFindInFilesSearchLocation

```
public class SPYFindInFilesSearchLocation
{
    public final static long      = 0;
    spyFindInFiles_Documents
    public final static long      = 1;
    spyFindInFiles_Project
    public final static long      = 2;
    spyFindInFiles_Folder
}
```


5.28.13 SPYFrequentElements

```
public class SPYFrequentElements
{
    public final static long    spyGlobalElements    = 0;
    public final static long    spyGlobalComplexType = 1;
}
```

5.28.14 SPYImageKind

```
public class SPYImageKind
{
    public final static long      spyImageType_PNG      = 0;
    public final static long      spyImageType_EMF      = 1;
}
```

5.28.15 SPYLibType

```
public class SPYLibType
{
    public final static long    spyLibType_static    = 0;
    public final static long    spyLibType_dll       = 1;
}
```

5.28.16 SPYLoading

```
public class SPYLoading
{
    public final static long    spyUseCacheProxy    = 0;
    public final static long    spyReload           = 1;
}
```

5.28.17 SPYNumberDateTimeFormat

```
public class SPYNumberDateTimeFormat
{
    public final static long    spySystemLocale    = 0;
    public final static long    spySchemaCompatible
}
```

5.28.18 SPYProgrammingLanguage

```
public class SPYLoading
{
    public final static long spyUndefinedLanguage = -1;
    public final static long spyJava              = 0;
    public final static long spyCpp                = 1;
    public final static long spyCSharp             = 2;
}
```

5.28.19 SPYProjectItemTypes

```
public class SPYProjectItemTypes
{
    public final static long    spyUnknownItem    = 0;
    public final static long    spyFileItem       = 1;
    public final static long    spyFolderItem     = 2;
    public final static long    spyURLItem        = 3;
}
```

5.28.20 SPYProjectType

```
public class SPYProjectType
{
    public final static long    spyVisualStudioProject    = 0;
    public final static long    spyVisualStudio2003Project = 1;
    public final static long    spyBorlandProject         = 2;
    public final static long    spyMonoMakefile           = 3;
}
```


5.28.21 SPYSampleXMLGenerationOptimization

```
public class SPYSampleXMLGenerationOptimization
{
    public final static long
    spySampleXMLGen_Optimized                = 0;
    public final static long
    spySampleXMLGen_NonMandatoryElements    = 1;
    public final static long
    spySampleXMLGen_Everything              = 2;
}
```

5.28.22 SPYSampleXMLGenerationSchemaOrDTDAssignment

```
public class SPYSampleXMLGenerationOptimization
{
    public final static long      = 0;
    spySampleXMLGen_AssignRelatively
    public final static long      = 1;
    spySampleXMLGen_AssignAbsolutely
    public final static long      = 2;
    spySampleXMLGen DoNotAssign
}
```

5.28.23 SPYSchemaDefKind

```
public class SPYSchemaDefKind
{
    public final static long spyKindElement = 0;
    public final static long spyKindComplexType = 1;
    public final static long spyKindSimpleType = 2;
    public final static long spyKindGroup = 3;
    public final static long spyKindModel = 4;
    public final static long spyKindAny = 5;
    public final static long spyKindAttr = 6;
    public final static long spyKindAttrGroup = 7;
    public final static long spyKindAttrAny = 8;
    public final static long spyKindIdentityUnique = 9;
    public final static long spyKindIdentityKey = 10;
    public final static long spyKindIdentityKeyRef = 11;
    public final static long spyKindIdentitySelector = 12;
    public final static long spyKindIdentityField = 13;
    public final static long spyKindNotation = 14;
    public final static long spyKindInclude = 15;
    public final static long spyKindImport = 16;
    public final static long spyKindRedefine = 17;
    public final static long spyKindFacet = 18;
    public final static long spyKindSchema = 19;
    public final static long spyKindCount = 20;
}
```

5.28.24 SPYSchemaDocumentationFormat

```
public class SPYSchemaDocumentationFormat
{
    public final static long      = 0;
    spySchemaDoc_HTML
    public final static long      = 1;
    spySchemaDoc_MSWord
    public final static long      = 2;
    spySchemaDoc_RTF
}
```

5.28.25 SPYTextDelimiters

```
public class SPYTextDelimiters
{
    public final static long      = 0;
    spyTabulator
    public final static long      = 1;
    spySemicolon
    public final static long spyComma = 2;
    public final static long spySpace = 3;
}
```

5.28.26 SPYTextEnclosing

```
public class SPYTextEnclosing
{
    public final static long    = 0;
    spyNoEnclosing
    public final static long    = 1;
    spySingleQuote
    public final static long    = 2;
    spyDoubleQuote
}
```

5.28.27 SPYTypeDetection

```
public class SPYTypeDetection
{
    public final static long      = 0;
    spyBestPossible
    public final static long      = 1;
    spyNumbersOnly
    public final static long      = 2;
    spyNoDetection
}
```

5.28.28 SPYURLTypes

```
public class SPYURLTypes
{
    public final static long    spyURLTypeAuto    = (-1);
    public final static long    spyURLTypeXML     = 0;
    public final static long    spyURLTypeDTD     = 1;
}
```


5.28.29 SpyViewModes

```
public class SPYViewModes
{
    public final static long      spyViewGrid      = 0;
    public final static long      spyViewText      = 1;
    public final static long      spyViewBrowser   = 2;
    public final static long      spyViewSchema    = 3;
    public final static long      spyViewContent   = 4;
    public final static long      spyViewAuthentic = 4;
    public final static long      spyViewWSDL      = 5;
    public final static long      spyViewZIP       = 6;
    public final static long      spyViewEditionInfo = 7;
}
```

5.28.30 SPYWhitespaceComparison

```
public class SPYWhitespaceComparison
{
    public final static long    spyCompareAsIs          = 0;
    public final static long    spyCompareNormalized    = 1;
    public final static long    spyStripAll              = 2;
}
```

5.28.31 SPYXMLDataKind

```
public class SPYXMLDataKind
{
    public final static long      spyXMLDataXMLDocStruct      = 0;
    public final static long      spyXMLDataXMLEntityDocStruct = 1;
    public final static long      spyXMLDataDTDDocStruct      = 2;
    public final static long      spyXMLDataXML              = 3;
    public final static long      spyXMLDataElement          = 4;
    public final static long      spyXMLDataAttr             = 5;
    public final static long      spyXMLDataText             = 6;
    public final static long      spyXMLDataCDATA            = 7;
    public final static long      spyXMLDataComment          = 8;
    public final static long      spyXMLDataPI               = 9;
    public final static long      spyXMLDataDefDoctype       = 10;
    public final static long      spyXMLDataDefExternalID    = 11;
    public final static long      spyXMLDataDefElement       = 12;
    public final static long      spyXMLDataDefAttlist       = 13;
    public final static long      spyXMLDataDefEntity        = 14;
    public final static long      spyXMLDataDefNotation      = 15;
    public final static long      spyXMLDataKindsCount       = 16;
}
```

6 AuthenticDesktop Integration

AuthenticDesktopControl ist ein Control, das ein Mittel zur Integration der AuthenticDesktop Benutzeroberfläche und der in diesem Abschnitt beschriebenen Funktionalitäten in die meisten Applikationsarten darstellt. Es wurde die ActiveX-Technologie gewählt, um die Integration mittels jeder beliebigen der vielen Programmiersprachen zu ermöglichen; dadurch können für die Integration C++, C#, VisualBasic oder HTML verwendet werden. Alle Komponenten sind vollständige OLE Controls, wodurch die Integration so einfach wie möglich gemacht wird. Es stehen zwei verschiedene Ebenen der Integration zur Verfügung, sodass die Integration an die verschiedensten Bedürfnisse angepasst werden kann.

Um AuthenticDesktop zu integrieren, müssen Sie das AuthenticDesktop Integrationspaket installieren. Stellen Sie sicher, dass Sie AuthenticDesktop zuerst installieren und dann erst das AuthenticDesktop Integrationspaket.

Für eine erfolgreiche Integration müssen beim Design die folgenden Hauptfaktoren berücksichtigt werden:

- Welche Technologie oder Programmiersprache kann die Host-Applikation zur Integration des AuthenticDesktopControl verwenden?
- Soll die integrierte Benutzeroberfläche genau wie AuthenticDesktop mit all seinen Menüs, Symbolleisten und Fenstern aussehen oder wäre eine Untermenge dieser Elemente - z.B. dass nur ein Dokument und eine eingeschränkte Anzahl an Befehlen verwendet werden kann - sinnvoller?
- Wie tief wird die Integration sein? Soll die AuthenticDesktop Benutzeroberfläche, wie sie ist, verwendet werden? Sind Erweiterungen zur Benutzeroberfläche und/oder Einschränkungen erforderlich? Können einige häufig verwendete Aufgaben automatisiert werden?

Die Abschnitte [Integration auf Applikationsebene](#) und [Integration auf Dokumentebene](#), von denen jeder Beispiele in den verschiedenen Programmiersprachen enthält, sollen Ihnen dabei helfen, rasch die richtige Entscheidung zu treffen. Der Abschnitt [Objektreferenz](#) enthält eine Beschreibung aller COM-Objekte mit ihren Eigenschaften und Methoden, die für die Integration verwendet werden können.

Für Automationsaufgaben kann die [AuthenticDesktop Automation-Schnittstelle](#) auch über das AuthenticDesktopControl aufgerufen werden.

Informationen zum Integrieren von AuthenticDesktop in Microsoft Visual Studio finden Sie im Abschnitt [AuthenticDesktop in Visual Studio](#).

6.1 Integration auf Applikationsebene

Die Integration auf Applikationsebene ist einfach. Sie können dadurch die gesamte Benutzeroberfläche von AuthenticDesktop in ein Fenster Ihrer Applikation einbetten. Da Sie die gesamte Benutzeroberfläche von AuthenticDesktop zur Verfügung haben, stehen Ihnen alle Menüs, Symbolleisten, die Statusleiste, die Dokumentfenster und Eingabehilfen zur Verfügung. Die Anpassung der Benutzeroberfläche der Applikation ist auf die Optionen eingeschränkt, die AuthenticDesktop bietet. Dazu gehören die Neuordnung und Anpassung der Größe der Eingabehilfen und die Anpassung von Menüs und Symbolleisten.

Das einzige ActiveX Control, das Sie integrieren müssen, ist [AuthenticDesktopControl](#). Der Wert seiner Eigenschaft [IntegrationLevel](#) ist standardmäßig "application-level". Sie können das Aussehen des Wrapper-Fensters des Control mit Hilfe von [Appearance](#) und [BorderStyle](#) konfigurieren. Bei Integration auf Applikationsebene dürfen [AuthenticDesktopControlDocument](#) oder [AuthenticDesktopControlPlaceHolder](#) ActiveX Controls nicht instantiiert oder aufgerufen werden.

Wenn Sie Initialisierungen vornehmen müssen oder ein bestimmtes Verhalten von AuthenticDesktop automatisieren wollen, verwenden Sie die für [AuthenticDesktopControl](#) beschriebenen Eigenschaften, Methoden und Events. Um komplexere AuthenticDesktop Funktionsaufrufe auszuführen, sollten Sie eventuell [AuthenticDesktopControl.Application](#) verwenden.

In diesem Abschnitt finden Sie ein Beispiel ([Beispiel: HTML](#)), in dem gezeigt wird, wie die AuthenticDesktop Applikation in eine HTML-Seite eingebettet werden kann. Eine Beschreibung zur Verwendung mit anderen Programmiersprachen oder kompliziertere Aufrufe finden Sie unter den [Beispielen](#) zur Integration auf Dokumentebene.

6.1.1 Beispiel: HTML

In diesem Beispiel wird eine einfache Integration des AuthenticDesktop Control auf Applikationsebene in eine HTML-Seite beschrieben. Die Integration wird in den folgenden Abschnitten beschrieben:

- Instantiieren Sie ein AuthenticDesktopControl in HTML-Code.
- Implementieren Sie Schaltflächen zum Laden von Dokumenten und automatisieren Sie Codegenerierungsaufgaben.
- Definieren Sie Aktionen für einige Applikationsevents.

Den Code für dieses Beispiel finden Sie in Ihrer AuthenticDesktop Installation im folgenden Ordner:

AuthenticExamples\ActiveX\HTML\AuthenticActiveX_ApplicationLevel.htm

Instantiieren des Control

Der HTML Object Tag dient zum Erstellen einer Instanz des AuthenticDesktopControl. Die Classid ist die von AuthenticDesktopControl. Mit "width" und "height" wird die Fenstergröße definiert. Es sind keine weiteren Parameter erforderlich, da die Standardeinstellung "application-level" ist.

Hinzufügen einer Schaltfläche zum Öffnen eines Standarddokuments

Als einfaches Beispiel, wie man manche Aufgaben automatisiert, fügen wir eine Schaltfläche zur Seite hinzu:

```
<input type="button" value="Open Marketing Expenses"
onclick="BtnOpenMEFile()">
```

Wenn Sie darauf klicken, wird ein vordefiniertes Dokument im AuthenticDesktopControl geöffnet. Wir verwenden eine Methode, um die Datei relativ zum AuthenticDesktopControl anzugeben, daher kann das Beispiel in unterschiedlichen Installationen ausgeführt werden.

Verbindung zu benutzerdefinierten Events herstellen

In diesem Beispiel werden zur Veranschaulichung des Prinzips zwei Event Callbacks für benutzerdefinierte AuthenticDesktopControl Events implementiert:

```
<!-- ----- -->
<!-- custom event 'OnDocumentOpened' of AuthenticDesktopControl object -->
<SCRIPT FOR="objAuthenticDesktopControl" event="OnDocumentOpened( objDocument
)" LANGUAGE="javascript">
    // alert("Document '" + objDocument.Name + "' opened!");
</SCRIPT>

<!-- ----- -->
<!-- custom event 'OnDocumentClosed' of AuthenticDesktopControl object -->
<SCRIPT FOR="objAuthenticDesktopControl" event="OnDocumentClosed( objDocument
)" LANGUAGE="javascript">
    // alert("Document '" + objDocument.Name + "' closed!");
</SCRIPT>
```

6.2 Integration auf Dokumentenebene

Bei der Integration auf Dokumentenebene haben Sie Freiheit bei der Instantiierung und Platzierung der folgenden Teile der AuthenticDesktop Benutzeroberfläche:

Falls erforderlich, muss in Ihrer Applikation ein Ersatz für die Menüs und Symbolleisten von AuthenticDesktop zur Verfügung gestellt werden.

Je nachdem, welche Teile der Benutzeroberfläche Sie wiederverwenden möchten, müssen Sie mehrere ActiveX Controls instantiieren und aufrufen. Alle diese Controls sind in der AuthenticDesktopControl OCX-Datei enthalten.

- [Mit Hilfe von AuthenticDesktopControl](#) können Sie die Integrationsebene definieren und auf applikationsweite Funktionalitäten zugreifen.
- [Mit Hilfe von AuthenticDesktopControlDocument](#) können Sie beliebig viele Editor-Fenster erstellen. Eventuell genügt es, nur ein Fenster zu erstellen und dieses wiederzuverwenden.
- Verwenden Sie optional [AuthenticDesktopControlPlaceholder](#) um AuthenticDesktop Eingabehilfenfenster, Validierungsausgabefenster oder andere oben erwähnte Fenster einzubetten.
- Rufen Sie Laufzeitinformationen über in AuthenticDesktopControl verfügbare Befehle, Menüs und Symbolleisten auf, um diese Befehle nahtlos in Ihre Applikationsmenüs und Symbolleisten zu integrieren. Nähere Informationen dazu finden Sie unter [Abfrage von AuthenticDesktop Befehlen](#).

Wenn Sie ein bestimmtes Verhalten von AuthenticDesktop automatisieren möchten, verwenden Sie die für [AuthenticDesktopControl](#), [AuthenticDesktopControlDocument](#) und [AuthenticDesktopControlPlaceholder](#) beschriebenen Eigenschaften, Methoden und Events. Bei Aufruf komplexerer AuthenticDesktop Funktionalitäten sollten Sie eventuell [AuthenticDesktopControl.Application](#), [AuthenticDesktopControlDocument.Document](#) und [AuthenticDesktopControlPlaceholder.Project](#) verwenden. Verwenden Sie allerdings zum Öffnen eines Dokuments immer [AuthenticDesktopControlDocument.Open](#) oder [AuthenticDesktopControlDocument.New](#) am entsprechenden Document Control. Verwenden Sie zum Öffnen eines Projekts immer [AuthenticDesktopControlPlaceholder.OpenProject](#) an einem Placeholder Control, das ein AuthenticDesktop Projektfenster einbettet.

Beschreibungen zum Instantiieren und Aufrufen der erforderlichen Controls in verschiedenen Programmierungsumgebungen finden Sie unter [Beispiele](#).

6.2.1 Verwendung von AuthenticDesktopControl

Zur Integration auf Dokumentebene instantiieren Sie zuerst ein [AuthenticDesktopControl](#). Setzen Sie die Eigenschaft [IntegrationLevel](#) auf `ICActiveXIntegrationOnDocumentLevel(=1)`. Setzen Sie die Fenstergröße des einbettenden Fensters auf `0x0`, damit eine andere Benutzeroberfläche gegebenenfalls vom Control verdeckt wird. Sie können das Aussehen des Wrapper-Fensters für das Control mittels [Appearance](#) und [BorderStyle](#) konfigurieren.

Vermeiden Sie nach Möglichkeit die Verwendung der Methode [Open](#), da dies zu unerwarteten Ergebnissen führen könnte. Verwenden Sie stattdessen die entsprechenden Öffnungsmethoden von [AuthenticDesktopControlDocument](#) und [AuthenticDesktopControlPlaceholder](#).

Eine Beschreibung zur Integration von AuthenticDesktop Befehlen in Ihre Applikation finden Sie unter [Abfragen von AuthenticDesktop Befehlen](#). Senden Sie Befehle über die Methode [Exec](#) an AuthenticDesktop. Führen Sie mit Hilfe der Methode [QueryStatus](#) eine Abfrage durch, um zu sehen, ob ein Befehl derzeit aktiviert oder deaktiviert ist.

6.2.2 Verwendung von AuthenticDesktopControlDocument

Mit Hilfe einer Instanz des `AuthenticDesktopControlDocument` ActiveX Control können Sie ein AuthenticDesktop Mapping-Bearbeitungsfenster in Ihre Applikation einbetten. Sie können so viele Instanzen verwenden, so viele Sie benötigen.

