ユーザーマニュアル



Unlimited XML Document Editing

Copyright@ 2018 Altova GmbH. All rights reserved. Use of this software is governed by an Altova license agreement. XMLSpy, MapForce, StyleVision, SchemaAgent, UModel, DatabaseSpy, DiffDog, Authentic, MissionKit, FlowForce, RaptorXML, MobileTogether, and Altova as well as their respective logos are either registered trademarks or trademarks of Altova GmbH. Protected by U.S. Patents 7,200,816 and other pending patents. This software contains third party software or material that is protected by copyright and subject to other terms and conditions as detailed on the Altova website at https://www.altova.com/legal/3rdparty.

Altova Authentic 2018 Desktop Enterprise Edition ユーザーマニュアル

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

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

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

発行日: 2018

(C) 2018 Altova GmbH

目次

1	Altov Editi	va Authentic 2018 Desktop Enterprise ion	3
2	ユー	ザーガイドとレファレンス	6
2.1	インタ	ーフェイスと環境	7
	2.1.1	グラフィカルユーザインターフェース (GUI)	
		- メインウインドウ	
		- プロジェクトウィンドウ	12
		- 情報ウインドウ	
		- 入力ヘルパー	
		- メニューバー、ツールバー、ステータスバー	
	2.1.2	アプリケーション環境	
2.2	Author	- Authentic Desktop 表面機能とAltova 表面	
2.2	Auther		
	2.2.1		
	2.2.2	Authentic View インターノエイス	
	2.2.3		
	2.2.4	Authentic View 内にナーダを挿入9る	
	2.2.5	馬性の値を入力する	
	2.2.6	エンティティを入力する	
2.2	2.2.7	ドキュメントを印刷9る	
2.3	Auther		
	2.3.1	GUIの概要	
	2.3.2	Authentic View ツールバーアイコン	41
	2.3.3	Authentic View メインウィンドウ	44
	2.3.4	Authentic View エントリヘルパー	47
	2.3.5	Authentic View コンテキストメニュー	51
2.4	Auther	ntic View 内で編集する	54
	2.4.1	基本的な編集	55
	2.4.2	Authentic View 内のテーブル	60
		- SPS テーブル	61
		- CALS/ HTML テーブル	
		- CALS/HTML テーブル編集アイコン	68

	2.4.3	DB の編集	71
		- DB テーブル内をナビゲー hする	72
		- DB クエリ	73
		- DB テーブルの変更	78
	2.4.4	日付と作業する	80
		- 日付の選択	81
		- テキス Юエンドリ	82
	2.4.5	エンティティを定義する	83
	2.4.6	XML 署名	85
	2.4.7	Authentic View 内のイメージ	87
	2.4.8	Authentic View 内のキーストロール	88
2.5	Authen	tic スクリプト	89
2.6	ブラウセ	ザービュー	91
2.7	Altova		92
	271	グローバル・リノースの定義	93
	2.,.1	- ファイル	96
		- フォルダー	101
		- データベース	103
	2.7.2	グローバルリソースを使用する	105
		- ファイルとフォルダーの割り当て	106
		- 構成の切り替え	109
2.8	Source	Control	110
	2.8.1	Setting Up Source Control	112
	2.8.2	Supported Source Control Systems	113
	2.8.3	Local Workspace Folder	115
	2.8.4	Application Project	116
	2.8.5	Add to Source Control	118
	2.8.6	Working with Source Control	120
	2.0.0	 Add to, Remove from Source Control	120
		- Check Out, Check In	122
		- Getting Files as Read-Only	124
		- Copying and Sharing from Source Control	126
		- Changing Source Control	129
	2.8.7	Source Control with Git	130
		- Enabling Git Source Control with GIT SCC Plug- in	131
		- Adding a Project to Git Source Control	132
		- Cloning a Project from Git Source Control	134
2.9	Visual S	Studio 内の Authentic Desktop	136
	2.9.1	Visual Studio のための Authentic Desktop プラグインのインストール	. 137
	2.9.2	スタンドアロンバージョンとの違い	138
2.10	Ec lips e	内の XMLSpy	139
	2.10.1	Eclipse のための Authentic Desktop プラグイン	140

	2.10.2	Eclipse 内の Authentic Desktop エントリポイント	
2.11	火ニュ-	-コマンド	
	2.11.1	ファイルメニュー	
		- 新規作成	153
		- 開く	
		- 再ロード	
		- エンコード	
		- 閉じる、全て閉じる、非アクティブを全て閉じる	
		- 保存、名前を付けて保存、全て保存	
		- メールで送信	
		- 印刷	171
		 印刷プレビュー、印刷設定	172
		- 最近使用されたファイル、終了	173
	2.11.2	編集メニュー	
		 一元に戻す、やり直し 	175
		- 検察、次を検察	
		- 直換	
	2.11.3		
		- プロジェクトの舟ロー [:	
		ノロノエノ 123 00	
		- ソーフ管理	
		シース管理から聞く	
		ソース管理を有効にする	
		して、「「」」	
		フォルダーの取得	
		チェックアウト チェックイン	
		チェックアウ Ю取り消し	193
		ソース管理へ追加	194
		ソース管理から削除	195
		ソース管理から共有	195
		履歴の表示	197
		差分の表示	
		プロパティの表示	199
		ステータスを最新の状態に更新	
		ソース管理マネジャー	200
		ソース管理の変更	200
		- プロジェクHこファイルを追加	202
		- プロジェクトにグローバルリソースを追加	
		- フロジェク Hこ URL を追加	

	- プロジェクHこアクティブなファイルを追加	205
	- プロジェクHこアクティブならびに関係するファイルを追加	206
	- プロジェクHこアクティブならびに関係するファイルを追加	207
	- プロジェクトに外部フォルダーを追加	208
	- プロジェクHこ外部ウェブフォルダーを追加	212
	- スクリプトの設定	217
	- プロパティ	218
	- 最近使用されたプロジェクト	221
2.11.4		222
	- 整形式のチェック	223
2.11.5	XMLの検証	225
2.11.6	XSL/ XQuery メニュー	227
	- XSL 変換	228
	- XSL-FO 変換	229
	- XSL パラメーター / XOuery 実行	231
2.11.7	Authentic メニュー	235
	- 新規 ドコメント	236
	- データベースの編集	237
	- Style Vision スタイルシー 1の編集	238
	- XML データの新たな行を選択し編集	239
	- XML 署名	240
	- XML エンティティの定義	242
	- マークアップの表示	244
	- 行の追加 /挿入 /複製 /削除	245
	- 上に 下に移動	246
	- HTML、RTF、PDF、Word 2007+ ドキュメントの生成	247
	- 信頼された場所	248
2.11.8	表示メニュー	249
	- Authentic View	250
	- ブラウザービュー	251
2.11.9	ブラウザーメニュー	252
	- 戻る	253
	- 進む	254
	- 中止	255
	- 最新の状態に更新	256
	- フォント	257
	- 別のウィンドウ	258
2.11.10	ツールメニュー	259
	- スペリング	260
	- スペルチェックのオプション	264
	- スクリプトエディター	268
	- マクロ	269
	- ユーザー定義ツール	270
	- グローバルリソース	271