Verwenden Sie die Methode [Open](#), um eine beliebige andere bereits vorhandene Datei zu laden.

Das Control unterstützt keinen schreibgeschützten Modus. Der Wert der Eigenschaft [ReadOnly](#) wird ignoriert.

Verwenden Sie zum Aufrufen von Dokumentfunktionalitäten [Path](#) und [Save](#) oder Methoden und Eigenschaften, die über die Eigenschaft [Document](#) aufgerufen werden.

6.2.3 Verwendung von AuthenticDesktopControlPlaceholder

Mit Hilfe von Instanzen des AuthenticDesktopControlPlaceholder ActiveX Control können Sie selektiv weitere Eingabehilfefenster von AuthenticDesktop in Ihre Applikation einbetten. Die Eigenschaft [placeholderWindowID](#) wählt das AuthenticDesktop Eingabehilfefenster aus, um es einzubetten. Verwenden Sie für jeden Fenster-Identifizier nur einen AuthenticDesktopControlPlaceholder. Eine Beschreibung gültiger Fenster-Identifizier finden Sie unter `Enumerations.AuthenticDesktopControlPlaceholderWindow`.

Für Platzhalter-Controls, die das AuthenticDesktop Projektfenster auswählen, stehen zusätzliche Methoden zur Verfügung. Verwenden Sie [OpenProject](#), um ein AuthenticDesktop Projekt zu laden. Verwenden Sie die Eigenschaft [Project](#) und die Methoden und Eigenschaften aus dem AuthenticDesktop Automation Interface, um andere projektbezogene Operationen auszuführen.

6.2.4 Abfragen von AuthenticDesktop Befehlen

Bei der Integration auf Dokumentenebene stehen Ihrer Applikation kein Menü und keine Symbolleiste aus AuthenticDesktop zur Verfügung. Stattdessen können Sie alle Befehle und die Struktur des Applikationsmenüs zur Laufzeit abfragen. Verwenden Sie zum Aufrufen dieser Informationen die Eigenschaft `AuthenticDesktopControl.CommandsStructure`. Bei professionellen Applikationen muss dieses Menü auf raffinierte Art in die eigene Menüstruktur integriert werden. In Ihrer Installation von AuthenticDesktop stehen Ihnen sogar Bilder für in AuthenticDesktop verwendete Befehle zur Verfügung. Symbole im GIF-Format finden Sie in Ihrem AuthenticDesktop Installationsordner unter `AuthenticExamples\ActiveX\Images of your AuthenticDesktop`. Die Dateinamen entsprechen den [labels](#) von Befehlen.

6.2.5 Beispiele

Dieser Abschnitt enthält Beispiele für die Integration von AuthenticDesktop auf Dokumentebene über unterschiedliche Container-Umgebungen und Programmiersprachen. Der Quellcode für alle Beispiele steht im Ordner `AuthenticExamples\ActiveX` Ihrer AuthenticDesktop Installation zur Verfügung.

C#

Im C#-Beispiel wird gezeigt, wie man das AuthenticDesktopControl in eine allgemeine Desktop-Applikation integriert, die mit C# über Visual Studio 2008 erstellt wurde.

-

Bitte beachten Sie, dass die Beispielapplikation bereits fertig ist. Sie müssen keine Änderungen vornehmen, wenn Sie sie ausführen und sehen möchten, wie sie funktioniert. In den folgenden Schritten wird beschrieben, welche Aktionen durchgeführt und welche Dinge berücksichtigt werden müssen, um ein Projekt wie dieses zu erstellen.

Einführung

Hinzufügen der AuthenticDesktop -Komponenten zur Toolbox

Bevor Sie einen Blick in das Beispielprojekt werfen, fügen Sie bitte die Assemblies zur .NET IDE Toolbox hinzu. Der AuthenticDesktop Installer hat die Assemblies bereits im .NET Global Assembly Cache (GAC) installiert. Wenn Sie das Toolbox-Dialogfeld unter **Tools |**

Add/Remove Toolbox Items öffnen, werden die Controls als

`AxAuthenticDesktopControl`, `AxAuthenticDesktopControlDocument` und `AxAuthenticDesktopControlPlaceholder` auf dem Register ".NET Framework Components" angezeigt. Aktivieren Sie alle, damit Sie in der IDE zur Verfügung stehen.

Platzieren des AuthenticDesktopControl

Sie benötigen eine AuthenticDesktopControl Instanz, um die Integrationsebene zu definieren und die Dokument- und Platzhalter-Controls der AuthenticDesktop Bibliothek zu verwalten. Das Control kann über den Abschnitt "General" des Toolbox-Eingabehilfenfensters in der IDE aufgerufen werden. Um es hinzuzufügen, müssen Sie die Komponente im Fenster "Toolbox" auswählen und ein Rechteck an der Stelle aufziehen, an der Sie es im Zielfenster haben möchten. Bei einer Applikation, die beim Start kein Fenster öffnet, können Sie ein einfaches unsichtbares Formular mit dem Control verwenden, an dem es im Code manuell erstellt wird.

Im Beispielprojekt wird diese Instanz zum Haupt-MdiContainer MDIMain hinzugefügt. Wenn Sie MDIMain in der Design-Ansicht über den Solution Explorer öffnen, sehen Sie im Client-Bereich des Frame-Fensters an der linken oberen Seite ein hellblaues Rechteck. Wenn Sie das Rechteck auswählen, sehen Sie die Eigenschaften des AuthenticDesktopControl. Die Eigenschaft `IntegrationLevel` muss unbedingt auf `ICActiveXIntegrationOnDocumentLevel` gesetzt werden, um die Unterstützung der AuthenticDesktop Bibliothek für Dokumente und Platzhalter zu aktivieren.

HTML

In diesem Beispiel wird eine Integration des AuthenticDesktop Control auf Dokumentebene in eine HTML-Seite gezeigt. Es werden die folgenden Themen behandelt:

- Instantiieren eines AuthenticDesktopControl ActiveX Control-Objekts in HTML-Code

- Instantiieren eines AuthenticDesktopControlDocument ActiveX Control für die Bearbeitung eines AuthenticDesktop Mappings
- Instantiieren eines AuthenticDesktopControlPlaceHolder für ein AuthenticDesktop-Projektfenster
- Instantiieren eines AuthenticDesktopControlPlaceHolder ActiveX Control für die Aufnahme eines der AuthenticDesktop-Eingabehilfefenster
- Erstellen einer benutzerdefinierten Symbolleiste für einige häufig gebrauchte AuthenticDesktop-Befehle
- Hinzufügen von weiteren Schaltflächen, die das COM Automation Interface von AuthenticDesktop verwenden
- Verwenden von Event Handlern zum Aktualisieren von Befehlsschaltflächen

Dieses Beispiel ist vollständig in der Datei `AuthenticDesktopActiveX_ApplicationLevel.htm` im Ordner `AuthenticExamples\ActiveX\HTML\` Ihrer AuthenticDesktop Installation enthalten.

Instantiieren des AuthenticDesktopControl

Zum Erstellen einer Instanz des AuthenticDesktopControl wird der HTML `OBJECT` Tag verwendet. Die Classid ist die von AuthenticDesktopControl. "width" und "height" werden auf 0 gesetzt, da wir dieses Control als Manager Control ohne Verwendung für seine Benutzeroberfläche verwenden. Die Integrationsebene wird in Form eines Parameters im `OBJECT` Tag definiert.

```
<OBJECT id=""
        width="0"
        height="0"
        VIEWASTEXT>
    <PARAM NAME="IntegrationLevel" VALUE="1">
</OBJECT>
```

Erstellen eines Bearbeitungsfensters

Der HTML `OBJECT` Tag wird zum Einbetten eines Dokumentenbearbeitungsfensters verwendet.

```
<OBJECT id="objDoc1"
        width="600"
        height="500"
        VIEWASTEXT>
</OBJECT>
```

Erstellen eines Projektfensters

Der HTML `OBJECT` Tag wird zum Erstellen eines AuthenticDesktopControl Placeholder-Fensters verwendet. Der erste zusätzliche benutzerdefinierte Parameter definiert, welcher Platzhalter das AuthenticDesktop-Projektfenster anzeigen soll. Der zweite Parameter lädt eines der Beispielprojekte aus Ihrer AuthenticDesktop Installation (aus dem Ordner `<yourusername>\MyDocuments`).

```
<OBJECT id="objProjectWindow"
        width="200"
        height="200"
        VIEWASTEXT>
</OBJECT>
```

Erstellen eines Placeholder für Eingabehilfenfenster

Der HTML `OBJECT` Tag wird zum Instantiieren eines `AuthenticDesktopControlPlaceholder` ActiveX Control verwendet, das die verschiedenen AuthenticDesktop Hilfsfenster aufnehmen kann. Anfangs wird kein Hilfsfenster angezeigt. Siehe Beispieldatei:

```
<OBJECT id="objEHWindow"
        width="200"
        height="200"
        VIEWASTEXT>
    <PARAM name="PlaceholderWindowID" value="">
</OBJECT>
```

Mit Hilfe von drei Schaltflächen kann zwischen dem tatsächlichen Fenster, das angezeigt werden soll, gewechselt werden. Mit der JavaScript-Anweisung die bei Klick auf eine Schaltfläche ausgeführt wird, wird die Eigenschaft `PlaceholderWindowID` auf den entsprechenden in `AuthenticDesktopControlPlaceholderWindow` definierten Wert gesetzt.

```
<SCRIPT ID="Javahandlers" LANGUAGE="javascript">
//
-----
// specify which of the windows shall be shown in the placeholder control.
function BtnHelperWindow(i_ePlaceholderWindowID)
{
    objEHWindow.PlaceholderWindowID = i_ePlaceholderWindowID;
}
</SCRIPT>
```

6.3 Befehlstabelle für Authentic Desktop

In den Tabellen in diesem Abschnitt werden die Namen und Identifier aller Befehle aufgelistet, die in AuthenticDesktop zur Verfügung stehen. In den einzelnen Unterabschnitten werden jeweils die Befehle aus dem entsprechenden Menü von AuthenticDesktop aufgelistet. In der Spalte ganz links wird der Text des Menübefehls angezeigt, damit Sie sofort wissen, um welche Funktionalität es sich hierbei handelt. Im letzten Unterabschnitt finden Sie all jene Befehle, die über das Hauptmenü nicht aufgerufen werden können.

Eine Beschreibung, wie man die aktuelle Ressourcenstruktur und die Verfügbarkeit von Befehlen abfragt, finden Sie unter [Abfragen von AuthenticDesktop Befehlen](#).

Verwenden Sie die Befehls-Identifier mit [XMLSpyControl.QueryStatus](#) oder [XMLSpyControlDocument.QueryStatus](#) um den aktuellen Status eines Befehls zu überprüfen. Verwenden Sie [XMLSpyControl.Exec](#) oder [XMLSpyControlDocument.Exec](#) um einen Befehl auszuführen.

[Menü "Datei"](#)
[Menü "Bearbeiten"](#)
[Menü "Projekt"](#)
[Menü "XML"](#)
[Menü "XSL/XQuery"](#)
[Menü "Authentic"](#)
[Menü "Ansicht"](#)
[Menü "Browser"](#)
[Menü "Extras"](#)
[Menü "Fenster"](#)
[Menü "Hilfe"](#)
[Diverse Menübefehle](#)

6.3.1 Menü "Datei"

Befehle aus dem Menü "Datei":

Menütext	Befehlsname	ID
Neu...	ID_FILE_NEW	57600
Öffnen...	ID_FILE_OPEN	57601
Globale Ressource öffnen...	IDC_OPEN_GLOBALRESOURCE	34112
Neu laden	IDC_FILE_RELOAD	34065
Kodierung...	IDC_ENCODING	34061
Schließen	ID_FILE_CLOSE	57602
Alles schließen	IDC_CLOSE_ALL	34050
Speichern	ID_FILE_SAVE	57603
Speichern unter...	ID_FILE_SAVE_AS	57604
Speichern unter URL...	ID_FILE_SAVE_TO_URL	34209
Alles speichern	ID_FILE_SAVE_ALL	34208
Als Mail senden...	ID_FILE_SEND_MAIL	57612
Drucken...	IDC_PRINT	34103
Druckvorschau	IDC_PRINT_PREVIEW	34104
Druckereinrichtung...	ID_FILE_PRINT_SETUP	57606
Letzte Datei	ID_FILE_MRU_FILE1	57616
Beenden	ID_APP_EXIT	57665

6.3.2 Menü "Bearbeiten"

Befehle aus dem Menü "Bearbeiten":

Menütext	Befehlsname	ID
Rückgängig	ID_EDIT_UNDO	57643
Wiederherstellen	ID_EDIT_REDO	57644
Ausschneiden	ID_EDIT_CUT	57635
Kopieren	ID_EDIT_COPY	57634
Einfügen	ID_EDIT_PASTE	57637
Löschen	ID_EDIT_CLEAR	57632
Alles markieren	ID_EDIT_SELECT_ALL	57642
Suchen...	ID_EDIT_FIND	57636
Weitersuchen	ID_EDIT_REPEAT	57640
Ersetzen...	ID_EDIT_REPLACE	57641

6.3.3 Menü "Projekt"

Befehle aus dem Menü "Projekt":

Menütext	Befehlsname	ID
Neues Projekt	IDC_PROJECTGUI_NEW	37200
Projekt öffnen...	IDC_PROJECTGUI_OPEN	37201
Projekt neu laden	IDC_PROJECTGUI_RELOAD	37202
Projekt schließen	IDC_PROJECTGUI_CLOSE	37203
Projekt speichern	IDC_PROJECTGUI_SAVE	37204
Versionskontrolle/Projekt öffnen...	IDC_SCC_OPEN_PROJECT	34140
Versionskontrolle/Versionskontrolle aktivieren	IDC_SCC_ENABLE	34137
Versionskontrolle/Aktuellste Version holen...	IDC_SCC_GET	34138
Versionskontrolle/Auschecken...	IDC_SCC_CHECK_OUT	34135
Versionskontrolle/Einchecken...	IDC_SCC_CHECK_IN	34134
Versionskontrolle/Auschecken rückgängig...	IDC_SCC_UNDO_CHECK_OUT	34145
Versionskontrolle/Zu Versionskontrolle hinzufügen...	IDC_SCC_ADD	34133
Versionskontrolle/Von Versionskontrolle ausgliedern...	IDC_SCC_REMOVE	34143
Versionskontrolle/Verlauf anzeigen...	IDC_SCC_HISTORY	34139
Versionskontrolle/Unterschiede anzeigen...	IDC_SCC_DIFF	34136
Versionskontrolle/Eigenschaften...	IDC_SCC_PROPERTIES	34141
Versionskontrolle/Status aktualisieren...	IDC_SCC_REFRESH	34142
Versionskontrolle/Native Oberfläche ausführen...	IDC_SCC_RUN	34144
Dateien zu Projekt hinzufügen...	IDC_ICPROJECTGUI_ADD_FILES_TO_PROJECT	37205
Globale Ressource zu Projekt hinzufügen...	IDC_ICPROJECTGUI_ADD_GLOBAL_RESOURCE_TO_PROJECT	37239
URL zu Projekt hinzufügen...	IDC_ICPROJECTGUI_ADD_URL_TO_PROJECT	37206
Aktive Datei zu Projekt hinzufügen	IDC_ICPROJECTGUI_ADD_ACTIVE_FILE_TO_PROJECT	37208
Aktive und verwandte Dateien zu Projekt hinzufügen	IDC_ICPROJECTGUI_ADD_ACTIVE_AND_RELATED_FILES_TO_PROJECT	37209
Projektordner zu Projekt hinzufügen...	IDC_ICPROJECTGUI_ADD_FOLDER_TO_PROJECT	37210
Externes Verzeichnis zum Projekt hinzufügen...	IDC_ICPROJECTGUI_ADD_EXT_FOLDER_TO_PROJECT	37211
Externes Webverzeichnis zum Projekt hinzufügen..	IDC_ICPROJECTGUI_ADD_EXT_URL_FOLDER_TO_PROJECT	37212
Projekteigenschaften...	IDC_ICPROJECTGUI_PROJECT_PROPERTIES	37223
Letztes Projekt	IDC_ICPROJECTGUI_RECENT	37224

6.3.4 Menü "XML"

Befehle aus dem Menü "XML":

Wohlgeformtheit prüfen	IDC_CHECK_WELL_FORM	34049
XML validieren	IDC_VALIDATE	32954

6.3.5 Menü "XSL/XQuery"

Befehle aus dem Menü "XSL/XQuery":

XSL-Transformation	IDC_TRANSFORM_XSL	33006
XSL:FO-Transformation	IDC_TRANSFORM_XSLFO	33007
XSL-Parameter/XQuery-Variablen...	IDC_TRANSFORM_XSL_PARAMS	33008

6.3.6 Menü "Authentic"

Befehle aus dem Menü "Authentic":

Menütext	Befehlsname	ID
Neues Dokument...	IDC_AUTHENTIC_NEW_FILE	34036
Datenbankdaten bearbeiten...	IDC_AUTHENTIC_EDIT_DB	34035
StyleVision Stylesheet zuweisen...	IDC_ASSIGN_SPS	34034
Neue Zeile mit XML-Daten für die Bearbeitung auswählen	IDC_CHANGE_WORKING_DB_XML_CELL	32861
XML Entities definieren...	IDC_DEFINE_ENTITIES	32805
Kleine Markup-Symbole einblenden	IDC_MARKUP_SMALL	34090
Große Markup-Symbole einblenden	IDC_MARKUP_LARGE	34088
Gemischte Markup-Symbole einblenden	IDC_MARKUP_MIX	34089
Zeile anhängen	IDC_ROW_APPEND	32806
Zeile einfügen	IDC_ROW_INSERT	32809
Zeile kopieren	IDC_ROW_DUPLICATE	32808
Zeile nach oben	IDC_ROW_MOVE_UP	32811
Zeile nach unten	IDC_ROW_MOVE_DOWN	32810
Zeile löschen	IDC_ROW_DELETE	32807

6.3.7 Menü "Ansicht"

Befehle aus dem Menü "Ansicht":

WSDL-Design-Ansicht	IDC_VIEW_WSDL	34117
XBRL-Taxonomieansicht	IDC_VIEW_XBRL	34118
Authentic-Ansicht	IDC_VIEW_CONTENT	34177
Browser-Ansicht	IDC_VIEW_BROWSER	34176
Einstellungen für Textansicht	IDC_TEXTVIEW_SETTINGS	34119