- カスタマ	?イズ	
コマンド	•	
ツールノ	۲ ۴ –	
ツール		
キーボー	ード	
メニュー	-	
マクロ		
プラグイ	でン	
オプショ	シ	
コンテキ	Fス トメニューのカスタマイズ	
- ツールノ	バーとウィンドウの復元	
- オプショ	シ	
ファイル	,	
ファイル	・タイプ	
エンコー	- F	
ビュー		
XSL		
スクリプ	パト	
ソース管	管理	
ネットフ	リーク プロキシ	
2.11.11 ウィンドウ	ーニー	
- 重ねて著	表示	
- 上下に対	並べて表示	
- 左右に対	並べて表示	
- プロジェ	<u>c</u> クドウィンドウ	
- 情報ウィ	心や	
- 入力へJ	ルパー	
- 出力ウィ	心や	
- プロジェ	፫ク はよび入力ヘルパー	
- 全てオン	ン/オフ	
- 現在開加	かれているウィンドウ	
2.11.12 ヘルプメ	ニュ ー	
- 目次、1	インデックス、検索	
- キーボ-	ードマップ	
- ライセン	ノス登録、注文フォーム、登録、最新情報のチェック	
- 他のコマ	マンド	
2.11.13 コマンドラ	ライン	

プログラマーのレファレンス 3

3.1	Scriptin	g Editor	325
	3.1.1	Overview	327
		- Scripting Projects in Authentic Desktop	. 328

322

		- The Scripting Editor GUI	
		- Components of a Scripting Project	
	3.1.2	Creating a Scripting Project	
	3.1.3	Global Declarations	
	3.1.4	Forms	
		- Creating a New Form	
		- Form Design and Form Objects	
		- Form Events	
	3.1.5	Events	
	3.1.6	Macros	351
	51110	 Creating and Editing a Macro 	
		- Running a Macro	
		- Debugging a Macro	
	317	Programming Points	358
	5.1.7	- Built- in Commands	360
		Form usage and commands	367
	318	Migrating to Scripting Editor 2010 and Later	368
32	IDF P1	hugins	371
5.2			
	3.2.1	Registration of IDE Plugins	
	3.2.2	Active X Controls	
	3.2.3	Configuration XML	
	3.2.4	ATL sample files	
		- Interface description (IDL)	
		- Class definition	
		- Implementation	
	3.2.5	IXMLSpyPlugIn	
		- OnCommand	
		- On Update Command	
		- OnEvent	
		- GetUIModifications	
		- GetDescription	
3.3	Applica	ation API	
	3.3.1	Overview	
		- Object Model	
		- Programming Languages	
		JScript	
		Start Application	
		Simple Document Access	
		It eration	
		Error Handling Events	
		VBScript	
		Events	
		Example: Using Events	401

	C#	
	Add Reference to Authentic Desktop API	407
	Application Startup and Shutdown	407
	Opening Documents	
	Iterating through Open Documents	
	Errors and COM Output Parameters	
	Events	
	Java	
	Example Java Project	415
	Simple Document Access	
	Iterations	416
	Use of Out-Parameters	416
	Event Handlers	417
3.3.2	Interfaces	
	- Application	419
	Events	
	OnBefore OpenDocument	
	OnBefore OpenProject	
	OnDocument Opened	
	On Project Opened	
	Active Document	
	AddMacroMenuItem	
	AddXSLT_XQParameter	
	Application	
	ClearMacroMenu	
	CreateXMLSchemaFromDBStructure	
	CurrentProject	
	Dialogs	
	Documents	
	Edit ion	
	FindInFiles	
	Get Dat abase Import Element List	
	GetDatabaseSettings	
	Get Dat abase Tables	
	GetExportSettings	
	GetTextImportElementList	
	GetTextImportExportSettings	
	GetXSLT XQParameterCount	
	GetXSLT XOParameterName	
	GetXSLT XOParameterXPath	
	Import From Database	428
	Import FromSchema	429
	Import From Text	430
	Import FromWord	
	Inport 110 m word	
	BAT Bupportea	
	Iviajor version	

	Minor Version	431
	NewProject	431
	OpenProject	432
	Parent	432
	Quit	432
	ReloadSettings	433
	Remove XSLT_XQP arameter	433
	RunMacro	433
	ScriptingEnvironment	434
	ServicePackVersion	434
	ShowApplic at ion	434
	ShowFindInFiles	434
	ShowForm	435
	Status	435
	URLDelete	435
	URLMake Directory	436
	Vis ible	436
	WarningNumber	436
	WarningText	436
_	AuthenticContextMenu	438
	Count Items	438
	Deleteltem	438
	GetltemText	438
	InsertItem	438
	Set ItemText	439
_	Authentic DataTransfer	440
	dropEffect	440
	getData	440
	ownDrag	441
	type	441
-	AuthenticEventContext	442
	Evaluate XP at h	442
	GetEventContextType	442
	Get NormalizedText Value	442
	Get Variable Value	443
	Get XMLNode	443
	Is Available	443
	Set Variable Value	444
-	AuthenticRange	445
	AppendRow	446
	Application	447
	CanPerformAction	447
	CanPerformActionWith	447
	Clone	448
	Collaps ToBe gin	448

Collaps To End	. 448
Сору	. 448
Cut	. 449
Delete	. 449
DeleteRow	. 449
Duplicate Row	. 450
EvaluateXPath	. 450
ExpandTo	. 451
FirstTextPosition	. 451
FirstXMLData	. 452
FirstXMLDataOffset	. 452
GetElementAttributeNames	. 453
GetElementAttributeValue	. 454
GetElementHierarchy	. 454
GetEntityNames	. 455
Get Variable Value	. 455
Goto	. 455
GotoNext	. 456
GotoNextCursorPosition	. 456
GotoPrevious	. 457
GotoPreviousCursorPosition	. 457
Has Element Attribute	. 458
InsertEntity	. 458
InsertRow	. 458
Is CopyEnable d	. 459
Is Cut Enable d	. 459
Is De let e Enable d	. 459
Is Empt y	. 460
Is Equal	. 460
IsFirstRow	. 460
Is In Dynamic Table	. 460
IsLastRow	. 461
Is Paste Enabled	. 461
IsSelected	. 461
Is Text State Applied	. 461
Last Text Position	. 462
Last XMLDat a	. 462
Last XMLDataOffset	. 463
Move Begin	. 464
MoveEnd	. 464
Move RowDown	. 465
Move RowUp	. 465
Parent	. 465
Paste	. 466
PerformAction	. 466

	Select	467
	SelectNext	467
	SelectPrevious	468
	SetElementAttributeValue	468
	SetFromRange	469
	Set Variable Value	470
	Text	470
_	AnthentiaView	471
	Fuents	471
	On Before Com.	471
	On Before Cut	472
	On Before Delete	472
	On Before Drop	472
	On Before Paste	473
	OnBefore Save	473
	OnDragOver	474
	On KeyboardEvent	474
	OnLoui OnMouse Event	475
	OnSelectionChanged	476
	On Toolbar Button Clicked	476
	On Toolbar Button Executed	477
	On Us e rAdde dXMLNo de	478
	Application	478
	As XMLS tring	478
	Context Menu	478
	CreateXMLNode	479
	Disable Attribute EntryHelper	479
	Disable Element EntryHelper	479
	Disable EntityEntryHelper	479
	DocumentBegin	480
	Document End	480
	DoNotPerformStandardAction.	480
	Fyaluate XPath	480
	Fvent	481
	EventContext	181
	CatTaalharDuttanStata	401
	Cete	401
		482
		482
	Is Undo Enable d.	483
	Markup V1s ibilit y	483
	Parent	483
	Print	483
	Re do	484
	Selection	484
		40.4
	Set I oolbar Button State	484
	Undo	484 485