6.3.8 Menü "Browser"

Befehle aus dem Menü "Browser":

Menütext	Befehlsname	ID
Zurück	IDC_BROWSER_BACK	34039
Vorwärts	IDC_BROWSER_FORWARD	34045
Abbrechen	IDC_BROWSER_STOP	34047
Aktualisieren	IDC_BROWSER_REFRESH	34046
Schriftgröße/Maximale Größe	IDC_BROWSER_FONT_LARGEST	34041
Schriftgröße/Größer	IDC_BROWSER_FONT_LARGE	34040
Schriftgröße/Mittel	IDC_BROWSER_FONT_MEDIUM	34042
Schriftgröße/Kleiner	IDC_BROWSER_FONT_SMALL	34043
Schriftgröße/Minimale Größe	IDC_BROWSER_FONT_SMALLEST	34044
Neues Fenster	IDC_BROWSER_USE_OWN_FRAME	34048

6.3.9 Menü "Extras"

Befehle aus dem Menü "Extras":

Menütext	Befehlsname	ID
Rechtschreibung...	IDC_SPELL_CHECK	34154
Rechtschreibung - Optionen...	IDC_SPELL_OPTIONS	34155
Skripting-Umgebung...	ID_WINDOW_SWITCHTOVBA	34231
Makros anzeigen...	ID_WINDOW_VBAMACROS	34232
Projekt/Skripts zuweisen	ID_SCRIPTINGPRJ_ASSIGN	34214
Projekt/Projektskripts entkoppeln	ID_SCRIPTINGPRJ_UNASSIGN	34215
Projekt/Projektskripts aktivieren	ID_SCRIPTINGPRJ_ACTIVE	34213
Globale Ressourcen	IDC_GLOBALRESOURCES	37401
Aktive Konfiguration/<plugin not loaded>	IDC_GLOBALRESOURCES_SUBMENUENTRY1	37408
Anpassen...	IDC_CUSTOMIZE	34055
Optionen...	IDC_SETTINGS	33300
<placeholder>	ID_SCRIPTING_MACROITEMS	34249

6.3.10 Menü "Fenster"

Befehle aus dem Menü "Fenster":

Menütext	Befehlsname	ID
Überlappend	ID_WINDOW_CASCADE	57650
Horizontal anordnen	ID_WINDOW_TILE_HORZ	57651
Vertikal anordnen	ID_WINDOW_TILE_VERT	57652
Projekt-Fenster	IDC_PROJECT_WINDOW	34128
Info-Fenster	IDC_INFO_WINDOW	34085
Eingabehilfen	IDC_ENTRY_HELPERS	34062
Ausgabefenster	IDC_OUTPUT_DIALOGBARS	34004
Projekt und Eingabehilfen	IDC_PROJECT_ENTRYHELPERS	34006
Alle ein/aus	IDC_ALL_BARS	34031

6.3.11 Menü "Hilfe"

Befehle aus dem Menü "Hilfe":

Menütext	Befehlsname	ID
Inhaltsverzeichnis...	IDC_HELP_CONTENTS	34076
Index...	IDC_HELP_INDEX	34077
Suchen...	IDC_HELP_SEARCH	34079
Tastaturzuordnungen...	IDC_HELP_KEYMAPDLG	34078
Software-Aktivierung...	IDC_ACTIVATION	34005
Bestellformular...	IDC_OPEN_ORDER_PAGE	34094
Registrieren...	IDC_REGISTRATION	34131
Auf Updates überprüfen...	IDC_CHECK_FOR_UPDATES	34275
Support Center...	IDC_OPEN_SUPPORT_PAGE	34096
Fragen und Antworten im Web...	IDC_SHOW_FAQ	34153
Komponenten und Gratistools downloaden...	IDC_OPEN_COMPONENTS_PAGE	34093
XMLSpy im Internet..	IDC_OPEN_XML_SPY_HOME	34098
XMLSpy Training...	ID_HELP_XMLSPYTRAINING	34210
Über XMLSpy...	ID_APP_ABOUT	57664

6.3.12 Diverse Menübefehle

Diverse andere Menü- und Kontextmenübefehle:

Datenbank	IDC_ADDRESOURCE_DATABASE	37405
Datei	IDC_ADDRESOURCE_FILE	37403
Ordner	IDC_ADDRESOURCE_FOLDER	37404
Attribute	IDC_ADD_CHILD_ATTRIBUTE	33402
CData	IDC_ADD_CHILD_CDATA	33403
Comment	IDC_ADD_CHILD_COMMENT	33404
ATTLIST	IDC_ADD_CHILD_DEF_ATTLIST	33405
DOCTYPE	IDC_ADD_CHILD_DEF_DOCTYPE	33406
ELEMENT	IDC_ADD_CHILD_DEF_ELEMENT	33407
ENTITY	IDC_ADD_CHILD_DEF_ENTITY	33408
ExternalID	IDC_ADD_CHILD_DEF_EXTERNAL_ID	33409
NOTATION	IDC_ADD_CHILD_DEF_NOTATION	33410
Processing Instruction	IDC_ADD_CHILD_PI	33411
Element	IDC_ADD_CHILD_STRUCT	33412
Text	IDC_ADD_CHILD_TEXT	33413
XInclude...	IDC_ADD_CHILD_XINCLUDE	34027
XML	IDC_ADD_CHILD_XML	33414
Attribute	IDC_APPEND_ATTRIBUTE	33415
CData	IDC_APPEND_CDATA	33416
Comment	IDC_APPEND_COMMENT	33417
ATTLIST	IDC_APPEND_DEF_ATTLIST	33418
DOCTYPE	IDC_APPEND_DEF_DOCTYPE	33419
ELEMENT	IDC_APPEND_DEF_ELEMENT	33420
ENTITY	IDC_APPEND_DEF_ENTITY	33421
ExternalID	IDC_APPEND_DEF_EXTERNAL_ID	33422

NOTATION	IDC_APPEND_DEF_NOTATION	33423
Processing Instruction	IDC_APPEND_PI	33424
Element	IDC_APPEND_STRUCT	33425
Text	IDC_APPEND_TEXT	33426
XInclude...	IDC_APPEND_XINCLUDE	34026
XML	IDC_APPEND_XML	33427
DTD zuweisen...	IDC_ASSIGN_DTD	34032

XML-Beispieldatei zuweisen...	IDC_ASSIGN_SAMPLE_XML	33000
Schema zuweisen...	IDC_ASSIGN_SCHEMA	34033
StyleVision Stylesheet zuweisen...	IDC_ASSIGN_SPS	34034
XSL zuweisen...	IDC_ASSIGN_XSL	33001
XSL:FO zuweisen...	IDC_ASSIGN_XSLFO	33002
Unselektierte reduzieren	IDC_COLLAPSE_UNSELECTED	33428
Alles in globale Ressourcen konvertieren	IDC_CONVERT_ALL_DATASOURCES_TO_GLOBAL_RESOURCES	36687
In globale Ressourcen konvertieren	IDC_CONVERT_DATASOURCE_TO_GLOBAL_RESOURCES	36684
In MapForce auf andere(s) DTD/Schema oder DB mappen	IDC_DTD_OPENIN_MAPFORCE	34056
HTML/PDF-Ausgabe in StyleVision erstellen	IDC_DTD_OPENIN_STYLEVISION	34057
Globale Ressource bearbeiten...	IDC_EDIT_GLOBAL_RESOURCES	36686
Dateipfad einfügen...	IDC_EDIT_INSERT_PATH_STRING	34013
XInclude einfügen...	IDC_EDIT_INSERT_XINCLUDE_STRING	34017
Als Element einbinden	IDC_ENCLOSE_IN_ELEMENT	33446
Einchecken...	IDC_FILE_CHECK_IN	32951
Auschecken...	IDC_FILE_CHECK_OUT	32952
Auschecken rückgängig...	IDC_FILE_UNDO_CHECK_OUT	32953
In Dateien suchen...	IDC_FIND_IN_FILES	34000

Cache-Speicher leeren	IDC_FLUSH_CACHED_FILES	34066
Programmcode generieren...	IDC_GENERATE_CODE_FROM_SCHEMA	34067
Durchsuchen...	IDC_GLOBALRESOURCESUI_CHOOSEFILEASFILE	37420
Andere globale Ressource auswählen...	IDC_GLOBALRESOURCESUI_CHOOSEFILEASGR	37419
Konfiguration hinzufügen	IDC_GLOBALRESOURCESUI_DETAILS_ADDCONFIG	37423
Konfigurationskopie hinzufügen	IDC_GLOBALRESOURCESUI_DETAILS_ADDCONFIGCOPY	37424
Konfiguration löschen	IDC_GLOBALRESOURCESUI_DETAILS_DELETECONFIG	37425
Konfiguration der aktiven globalen Ressource	IDC_GLOBALRESOURCES_ACTIVECONFIG	37400
Löschen	IDC_GLOBALRESOURCES_MAINDLG_DELETERESOURCE	37422
Zum nächsten Lesezeichen	IDC_GOTONEXTBOOKMARK	34070

Zum vorherigen Lesezeichen	IDC_GOTOPREVBOOKMARK	34071
Gehe zu Definition	IDC_GOTO_DEFINITION	33447
Gehe zu DTD	IDC_GOTO_DTD	34072
Gehe zu Datei	IDC_GOTO_FILE	33448
Gehe zu Zeile/Zeichen	IDC_GOTO_LINE	34073
Gehe zu Schema	IDC_GOTO_SCHEMA	34074
Gehe zu XSL	IDC_GOTO_XSL	33004
Als Tabelle anzeigen	IDC_GRID_VIEW_AS_TABLE	34075
SELECT-Anweisung bearbeiten...	IDC_ICDBWNDS_EDIT_LOCALVIEW	36676
SELECT-Anweisung entfernen	IDC_ICDBWNDS_REMOVE_LOCALVIEW	36680
SELECT-Anweisung hinzufügen...	IDC_ICDBWND_CREATE_LOCALVIEW	36677
SELECT-Anweisung erzeugen und hinzufügen...	IDC_ICDBWND_GENEATE_SQL_FOR_LOCALVIEW	36678
Aus Online Browser entfernen	IDC_ICDBWND_REMOVE_FROM_BROWSERVIEW	36683
Aktive und verwandte Dateien hinzufügen	IDC_ICPROJECTGUI_ADD_ACTIVE_AND_RELATED_FILES	37217
Aktive Datei hinzufügen	IDC_ICPROJECTGUI_ADD_ACTIVE_FILE	37216

Externes Verzeichnis hinzufügen...	IDC_ICPROJECTGUI_ADD_EXT_FOLDER	37219
Externes Webverzeichnis hinzufügen...	IDC_ICPROJECTGUI_ADD_EXT_URL_FOLDER	37220
Dateien hinzufügen...	IDC_ICPROJECTGUI_ADD_FILES	37213
Projektordner hinzufügen...	IDC_ICPROJECTGUI_ADD_FOLDER	37218
Globale Ressource hinzufügen...	IDC_ICPROJECTGUI_ADD_GLOBAL_RESOURCE	37238
URL hinzufügen...	IDC_ICPROJECTGUI_ADD_URL	37214
Eigenschaften...	IDC_ICPROJECTGUI_PROPERTIES	37222
Aktualisieren	IDC_ICPROJECTGUI_REFRESH_EXT_FOLDER	37221
Weitere DTD einfügen...	IDC_INCLUDE_DTD	34084
Höher stufen	IDC_MOVE_LEFT	34091
Tiefer stufen	IDC_MOVE_RIGHT	34092
Namespace-Präfix...	IDC_NAMESPACE	33462
Optimale Breite	IDC_OPTIMAL_WIDTHS	34099

0 oder mehr Übereinstimmungen	IDC_RECM_0MATCH	33901
1 oder mehr Übereinstimmungen	IDC_RECM_1MATCH	33902
Beliebiges Zeichen	IDC_RECM_ANYCHAR	33903
Zeichen in Bereich	IDC_RECM_CHARINRANGE	33904
Zeichen außerhalb des Bereichs	IDC_RECM_CHARNOTINRANGE	33905
Zeilenende	IDC_RECM_ENDLINE	33906
Wortende	IDC_RECM_ENDWORD	33907
Zeilenanfang	IDC_RECM_STARTLINE	33908
Wortanfang	IDC_RECM_STARTWORD	33909
Getaggtter Ausdruck	IDC_RECM_TAGGEDEXP	33910
Reduzieren -	IDC_SEL_COLLAPSE	34151
Erweitern +	IDC_SEL_EXPAND	34152
Alles erweitern	IDC_SEL_EXPAND_ALL	33463

Zeile anhängen	IDC_TABLE_APPEND_ROW	34157
Zeile einfügen	IDC_TABLE_INSERT_ROW	34158
Aufsteigend sortieren	IDC_TABLE_SORT_ASC	33464
Absteigend sortieren	IDC_TABLE_SORT_DESC	33465
Lesezeichen einfügen/löschen	IDC_TOGGLE_BOOKMARK	34162
Ein-/Auskommentieren	IDC_TOGGLE_XML_COMMENT	34029
Eingabehilfen aktualisieren	IDC_UPDATE_ELEMENT_CHOICE	34173
Meldungsfenster	IDC_VALIDATOR_OUTPUT	34175
Zeilenumbruch	IDC_WORD_WRAP	34181
Aktive Datei zu Projekt hinzufügen	ID_ADDACTIVEFILETOPROJECT	36549
Dateien zu Projekt hinzufügen ...	ID_ADDFILESTOPROJECT	36550
Ordner erstellen	ID_CREATEFOLDER	36555
Verbindung trennen	ID_DISCONNECT	36559
Verbindung zu allen Datenquellen trennen	ID_DISCONNECTFROMALLDATASOURCES	36560
Daten bearbeiten	ID_EDITRESULTDATA	36646
Umbenennen	ID_EDIT_RENAME	36563
Projekt öffnen	ID_FILE_LOAD_PROJECT	36565
Fügt eine Konfiguration hinzu	ID_GLOBALRESOURCES_ADDCONFIG	37429

Fügt eine Konfiguration als Kopie der ausgewählten Konfiguration hinzu	ID_GLOBALRESOURCES_ADDCONFIGCOPY	37430
Löscht eine Konfiguration	ID_GLOBALRESOURCES_DELCONFIG	37431
Referenzierte Tabelle anzeigen	ID_ICDBWND_FGNKEY_GOTO_REFERENCE	36567
In XMLSpy anzeigen	ID_ICDBWND_VIEW_XMLSCHEMA_IN_XMLSPY	36500
Neue Datenquelle hinzufügen	ID_ICDBWND_ADDANEWDATASOURCE	36568
Zu Design Editor hinzufügen	ID_ICDBWND_BROWSER_ADD_TO_DESIGNVIEW	36569
Zu Favoriten hinzufügen/aus Favoriten löschen	ID_ICDBWND_BROWSER_ADD_TO_FAVOURITES	36570
Aktualisieren	ID_ICDBWND_BROWSER_REFRESH_ROOT	36571

Alle	ID_ICDBWND_BROWSER_SEARCH_MODE_ALL	36575
Aus der aktuellen Datenquelle	ID_ICDBWND_BROWSER_SEARCH_MODE_DATASOURCE	36576
Aus dem aktiven Objekt	ID_ICDBWND_BROWSER_SEARCH_MODE_FOCUSED_ITEM	36577
In neuem Design Editor anzeigen	ID_ICDBWND_BROWSER_SHOW_IN_DESIGNVIEW	36578
Aktivieren	ID_ICDBWND_CHECK	36579
Children aktivieren	ID_ICDBWND_CHECK_ALL	36580
Löschen	ID_ICDBWND_CLEAR_ROWCOUNT	36538
Children	ID_ICDBWND_COLLAPSE_CHILDREN	36581
Gleichrangige	ID_ICDBWND_COLLAPSE_SIBLING	36582
SQL ausführen	ID_ICDBWND_EXECUTE	36583
Children	ID_ICDBWND_EXPAND_CHILDREN	36584
Gleichrangige	ID_ICDBWND_EXPAND_SIBLINGS	36585
Datenbankdaten exportieren	ID_ICDBWND_EXPORT	36586
Hinzufügen	ID_ICDBWND_FILEDSN_ADD	36588
Löschen	ID_ICDBWND_FILEDSN_DELETE	36589
Enthält	ID_ICDBWND_FILTER_CONTAINS	36591
Enthält nicht	ID_ICDBWND_FILTER_DOES_NOT_CONTAIN	36592
Endet mit	ID_ICDBWND_FILTER_ENDS_WITH	36593
Ist gleich	ID_ICDBWND_FILTER_EQUALS	36594
Kein Filter	ID_ICDBWND_FILTER_INACTIVE	36596
Beginnt mit	ID_ICDBWND_FILTER_STARTS_WITH	36597
Verbindung herstellen	ID_ICDBWND_MENU_DATASOURCE_CONNECT	36605

Verbindung trennen	ID_ICDBWND_MENU_DATASOURCE_DISCONNECT	36606
Tabellen abrufen	ID_ICDBWND_MENU_DATASOURCE_GETTABLES	36607
Vorschau	ID_ICDBWND_PREVIEWITEM	36608
Alle Favoriten entfernen	ID_ICDBWND_REMOVE_ALL_FAVORITES	36609
Ein-/ausblenden	ID_ICDBWND_TOGGLE	36610
Deaktivieren	ID_ICDBWND_UNCHECK	36611

Children deaktivieren	ID_ICDBWND_UNCHECK_ALL	36612
Anzeigen/Aktualisieren	ID_ICDBWND_UPDATE_ROWCOUNT	36537
Öffnen	ID_OPEN	36619
Dateierweiterung hinzufügen	ID_POPUP_ADDEXTENSION	35003
Suchpfad hinzufügen	ID_POPUP_ADDSEARCHPATH	35004
Dateierweiterung löschen	ID_POPUP_DELETEEXTENSION	35005
Suchpfad löschen	ID_POPUP_DELETESEARCHPATH	35006
Dateierweiterung bearbeiten	ID_POPUP_EDITEXTENSION	35007
Suchpfad bearbeiten	ID_POPUP_EDITSEARCHPATH	35008
Suchpfade neu laden	ID_POPUP_RELOAD_SEARCH_PATHS	35009
Suchpfade zurücksetzen	ID_POPUP_RESET_SEARCH_PATHS	33810
Debugger starten	ID_PROCESS_XSL	34212
Neues Projekt	ID_PROJECT_NEWPROJECT	36620
Anhängen	ID_RELDISP_APPEND_LAYER	33201
Löschen	ID_RELDISP_DELETE_LAYER	33202
Nächstes Objekt auf aktiver Ebene aktiv setzen	ID_RELDISP_FOCUS_NEXT_ITEM_ON_LAYER	33227
Vorheriges Objekt auf aktiver Ebene aktiv setzen	ID_RELDISP_FOCUS_PREV_ITEM_ON_LAYER	33228
Ausgewählte Ebene	ID_RELDISP_HIDE_LAYER	33203
Andere	ID_RELDISP_HIDE_LAYER_OTHER	33204
Einfügen	ID_RELDISP_INSERT_LAYER	33205
Ausgewählte Ebene	ID_RELDISP_LOCK_LAYER	33206
Andere	ID_RELDISP_LOCK_LAYER_OTHER	33207
Zurück	ID_RELDISP_MOVE_LAYER_ITEMS_BACK	33208
Nach vorne	ID_RELDISP_MOVE_LAYER_ITEMS_FRONT	33209
Ausgewählte Ebene	ID_RELDISP_PALE_LAYER	33210

Andere	ID_RELDISP_PALE_LAYER_OTHER	33211

Ausgewählte Ebene	ID_RELDISP_RECOLOR_LAYER	33212
Andere	ID_RELDISP_RECOLOR_LAYER_OTHER	33213
Umbenennen	ID_RELDISP_RENAME_LAYER	33214
Zustand aller Ebenen zurücksetzen	ID_RELDISP_RESET_LAYER_STATES	33231
Auswählen	ID_RELDISP_SELECT_ITEMS	33215
Andere auswählen	ID_RELDISP_SELECT_OTHER_ITEMS	33216
Ausgewählte Ebene	ID_RELDISP_SHOW_LAYER	33217
Anzahl der Ebenenobjekte anzeigen	ID_RELDISP_SHOW_LAYER_ITEM_COUNT	33229
Andere	ID_RELDISP_SHOW_LAYER_OTHER	33218
Ausgewählte Ebene	ID_RELDISP_TOGGLE_COLORING	33219
Andere	ID_RELDISP_TOGGLE_COLORING_OTHER	33220
Ausgewählte Ebene	ID_RELDISP_TOGGLE_LAYER_VISIBILITY	33221
Andere	ID_RELDISP_TOGGLE_LAYER_VISIBILITY_OTHER	33222
Ausgewählte Ebene	ID_RELDISP_TOGGLE_LOCK	33223
Andere	ID_RELDISP_TOGGLE_LOCK_OTHER	33224
Ausgewählte Ebene	ID_RELDISP_UNLOCK_LAYER	33225
Andere	ID_RELDISP_UNLOCK_LAYER_OTHER	33226
Alle Datenquellen entfernen	ID_REMOVEALLDATASOURCES	36621
Entfernen	ID_REMOVE_DATASOURCE	36622
Aus Favoriten entfernen	ID_REMOVE_FAVORITE_ITEM	36623
Entfernen	ID_REMOVE_FROM_PROJECT	36624
Alle Zeilen	ID_RETRIEVE_ALLROWS	36625
Erste n Zeilen	ID_RETRIEVE_FIRSTROWS	36626
Projekt speichern	ID_SAVEPROJECT	36627
Projekt speichern unter...	ID_SAVEPROJECTAS	36628

6.4 Aufrufen von AuthenticDesktopAPI

Diese Dokumentation beschäftigt sich hauptsächlich mit den ActiveX Controls und Schnittstellen, die für die Integration der AuthenticDesktop Benutzeroberfläche in Ihre Applikation benötigt werden. Über die folgenden Eigenschaften haben Sie Zugriff auf das AuthenticDesktop Automation Interface (AuthenticDesktopAPI), um die Funktionalitäten der integrierten Komponenten zu automatisieren oder zu definieren.

[AuthenticDesktopControl.Application](#)

[AuthenticDesktopControlDocument.Document](#)

[AuthenticDesktopControlPlaceholder.Project](#)

Für die Verwendung des AuthenticDesktop Automation Interface gelten beim Integrieren von AuthenticDesktopControl auf Dokumentebene einige Einschränkungen. Nähere Informationen dazu finden Sie unter [Integration auf Dokumentebene](#).

6.5 Objektreferenz

Objekte:

[AuthenticDesktopCommand](#)

[AuthenticDesktopCommands](#)

[AuthenticDesktopControl](#)

[AuthenticDesktopControlDocument](#)

[AuthenticDesktopControlPlaceHolder](#)

Um Zugriff auf die AuthenticDesktop Standardfunktionalitäten zu erhalten, können auch Objekte des **AuthenticDesktop Automation Interface** aufgerufen werden. Nähere Informationen dazu finden Sie unter [AuthenticDesktopControl.Application](#), [AuthenticDesktopControlDocument.Document](#) und [AuthenticDesktopControlPlaceHolder.Project](#).

6.5.1 AuthenticDesktopCommand

Eigenschaften:

[ID](#)
[Label](#)
[IsSeparator](#)
[ToolTip](#)
[StatusText](#)
[Accelerator](#)
[SubCommands](#)

Beschreibung:

Jedes `MapForceCommand` Objekt kann einer von drei möglichen Typen sein:

- **Befehl:** `ID` ist auf einen Wert größer als 0 gesetzt und `Label` ist auf "Command Name" gesetzt. `IsSeparator` ist `false` und die `SubCommands` Collection ist leer.
- **Trennzeichen:** `IsSeparator` ist `true`. `ID` ist 0 und `Label` ist nicht definiert. Die `SubCommands` Collection ist leer.
- **(Unter)-Menü:** Die `SubCommands` Collection enthält [Command](#) Objekte und `Label` ist der Name des Menüs. `ID` ist auf 0 gesetzt und `IsSeparator` ist `false`.

Accelerator

Eigenschaft: `Label` als `string`

Beschreibung:

Dies ist bei Befehlsobjekten, die Children der `ALL_COMMANDS` Collection sind, die für den Befehl definierte Zugriffstaste. Wenn dem Befehl keine Zugriffstaste zugewiesen wurde, gibt diese Eigenschaft den leeren String zurück.

Die String-Darstellung der Zugriffstaste hat das folgenden Format:

[ALT+] [CTRL+] [SHIFT+] key

key wird mittels der Windows Platform SDK-Funktion `GetKeyNameText` konvertiert.

ID

Eigenschaft: `ID` als `long`

Beschreibung:

`ID` ist 0 für Trennzeichen und Menüs.

Bei Befehlen ist dies die ID, die mit [Exec](#) und [QueryStatus](#) verwendet werden kann.

IsSeparator

Eigenschaft: `IsSeparator` als `boolean`

Beschreibung:

True, wenn der Befehl ein Trennzeichen ist.

Label

Beschreibung:

Label ist bei Trennzeichen leer.

Bei Befehlsobjekten, die Children der ALL_COMMANDS Collection sind, ist dies ein eindeutiger Name. Die Symbole für die Befehlsschaltflächen sind in Dateien mit diesem Namen gespeichert. Nähere Informationen dazu siehe [Abfragen von Befehlen](#).

Bei Befehlsobjekten, die Children von Menüs sind, enthält die Eigenschaft "Label" den Menüttext des Befehls. Bei Untermenüs enthält diese Eigenschaft den Menüttext.

StatusText

Eigenschaft: Label als [string](#)

Beschreibung:

Bei Befehlsobjekten, die Children der ALL_COMMANDS Collection sind, ist dies der Text, der in der Statusleiste angezeigt wird, wenn der Befehl ausgewählt ist.

SubCommands

Eigenschaft: SubCommands als [Commands](#)

Beschreibung:

Die SubCommands Collection enthält alle untergeordneten Befehle, wenn dieser Befehl tatsächlich ein Menü oder Untermenü ist.

ToolTip

Eigenschaft: ToolTip als [string](#)

Beschreibung:

Bei Befehlsobjekten, die Children der ALL_COMMANDS Collection sind, ist dies der Text, der als Tooltipp angezeigt wird.

6.5.2 AuthenticDesktopCommands

Eigenschaften:

[Count](#)

[Item](#)

Beschreibung:

Sammlung von [Command](#)-Objekten für den Zugriff auf Befehlsbezeichnungen und IDs des AuthenticDesktopControl. Diese Befehle können mit der [Exec](#) Methode ausgeführt werden und ihr Status kann mittels [QueryStatus](#) abgefragt werden.

Count

Eigenschaft: `Count` als `long`

Beschreibung:

Anzahl der [Command](#) Objekte auf dieser Ebene der Collection

Item

Eigenschaft: `Item (n als long)` als [Command](#)

Beschreibung:

Ruft den Befehl mit den Index `n` in dieser Sammlung auf. Der Index basiert auf 1.

6.5.3 AuthenticDesktopControl

Eigenschaften:[IntegrationLevel](#)[Appearance](#)[Application](#)[BorderStyle](#)[CommandsList](#)

CommandsStructure (deprecated)

[EnableUserPrompts](#)[MainMenu](#)[Toolbars](#)**Methoden:**[Open](#)[Exec](#)[QueryStatus](#)**Events:**[OnUpdateCmdUI](#)[OnOpenedOrFocused](#)[OnCloseEditingWindow](#)[OnFileChangedAlert](#)[OnContextChanged](#)[OnDocumentOpened](#)[OnValidationWindowUpdated](#)

Dieses Objekt ist ein vollständiges ActiveX Control und sollte nur sichtbar sein, wenn die AuthenticDesktop Bibliothek im Applikationsebenenmodus verwendet wird.

Eigenschaften

Es sind die folgenden Eigenschaften definiert:

[IntegrationLevel](#)[EnableUserPrompts](#)[Appearance](#)[BorderStyle](#)

Befehlsbezogene Eigenschaften:

[CommandsList](#)[MainMenu](#)[Toolbars](#)

CommandsStructure (nicht mehr verwendet)

Zugriff auf die AuthenticDesktopAPI:

[Application](#)

Appearance

Eigenschaft: Appearance als [short](#)

Dispatch Id: -520

Beschreibung:

Bei einem Wert, der nicht gleich 0 ist, wird ein Client-Rand rund um das Control angezeigt. Der Standardwert ist 0.

Application

Eigenschaft: `Application` als [Application](#)

Dispatch Id: 1

Beschreibung:

Mit der Eigenschaft `Application` erhalten Sie Zugriff auf das `Application` Objekt der vollständigen AuthenticDesktop Automation Server API. Die Eigenschaft ist schreibgeschützt.

BorderStyle

Eigenschaft: `BorderStyle` als [short](#)

Dispatch Id: -504

Beschreibung:

Bei einem Wert 1 wird das Control mit einer dünnen Umrandung angezeigt. Der Standardwert ist 0.

CommandsList

Eigenschaft: `CommandList` als [Commands](#) (schreibgeschützt)

Dispatch Id: 1004

Beschreibung:

Diese Eigenschaft gibt eine flache Liste aller Befehle zurück, die mit `AuthenticDesktopControl` verfügbar sind.

EnableUserPrompts

Eigenschaft: `EnableUserPrompts` als [boolean](#)

Dispatch Id: 1006

Beschreibung:

Wenn Sie diese Eigenschaft auf *false* setzen, wird die Eingabeaufforderung im Control deaktiviert. Der Standardwert ist *true*.

IntegrationLevel

Eigenschaft: `IntegrationLevel` als [ICActiveXIntegrationLevel](#)

Dispatch Id: 1000

Beschreibung:

Die Eigenschaft `IntegrationLevel` bestimmt den Operationsmodus des Control. Nähere Informationen dazu siehe auch [Integration auf Applikationsebene](#) und [Integration auf Dokumentenebene](#).

Anmerkung: Diese Eigenschaft muss unbedingt sofort nach Erstellung des `AuthenticDesktopControl` Objekts definiert werden.

MainMenu

Eigenschaft: `MainMenu` als [Command](#) (schreibgeschützt)

Dispatch Id: 1003

Beschreibung:

Mit dieser Eigenschaft erhalten Sie Zugriff auf die Beschreibung des `AuthenticDesktopControl` Hauptmenüs.

Toolbars

Eigenschaft: `Toolbars` als [Commands](#) (schreibgeschützt)

Dispatch Id: 1005

Beschreibung:

Diese Eigenschaft gibt eine Liste aller Symbolleistenbeschreibungen zurück, die alle mit `AuthenticDesktopControl` verfügbaren Symbolleisten beschreiben.

Methoden

Es sind die folgenden Methoden definiert:

[Open](#)

[Exec](#)

[QueryStatus](#)

Exec

Methode: `Exec` (`nCmdID` als [long](#)) als [boolean](#)

Dispatch Id: 6

Beschreibung:

`Exec` ruft den `AuthenticDesktop` Befehl mit der ID `nCmdID` auf. Wenn der Befehl ausgeführt werden kann, gibt die Methode `true` zurück. Eine Liste aller verfügbaren Befehle finden Sie unter `CommandsStructure` und eine Beschreibung, wie man den Status eines Befehls abrufen, finden Sie unter [QueryStatus](#).

Open

Methode: `Open` (`strFilePath` als [string](#)) als [boolean](#)

Dispatch Id: 5

Beschreibung:

Das Ergebnis der Methode ist von der Erweiterung abhängig, die im Argument `strFilePath` übergeben wird. Ist die Dateierweiterung `.mfd`, wird ein neues Dokument geöffnet. Ist die Dateierweiterung `.mfp`, wird das entsprechende Projekt geöffnet. Wird eine andere

Dateierweiterung in die Methode übergeben, versucht das Control, die Datei als neue Komponente in das aktive Dokument zu laden.

Verwenden Sie diese Methode nicht, um Dokumente oder Projekte bei Verwendung des Control auf Dokumentebene zu laden. Verwenden Sie statt dessen

[AuthenticDesktopControlDocument.Open](#) und
[AuthenticDesktopControlPlaceHolder.OpenProject](#).

QueryStatus

Methode: `QueryStatus` (`nCmdID` als `long`) als `long`

Dispatch Id: 7

Beschreibung:

`QueryStatus` gibt den Status "enabled/disabled" und "checked/unchecked" des von `nCmdID` definierten Befehls zurück. Der Status wird als Bitmaske zurückgegeben.

Bit	Wert	Name	Bedeutung
0	1	Supported	Setzen, wenn der Befehl unterstützt wird.
1	2	Enabled	Setzen, wenn der Befehl aktiviert ist (ausgeführt werden kann).
2	4	Checked	Setzen, wenn der Befehl angehakt ist.

Das bedeutet, dass die Befehls-ID bei Rückgabe von 0 durch `QueryStatus` nicht als gültiger AuthenticDesktop Befehl erkannt wird. Wenn `QueryStatus` einen Wert 1 oder 5 zurückgibt, wird der Befehl deaktiviert.

Events

Das AuthenticDesktopControl ActiveX Control stellt die folgenden Verbindungspunkt-Events bereit:

[OnUpdateCmdUI](#)
[OnOpenedOrFocused](#)
[OnCloseEditingWindow](#)
[OnFileChangedAlert](#)
[OnContextChanged](#)

[OnDocumentOpened](#)
[OnValidationWindowUpdated](#)

OnCloseEditingWindow

Event: `OnCloseEditingWindow` (`i_strFilePath` als `String`) als `boolean`

Dispatch Id: 1002

Beschreibung:

Dieses Event wird ausgelöst, wenn AuthenticDesktop ein bereits geöffnetes Dokument schließen muss. Als Antwort auf dieses Event müssen Clients das mit `i_strFilePath` verknüpfte Bearbeitungsfenster schließen. Bei Rückgabe von `true` von diesem Event, wird angezeigt, dass der Client das Dokument geschlossen hat. Clients können `false` zurückgeben, wenn keine bestimmte Behandlung erforderlich ist und AuthenticDesktopControl versuchen soll, das

Bearbeitungsfenster zu schließen und das damit verknüpfte Dokument-Control zu zerstören.

OnContextChanged

Event: OnContextChanged (i_strContextName als [String](#), i_bActive als [bool](#)) als [bool](#)

Dispatch Id: 1004

Beschreibung:

Dieses Event wird in AuthenticDesktop nicht verwendet

OnDocumentOpened

Event: OnDocumentOpened (objDocument als [Document](#))

Dispatch Id: 1

Beschreibung:

Dieses Event wird immer, wenn ein Dokument geöffnet wird, ausgelöst. Das Argument `objDocument` ist ein `Document` Objekt aus dem AuthenticDesktop Automation Interface und kann dazu verwendet werden, weitere Details zum Dokument abzurufen oder weitere Operationen durchzuführen. Bei Integration auf Dokumentebene ist es oft besser, stattdessen das Event [AuthenticDesktopControlDocument.OnDocumentOpened](#) zu verwenden.

OnFileChangedAlert

Event: OnFileChangedAlert (i_strFilePath als [String](#)) als [bool](#)

Dispatch Id: 1001

Beschreibung:

Dieses Event wird ausgelöst, wenn eine mit AuthenticDesktopControl geladene Datei auf der Festplatte von einer anderen Applikation geändert wurde. Clients sollten `true` zurückgeben, wenn Sie das Event behandelt haben oder `false`, wenn AuthenticDesktop es auf die übliche Art behandeln soll, also fragen soll, ob das Dokument neu geladen werden soll.

OnLicenseProblem

Event: OnLicenseProblem (i_strLicenseProblemText als [String](#))

Dispatch Id: 1005

Beschreibung:

Dieses Event wird ausgelöst, wenn AuthenticDesktopControl feststellt, dass für dieses Control keine gültige Lizenz vorhanden ist. Wenn die Lizenz auf eine bestimmte Anzahl an Benutzern beschränkt ist, kann dies auch einige Zeit nach Initialisierung des Control geschehen. Dieses Event sollte dazu verwendet werden, um den Zugriff auf die Funktionalität dieses Control zu deaktivieren. Nach diesem Event blockiert dieses Control den Zugriff auf seine Funktionalitäten (und zeigt z.B. leere Fenster in seinen Controls an und gibt bei Anfragen Fehlermeldungen zurück).

OnOpenedOrFocused

Event: OnOpenedOrFocused (i_strFilePath als [String](#), i_bOpenWithThisControl als [bool](#))

Dispatch Id: 1000**Beschreibung:**

Bei Integration auf Applikationsebene informiert dieses Event Clients, dass ein Dokument von AuthenticDesktop geöffnet oder aktiv gemacht wurde.