Update XMLInstance Entities	485
Whole Document	486
XMLDataRoot	486
- CodeGeneratorDlg	487
Application	487
CPPSettings_DOMType	488
CPPSettings_GenerateVC6ProjectFile	488
CPPSettings_GenerateGCCMakefile	488
CPPSettings_GenerateVSProjectFile	488
CPPSettings_LibraryType	489
CPPSettings_UseMFC	489
CSharpSettings_ProjectType	489
OutputPath	489
OutputPathDialogAction	490
Output Result DialogAction	490
Parent	490
ProgrammingLanguage	491
PropertySheetDialogAction	491
TemplateFileName	491
- DatabaseConnection	492
ADOConnection	492
As Attributes	493
Comment Included	493
CreateMissingTables	493
CreateNew	493
Dat abase Kind	494
Dat abase Schema	494
Exc lude Ke ys	494
File	494
ForeignKeys	495
Import Columns Type	495
Include EmptyElements	495
NullReplacement	495
NumberDateTimeFormat	496
ODBCConnection	496
PrimaryKeys	496
SchemaExtensionType	496
SchemaFormat	497
SQLSelect	497
TextFieldLen	497
Unique Ke ys	497
- Dialogs	498
Application	498
CodeGeneratorDlg	498
FileSelectionDlg	499

	Parent	499
	SchemaDocumentationDlg	499
	GenerateSampleXMLDlg	499
	DTDSchemaGeneratorDlg	500
	FindInFilesDlg	500
	WSDLDocumentationDlg	500
	WSDL20DocumentationDlg	500
	XBRLDocumentationDlg	501
_	Document	502
	Events	503
	OnBeforeSaveDocument	503
	OnBefore Close Document	504
	OnBefore Validat e	504
	On View Activation	505
	Application	506
	Assim DED	506
	AssignDID	506
	Assignschema.	500
	ASSIGNABL	500
	AssignXSLFO	507
	As XMLString	507
	Authentic View	507
	Close	508
	Convert DTDOr Schema	508
	Convert DTDOr Schema Convert DTDOr Schema Ex	508 509
	Convert DTDOrSchema Convert DTDOrSchemaEx Convert ToWSDL20	508 509 509
	Convert DTDOrSchema Convert DTDOrSchemaEx Convert ToWSDL20 Create Child	508 509 509 510
	Convert DTDOrSchema Convert DTDOrSchemaEx Convert To WSDL20 Create Child Create DBStructure FromXMLSchema	508 509 509 510 510
	Convert DTDOrSchema Convert DTDOrSchemaEx Convert To WSDL20 Create Child Create DBStructure FromXMLSchema Create Schema Diagram	508 509 509 510 510 511
	Convert DTDOrS chema Convert DTDOrS chemaEx Convert To WS DL20 Create Child Create DBS tructure From XMLS chema Create S chema Diagram Current Vie wMode	508 509 509 510 510 511 511
	Convert DTDOrSchema Convert DTDOrSchemaEx Convert To WSDL20 Create Child Create DBStructure FromXMLSchema Create SchemaDiagram Current Vie wMode DataRoot	508 509 510 510 511 511 511
	Convert DTDOrSchema Convert DTDOrSchemaEx Convert ToWSDL20 Create Child Create DBStructure FromXMLSchema Create SchemaDiagram Current Vie wMode DataRoot Doc Edit Vie w	508 509 510 510 511 511 511 511
	Convert DTDOrS chema Convert DTDOrS chemaEx Convert To WSDL20 Create Child Create DBS tructure From XMLS chema Create S chema Diagram Current Vie wMode DataRoot Doc Edit Vie w Encoding	508 509 510 510 511 511 511 511 512
	Convert DTDOrS chema Convert DTDOrS chemaEx Convert To WSDL20 Create Child Create DBS tructure From XMLS chema Create Schema Diagram Current Vie wMode DataRoot Doc Edit Vie w Encoding End Changes	508 509 510 511 511 511 511 511 512 512
	Convert DTDOrSchema Convert DTDOrSchemaEx Convert To WSDL20 Create Child Create DBStructure FromXMLSchema Create SchemaDiagram Current Vie wMode DataRoot DataRoot Doc Edit Vie w Encoding EndChanges Execute XQuery	508 509 510 511 511 511 511 511 512 512 513
	Convert DTDOrS chema Convert DTDOrS chemaEx Convert To WSDL20 Create Child Create DBS tructure From XMLS chema Create S chema Diagram Current Vie wMode DataRoot Doc Edit Vie w Enc oding End Changes Exe cute XQuery Export To Database	508 509 510 510 511 511 511 511 512 512 513
	Convert DTDOrS chema Convert DTDOrS chemaEx Convert To WSDL20 Create Child Create DBS tructure FromXMLS chema Create S chema Diagram Current Vie wMode DataRoot DataRoot Doc Edit Vie w Encoding EndChanges Export To Dat abase Export To Text	508 509 510 511 511 511 511 512 512 512 513 513 514
	Convert DTDOrSchema. Convert DTDOrSchemaEx. Convert To WSDL20. Create Child. Create DBStructure FromXMLSchema. Create SchemaDiagram. Current Vie wMode. DataRoot Doc Edit Vie w. Encoding EndChanges. Execute XQuery. Export To Database. Export To Text. Flatten DTDOrSchema.	508 509 510 511 511 511 511 512 512 512 513 513 514 515
	Convert DTDOrS chema. Convert DTDOrS chemaEx. Convert To WSDL20. Create Child. Create DBS tructure From XMLS chema. Create S chema Diagram. Current Vie wMode. DataRoot DataRoot Doc Edit Vie w. Encoding EndChanges. Execute XQuery. Export To Database. Export To Text. Flatt en DTDOrS chema. FullName	508 509 510 511 511 511 511 512 512 513 513 513 514 515
	Convert DTDOrS chema Convert DTDOrS chema Ex Convert To WSDL20 Create Child Create DBS tructure From XMLS chema Create S chema Diagram Current Vie wMode DataRoot Doc Edit Vie w Enc oding End Changes Execute XQuery Export To Dat abase Export To Text Flatten DTDOrS chema FullName Generate DTDOrS chema	508 509 510 511 511 511 511 512 512 512 513 513 513 514 515 515
	Convert DTDOrSchema. Convert DTDOrSchemaEx. Convert ToWSDL20. Create Child. Create DBStructure FromXMLS chema. Create SchemaDiagram. Current Vie wMode. DataRoot Doc Edit Vie w. Encoding EndChanges. Execute XQuery. Export ToDatabase. Export ToDatabase. Export ToText. Flatten DTDOrSchema. FullName Generate DTDOrSchemaEx.	508 509 510 511 511 511 511 512 512 513 514 515 515 515 516
	Convert DTDOrS che ma. Convert DTDOrS che maEx. Convert To WSDL20. Create Child. Create DBStructure From XMLS che ma. Create S che maDiagram. Current Vie wMode. DataRoot Doc Edit Vie w. Encoding EndChanges. Execute XQuery. Export ToDat abase. Export ToText. Flatten DTDOrS che ma. FullName Generate DTDOrS che ma. Generate DTDOrS che ma. Generate DTDOrS che ma. Generate DTDOrS che ma.	508 509 510 511 511 511 511 512 512 513 513 514 515 515 516 516
	Convert DTDOrS che ma. Convert DTDOrS che maEx. Convert To WSDL20. Create Child. Create DBStructure From XMLS che ma. Create Sche maDiagram. Current Vie wMode. DataRoot Doc Edit Vie w. Encoding EndChanges. Execute XQuery. Export ToDatabase. Export ToDatabase. Export ToText. Flatten DTDOrS che ma. FullName Generate DTDOrS che maEx. Generate PTDOrS che maEx. Generate ProgramCode.	508 509 510 511 511 511 511 512 512 513 513 514 515 515 515 516 516 517
	Convert DTDOrS chema Convert DTDOrS chema Ex	508 509 510 511 511 511 511 511 512 512 513 513 514 515 515 516 516 516 517 517