Bei Integration auf Dokumentebene gibt dieses Event dem Client die Anweisung, die Datei `i_strFilePath` in einem Dokumentfenster zu öffnen. Wenn die Datei bereits offen ist, sollte das entsprechende Dokumentfenster zum aktiven Fenster gemacht werden.

Wenn `i_bOpenWithThisControl true` ist, muss das Dokument mit `AuthenticDesktopControl` geöffnet werden, da ein interner Zugriff erforderlich ist. Andernfalls kann die Datei mit anderen Editoren geöffnet werden.

OnToolWindowUpdated

Event: `OnToolWindowUpdated( pToolWnd als long )`

Dispatch Id: 1006**Beschreibung:**

Dieses Event wird ausgelöst, wenn das Fenster "Extras" aktualisiert wird.

OnUpdateCmdUI

Event: `OnUpdateCmdUI ( )`

Dispatch Id: 1003**Beschreibung:**

Wird häufig aufgerufen, um dem Programmierer die Möglichkeit zu geben, den Status von AuthenticDesktop Befehls mittels [AuthenticDesktopControl.QueryStatus](#) zu überprüfen. Führen Sie an diesem Callback keine langen Operationen durch.

OnValidationWindowUpdated

Event: `OnValidationWindowUpdated ( )`

Dispatch Id: 3**Beschreibung:**

Dieses Event wird ausgelöst, wenn das Fenster "Validierungsausgabe" mit neuen Informationen aktualisiert wird.

6.5.4 AuthenticDesktopControlDocument

Eigenschaften:

[Appearance](#)
[BorderStyle](#)
[Document](#)
[IsModified](#)
[Path](#)
[ReadOnly](#)

Methoden:

[Exec](#)
[New](#)
[Open](#)
[QueryStatus](#)
[Reload](#)
[Save](#)
[SaveAs](#)

Events:

[OnDocumentOpened](#)
[OnDocumentClosed](#)
[OnModifiedFlagChanged](#)
[OnContextChanged](#)
[OnFileChangedAlert](#)
[OnActivate](#)

Wenn das AuthenticDesktopControl im Modus auf Dokumentebene integriert ist, wird jedes Dokument in einem eigenen Objekt vom Typ `AuthenticDesktopControlDocument` angezeigt. Das `AuthenticDesktopControlDocument` enthält immer nur ein Dokument, kann aber wiederverwendet werden, um hintereinander mehrere Dateien anzuzeigen.

Dieses Objekt ist ein komplettes ActiveX Control.

Eigenschaften

Es sind die folgenden Eigenschaften definiert:

[ReadOnly](#)
[IsModified](#)
[Path](#)
[Appearance](#)
[BorderStyle](#)

Zugriff auf die AuthenticDesktopAPI:

[Document](#)

Appearance

Eigenschaft: Appearance als `short`

Dispatch Id: -520

Beschreibung:

Bei einem Wert, der nicht gleich 0 ist, wird ein Client-Rand rund um das Control angezeigt. Der Standardwert ist 0.

BorderStyle

Eigenschaft: `BorderStyle` als `short`

Dispatch Id: -504

Beschreibung:

Bei einem Wert 1 wird das Control mit einer dünnen Umrandung angezeigt. Der Standardwert ist 0.

Document

Eigenschaft: `Document` als `Document`

Dispatch Id: 3

Beschreibung:

Mit der Eigenschaft `Document` erhalten Sie Zugriff auf das `Document` Objekt der AuthenticDesktop Automation Server API. Diese Schnittstelle bietet zusätzliche Funktionalitäten, die mit dem im Control geladenen Dokument verwendet werden können. Die Eigenschaft ist schreibgeschützt.

IsModified

Eigenschaft: `IsModified` als `boolean` (schreibgeschützt)

Dispatch Id: 1006

Beschreibung:

`IsModified` ist `true`, wenn der Dokumentinhalt seit dem letzten Öffnen, Neuladen oder Speichern geändert wurde. Andernfalls ist es `false`.

Path

Eigenschaft: `Path` als `String`

Dispatch Id: 1005

Beschreibung:

Definiert den vollständigen Pfadnamen des im Control geladenen Dokuments bzw. ruft diesen ab.

ReadOnly

Eigenschaft: `ReadOnly` als `boolean`

Dispatch Id: 1007

Beschreibung:

Mit Hilfe dieser Eigenschaft können Sie den schreibgeschützten Status des Dokuments aktivieren und deaktivieren. Wenn `ReadOnly true` ist, können keine Änderungen vorgenommen werden.

Methoden

Es sind die folgenden Methoden definiert:

Behandlung von Dokumenten:

[New](#)

[Open](#)

[Reload](#)

[Save](#)

[SaveAs](#)

Behandlung von Befehlen:

[Exec](#)

[QueryStatus](#)

Exec

Methode: `Exec (nCmdID as long) als boolean`

Dispatch Id: 8

Beschreibung:

`Exec` ruft den AuthenticDesktop Befehl mit der ID `nCmdID` auf. Wenn der Befehl ausgeführt werden kann, gibt die Methode `true` zurück. Der Client sollte die `Exec` Methode des Dokuments aufrufen, wenn in der Applikation gerade ein aktives Dokument verfügbar ist.

Eine Liste aller verfügbaren Befehle finden Sie unter `CommandsStructure` und eine Beschreibung, wie man den Status eines Befehls abrufen, finden Sie unter [QueryStatus](#).

New

Methode: `New () als boolean`

Dispatch Id: 1000

Beschreibung:

Diese Methode initialisiert ein neues Mapping innerhalb des Control.

Open

Methode: `Open (strFilePath als string) als boolean`

Dispatch Id: 1001

Beschreibung:

`Open` lädt die Datei `strFileName` als das neue Dokument in das Control.

QueryStatus

Methode: `QueryStatus (nCmdID als long) als long`

Dispatch Id: 9

Beschreibung:

`QueryStatus` gibt den Status "enabled/disabled" und "checked/unchecked" des von `nCmdID` definierten Befehls zurück. Der Status wird als Bitmaske zurückgegeben.

Bit	Wert	Name	Bedeutung
0	1	Supported	Setzen, wenn der Befehl unterstützt wird.
1	2	Enabled	Setzen, wenn der Befehl aktiviert ist (ausgeführt werden kann).
2	4	Checked	Setzen, wenn der Befehl angehakt ist.

Das bedeutet, dass die Befehls-ID bei Rückgabe von 0 durch `QueryStatus` nicht als gültiger AuthenticDesktop Befehl erkannt wird. Wenn `QueryStatus` einen Wert 1 oder 5 zurückgibt, ist der Befehl deaktiviert. Der Client sollte die `QueryStatus` Methode des Dokument-Control aufrufen, wenn in der Applikation gerade ein aktives Dokument vorhanden ist.

Reload

Methode: `Reload ()` als `boolean`

Dispatch Id: 1002

Beschreibung:

`Reload` aktualisiert den Dokumentinhalt aus dem Dateisystem.

Save

Methode: `Save ()` als `boolean`

Dispatch Id: 1003

Beschreibung:

`Save` speichert das aktuelle Dokument unter dem Pfad [Path](#).

SaveAs

Methode: `OpenDocument (strFileName als string) als boolean`

Dispatch Id: 1004

Beschreibung:

`SaveAs` setzt [Path](#) auf `strFileName` und speichert das Dokument anschließend unter diesem Pfad.

Events

Das AuthenticDesktopControlDocument ActiveX Control stellt die folgenden Verbindungspunkt-Events zur Verfügung:

[OnDocumentOpened](#)
[OnDocumentClosed](#)
[OnModifiedFlagChanged](#)
[OnContextChanged](#)

[OnFileChangedAlert](#)
[OnActivate](#)
[OnSetEditorTitle](#)

OnActivate

Event: OnActivate ()

Dispatch Id: 1005

Beschreibung:

Dieses Event wird ausgelöst, wenn das Dokument-Control aktiviert ist, den Fokus hat und bereit für die Benutzereingabe ist.

OnContextChanged

Event: OnContextChanged (i_strContextName als [String](#), i_bActive als [bool](#)) als [bool](#)

Dispatch Id: 1004

Beschreibung: Keine

OnDocumentClosed

Event: OnClosed ()

Dispatch Id: 1001

Beschreibung:

Dieses Event wird immer dann ausgelöst, wenn das in dieses Control geladene Event geschlossen wird. Das Argument `objDocument` ist ein `Document` Objekt aus dem AuthenticDesktop Automation Interface und sollte mit Vorsicht verwendet werden.

OnDocumentOpened

Event: OnDocumentOpened (objDocument als [Document](#))

Dispatch Id: 1000

Beschreibung:

Dieses Event wird immer dann ausgelöst, wenn ein Dokument in diesem Control geöffnet wird. Das Argument `objDocument` ist ein `Document` Objekt aus dem AuthenticDesktop Automation Interface und dient dazu, nähere Informationen über das Dokument abzurufen oder weitere Operationen auszuführen.

OnDocumentSaveAs

Event: OnContextDocumentSaveAs (i_strFileName als [String](#))

Dispatch Id: 1007

Beschreibung:

Dieses Event wird ausgelöst, wenn dieses Dokument intern unter einem neuen Namen gespeichert wird.

OnFileChangedAlert

Event: OnFileChangedAlert () als bool

Dispatch Id: 1003

Beschreibung:

Dieses Event wird ausgelöst, wenn eine in dieses Control geladene Datei auf der Festplatte von einer anderen Applikation geändert wurde. Clients sollten true zurückgeben, wenn Sie das Event behandelt haben oder false, wenn AuthenticDesktop es auf die übliche Art behandeln soll, also fragen soll, ob das Dokument neu geladen werden soll.

OnModifiedFlagChanged

Event: OnModifiedFlagChanged (i_bIsModified als boolean)

Dispatch Id: 1002

Beschreibung:

Dieses Event wird immer dann ausgelöst, wenn das Dokument zwischen dem Status "geändert" und "nicht geändert" wechselt. Der Parameter *i_bIsModified* ist *true*, wenn sich der Inhalt des Dokuments vom ursprünglichen Inhalt unterscheidet und ist andernfalls *false*.

OnSetEditorTitle

Event: OnSetEditorTitle ()

Dispatch Id: 1006

Beschreibung:

Dieses Event wird ausgelöst, wenn das enthaltene Dokument intern umbenannt wird.

6.5.5 AuthenticDesktopControlPlaceHolder

Eigenschaften für alle Arten von Platzhalterfenstern:

[PlaceholderWindowID](#)

Eigenschaften für das Projekt-Platzhalterfenster:

[Project](#)

Methoden für das Projekt-Platzhalterfenster:

[OpenProject](#)

[CloseProject](#)

Das `AuthenticDesktopControlPlaceHolder` Control dient dazu, die zusätzlichen AuthenticDesktop Fenster, wie Übersicht, Bibliothek oder das Projektfenster anzuzeigen. Es wird wie jedes andere ActiveX Control verwendet und kann überall in die Client-Applikation platziert werden.

Eigenschaften

Es sind die folgenden Eigenschaften definiert:

[PlaceholderWindowID](#)

Zugriff auf die AuthenticDesktopAPI:

[Project](#)

Label

Eigenschaft: `Label` als `String` (schreibgeschützt)

Dispatch Id: 1001

Beschreibung:

Mit dieser Eigenschaft erhalten Sie Zugriff auf den Titel des Platzhalters. Diese Eigenschaft ist schreibgeschützt.

PlaceholderWindowID

Eigenschaft: `PlaceholderWindowID` als `XMLSpyControlPlaceholderWindow`

Dispatch Id: 1

Beschreibung:

Mit Hilfe dieser Eigenschaft weiß das Objekt, welches AuthenticDesktop Fenster im Client-Bereich des Control angezeigt werden soll. Die `PlaceholderWindowID` kann jederzeit auf jeden gültigen Wert der `AuthenticDesktopControlPlaceholderWindow` gesetzt werden. Das Control ändert seinen Status sofort und zeigt das neue AuthenticDesktop Fenster an.

Project

Eigenschaft: `Project` als `Project` (schreibgeschützt)

Dispatch Id: 2

Beschreibung:

Mit der Eigenschaft `Project` erhalten Sie Zugriff auf das `Project` Objekt der AuthenticDesktop Automation Server API. Diese Schnittstelle bietet zusätzliche Funktionalitäten, die mit dem in das Control geladenen Projekt verwendet werden können. Die Eigenschaft gibt nur dann eine gültige Projektschnittstelle zurück, wenn das Platzhalterfenster [PlaceholderWindowID](#) mit einem Wert von `AuthenticDesktopXProjectWindow` (=3) hat. Die Eigenschaft ist schreibgeschützt.

Methoden

Es sind die folgenden Methoden definiert:

[OpenProject](#)

[CloseProject](#)

OpenProject

Methode: `OpenProject` (`strFileName` als `string`) als `boolean`

Dispatch Id: 3

Beschreibung:

`OpenProject` lädt die Datei `strFileName` als das neue Projekt in das Control. Wenn das Platzhalterfenster eine [PlaceholderWindowID](#) hat, die nicht mit `XMLSpyXProjectWindow` (=3) übereinstimmt, schlägt die Methode fehl.

CloseProject

Methode: `CloseProject` ()

Dispatch Id: 4

Beschreibung:

`CloseProject` schließt das in das Control geladene Projekt. Wenn das Platzhalterfenster eine [PlaceholderWindowID](#) hat, die nicht mit `AuthenticDesktopXProjectWindow` (=3) übereinstimmt, schlägt die Methode fehl.

Events

Das `AuthenticDesktopControlPlaceholder` ActiveX Control stellt die folgenden Verbindungspunkt-Events zur Verfügung:

[OnModifiedFlagChanged](#)

OnModifiedFlagChanged

Event: OnModifiedFlagChanged (i_bIsModified als [boolean](#))

Dispatch Id: 1

Beschreibung:

Dieses Event wird nur bei Platzhalter-Controls mit einer [PlaceholderWindowID](#) von AuthenticDesktopXProjectWindow (=3) ausgelöst. Das Event wird immer dann ausgelöst, wenn der Projektinhalt sich zwischen dem Status "geändert" und "nicht geändert" ändert. Der Parameter *i_bIsModified* ist *true*, wenn sich der Projektinhalt vom ursprünglichen Inhalt unterscheidet und *false*, wenn dies nicht der Fall ist.

OnSetLabel

Event: OnSetLabel (i_strLabel als [String](#))

Dispatch Id: 1000

Beschreibung:

Dieses Event wird immer dann ausgelöst, wenn die Bezeichnung eines Platzhalter-Control-Fensters geändert wird.

6.5.6 Enumerationen

Es sind die folgenden Enumerationen definiert:

[ICActiveXIntegrationLevel](#)

AuthenticDesktopControlPlaceholderWindow

ICActiveXIntegrationLevel

Mögliche Werte für die Eigenschaft [IntegrationLevel](#) für das AuthenticDesktopControl.

ICActiveXIntegrationOnApplicationLevel	= 0
ICActiveXIntegrationOnDocumentLevel	= 1

AuthenticDesktopControlPlaceholderWindow

Enter topic text here.

Kapitel 4

Anhänge

Anhänge

Diese Anhänge enthalten technische Informationen über Authentic Desktop und wichtige Lizenzinformationen. Die einzelnen Anhänge sind in die folgenden Unterabschnitte gegliedert:

Technische Daten

- Betriebssystem und erforderlicher Arbeitsspeicher
- Altova XML Parser
- Altova XSLT- und XQuery-Prozessor
- Unicode-Unterstützung
- Internet-Verwendung

Lizenzinformationen

- Electronic Software Distribution (EDI)
- Copyright
- Endbenutzer-Lizenzvereinbarung (EULA)

1 Technische Daten

Dieses Kapitel enthält nützliche Hintergrundinformationen zu den technischen Aspekten Ihrer Software. Es ist in die folgenden Abschnitte gegliedert:

- [OS- und Arbeitsspeichieranforderungen](#)
- [Altova XML Parser](#)
- [Altova XSLT- und XQuery-Prozessor](#)
- [Unicode-Unterstützung](#)
- [Internet-Verwendung](#)

1.1 OS- und Arbeitsspeichieranforderungen

Betriebssystem

Diese Software-Anwendung ist eine 32-Bit Windows-Applikation, die auf Windows 2000, Windows XP Windows Server 2003 und 2008 und Windows Vista läuft.

Arbeitsspeicher

Da die Software in C++ geschrieben wurde, wird dafür nicht so viel Platz wie in einer Java Runtime Umgebung benötigt und normalerweise wird dafür weniger Arbeitsspeicher als bei einer vergleichbaren Java-basierten Applikation benötigt.

Es ist notwendig, dass die einzelnen Dokumente in den Hauptarbeitsspeicher geladen werden, damit jedes Dokument zur Gänze geparkt und analysiert werden kann und die Anzeige- und Bearbeitungsgeschwindigkeit während der normalen Arbeit verbessert wird.

Auch die unbegrenzte Rückgängig-Funktion kann einiges an Arbeitsspeicher in Anspruch nehmen. Wenn Sie große Abschnitte in großen Dokumenten immer wieder ausschneiden und einfügen, kann dies enorm viel Speicherplatz verbrauchen.

1.2 Altova XML Parser

Wenn Sie ein XML-Dokument öffnen, verwendet die Applikation den integrierten Validierungsparser (den Altova XML Parser), um das Dokument auf Wohlgeformtheit zu prüfen, es gegen ein Schema zu validieren (falls eines angegeben wurde) und Baumstrukturen und Infosets zu erstellen. Der Altova XML Parser dient auch dazu, beim Editieren von Dokumenten intelligente Eingabehilfen zur Verfügung zu stellen und etwaige Validierungsfehler dynamisch anzuzeigen.

Im integrierten Altova XML Parser ist die Final Recommendation der W3C XML Schema Spezifikation implementiert. Neue Entwicklungen, die von der XML Schema-Arbeitsgruppe empfohlen werden, werden ständig in den Altova Parser integriert, sodass Ihnen mit Altova-Produkten immer eine Entwicklungsumgebung auf dem neuesten Stand der Technik zur Verfügung steht.

1.3 Altova XSLT- und XQuery-Prozessor

Die Altova Produkte arbeiten mit dem Altova XSLT 1.0-Prozessor, dem Altova XSLT 2.0-Prozessor und dem Altova XQuery-Prozessor. Nähere Informationen zu diesen Prozessoren finden Sie im Anhang 1 zu dieser Dokumentation im Abschnitt über Prozessoren.