Generate	WS DLDocumentation	. 518
Generate	XBRLDocumentation	. 518
GetDBStr	uctureList	. 518
GetExport	Element List	. 519
GetPathN	ame (obsolete)	. 519
GridView		. 520
Is Modified		. 520
Is Valid		. 520
Is ValidEx		. 520
Is WellForn	ne d	. 521
Name		. 521
Parent		. 522
Path		. 522
RootElem	ent	. 522
Save		. 523
Save As		. 523
Saved		. 523
SaveInStr	ng	. 523
SaveToUF	L	. 524
SetActive	Document	. 524
SetEncod	ng (obsolete)	. 524
SetExtern	alls Valid	. 526
SetPathN	ame (obsolete)	. 526
StartChar	ge s	. 526
Suggestio	ns	. 527
SwitchVie	wMode	. 527
Text View		. 527
Title		. 527
Transform	XSL	. 528
Transform	XSLEx	. 528
Transform	XSLFO	. 528
Treat XBR	LInconsistencies As Errors	. 528
Updat e Vie	WS	. 529
Updat e XM	LData	. 529
- Document	s	. 530
Count		. 530
ltem		. 531
NewAuthe	nt ic File	. 531
NewFile		. 531
NewFile Fr	omText	. 532
OpenAuth	ent ic File	. 532
OpenFile		. 532
OpenURL		. 533
OpenURLI	Dialog	. 533
- DTDSchei	naGeneratorDlg	. 535

	Application	535
	AttributeTypeDefinition	535
	DTDS chemaFormat	535
	Frequent Elements	536
	GlobalAttributes	536
	MaxEnumLength	536
	Merge AllEqualNamed	536
	OnlyStringEnums	537
	Output Path	537
	Output Path Dialog Action	537
	Parent	537
	Resolve Entities	537
	Type Detection	538
	Value List	538
_		520
	Count	520
	L	539
		539
	Remove Element	539
-	ElementListItem	540
	Element Kind.	540
	FieldCount	540
	Name	540
	RecordCount	540
-	ExportSettings	541
	CreateKeys	541
	ElementList	541
	EntitiesToText	541
	Export AllEle ments	542
	Export Complete XML	542
	FromAttributes	542
	From Single SubElements	542
	FromText Values	542
	Independent PrimaryKey	542
	Namespace	543
	StartFromElement	543
	SubLe ve lLimit	543
_	FileSelectionDlg	544
	Application	544
	DialogAction	544
	FullName	544
	Parent	545
_	FindIn Files Dlg	546
	AdvancedXMLSearch	546
	Application	546
	DoReplace	547

	FileExtension	547
	Find	547
	Include Subfolders	547
	MatchCase	547
	Match Whole Word	548
	Parent	548
	RegularExpression	548
	Replace	548
	ReplaceOnDisk	548
	SearchInProjectFilesDoExternal	549
	SearchLocation	549
	ShowResult	549
	StartFolder	549
	XMLAttributeContents	549
	XMLAttributeNames	550
	XMLCData	550
	XMLComments	550
	XMLElementContents	550
	XMLElement Names	550
	XMLPI	551
	XMLRest	551
_	FindInFilesResult	552
	Application	552
	Count	552
	Document	552
	ltem	552
	Parent	553
	Path	553
-	FindInFilesResultMatch	554
	Application	554
	Length	554
	Line	554
	LineText	555
	Parent	555
	Position	555
	Replaced	555
-	FindInFilesResults	556
	Application	556
	Count	556
	ltem	556
	Parent	556
-	GenerateSampleXMLDlg	557
	Application	557
	Choice Mode	557
	ConsiderSampleValueHints	558

	ContentOfNillableElementsIsNonMandatory	558
	FillAttributesWithSampleData	558
	FillElementsWithSampleData	558
	FillWithSampleData - obsolete	559
	LocalNameOfRootElement	559
	NamespaceURIOfRootElement	559
	NonMandatoryAttributes	559
	NonMandatoryElements	559
	Optimization - obsolete	560
	Options DialogAction	560
	Parent	560
	RepeatCount	560
	Sample Value Hints	560
	SchemaOrDTDAssignment	561
	TakeFirstChoice - obsolete	561
	TryToUseNonAbstractTypes	561
_	GridView	562
	Events	562
	On Before Drag	562
	On Before Drop-	562
	On Before Start Editing	563
	OnEditingFinished	563
	$(n F \cap C \cap S \cap C \cap A \cap C \cap C$	504
	Current Es and	561
	Current Focus	564
	CurrentFocus	564 564
	CurrentFocus Deselect	564 564 564
	CurrentFocus Deselect Is Visible Select	564 564 565
	CurrentFocus Deselect Is Visible Select SetFocus	564 564 565 565
_	CurrentFocus Deselect Is Visible Select SetFocus SchemaDocumentationDlg	564 564 565 565 566
_	CurrentFocus Deselect Is Visible Select SetFocus SchemaDocumentationDlg	564 564 565 565 566 567
_	CurrentFocus Deselect Is Visible Select SetFocus SchemaDocumentationDlg	564 564 565 565 566 567 567
_	CurrentFocus Deselect Is Visible Select SetFocus SchemaDocumentationDlg	564 564 565 565 566 567 567 567
_	CurrentFocus Deselect Is Visible Select SetFocus SchemaDocumentationDlg	564 564 565 565 566 567 567 567
_	CurrentFocus Deselect Is Visible Select SetFocus SchemaDocumentationDlg	564 564 565 565 566 567 567 567 567
_	Current Focus Deselect Is Visible Select Set Focus SchemaDocumentationDlg	564 564 565 565 566 567 567 567 567 568 568
_	Current Focus Deselect Is Visible Select Set Focus SchemaDocumentationDlg	564 564 565 565 565 567 567 567 567 568 568 568
_	CurrentFocus Deselect Is Visible Select SetFocus SchemaDocumentationDlg	564 564 565 565 566 567 567 567 567 568 568 568 568
_	Current Focus Deselect Is Visible Select Set Focus SchemaDocumentationDlg. AllDetails Application Create Diagrams Folder. DiagramFormat Embed CS S In HTML Embed Diagrams. Generate Relative Links Include All Include Attribute Groups.	564 564 565 565 566 567 567 567 567 568 568 568 568 569 569
_	Current Focus Deselect Is Visible Select Set Focus SchemaDocumentationDlg	564 564 565 565 566 567 567 567 567 568 568 568 568 568 569 569
_	Current Focus Dese lect Svisible Select Set Focus SchemaDocumentationDlg	564 564 565 565 566 567 567 567 567 568 568 568 568 569 569 569 569
_	Current Focus	564 564 565 565 566 567 567 567 567 568 568 568 568 568 569 569 569 569 569 569
_	CurrentFocus	564 564 565 565 566 567 567 567 567 568 568 568 568 568 569 569 569 569 569 570 570
_	Current Focus Deselect Is Visible Select Set Focus SchemaDocument at ion Dlg	564 564 565 565 566 567 567 567 567 568 568 568 569 569 569 569 569 569 570 570
_	CurrentFocus	564 564 565 565 567 567 567 567 567 568 568 568 568 569 569 569 569 569 570 570 570 570

	Include Re de fines	571
	Include ReferencedSchemas	571
	Include Simple Types	572
	Multiple Output Files	572
	Options DialogAction	572
	Output File	573
	Output File DialogAction	573
	Output Format	573
	Parent	573
	ShowAnnotations	574
	ShowAttributes	574
	ShowChildren	574
	ShowDiagram	575
	ShowEnumerations	575
	ShowIdentityConstraints	575
	ShowNamespace	575
	ShowPatterns	576
	ShowProgressBar	576
	ShowProperties	576
	ShowResult	577
	ShowSingleFacets	577
	ShowSourceCode	577
	ShowType	577
	ShowUsedBy	578
	SPSFile	578
	Use Fixe dDe sign	578
-	SpyProject	579
	CloseProject	579
	ProjectFile	579
	Rootltems	579
	SaveProject	580
	SaveProjectAs	580
_	SpyProjectItem	581
	ChildItems	581
	FileExtensions	581
	ItemType	581
	Name	581
	Open	582
	Parentltem	582
	Path	582
	ValidateWith	582
	XMLForXSLTransformation	582
	XSLForXMLTransformation	582
	XSLTransformationFileExtension	583
	XSLTransformationFolder	583

AddFile 584 AddFolder 584 AddURL 584 AddURL 585 Count 585 hem 585 Removeltem 585 - TextImportExportSettings 586 CommentIncluded 586 DestinationFolder 586 EnclosingCharacter 586 EncodingByteOrder 587 FileIDelimiter 587 FileExtension 587 HeaderRow 587 ImportFile 587 RemoveNewline 588 - Text View 589 OnBeforeShowSuggestions 589 OnChar 590 GoToLineChar 591 LineCount 591 LineCount 591 LineCount 592 Parent 592 SelectionEnd 592 Parent 592 SelectionEnd 592 Parent 592 SelectionEnd 593 SelectionEnd 593 <t< th=""><th>-</th><th>SpyProjectItems</th><th>584</th></t<>	-	SpyProjectItems	584
AddFolder 584 AddURL 584 Count 585 Item 585 Remove Item 585 TextImportExportSettings 586 CommentIncluded 586 DestinationFolder 586 Encoding 586 EncodingByteOrder 587 FieldDelimiter 587 FieldDelimiter 587 FieldDelimiter 587 HeaderRow 587 ImportFile 587 RemoveNewline 588 RemoveNewline 588 OnBeforeShowSuggestions 589 OnChar 590 Application 590 Got OLineChar 591 LineCount 591 LineCount 591 LineCount 592 Parent 592 SelectionEnd 592 SelectionEnd 592 SelectionEnd 593 SelectText 593 SelectText 593 SelectText 593 <		AddFile	584
AddURL 584 Count 585 Iem 585 Remove Item 585 Remove Item 585 Comment Included 586 DestinationFolder 586 EnclosingCharacter 586 EncodingByteOrder 587 FileDelimiter 587 FileDelimiter 587 HeaderRow 587 HeaderRow 587 ImportFile 587 Remove Newline 588 TextView 589 OnBeforeShowSuggestions 589 OnChar 590 GoToLineChar 591 Line FormPosition 591 Line Length 592 Parent 592 Parent 592 Parent 592 SelectionEnd 593 SelectionEnd 593 SelectionEnd 592 Parent 592 Parent 593 SelectionEnd 593 SelectionEnd 593 Se		AddFolder	584
Count 585 Item 585 Removellem 585 Text Import Export Settings 586 Comment Included 586 DestinationFolder 586 Encoding 586 EncodingByteOrder 586 FieldDe limiter 587 FieldDe limiter 587 FieldDe limiter 587 Header Row 587 Import File 587 Remove De limit er 588 Remove Newline 588 Text View 589 OnBefore ShowSuggestions 589 On Defore ShowSuggestions 589 On Char 590 Application 590 Go To Line Char 591 Line Count 591 Line Count 591 Line Count 591 Line Chard 592 Parent 592 PositionFromLine 592 Select Toxt 593 Select Toxt 593<		AddURL	584
Item 585 Remove Item 585 Text Import Export Settings 586 Comment Included 586 Destination Folder 586 Enclosing Character 586 Encoding 586 Encoding Deter 587 FieldDe limiter 587 FieldDe limiter 587 HeaderRow 587 Import File 587 Remove De limit er 588 Remove Newline 588 Text View 589 OnDefore ShowSugge stions 589 On Char 590 Application 590 Got ToLine Char 591 Line From Position 591 Line From Position 592 Parent 592 Parent 592 Select Text 593 Select Text 5		Count	. 585
Remove h em. 585 Text Import Export Settings 586 Comment Included 586 Destination Folder 586 Encoding Character. 586 Encoding Byte Order 587 FieldDe limiter. 587 FieldDe limiter. 587 Fiel Ext ension 587 Header Row 587 Import File 587 Remove De limit er 588 Remove De limit er 588 Itext View 589 On De fore Show Suggestions 589 On Char 590 Application 590 Go To Line Char 591 Line From Position 591 Line Count 591 Line From Position 592 Select Text 592 Select Text 593 Select ton Stat 593		Item	. 585
 TextImportExportSettings TextImportExportSettings S86 CommentIncluded S86 DestinationFolder S86 EncodingCharacter S87 FicelDE limiter S87 FileExtension File Extension S87 Remove De limiter S88 Remove Ne wine S88 OnBefore ShowSuggestions S89 OnChar OnBefore ShowSuggestions S90 GoToLineChar S91 Line FromPosition S91 Line Count S91 Line FromPosition S92 PositionFromLine S92 SelectText S93 SelectText S93 Text S93 SelectText S93 Text S93 SelfText S93 Text S94 AllDetails S95 Application S95 Create Diagrams Folder S95 Create Diagrams Folder S95 DiagramFormat S95 Create Diagrams Folder S95 DiagramFormat S95 DiagramFormat S95 Create Diagrams Folder S95 DiagramFormat S95 Create Diagrams Folder S95 DiagramFormat S95 Create Diagrams Folder S95 Create Diagrams Folder S95 DiagramFormat S95 Create Diagrams Folder S95 Create Diagrams F		Remove It em	. 585
Comment Included 586 DestinationFolder 586 EnclosingCharacter 586 Encoding 586 EncodingByteOrder 587 FieldDelimiter 587 FieldDelimiter 587 FieldDelimiter 587 FieldDelimiter 587 RemoveDelimiter 588 Remove Delimiter 588 Remove Ne wline 588 Text View 589 OnBefore ShowSuggestions 589 OnChar 590 Application 590 Go To Line Char 591 Line Count 591 Line Count 591 Line FromPosition 591 Line FromDosition 592 Parent 592 SelectionEnd 592 SelectionEnd 593 SelectionEnd 592 SelectionEnd 592 SelectionEnd 592 SelectionEnd 593 SelectionEnd	_	TextImportExportSettings	. 586
DestinationFolder 586 EnclosingCharacter 586 Encoding 586 EncodingByteOrder 587 FieldDelimiter 587 FieldDelimiter 587 HeaderRow 587 ImportFile 587 Remove Delimiter 588 Remove Delimiter 588 Remove Newline 588 Text View 589 OnBeforeShowSuggestions 589 OnChar 590 Application 590 GoToLineChar 591 Length 591 Line Count 591 Line Count 591 Line FromPosition 592 Parent 592 SelectionEnd 592 SelectionEnd 593 SelectionEnd 593 SelectionStart 593 SelectionEnd 593 SelectionEnd 593 SelectionEnd 593 SelectionEnd 593		Comment Inc lude d	. 586
EnclosingCharacter 586 Encoding 586 EncodingByteOrder 587 FieldDelimiter 587 FileExtension 587 HeaderRow 587 ImportFile 587 Remove Delimiter 588 Remove Newline 588 TextView 589 DoBeforeShowSuggestions 589 OnChar 590 Application 590 GoToLineChar 591 Length 591 LineCount 591 Line FromPosition 592 Parent 592 Parent 592 SelectionStart 593 SelectionStart 593 SelectionStart 593 SelectionStart 593 Text 593 SelectionEnd 593 SelectionStart 593 SelectionStart 593 SelectionEnd 593 Text 593 SelectionEnd 593 SelectionStart 593 <		DestinationFolder	. 586
Encoding 586 EncodingByteOrder 587 FieldDelimiter 587 FieldDelimiter 587 FileExtension 587 HeaderRow 587 HeaderRow 587 ImportFile 587 RemoveDelimiter 588 RemoveNewine 588 TextView 589 DoBeforeShowSuggestions 589 OnChar 590 Application 590 GoToLineChar 591 Length 591 Line Count 591 Line FromPosition 591 Line Ingth 592 PositionFromLine 592 SelectionEnd 592 SelectionEnd 593 SelectText 593 <t< td=""><td></td><td>EnclosingCharacter</td><td>586</td></t<>		EnclosingCharacter	586
EncodingByteOrder 587 FieldDelimiter 587 FileExtension 587 FileExtension 587 HeaderRow 587 ImportFile 587 RemoveDelimiter 588 RemoveNewline 588 TextView 589 Events 589 OnBeforeShowSuggestions 589 OnChar 590 Application 590 GoToLineChar 591 Length 591 LineCount 591 Line FromPosition 591 MoveCaret 592 Parent 592 SelectionEnd 592 SelectionEnd 592 SelectionEnd 592 SelectionEnd 592 SelectionStart 593 SelectText 593		Encoding	586