Diese drei Prozessoren stehen auch als eigenständige Applikationen in Form des AltovaXML-Pakets zum kostenlosen Download von der [Altova Website](#) zur Verfügung. Die Dokumentation zur Verwendung der Prozessoren wird mit dem AltovaXML-Paket mitgeliefert.

1.4 Unicode-Unterstützung

Unicode ist ein neuer 16-Bit-Zeichensatzstandard, der vom [Unicode-Konsortium](#) definiert wurde und der eine eindeutige Zahl für jedes Zeichen zur Verfügung stellt, und zwar:

- plattformunabhängig
- programmunabhängig
- unabhängig von der Sprache

Im Prinzip arbeiten Computer nur mit Zahlen. Sie speichern Buchstaben und andere Zeichen unter einer Zahl, die sie diesen zuweisen. Vor der Entwicklung von Unicode gab es hunderte von verschiedenen Codiersystemen für die Zuweisung dieser Zahlen. Es gab keine einzige Codierung, die Platz für genügend Zeichen bot: So werden z.B. in der Europäischen Union mehrere verschiedene Codierungen benötigt, um alle ihre Sprachen abzudecken. Selbst für eine einzige Sprache wie z.B. Englisch reichte eine einzelne Codierung nicht aus, um alle gebräuchlichen Buchstaben, Satzzeichen und technischen Symbole abzudecken.

Diese Codierungssysteme standen früher miteinander in Konflikt, d.h. zwei Codierungen verwendeten dieselbe Zahl für zwei verschiedene Zeichen oder verschiedene Zahlen für dasselbe Zeichen. Jeder Computer (v.a. Server) muss viele verschiedene Codierungen unterstützen. Wenn aber Daten zwischen verschiedenen Codierungen oder Plattformen ausgetauscht werden, besteht immer das Risiko, dass Fehler entstehen.

Unicode ist die Lösung für diese Probleme!

Unicode enthält eine eindeutige Zahl für jedes Zeichen, egal für welche Plattform, welches Programm oder welche Sprache. Der Unicode-Standard wurde von branchenführenden Unternehmen wie Apple, HP, IBM, JustSystem, Microsoft, Oracle, SAP, Sun, Base und vielen anderen übernommen.

Unicode ist für moderne Standards wie XML, Java, ECMAScript (JavaScript), LDAP, CORBA 3.0, WML, usw. erforderlich und ist das offizielle Verfahren zum Implementieren von ISO/IEC 10646. Unicode wird von vielen Betriebssystemen, allen modernen Browsern und vielen anderen Produkten unterstützt. Die Entwicklung des Unicode-Standards und die Verfügbarkeit von Tools, die diesen unterstützen, gehören zu den wichtigsten globalen Software Technologie-Trends in letzter Zeit.

Die Integration von Unicode in Client-Server oder Multi-Tier-Applikationen und Websites bietet beträchtliche Kostenvorteile im Vergleich zur Verwendung herkömmlicher Zeichensätze. Dank Unicode kann ein einziges Software-Produkt oder eine einzige Website für mehrere Plattformen, Sprachen und Länder konzipiert werden, ohne dass es neu entwickelt werden muss. Daten können dadurch durch viele verschiedene Systeme übertragen werden, ohne dass dabei Fehler auftreten.

1.4.1 Windows 2000 und Windows XP

Die XML-Produkte von Altova bieten vollständige Unicode-Unterstützung. Um ein XML-Dokument von einem System mit nicht lateinischer Zeichensatzcodierung bearbeiten zu können, benötigen Sie allerdings auch eine Schriftart, die die von diesem Dokument verwendeten Unicode-Zeichen unterstützt.

Beachten Sie bitte, dass die meisten Schriftarten nur eine bestimmte Untergruppe des gesamten Unicode-Bereichs enthalten und normalerweise für das entsprechende Schriftsystem ausgelegt sind. Aus diesem Grund kann es vorkommen, dass ein XML-Dokument "nicht druckbare" Zeichen enthält, da die gewählte Schriftart die erforderlichen Glyphen nicht enthält. Es ist daher manchmal nützlich, wenn man eine Schriftart hat, die den gesamten Unicode-Bereich abdeckt - v.a. wenn Sie XML-Dokumente aus der ganzen Welt editieren.

Die größte uns bekannte Universalschriftart ist "Arial Unicode MS", eine Schriftart, die von Agfa Monotype für Microsoft entwickelt wurde. Diese Schriftart enthält mehr als 50.000 Glyphen und deckt das gesamte im Unicode 2.1.-Standard definierte Zeichenspektrum ab. Die Schriftart ist in Microsoft Office 2000 enthalten und beansprucht 23 MB.

Wir empfehlen Ihnen dringend, diese Schriftart auf Ihrem System zu installieren und mit der Applikation zu verwenden, wenn Sie öfter Dokumente bearbeiten, die in unterschiedlichen Schriftsystemen verfasst wurden. Diese Schriftart wird bei Verwendung der Standardinstallation von Microsoft Office nicht installiert, Sie können jedoch die Option "Benutzerdefinierte Installation" wählen, um sie zu installieren.

Im Ordner "Examples" finden Sie außerdem eine neue XHTML-Datei mit dem Namen `Unicode-UTF8.html`, die den Satz "Wenn die Welt miteinander spricht, spricht sie Unicode" in vielen verschiedenen Sprachen und Schriftsystemen enthält ("When the world wants to talk, it speaks Unicode") (世界的に話すなら、Unicode です)。 - Diese Zeile stammt aus der 10. Unicode-Konferenz von 1997 und ist ein schönes Beispiel für die Bedeutung von Unicode für den XML-Standard. Wenn Sie diese Datei öffnen, erhalten Sie einen kurzen Eindruck davon, was mit Unicode möglich ist und welche Schriftsysteme von den auf Ihrem PC verfügbaren Schriftarten unterstützt werden.

1.4.2 Rechts nach Links-Schreibrichtung

Beachten Sie bitte, dass selbst unter Windows NT 4.0 Texte aus Schriftsystemen mit einer Schreibrichtung von rechts nach links (z.B. Hebräisch oder Arabisch) nur in den Ländern korrekt wiedergegeben werden, in denen solche Schriftsysteme tatsächlich verwendet werden. Dies liegt daran, dass nur die hebräische und arabische Version von Windows NT die Wiedergabe und Bearbeitung von Text in Rechts-links-Schreibweise auf Betriebssystemebene unterstützen.

1.5 Internet-Verwendung

Altova-Applikationen können für Sie auch eine Verbindung mit dem Internet herstellen. Dies geschieht in den folgenden Fällen:

- Wenn Sie im Registrierungsdialogfeld (**Hilfe | Software-Aktivierung**) auf "Kostenlosen Evaluierungs-Key anfordern" klicken, werden die drei Felder im Registrierungsdialogfeld über eine normale HTTP-Verbindung (Port 80) an unseren Webserver übertragen. Sie erhalten dann per E-Mail (normales SMTP) den kostenlosen Evaluierungs-Keycode zugesandt.
- Falls Sie den URL-Modus des Dialogfelds "Öffnen" verwenden, um ein Dokument direkt über eine URL zu öffnen (**Datei | Öffnen | Zu URL wechseln**), wird dieses Dokument über eine HTTP-Verbindung (Port 80) aufgerufen. (*Diese Funktionalität steht in XMLSpy und Authentic Desktop zur Verfügung.*)
- Wenn Sie ein XML-Dokument öffnen, das sich auf ein XML-Schema oder eine DTD bezieht und das Dokument durch eine URL definiert wird, wird es ebenfalls über eine HTTP-Verbindung (Port 80) aufgerufen, sobald Sie das XML-Dokument validieren. Dies kann auch automatisch beim Öffnen eines Dokuments vorkommen, wenn Sie die Applikation so konfiguriert haben, dass Dateien beim Öffnen automatisch validiert werden (Register "Datei" des Dialogfelds **Extras | Optionen**).
- Wenn Sie den Befehl "Als Mail senden..." (**Datei | Als Mail senden**) verwenden, wird die aktuelle Auswahl bzw. Datei über ein MAPI-kompatibles Mail-Programm, das auf dem PC des Benutzers installiert ist, versendet
- Im Rahmen der in diesem Handbuch und der Altova-Software-Lizenzvereinbarung beschriebenen Software-Aktivierung und beim Live-Update..

2 Lizenzinformationen

Dieser Abschnitt enthält die folgenden Informationen:

- Informationen über den [Vertrieb dieses Software-Produkts](#)
- [Copyright](#)-Informationen im Zusammenhang mit diesem Software-Produkt
- Die [Endbenutzer-Lizenzvereinbarung \(EULA\)](#) zu diesem Software-Produkt

Lesen Sie die Informationen bitte sorgfältig - sie sind rechtlich bindend, da Sie sich bei der Installation dieses Software-Produkts damit einverstanden erklärt haben.

2.1 Electronic Software Distribution

Dieses Produkt ist über EDS (Electronic Software Distribution), also auf elektronischem Weg erhältlich, eine Methode, die die folgenden einzigartigen Vorteile bietet:

- Sie können die Software kostenlos testen, bevor Sie sich zu einem Kauf entscheiden.
- Wenn Sie sich entschieden haben, die Software zu kaufen, können Sie Ihre Bestellung online auf der [Altova Website](#) abgeben und das Produkt steht Ihnen in Minutenschnelle im vollen Lizenzumfang zur Verfügung.
- Sie erhalten immer die neueste Version unserer Software
- Die Software enthält ein umfassendes Online-Hilfesystem. Die neueste Version des Benutzerhandbuchs steht auf unserer Website unter http://www.altova.com/support_help.html (i) im HTML-Format zum Aufrufen online und (ii) im PDF-Format zum Download und Ausdrucken zur Verfügung.

30-Tage-Evaluierungszeitraum

Nachdem Sie dieses Software-Produkt heruntergeladen haben, können Sie es 30 Tage lang kostenlos testen. Während dieses Zeitraums werden Sie nach etwa 20 Tagen in regelmäßigen Abständen daran erinnert, dass die Software noch nicht lizenziert wurde. Diese Erinnerungsmeldung wird allerdings nur einmal, nämlich bei jedem Start des Programms, angezeigt. Wenn Sie das Programm nach Ablauf des 30-tägigen Evaluierungszeitraums weiterverwenden möchten, müssen Sie eine [Endbenutzer-Lizenz](#) erwerben, die Sie in Form eines Keycodes erhalten, der in das Dialogfeld "Software-Aktivierung" eingegeben werden muss, um das Produkt zu aktivieren. Sie können Ihre Lizenz online in unserem Web-Shop unter [Altova Shop](#) erwerben.

Weitergabe der Software an andere Mitarbeiter in Ihrem Unternehmen zu Testzwecken

Wenn Sie die Evaluierungsversion der Software auch anderen Personen in Ihrem Unternehmen über das Netzwerk zur Verfügung stellen möchten oder wenn Sie sie auf einem PC installieren möchten, der nicht mit dem Internet verbunden ist, sollten Sie nur das Installationsprogramm weitergeben - vorausgesetzt es wurde nicht modifiziert. Jeder, der das von Ihnen zur Verfügung gestellte Installationsprogramm aufruft, muss einen eigenen Evaluierungs-Keycode für 30 Tage anfordern. Nach Ablauf des Testzeitraums, muss eine Lizenz erworben werden, damit das Produkt weiter verwendet werden kann.

Nähere Informationen finden Sie in der [Endbenutzer-Lizenzvereinbarung](#) am Ende dieses Handbuchs.

2.2 Rechte am geistigen Eigentum

Die Software und alle Kopien, deren Anfertigung Ihnen von Altova gestattet ist, sind das geistige Eigentum von Altova und dessen Lieferanten und stehen in deren Besitz. Struktur, Organisation und Code der Software stellen wertvolle Betriebsgeheimnisse und vertrauliche Informationen von Altova und dessen Lieferanten dar. Die Software ist durch gesetzliche Bestimmungen urheberrechtlich geschützt. Diese gesetzlichen Bestimmungen beinhalten (ohne darauf beschränkt zu sein) das Urheberrecht der Vereinigten Staaten, internationale Verträge und das in den Ländern, in denen die Software genutzt wird, geltende Recht. Altova behält sich das Eigentum an allen Patenten, Urheberrechten, Branchengeheimnissen, Warenzeichen und sonstigen geistigen Eigentumsrechten, die hinsichtlich der Software bestehen, vorbehält. Das Eigentumsrecht von Altova erstreckt sich auch auf alle Bilder, Fotografien, Animationen, Videos, Audioaufzeichnungen, Musikstücke, Texte und „Applets“, die Teil der Software sind, und alle dazugehörigen Unterlagen in gedruckter Form. Mitteilungen über geltend gemachte Verstöße gegen geistige Eigentumsrechte sind an den Copyright Agenten von Altova zu richten (nähere Angaben dazu finden Sie auf der Website von Altova).

Altova Software enthält Software-Komponenten Dritter, die ebenfalls urheberrechtlich geschützt sind. Unter anderem sind diese Urheberrechte ausführlich unter http://www.altova.com/legal_3rdparty.html aufgelistet.

Alle anderen Namen oder Warenzeichen sind das Eigentum ihrer jeweiligen Inhaber.

2.3 Endbenutzer-Lizenzvereinbarung

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
Rudolfplatz 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 Software License Agreement. Altova may use your internal network for license metering between installed versions of the Software. **Altova's Software may use your internal network and Internet connection for the purpose of transmitting license-related data at the time of installation, registration, use, 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 Software and to improve customer service. 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.** Altova provides a new LiveUpdate notification service to you, which is free of charge. 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 Software License Agreement. By your acceptance of the terms of this Software License Agreement or use of the Software, you authorize the collection, use and disclosure of information collected by Altova for the purposes provided for in this Software License Agreement 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 Software License Agreement 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, 10 and 11 survive termination of this agreement as applicable.

10. THIRD PARTY SOFTWARE

The Authentic Software may contain third party software which requires notices and/or additional terms and conditions. Such required third party software notices and/or additional terms and conditions are located at our Website at http://www.altova.com/legal_3rdparty.html and are made a part of and incorporated by reference into this Agreement. By accepting this Agreement, you are also accepting the additional terms and conditions, if any, set forth therein.

11. 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 2006-09-12

Index

■

.NET,

- Integration von Authentic Desktop, 99
- Unterschiede zu Authentic Desktop Standalone, 101

A

Abfragen,

- für DB-Anzeige in der Authentic-Ansicht, 67

ActiveX Controls,

- Unterstützung, 333

Aktive Konfiguration,

- für globale Ressourcen, 234

Alias,

- siehe Globale Ressourcen, 84

Alles markieren,

- Befehl, 131

Altova Globale Ressourcen,

- siehe Globale Ressourcen, 84

Altova Support, 22

Altova XML Parser,

- Info, 777

Altova-Produkte, 22

Altova-Prozessoren,

- in Altova-Produkten, 778

Änderungen in der Version,

- XMLSpy API, 281

Änderungen verfolgen, 250

Angedockte Fenster, 8

Angedocktes Fenster, 261, 262

Anhängen, 183

- Zeile (in Authentic-Ansicht), 209

Anmerkungen zur Programmierung,

- in der Skripting-Umgebung, 330

Anmerkungen zur Release,

- XMLSpy API, 281

Anpassen,

- Kontextmenü, 242
- Kontextmenü 'Anpassen', 247

Makros, 244

Menü, 242

Symbolleiste/Menü Befehle, 235

Anpassung, 20

Ansicht,

Browser-Ansicht, 215

Menü, 211

Ansicht wechseln,

zur Authentic-Ansicht, 42

Anzeigen, 261, 262, 263, 266

API,

aufrufen, 752

Dokumentation, 350

JAVA, 653

JAVA Classpath, 653

Übersicht, 351

Application,

ActiveDocument, 383

AddMacroMenuItem, 383

Application, 383

ClearMacroMenu, 384

CurrentProject, 384

Dialogs, 384

Documents, 384

GetDatabaseImportElementList, 385

GetDatabaseSettings, 386

GetDatabaseTables, 386

GetExportSettings, 387

GetTextImportElementList, 387

GetTextImportExportSettings, 388

ImportFromDatabase, 388

ImportFromSchema, 389

ImportFromText, 390

ImportFromWord, 390

NewProject, 391

OnBeforeOpenDocument, 381

OnBeforeOpenProject, 381

OnDocumentOpened, 382

OnProjectOpened, 382

OpenProject, 391

Parent, 392

Quit, 392

ReloadSettings, 392

RunMacro, 393

ScriptingEnvironment, 393

ShowApplication, 393

ShowForm, 394

URLDelete, 394

Application,

- URLMakeDirectory, 395
- WarningNumber, 395
- WarningText, 395

Applikationsebene,

- Integration von AuthenticDesktop, 723

Arabisch, 781**Arbeitsspeicher-Anforderungen, 776****Arial Unicode MS, 780****ATL,**

- Plug-In Beispieldateien, 337

Attribut-Eingabehilfe,

- in der Authentic-Ansicht, 48

Attribut-Vorschau, 252**Attributwerte,**

- in der Authentic-Ansicht eingeben, 36

Ausgabefenster,

- Anzeige ein-/aus, 264

Ausgabeformatierung, 250**Authentic,**

- CDATA-Abschnitte, 33
- Daten eingeben, 33
- Daten in einer XML-Datenbank bearbeiten, 205
- Dateneingabe-Steuerelemente, 33
- DB-Daten bearbeiten, 203
- Eingabehilfen, 27
- Elemente anwenden, 30
- Elemente löschen, 30
- Entities, 33
- Kontextmenü, 27
- Markup-Tags anzeigen, 27
- Menü, 201
- Nodes einfügen, 30
- Nodes entfernen, 30
- Nodes hinzufügen, 30
- Sonderzeichen, 33
- Tabellen, 30

Authentic Integrationspaket, 103**Authentic Desktop,**

- Features, 22
- Hilfe, 22
- user manual, 3

Authentic Desktop Befehlstabelle, 733**Authentic Integrationspaket, 100****Authentic Perspektive in Eclipse, 107****Authentic Plugin für Eclipse,**