FieldDelimiter 587 FileExtension 587 FileExtension 587 HeaderRow 587 ImportFile 587 Remove Delimiter 588 Remove Delimiter 588 Remove Newline 588 Text View 589 Events 589 OnBefore ShowSuggestions 589 OnChar 590 Application 590 GoToLine Char 590 GoToLine Char 591 Length 591 Line FromPosition 591 Line FormPosition 592 Parent 592 PositionFromLine 592 SelectionEnd 592 SelectionEnd 593 SelectText 593 SelectText 593 Text 593 SelectText 593 SelectText 593 SelectText 593 SelectText 593 SelectText 593 SelectText 595		EncodingByteOrder	. 587
File Extension 587 HeaderRow 587 Import File 587 Remove De limit er 588 Remove Ne wline 588 Text Vie w 589 Events 589 OnBefore ShowSuggestions 589 On Char 590 Application 590 Got ToLine Char 591 Line Count 591 Line FromPosition 591 Line Length 592 Parent 592 PositionFromLine 592 SelectionEnd 593 Select Text 593		FieldDelimiter	. 587
HeaderRow 587 ImportFile 587 Remove De limit er 588 Remove Ne wline 588 Text Vie w 589 Events 589 OnBeforeShowSuggestions 589 OnChar 590 Application 590 Got OLine Char 591 Length 591 Line Count 591 Line Count 591 Line From Position 591 Move Caret 592 Parent 592 Position FromLine 592 Select Tost 593 Select Text 593 Select Text 593 Sel Text 593 Text 593 Sel Text 5		FileExtension	. 587
Import File 587 Remove De limit er 588 Remove Ne wline 588 Text View 589 Events 589 OnBefore ShowSuggestions 589 OnChar 590 Application 590 Get Range Text 590 Go To Line Char 591 Length 591 Line Count 591 Line Count 591 Line Length 592 Parent 592 Position FromLine 592 Parent 592 Select Text 593 Select Text 593 Select Text 593 Sel Text 593 </td <td></td> <td>HeaderRow</td> <td>. 587</td>		HeaderRow	. 587
remove De limit er 588 Remove Newline 588 Text View 589 Events 589 OnBefore ShowSuggestions 589 OnChar 590 Application 590 Got OLine Char 591 Length 591 Line Count 591 Line FromPosition 591 Line FromPosition 592 Parent 592 PositionFromLine 592 SelectionEnd 592 SelectionEnd 593 Select Text 593		Import File	. 587
RemoveNewline.588Text View589Events589OnBeforeShowSuggestions589OnChar590Application590GetRangeText590GoToLineChar591Length591LineCount591Line FromPosition591Line FromPosition591MoveCaret592Parent592PositionFromLine592SelectionEnd593SelectionEnd593SelectText593SelectText593Text593VSDLDocumentationDlg594AllDetails595Application595DiagramFormat595EmbedCSShHITML596		Remove De limit er	. 588
- Text View 589 Events 589 OnBefore ShowSuggestions 589 OnChar 590 Application 590 Get Range Text 590 Go To Line Char 591 Length 591 Line Count 591 Line FromPosition 591 Move Caret 592 Parent 592 PositionFromLine 592 Replace Text 593 SelectionEnd 592 SelectionEnd 593 SelectText 593 SelectionStart 593 Seletails 595 AllDetails 595 Application 595 DiagramFormat 595 EmbedCSSInHTML 596		Remove Newline	. 588
Events 589 OnBefore ShowSuggestions 589 OnChar 590 Application 590 Get Range Text 590 Go To Line Char 591 Length 591 Line Count 591 Line FromPosition 591 Line FromPosition 591 Move Caret 592 Parent 592 PositionFromLine 592 Replace Text 592 SelectionEnd 592 SelectionEnd 593 SelectText 593 SelectText 593 Text 593 VSDLDocument at ionDlg 594 AllDetails 595 DiagramFormat 595 DiagramFormat 595	_	TextView	. 589
OnBeforeShowSuggestions 589 OnChar 590 Application 590 GetRangeText 590 Go To Line Char 591 Length 591 Line Count 591 Line FromPosition 591 Line Length 591 Move Caret 592 Parent 592 PositionFromLine 592 Replace Text 592 SelectionStart 593 Select Text 593 Select Text 593 Text 593 VSDLDocument at ionDig 594 AllDet ails 595 Application 595 DiagramsFolder 595 EmbedCSS In HTML 596		Events	589
OnChar 590 Application 590 GetRangeText 590 Go To Line Char 591 Length 591 Line Count 591 Line FromPosition 591 Line Length 591 Move Caret 592 Parent 592 PositionFromLine 592 Replace Text 592 SelectionEnd 592 Select Text 593 Select Text 593 Select Text 593 Text 593 VSDLDocumentationDig 594 AllDetails 595 Application 595 Diagrams Folder 595 EmbedCSS In HTML 596		OnBeforeShowSuggestions	589
Application590GetRangeText590GoToLineChar591Length591LineCount591LineFromPosition591LineLength591MoveCaret592Parent592PositionFromLine592ReplaceText593SelectionEnd593SelectText593SelectText593Text593VSDLDocumentationDlg594AllDetails595CreateDiagramsFolder595DiagramFormat595EmbedCSSInHTML596		On Char	590
Get Range Text590Go To Line Char.591Length591Line Count591Line FromPosition591Line Length591Move Caret592Parent592Position FromLine592Replace Text592SelectionEnd593Select Text593Select Text593Text593VSDLDoc umentationDig594AllDetails595Create Diagrams Folder595Diagram Format595Embed CSS In HTML596		Application	590
GoToLineChar.591Length591Line Gount591Line Count591Line FromPosition591Line Length591Move Caret592Parent592PositionFromLine592Replace Text592SelectionEnd592Select Text593Select Text593Select Text593Select Text593Select Text593Select Text593Create Diagrams Folder595Diagram Format595Embed CSS In HTML596		GetRangeText	590
Length591Line Count591Line Count591Line FromPosition591Line Length591Move Caret592Parent592Position From Line592Replace Text592Selection End592Select Text593Select Text593Select Text593Select Text593Select Text593Select Text593Create Diagrams Folder595Diagram Format595Embed CSS In HTML596		GoToLineChar	591
Line Count591Line FromPosition591Line Length591Move Caret592Parent592Position From Line592Replace Text592Selection End592Select Text593Select Text593Select Text593Sel Text593Text593Text593VS DL Document at ion Dlg.594All Details595Create Diagrams Folder595Diagram Format595Embed CS SIn HT ML596		Length	591
Line FromPosition591Line Length591Move Caret592Parent592Position From Line592Replace Text592Selection End592Selection Start593Select Text593Select Text593Select Text593Text593Text593VSDL Document at ion Dlg594All Details595Application595Diagram Format595Embed CSS In HTML596		LineCount	591
Line Length591Move Caret592Parent592Position From Line592Replace Text592Selection End592Selection Start593Select Text593Select Text593Text593Text593All Details595Application595Diagram Format595Embed C S S In HTML596		LineFromPosition	591
MoveCaret592Parent592PositionFromLine592ReplaceText592SelectionEnd592SelectionStart593SelectText593SelText593Text593Text593AllDetails595Application595DiagramFormat595EmbedCSSInHTML596		LineLength	591
Parent592PositionFromLine592ReplaceText592SelectionEnd592SelectionStart593SelectText593SelText593Text593Text593VSDLDocumentationDlg594AllDetails595Application595DiagramFormat595EmbedCSSInHTML596		MoveCaret	592
PositionFromLine592ReplaceText592SelectionEnd592SelectionStart593SelectText593SelText593Text593Text593AllDetails595Application595DiagramFormat595EmbedCSSInHTML596		Parent	592
Replace Text592SelectionEnd592SelectionStart593SelectText593SelText593Text593- WSDLDocumentationDlg594AllDetails595Application595Create DiagramsFolder595DiagramFormat595EmbedCSSInHTML596		PositionFromLine	592
SelectionEnd.592SelectionStart.593SelectText593SelText593Text593- WSDLDocumentationDlg.594AllDetails595Application595CreateDiagramsFolder.595DiagramFormat595EmbedCSSInHTML596		ReplaceText	592
SelectionStart		SelectionEnd	592
SelectText593SelText593Text593- WSDLDocumentationDlg594AllDetails595Application595CreateDiagramsFolder595DiagramFormat595EmbedCSSInHTML596		SelectionStart	593
SelText593Text593- WSDLDocumentationDlg.594AllDetails595Application595CreateDiagramsFolder595DiagramFormat595EmbedCSSInHTML596		SelectText	. 593
Text593- WS DLDoc umentation Dlg.594AllDetails595Application595Create Diagrams Folder595Diagram Format595Embed CSS In HTML596		SelText	. 593
- WSDLDocumentationDlg		Text	593
AllDetails 595 Application 595 CreateDiagramsFolder 595 DiagramFormat 595 EmbedCSSInHTML 596	-	WS DLDocumentationDlg	. 594
Application595CreateDiagramsFolder595DiagramFormat595EmbedCSSInHTML596		AllDetails	595
CreateDiagramsFolder		Application	595
DiagramFormat		CreateDiagramsFolder	. 595
EmbedCSSInHTML		DiagramFormat	595
		Embe dCSSInHTML	596