- installieren, 103

Authentic Plugin für VS .NET,

- installieren, 100

Authentic-Ansicht,

- Attributwerte eingeben, 36
- Benutzeroberfläche, 39
- Dokumentanzeige, 45
- Eingabehilfen, 48
- Entities einfügen, 37
- Hauptfenster, 45
- Kontextmenüs, 52
- Markup-Anzeige, 45
- neue XML-Datei öffnen, 202
- SPS-Tabellen, 59
- Symbolleisten-schaltflächen, 42
- Tabellen (SPS und XML), 58
- Text formatieren, 42
- Tutorial, 23
- Übersicht über die GUI, 40
- Verwendung von XML-Tabellen, 61
- Verwendung wichtiger Features, 54
- Voraussetzungen für das Tutorial, 23
- Vorlage, 25
- wechseln zur, 42, 214
- XML-Dokument drucken, 38
- XML-Dokument öffnen, 25
- XML-Tabellen, 61
- XML-Tabellensymbole, 74

AuthenticDataTransfer,

- dropEffect, 397
- getData, 398
- ownDrag, 398
- type, 398

AuthenticDesktop,

- Integration, 722

AuthenticDesktop API,

- aufrufen, 752

AuthenticDesktop Integration,

- Beispiel, 724

AuthenticDesktopCommand,

- in AuthenticDesktopControl, 754

AuthenticDesktopCommands,

- in AuthenticDesktopControl, 756

AuthenticDesktopControl, 757

- Beispiel für die Integration auf Applikationsebene, 724
- Beispiel für Integration auf Applikationsebene, 724
- Beispiele für die Integration auf Dokumentenebene, 730
- Dokumentation, 722
- Integration auf Applikationsebene, 723
- Integration auf Dokumentenebene, 725, 726, 727, 728, 729

AuthenticDesktopControl, 757

Integration mittels C#, 730
Integration mittels HTML, 730
integration using C#, 730
object reference, 753

AuthenticDesktopControlDocument, 763**AuthenticDesktopControlPlaceHolder, 769****Authentic-Menü,**

Anzeige von Markup-Symbolen, 42
Bearbeiten dynamischer Tabellen, 42

AuthenticRange,

AppendRow, 400
Application, 401
CanPerformAction, 401
CanPerformActionWith, 402
Close, 402
CollapsToBegin, 402
CollapsToEnd, 403
Copy, 403
Cut, 403
Delete, 404
DeleteRow, 404
DuplicateRow, 404
ExpandTo, 405
FirstTextPosition, 405
FirstXMLData, 406
FirstXMLDataOffset, 407
GetElementAttributeNames, 408
GetElementAttributeValue, 408
GetElementHierarchy, 409
GetEntityNames, 409
Goto, 410
GotoNext, 410
GotoNextCursorPosition, 411
GotoPrevious, 411
GotoPreviousCursorPosition, 412
HasElementAttribute, 412
InsertEntity, 412
InsertRow, 413
IsCopyEnabled, 413
IsCutEnabled, 414
IsDeleteEnabled, 414
IsEmpty, 414
IsEqual, 415
IsFirstRow, 415
IsInDynamicTable, 415
IsLastRow, 415
IsPasteEnabled, 416

IsTextStateApplied, 416
LastTextPosition, 416
LastXMLData, 417
LastXMLDataOffset, 418
MoveBegin, 419
MoveEnd, 419
MoveRowDown, 420
MoveRowUp, 420
Parent, 420
Paste, 421
PerformAction, 421
Select, 422
SelectNext, 422
SelectPrevious, 423
SetElementAttributeValue, 424
SetFromRange, 425
Text, 425

AuthenticView, 439

Application, 434
AsXMLString, 434
DocumentBegin, 435
DocumentEnd, 435
Event, 435
Goto, 436
IsRedoEnabled, 436
IsUndoEnabled, 437
MarkupVisibility, 437
OnBeforeCopy, 427
OnBeforeCut, 427
OnBeforeDelete, 428
OnBeforeDrop, 428
OnBeforePaste, 429
OnDragOver, 430
OnKeyboardEvent, 431
OnMouseEvent, 432
OnSelectionChanged, 433
Parent, 437
Print, 437
Redo, 438
Selection, 438
Undo, 438
WholeDocument, 439
XMLDataRoot, 440

Automatische Validierung, 251**Automatische Validierung ausschalten, 251**

B

Bearbeiten,

- Makro Schaltfläche, 247
- Menü, 128

Bearbeitet mit XMLSPY, 250

Befehl,

- hinzufügen zu Symbolleiste/Menü, 235
- Kontextmenü, 242
- löschen aus Menü, 242
- Menü zurücksetzen, 242

Befehle,

- Auflistung in Tastaturzuordnung, 272

Beispiele,

- Pfad installierter Dateien, 21

Benutzerdefiniert,

- Wörterbuch, 224

Benutzerhandbuch Authentic Desktop, 6

Benutzeroberfläche,

- Beschreibung, 8
- mittels Plug-In konfigurieren, 334

Beschreibung der Benutzeroberfläche, 8

Bestellen der Altova-Software, 273

Betriebssysteme für Altova-Produkte, 776

Big-endian, 253

Browser, 252

- Ansicht, 215
- Menü, 216

Browser-Ansicht,

- Beschreibung, 83
- Inhalt aktualisieren, 220
- Laden einer Seite abbrechen, 219
- neues Fenster, 222
- Schriftgröße, 221
- weiter, 218
- zurück, 217

C

C#,

- integration of AuthenticDesktop, 730
- Integration von AuthenticDesktop, 730

CDATA-Abschnitte,

- in der Authentic-Ansicht einfügen, 55

Child anhängen,

- in Grid-Ansicht, 186

Class ID,

- in AuthenticDesktop Integration, 724

ClassPath Anweisung, 653

CodeGeneratorDlg,

- Application, 441
- CPPSettings_DOMType, 442
- CPPSettings_LibraryType, 443
- CPPSettings_UseMFC, 443
- CSharpSettings_ProjectType, 443
- OutputPath, 444
- OutputPathDialogAction, 444
- OutputResultDialogAction, 444
- Parent, 445
- ProgrammingLanguage, 445
- PropertySheetDialogAction, 445
- TemplateFileName, 445

COM-API,

- Dokumentation, 350

Copy command, 130

Copyright-Informationen, 783, 785

CR&LF, 250

CustomCatalog, 190

Cut command, 130

D

Darstellung, 252

DatabaseConnection,

- ADOConnection, 447
- AsAttributes, 448
- CreateMissingTables, 448
- CreateNew, 448
- DatabaseKind, 449
- ExcludeKeys, 449
- File, 449
- IncludeEmptyElements, 450
- NumberDateTimeFormat, 450
- ODBCConnection, 451
- SQLSelect, 451
- TextFieldLen, 451

Datei,

- als Mail senden, 124
- Druckoptionen, 125

Datei,

- Kodierung, 118
- Neu laden, 117
- Öffnen, 114
- Optionen beim Öffnen, 250
- Register, 250
- schließen, 119
- speichern, 120
- Standard-Encoding, 253

Datei (Menü), 112**Datei abrufen,**

- Versionskontrolle, 154

Dateiarten, 251**Dateien,**

- Zu Versionskontrolle hinzufügen, 158
- zuletzt verwendete, 127

Dateien speichern,

- Kodierung, 118

Dateierweiterungen,

- anpassen, 190

Datenbanken,

- in Authentic-Ansicht bearbeiten, 203
- siehe auch DB, 65

Datumswahl,

- in der Authentic-Ansicht verwenden, 77

DB,

- Abfragen erstellen, 67
- Abfragen in der Authentic-Ansicht, 65
- Anzeige in der Authentic-Ansicht filtern, 67
- Bearbeiten in der Authentic-Ansicht, 65
- in der Authentic-Ansicht bearbeiten, 72
- Navigieren in Tabellen in der Authentic-Ansicht, 66
- Parameters in DB-Abfragen, 67

Delete,

- Application.URLDelete, 394

Delete command, 130**Dialogs,**

- Application, 452
- CodeGeneratorDlg, 452
- DTDSchemaGeneratorDlg, 454
- FileSelectionDlg, 453
- GenerateSampleXMLDlg, 453
- Parent, 453
- SchemaDocumentationDlg, 453

directories,

- creating with Application.URLMakeDirectory, 395

DocEditEvent (obsolete),

- altKey (obsolete), 568

- altLeft (obsolete), 569
- button (obsolete), 570
- cancelBubble (obsolete), 571
- clientX (obsolete), 572
- clientY (obsolete), 573
- ctrlKey (obsolete), 574
- ctrlLeft (obsolete), 575
- dataTransfer (obsolete), 576
- fromElement (obsolete), 577
- keyCode (obsolete), 577
- propertyName (obsolete), 578
- repeat (obsolete), 578
- returnValue (obsolete), 578
- shiftKey (obsolete), 579
- shiftLeft (obsolete), 580
- srcElement (obsolete), 581
- type (obsolete), 582

DocEditView (obsolete),

- ApplyTextState (obsolete), 590
- CurrentSelection (obsolete), 591
- EditClear (obsolete), 591
- EditCopy (obsolete), 592
- EditCut (obsolete), 592
- EditPaste (obsolete), 593
- EditRedo (obsolete), 593
- EditSelectAll (obsolete), 594
- EditUndo (obsolete), 594
- event (obsolete), 595
- GetAllowedElements (obsolete), 595
- GetNextVisible (obsolete), 598
- GetPreviousVisible (obsolete), 599
- IsEditClearEnabled (obsolete), 600
- IsEditCopyEnabled (obsolete), 600
- IsEditCutEnabled (obsolete), 601
- IsEditPasteEnabled (obsolete), 601
- IsEditRedoEnabled (obsolete), 602
- IsEditUndoEnabled (obsolete), 602
- IsRowAppendEnabled (obsolete), 603
- IsRowDeleteEnabled (obsolete), 603
- IsRowDuplicateEnabled (obsolete), 604
- IsRowInsertEnabled (obsolete), 604
- IsRowMoveDownEnabled (obsolete), 606
- IsRowMoveUpEnabled (obsolete), 606
- IsTextStateApplied (obsolete), 607
- IsTextStateEnabled (obsolete), 607
- LoadXML (obsolete), 608
- RowAppend (obsolete), 610
- RowDelete (obsolete), 610

DocEditView (obsolete),

- RowDuplicate (obsolete), 611
- RowInsert (obsolete), 611
- RowMoveDown (obsolete), 612
- RowMoveUp (obsolete), 612
- SaveXML (obsolete), 613
- SelectionMoveTabOrder (obsolete), 614
- SelectionSet (obsolete), 615
- XMLRoot (obsolete), 616

Document,

- Application, 460
- AssignDTD, 460
- AssignSchema, 461
- AssignXSL, 461
- AssignXSLFO, 461
- AuthenticView, 462
- Close, 463
- ConvertDTDOOrSchema, 463
- CreateChild, 464
- CreateSchemaDiagram, 464
- CurrentViewMode, 465
- DataRoot, 465
- DocEditView, 465
- Encoding, 465
- EndChanges, 466
- ExecuteXQuery, 466
- ExportToDatabase, 467
- ExportToText, 467
- FullName, 468
- GenerateDTDOOrSchema, 469
- GenerateProgramCode, 470
- GenerateSampleXML, 470
- GenerateSchemaDocumentation, 470
- GetExportElementList, 471
- GetPathName, 472
- GridView, 472
- IsModified, 472
- IsValid, 472
- IsWellFormed, 474
- Name, 474
- OnBeforeCloseDocument, 458
- OnBeforeSaveDocument, 457
- OnBeforeValidate, 459
- OnCloseDocument, 459
- OnViewActivation, 460
- Path, 475
- RootElement, 475
- Save, 475

- SaveAs, 476
- Saved, 476
- SaveInString, 476
- SaveToURL, 477
- SetActiveDocument, 477
- SetEncoding, 477
- SetExternalsValid, 479
- SetPathName, 479
- StartChanges, 479
- SwitchViewMode, 480
- Title, 480
- TransformXSL, 481
- TransformXSLFO, 481
- UpdateViews, 482
- UpdateXMLData, 482
- XQuery, 466

Documents,

- Count, 483
- Item, 483
- NewFile, 484
- NewFileFromText, 484
- OpenFile, 484
- OpenURL, 485
- OpenURLDialog, 485

Dokument,

- Rechtschreibprüfung, 224

Dokumente im Hauptfenster, 10**Dokumentebene,**

- Beispiele für die Integration von AuthenticDesktop, 730
- Integration des AuthenticDesktopControl, 725
- Integration von AuthenticDesktop, 726, 727, 728, 729

Draw (Menü),

- in der Skripting-Umgebung, 328

Drucken,

- von der Authentic-Ansicht, 38

Druckereinrichtung, 126**DTD, 250, 251****DTDSchemaGeneratorDlg,**

- Application, 487
- AttributeTypeDefinition, 487
- DTDSchemaFormat, 488
- FrequentElements, 488
- GlobalAttributes, 488
- MaxEnumLength, 488
- MergeAllEqualNamed, 489
- OnlyStringEnums, 489
- OutputPath, 489
- OutputPathDialogAction, 489

DTDSchemaGeneratorDlg,

- Parent, 490
- ResolveEntities, 490
- TypeDetection, 490
- ValueList, 490

Dynamische (SPS) Tabellen in der Authentic-Ansicht,

- Verwendung, 59

Dynamische Tabellen,

- bearbeiten, 42

E

Eclipse Plattform,

- und Authentic Desktop, 102
- und Authentic Integrationspaket, 103

Eclipse-Plattform,

- Authentic Perspektive, 107

Edit (Menü),

- in der Scripting-Umgebung, 322

Eigenschaftsblatt, 300**Einfügen,**

- in Grid-Ansicht, 180
- Zeile (in Authentic-Ansicht), 209

Eingabehilfe, 263, 266**Eingabehilfen, 15**

- aktualisieren, 192
- Anzeige ein/aus, 265

Eingabehilfen aktualisieren,

- Befehl, 192

Eingabemethoden, 780**Eingabetaste,**

- drücken, 82

Einstellungen, 20, 250

- Skripting, 255

Element-Eingabehilfe,

- in der Authentic-Ansicht, 48

ElementList,

- Count, 491
- Item, 491
- RemoveElement, 491

ElementListItem,

- ElementKind, 493
- FieldCount, 493
- Name, 493
- RecordCount, 493

E-Mail,

- Dateien senden, 124

Encoding, 118

- Standard, 253

Endbenutzer-Lizenzvereinbarung, 783, 786**Entities,**

- in der Authentic-Ansicht definieren, 55, 79
- in der Authentic-Ansicht einfügen, 55
- in der Authentic-Ansicht Entities einfügen, 37

Entities-Eingabehilfe,

- in der Authentic-Ansicht, 48

Enumerationen,

- in AuthenticDesktopControl, 772

Enumerations,

- SPYAttributeTypeDefinition, 619
- SPYAuthenticActions, 620
- SPYAuthenticDocumentPosition, 621
- SpyAuthenticElementActions, 622
- SPYAuthenticElementKind, 623
- SPYAuthenticMarkupVisibility, 624
- SPYDatabaseKind, 625
- SPYDialogAction, 626
- SPYDOMType, 627
- SPYDTDSchemaFormat, 628
- SPYEncodingByteOrder, 629
- SPYExportNamespace, 630
- SPYFrequentElements, 632
- SPYKeyEvent, 634
- SPYLibType, 635
- SPYLoading, 636
- SPYMouseEvent, 637
- SPYNumberDateTimeFormat, 638
- SPYProgrammingLanguage, 639
- SPYProjectItemTypes, 640
- SPYProjectType, 641
- SPYSampleXMLGenerationOptimization, 642
- SPYSampleXMLGenerationSchemaOrDTDAssignment, 643
- SPYSchemaDefKind, 644
- SPYSchemaDocumentationFormat, 645
- SPYTextDelimiters, 646
- SPYTextEnclosing, 647
- SPYTypeDetection, 648
- SPYURLTapes, 649
- SPYViewModes, 650
- SPYVirtualKeyMask, 651
- SPYXMLDataKind, 652

Ersetzen,

- Text, 132

Ersetzen,

Text in Dokument, 133

Evaluierungs-Keycode,

für Ihre Altova-Software, 273

Evaluierungszeitraum,

für Altova-Software-Produkte, 784

von Altova Software-Produkten, 783

Event, 381, 382, 427, 428, 429, 430, 431, 432, 433, 457, 458, 459, 460, 515, 516, 517**Event Handler,**

in der Skripting-Umgebung, 310

Events, 358**Explorer, 251****ExportSettings,**

CreateKeys, 495

ElementList, 495

EntitiesToText, 496

ExportAllElements, 496

FromAttributes, 496

FromSingleSubElements, 496

FromTextValues, 497

IndependentPrimaryKey, 497

Namespace, 497

SubLevelLimit, 497

Extern geparste Entitäten, 251**Externer XSL Prozessor, 254****Extras,**

Menü, 223

F

Favoriten, 269**Fenster,**

Anzeige verwalten, 8

automatisch ausblenden, 8

Eingabehilfe, 263, 266

frei schwebend, andocken, als Register, 8

horizontal nebeneinander, 259

Info, 262, 266

Menü, 257

öffnen, 267

Projekt, 261, 266

überlappend, 258

vertikal anordnen, 260

Fenster automatisch ausblenden, 8**File (Menü),**

in der Skripting-Umgebung, 321

FileSelectionDlg,

Application, 498

DialogAction, 498

FullName, 499

Parent, 499

Finden,

siehe Suchen, 133

Formulare,

Beispiel, 304

Design-Ansicht, in der Skripting-Umgebung, 295

Eigenschaften, in der Skripting-Umgebung, 300

Einstellen der Tabulatorreihenfolge von Objekten, 298

Farben in Formularen verwenden, 309

Handler für Formular-Events, in der Skripting-Umgebung, 300

in der Skripting-Umgebung, 294

Skript-Ansicht, in der Skripting-Umgebung, 295

TheView Objekt, 303

und Ebenen, 298

und Editor-Ansichten in der Skripting-Umgebung, 295

und Formularobjekte, 298

Formularobjekte,

Eigenschaften, 300

in Scripting Environment, 298

Formularobjektliste, 298**Frei schwebende Fenster, 8****Freigegeben,**

aus Versionskontrolle, 159

Funktionstasten, 237

G

Gemischte Markup-Symbole in (in Authentic)-Ansicht, 208**Generate Sample XML, 619, 642, 643****GenerateSampleXMLDlg,**