	Embe dDiagrams	596
	GlobalElementsAndTypesOnly	596
	Include All	597
	Include Binding	597
	Include ImportedWSDLFiles	597
	Include Messages	597
	Include Overview	598
	Include Port Type	598
	Include Service	598
	Include Types	599
	Multiple Output Files	599
	Options DialogAction	599
	OutputFile	599
	Output File DialogAction	600
	Output Format	600
	Parent	600
	SeparateSchemaDocument	601
	ShowBindingDiagram	601
	ShowExtensibility	601
	ShowMessageParts	601
	ShowPort	602
	ShowPortTypeDiagram	602
	ShowPort Type Operations	602
	ShowProgressBar	603
	ShowResult	603
	ShowServiceDiagram	603
	ShowSourceCode	603
	ShowTypesDiagram	604
	ShowUsedBy	604
	Use Fixe dDesign	604
	SPSFile	605
_	WSDL20DocumentationDlg	606
	AllDetails	607
	Application	607
	Create Diagrams Folder	607
	DiagramFormat	607
	Embe dCSSInHTML	608
	Embe dDiagrams	608
	GlobalElementsAndTypesOnly	608
	Include All	609
	Include Binding	609
	Include ImportedWSDLFiles	609
	Include Interface	609
	Include Overview	610
	Include Service	610

	Include Types	610
	Multiple Output Files	610
	Options DialogAction	611
	Output File	611
	Output File DialogAction	611
	Output Format	612
	Parent	612
	SeparateSchemaDocument	612
	ShowBindingDiagram	612
	ShowEndpoint	613
	ShowExtensibility	613
	ShowFault	613
	ShowInterfaceDiagram	614
	ShowOperation	614
	ShowProgressBar	614
	ShowResult	614
	ShowServiceDiagram	615
	ShowSourceCode	615
	ShowTypes Diagram	615
	ShowUsedBy	616
	SPSFile	616
	Like Fixed Design	616
_	XBRI Documentation Dla	617
	AllDetails	618
	Application	618
	Create Diagrams Folder	618
	DiagramFormat	618
	EmbedCSSInHTML	619
	EmbedDiagrams	619
	Include All	619
	Include Calculation Linkroles	620
	Include Definition Linkroles	620
	Include GlobalElements	620
	Include Namespace Prefixes	620
	Include Overview	621
	Include Presentation Linkroles	621
	Ontions Dialog Action	621
	Output File	622
	Output File Dialog Action	622
	Output Format	622
	Parent	622
	Short QualifiedName	622
	ShowAbstract	623
	ShowBalance	623
	ShowDiagram	624
	5110 wDiagram	024