Application, 511

FillWithSampleData, 512

NonMandatoryAttributes, 512

NonMandatoryElements, 512

Parent, 511

RepeatCount, 512

TakeFirstChoice, 512

Global, 779**Globale,**

Globale,

Einstellungen, 250

Globale Ressourcen, 84

definieren, 85, 233

Konfigurationen ändern, 98

Konfigurationen kopieren, 93

Symbolleiste aktivieren, 236

Typ Datei definieren, 87

Typ Datenbank definieren, 91

Typ Ordner definieren, 90

verwenden, 94, 98

vom Typ Datei und Ordner verwenden, 95

Grafikformate,

in der Authentic-Ansicht, 81

Grammatik, 251**GridView,**

CurrentFocus, 517

Deselect, 518

IsVisible, 518

OnBeforeDrag, 515

OnBeforeDrop, 515

OnBeforeStartEditing, 516

OnEditingFinished, 516

OnFocusChanged, 517

Select, 518

SetFocus, 518

Große Markup-Symbole anzeigen, 42, 45**Großen Markup-Symbole in (in Authentic)-Ansicht, 208**

H

Hauptfenster, 10

in der Skripting-Umgebung, 289

Hebräisch, 781**Hilfe,**

Index, 270

Inhaltsverzeichnis, 269

Menü, 268

Suchen, 271

Tastaturzuordnungen, 272

Hilfesystem, 269**Hin- und Herschalten, 262, 263, 266****Hintergrundinformationen, 775****Hin-und Herschalten, 261****Hinzufügen,**

Dateien zu Projekt, 164

Ordner zu Projekt, 169

Hiragana, 780**HTML,**

Integration von AuthenticDesktop, 730

HTML-Beispiel,

für AuthenticDesktopControl Integration, 724

I

Index,

Hilfe, 270

Info-Fenster, 14, 262, 266**Inhaltsverzeichnis, 269****Installation,**

Pfad von Beispieldateien, 21

Integrating,

AuthenticDesktop in Applikationen, 722

Internet, 275, 276**Internet-Verwendung,**

in Altova-Produkten, 782

ISO/IEC 10646, 779

J

JAVA,

API, 653

ClassPath, 653

JRE,

für Authentic Plugin für Eclipse, 103

K

Kanji, 780**Kapitel, 269****Katakana, 780****Katalog,**

Oasis XML, 190

Keycodes,

für Ihre Altova-Software, 273

Klasse,

JAVA, 653

Kleine Markup-Symbole anzeigen, 42, 45**Kleine Markup-Symbole in (in Authentic)-Ansicht, 208**

Kodierung,

von Dateien, 118

Konfigurationen,

einer globalen Ressource, 85, 234

Konfigurationen in Globalen Ressourcen, 98**Konfigurieren,**

XMLSPY-Benutzeroberfläche, 334

Kontextmenü,

anpassen, 247

Befehle, 242

Kontextmenüs,

in der Authentic-Ansicht, 52

Kopieren,

Zeile (in Authentic-Ansicht), 209

Kürzel, 237

anzeigen in Tooltip, 246

zuweisen/löschen, 237

L**Layout (Menü),**

in der Skripting-Umgebung, 325

Leere Elemente, 251**Lesezeichen, 269****Links,**

in der Authentic-Ansicht verwenden, 55

Little-endian, 253**Lizenz, 786**

für Ihre Altova-Software, 273

Informationen, 783

loading, 485**Löschen,**

Befehl aus Symbolleiste, 235

Befehl im Kontextmenü, 242

Kürzel zu Befehl, 237

Symbol aus Symbolleiste, 235

Symbolleiste, 236

Zeile (in Authentic-Ansicht), 209

M**MainCatalog, 190****Makro,**

Hinzufügen zu Menü/Symbolleiste, 244

Schaltfläche bearbeiten, 247

Makros,

ändern, 319

ausführen, 317

Funktionen dafür erstellen, 315

in der Skripting-Umgebung erstellen, 314

Verwendung in der Skripting-Umgebung, 313

Makros anzeigen,

Skripting, 231

Markup,

in der Authentic-Ansicht, 42

Markup-Symbole,

in der Authentic-Ansicht, 45

Markup-Symbole (in Authentic)-Ansicht,

ausblenden, 208

kleine/große/gemischte einblenden, 208

Markup-Symbole ausblenden, 42, 45**Maximale Zellenbreite, 252****Meldungen (Fenster), 16****Menü,**

anpassen, 242

Ansicht, 211

Bearbeiten, 128

Befehl hinzufügen/löschen, 235

Befehle löschen, 242

Browser, 216

Extras, 223

Fenster, 257

Hilfe, 268

Makro hinzufügen, 244

Projekt, 134

Standard/XMLSPY, 242

Menüleiste, 17**MIME, 251****Mixed Markup-Symbole anzeigen, 42, 45****Module in der Skripting-Umgebung, 289****MSXML, 254****Multi-user, 250****N****Nach oben/nach unten,**

Zeile verschieben (Authentic-Ansicht), 210

Nebeneinander, 252

horizontal, 259

vertikal, 260

Neu laden, 250

geänderte Dateien, 117

Neue Datei,

erstellen, 113

Nicht lateinischen Schriftsysteme, 780**Nicht-XML Dateien, 251**

O

OASIS,

XML-Katalog, 190

Öffnen, 114**Optimale Breite, 252****Optionen beim Öffnen,**

Datei, 250

Ordner abrufen,

Versionskontrolle, 155

OrgChart.sps,

im Tutorial der Authentic-Ansicht, 23

OrgChart.xml,

im Tutorial der Authentic-Ansicht, 23

OrgChart.xsd,

im Tutorial der Authentic-Ansicht, 23

P

Parameter,

in DB-Abfragen, 67

Übergabe an Stylesheet über Schnittstelle, 197

Parent, 475**Parser,**

in Altova-Produkte integrierter, 777

XSLT, 254

Paste command, 130**Plattformen für Altova-Produkte, 776****Plug-in, 245**

ATL-Beispieldateien, 337

Konfiguration der Benutzeroberfläche, 334

Registrierung, 332

XMLSPY, 331

PRJ-Dateien, 286**Programmeinstellungen, 250****Programmers' Reference, 280****Project (Menü),**

in der Skripting-Umgebung, 324

Projekt,

aktive Dateien hinzufügen, 167

Dateien zu Projekt hinzufügen, 164

Eigenschaften, 176

externe Ordner hinzufügen, 170

externen Webordner hinzufügen, 173

Hinzufügen aktiver und verwandter Dateien, 168

Hinzufügen zu Versionskontrolle, 158

Menü, 134

Neu, 136

Neu laden, 138

öffnen, 137

Ordner hinzufügen, 169

schließen, 139

Skripting, 232

Speichern, 140

URL zu Projekt hinzufügen, 166

zuletzt verwendete, 178

Projekte,

globale Ressourcen hinzufügen, 165

Pfad installierter Dateien, 21

Übersicht, 134

Projektfenster, 12

Anzeige ein/aus, 265

in der Skripting-Umgebung, 286

Projekt-Fenster, 261, 266**PUBLIC,**

Identifizier - Katalog, 190

R

Rechtschreibprüfung, 224

benutzerdefiniertes Wörterbuch, 224

Rechtliches, 783**Rechtschreibprüfung,**

Optionen, 226

Referenz, 111**Register, 250****Registrierung,**

Plug-In, 332

Registrierung der Altova-Software, 273**Registry,**

Einstellungen, 250

Regular Expression,

in Suchstring, 132

Rückgängig, 129

S

save, 477

schema, 389

Einstellungen, 250

SchemaDocumentationDlg,

AllDetails, 520

Application, 520

IncludeAll, 521

IncludeAttributeGroups, 521

IncludeComplexTypes, 521

IncludeGlobalElements, 522

IncludeGroups, 522

IncludeIndex, 523

IncludeLocalElements, 523

IncludeRedefines, 524

IncludeSimpleTypes, 524

OptionsDialogAction, 525

OutputFile, 525

OutputFileDialogAction, 525

OutputFormat, 526

Parent, 526

ShowAnnotations, 526

ShowAttributes, 527

ShowChildren, 527

ShowConstraints, 528

ShowDiagram, 527

ShowEnumerations, 528

ShowNamespace, 528

ShowPatterns, 529

ShowProgressBar, 529

ShowProperties, 529

ShowResult, 529

ShowSingleFacets, 530

ShowSourceCode, 530

ShowType, 530

ShowUsedBy, 531

Schließen, 119

Projekt, 139

Schriftart, 780

Schriftgröße,

in Browser-Ansicht, 221

Schriftsystem, 780

Schriftsysteme, 781

Skripting,

Makros anzeigen, 231

Projekt, 232

Zu Skripting-Umgebung wechseln, 230

Skripting-Projekte, 286

Skripting-Umgebung, 283

Anmerkungen zur Programmierung, 330

Application Event Handler, 310

Einstellungen, 293

Event Handler, 291

Formulare, 291, 295

Formularebenen, 298

Formularobjekte, 298

Globale Deklarationen, 291

Hauptfenster, 289

Makros, 291, 313

Projektfenster, 286

Übersicht über die Verwendung, 285

Skripts, 255

Skriptsprache, 255

Software-Produktlizenz, 786

Source control, 256

Speichern,

Projekt, 140

Splashscreen, 252

SPP Datei-Speicherpfade, 134

SPS,

neuer XML-Datei zuweisen, 113

SPS-Datei,

zu XML-Datei zuweisen, 204

SPS-Tabellen,

dynamischen Tabellen bearbeiten, 42

SPS-Tabellen in der Authentic-Ansicht,

Verwendung, 59

SpyProject,

CloseProject, 532

ProjectFile, 532

RootItems, 532

SaveProject, 533

SaveProjectAs, 533

SpyProjectItem,

ChildItems, 534

FileExtensions, 534

ItemType, 534

Name, 535

Open, 535

ParentItem, 535

Path, 535

SpyProjectItem,

- ValidateWith, 535
- XMLForXSLTransformation, 536
- XSLForXMLTransformation, 536
- XSLTransformationFileExtension, 536
- XSLTransformationFolder, 536

SpyProjectItems,

- AddFile, 537
- AddFolder, 537
- AddURL, 537
- Count, 538
- Item, 538
- RemoveItem, 538

Standard,

- Encoding, 253
- Menü, 242

Standardansicht,

- Einstellung im Hauptfenster, 251

Standardmäßiger Editor, 251**Startgruppe,**

- hinzufügen (Kontextmenü), 247

Statische (SPS) Tabellen in der Authentic-Ansicht,

- Verwendung, 59

Statusleiste, 17**StyleVision,**

- zur Bearbeitung eines StyleVision Power Stylesheet, 204

StyleVision Power Stylesheet,

- in %STYLEVISION%> bearbeiten, 204
- zu XML-Datei zuweisen, 204

StyleVision Power Stylesheet zu XML-Datei zuweisen, 204**Suchen,**

- Hilfe, 271
- Text in Dokument, 132
- und Text in Dokument ersetzen, 133

Support Center, 275**Support-Optionen, 22****Symbol,**

- Hinzufügen zu Symbolleiste/Menü, 235

Symbole,

- große anzeigen, 246

Symbolleiste, 17

- Befehle anpassen, 235
- Ein/Aus, 236
- große Symbole anzeigen, 246
- Makro hinzufügen, 244
- Neue erstellen, 236
- Symbolleiste & Menübefehle zurücksetzen, 236

Syntaxfärbung, 251, 252

T

Tabelle,

- automatisch erzeugen, 251

Tabellen,

- dynamische (SPS)-Tabellen (SPS) bearbeiten, 42
- in der Authentic-Ansicht, 30

Tabellen in der Authentic-Ansicht,

- Arbeiten mit (statischen und dynamischen) SPS Tabellen, 59
- Arbeiten mit XML-Tabellen, 61
- Symbole zum Bearbeiten von XML-Tabellen, 74
- Verwendung, 58

Tastaturzuordnungen, 272**Tatstaturkürzel, 237****Technische Informationen, 775****Technischer Kundendienst, 275****terminate, 392****Text,**

- in der Authentic-Ansicht bearbeiten, 55
- in der Authentic-Ansicht formatieren, 55
- in Dokument suchen, 132
- suchen und ersetzen, 133

TextImportExportSettings,

- DestinationFolder, 539
- EnclosingCharacter, 539
- Encoding, 540
- EncodingByteOrder, 540
- FieldDelimiter, 540
- FileExtension, 540
- HeaderRow, 540
- ImportFile, 541

Themen,

- Ansicht im Inhaltsverzeichnis, 269

TheView, 303**Tooltip,**

- anzeigen, 246
- Kürzel anzeigen, 246

Transformation,

- siehe XSLT-Transformation, 195

Tutorials,

- Pfad installierter Dateien, 21

U

Überlappend,

Fenster, 258

Übernehmen, 250**Überprüfen,**

Rechtschreibung, 224

Übersicht,

von XMLSpy API, 351

UCS-2, 253, 779**UCS-4, 779****Unicode,**

Konsortium, 779

Standard, 779

Unterstützung, 780

Unicode-Zeichen, 780**Universalschriftart, 780****Urheberrecht, 277****URL, 394, 395, 477, 485**

per E-Mail senden, 124

User manual, 3**UTF-16, 253****UTF-8, 779**

V

Validator,

in Altova-Produkten, 777

Validieren, 190**Validierung, 20****Validierungsmeldungen, 16****Verbergen, 261, 262, 263, 266****Versionskontrolle,**

aktivieren /deaktivieren, 153

aktuellste Version holen, 154

auschecken, 156

Auschecken rückgängig, 157

ausgliedern, 159

Eigenschaften, 163

Einchecken, 156

Hinzufügen in Versionskontrolle, 158

Projekt öffnen, 152

Status aktualisieren, 163

Unterschiede anzeigen, 161

unterstützte Anbieter, 141

Verlauf anzeigen, 160

Versionskontrollmanager, 163**Versionsnummer, 277****Vertrieb,**

von Altova Software-Produkten, 783

von Altova-Software-Produkten, 784

View (Menü),

in der Skripting-Umgebung, 323

Visual Studio .Net,

und Authentic Desktop, 99

und Authentic DesktopUnterschiede, 101

Volltextsuche, 271**Von rechts nach links, 781****Vorlagen,**

von XML-Dokumenten in Authentic-Ansicht, 202

Vorlagendateien,

für neue Dokumente, 113

VS .NET,

und Authentic Integrationspaket, 100

W

Web Server, 275, 276**Wechseln,**

Zu Skript-Umgebung, 230

Wiedergabe, 781**Window (Menü),**

in der Skripting-Umgebung, 329

Windows,

Unterstützung für Altova-Produkte, 776

Windows 2000, 780**Windows NT, 780****Wohlgeformtheitsprüfung, 189****Wörterbuch,**

aktuelles ändern, 224

benutzerdefiniertes Wörterbuch hinzufügen, 224

Rechtschreibprüfung, 224

X

XBRL-Schriftarten, 252**XInclude, 180, 183, 186**

XInclude, 180, 183, 186

- appending in Grid View, 183
- in der Grid-Ansicht als Child anhängen, 186
- in die Grid-Ansicht einfügen, 180

XML, 779

- Menü, 179
- Oasis-Katalog, 190
- Rechtschreibprüfung, 224

XML Parser,

- Info, 777

xml:base,

- and XInclude, 180, 183, 186

XMLData,

- AppendChild, 558
- EraseAllChildren, 559
- EraseCurrentChild, 559
- GetChild, 560
- GetChildKind, 560
- GetCurrentChild, 561
- GetFirstChild, 561
- GetNextChild, 561
- HasChildren, 563
- HasChildrenKind, 562
- InsertChild, 563
- IsSameNode, 564
- Kind, 564
- MayHaveChildren, 564
- Name, 564
- Parent, 565
- TextValue, 565

XML-Datei Global Resources, 85**XML-Datenbank,**

- neue Datenzeile in die Authentic-Ansicht laden, 205

XML-DB,

- neue XML-Datenzeile laden, 66

XML-Dokument,

- in der Authentic-Ansicht öffnen, 25

XML-Konformität, 251**XMLSPY, 111, 276, 277**

- allgemeine Einführung, 7
- Plug-In Registrierung, 332

XMLSpy API,

- Anmerkungen zur Release, 281
- Dokumentation, 350
- Übersicht, 351

XMLSPY Plug-In, 331**XMLSpyDocumentEditor,**

- MarkUpView, 608

XMLSpyLib, 350

- Application, 380
- AuthenticDataTransfer, 397
- AuthenticRange, 399
- AuthenticSelection (obsolete), 584
- AuthenticView, 426
- CodeGeneratorDlg, 441
- DatabaseConnection, 447
- Dialogs, 452
- DocEditEvent (obsolete), 567
- DocEditView (obsolete), 588
- Document, 456
- Documents, 483
- DTDSchemaGeneratorDlg, 487
- ElementList, 491
- ElementListItem, 493
- ExportSettings, 495
- FileSelectionDlg, 498
- GenerateSampleXMLDlg, 511
- GridView, 515
- ProjectItem, 534
- SchemaDocumentationDlg, 519
- SpyProject, 532
- SpyProjectItems, 537
- TextImportExportSettings, 539
- XMLData, 557

XML-Tabellen in der Authentic-Ansicht,

- Editiersymbole, 74
- Verwendung, 61

XML-Vorlagendatei,

- in der Authentic-Ansicht, 25

XPath zum ausgewählten Node, 40**XPointers, 180, 183, 186****XQuery,**

- Variablen an XQuery-Dokument übergeben, 197

XQuery-Prozessor,

- in Altova-Produkten, 778

XSL/Query Menü, 193**XSL:FO,**

- Transformation, 195

XSLT,

- Prozessor, 254

XSLT-Parameter,

- über Schnittstelle an Stylesheet übergeben, 197

XSLT-Prozessoren,

- in Altova-Produkten, 778

XSLT-Transformation, 194

- in FO, 195

XSLT-Transformation, 194

in PDF, 195

Z

Zeichensatz,

Kodierung, 253

Zeile,

anhängen (in Authentic-Ansicht), 209
einfügen (in Authentic-Ansicht), 209
kopieren (in Authentic-Ansicht), 209
löschen (in Authentic-Ansicht), 209
nach oben/nach unten verschieben, 210

Zeilenschaltung,

siehe Eingabetaste, 82

Zeilenumbruch, 250**Zuletzt verwendetet Dateien,**

Liste, 127

Zurücksetzen,

Kürzel, 237
Menübefehle, 242
Symbolleiste & Menübefehle, 236

Zuweisen,

Kürzel zu Befehl, 237