	ShowImportedElements	624
	ShowItemtype	624
	ShowLabels	624
	ShowLinkbaseReferences	625
	ShowNillable	625
	ShowPeriod	625
	ShowProgressBar	626
	ShowReferences	626
	ShowResult	626
	ShowSubstitutiongroup	626
	SPSFile	627
	Use Fixe dDe sign	627
	- XMLData	628
	AppendChild	629
	CountChildren	629
	Count Children Kind	629
	Erase AllChildren	630
	EraseChild	630
	EraseCurrentChild	630
	GetChild	631
	GetChildAttribute	631
	GetChildElement	632
	GetChildKind	632
	GetCurrentChild	632
	GetFirstChild	633
	GetNamespacePrefixForURI	633
	GetNextChild	633
	Get Text Value XMLDecoded	634
	HasChildren	635
	Has Children Kind	635
	InsertChild	635
	Insert ChildAfter	636
	Insert ChildBefore	636
	Is Same Node	636
	Kind	636
	MayHave Children	637
	Name	637
	Parent	637
	Set Text Value XMLEncoded	638
	Text Value	638
3.3.3	Enumerations	639
	- ENUMApplicationStatus	640
	- SPYAttributeTypeDefinition	641
	- SPYAuthentic Actions	642
	- SPYAuthenticDocumentPosition	643

	- SPYAuthenticElementActions	644
	- SPYAuthenticElementKind	645
	- SPYAuthentic MarkupVisibility	646
	- SPYAuthenticToolbarButtonState	647
	- SPYDatabaseKind	648
	- SPYDialogAction	649
	- SPYDOMType	650
	- SPYDTDSchemaFormat	651
	- SPYEncodingByteOrder	652
	- SPYExport Namespace	653
	- SPYFindInFilesSearchLocation	654
	- SPYFrequentElements	655
	- SPYImageKind	656
	- SPYImportColumnsType	657
	- SPYKeyEvent	658
	- SPYKeyStatus	659
	- SPYLibType	660
	- SPYLoading	661
	- SPYMouseEvent	662
	- SPYNumberDateTimeFormat	663
	- SPYProgrammingLanguage	664
	- SPYProjectItemTypes	665
	- SPYProjectType	666
	- SpySampleXMLGenerationChoiceMode	667
	- SPYSampleXMLGenerationOptimization (Obsolete)	668
	- SpySample XMLGenerationSample Value Hints	669
	- SPYSample XMLGenerationSchemaOrDTDAssignment	670
	- SPYSchemaDefKind	671
	- SPYSchemaDocumentationFormat	672
	- SPYSchemaExtensionType	673
	- SPYSchemaFormat	674
	- SPYTextDelimiters	675
	- SPYTextEnclosing	676
	- SPYTypeDetection	677
	- SPYURLTypes	678
	- SPYValidateXSDVersion	679
	- SPYValidateErrorFormat	680
	- SPYViewModes	681
	- SPYVirtualKeyMask	
	- SPYXMLDataKind	683
Active	X Integration	684
3 4 1	· · · · · · · · · · · · · · · · · · ·	685
347	るctiveX コントロールをツールボックスに追加する	
2/2	マプリケーションノベルのな今	
J.T.J		

3.4

3.4.4	ドキュメントレベルの統合	692
3.4.5	ActiveX 統合のサンプル	696
	- C#	697
	サンプル C# ソリューションの実行	697
	- HTML	
	アプリケーションレベルの HTML 統合	
	コントロールインスタンスを作成する	
	デフォルトのトキュメントを開くボタンを追加する	
	トキュシトレインレの HIML 統合	
	エディターウィンドウの作成	
	ヘルパーウィンドウのためにプレースホルダを作成する	
	- Java	
	サンプル Java プロジェクト	
	ActiveX コントロールの作成	
	コントロール内のデータのロード	
	基本のイベント(ンドリング	
	メニュー	
	⋃ アップデー ┠イベン ┠∖ンドリング	
	XML ツリーの作成	
3.4.6	コマンド レファレンス	
	- 「ファイル」メニュー	
	- 編集」メニュー	
	- 「プロジェクト」メニュー	
	- 「XML」ノー	
	ר [−] [¶] XSL/ XQuery אב⊐ −	
	- 「Authentic」メニュー	
	- 表示」メニュー	
	- 「ブラウザー」メニュー	
	- 『ソール」メニュー	
	- ヘルプ」メニュー	
	- ヘヘルプ」メニュー	
3.4.7	オブジェクトレファレンス	
	- Authentic DesktopCommand	
	Accelerator	
	D	
	kSeparator	
	Label	
	Name	
	StatusText	
	SubCommands	
	ToolTip	
	- Authentic DesktopCommands	

Count	730
ltem	730
- Authentic DesktopControl	731
プロパティ	731
Appearance	731
Applic at ion	732
BorderStyle	732
CommandsList	732
Enable UserPrompts	733
IntegrationLevel	733
MainMenu	733
Toolbars	/34
メソッド	735
<i>Exec</i>	735
Ope n	735
QueryStatus	735
Events	736
On Close Edit ing Window	736
On Context Changed	736
OnDocumentOpened	736
On File Change dAle rt	737
OnLicense Problem	737
OnOpenedOrFocused	737
On ToolWindowUpdated	/3/
()n l h dat o ('m dl ll	138
	728
On Validat ion WindowUpdat e d	738
On Opticite Child of Mindow Updated	738 739
ののDate CMuOr On Validation WindowUpdated Authentic DesktopControlDocument プロパティ	738 739 739
On Opticite Child on Window Updat e d On Validation Window Updat e d Authentic Desktop Control Document プロパティ Appearance	738 739 739 739
On Opticite Child on Window Updat e d On Validation Window Updat e d Authentic Desktop Control Document プロパティ Appe arance Border Style	738 739 739 739 739 740
On Validation Window Updated Authentic DesktopControlDocument $\mathcal{J} \square \mathcal{I} \neg \mathcal{I}$ Appearance BorderStyle Document	738 739 739 739 740 740 740
On Validation Window Updat e d Authentic DesktopControlDocument $\mathcal{J}\square \mathcal{I} \neg \mathcal{I}$ Appe arance BorderStyle Document Is Modified	738 739 739 739 740 740 740 740
On Validation Window Updat e d Authentic DesktopControlDocument $\mathcal{J} \Box \mathcal{N} \mathcal{F} \mathcal{I}$ Appearance BorderStyle Document Is Modified Path Describer	738 739 739 739 740 740 740 740 740
On Validation Window Updat e d Authentic DesktopControlDocument $\mathcal{J} \square \mathcal{K} \mathcal{F} \mathcal{I}$ Appearance BorderStyle Document Is Modified Path ReadOnly	738 739 739 739 740 740 740 740 740
On Validation Window Updat ed Authentic DesktopControlDocument プロパティ Appe arance BorderStyle Document Is Modified Path ReadOnly	738 739 739 739 739 740 740 740 740 740 741
On Validation Window Updat ed Authentic DesktopControlDocument プロパティ Appearance BorderStyle Document Is Modified Path ReadOnly メソッド Exec	738 739 739 739 740 740 740 740 740 741 741
On Update Chulor On Validation Window Updated Authentic DesktopControlDocument プロパティ Appearance BorderStyle Document Is Modified Path ReadOnly メソッド Exec New	738 739 739 739 740 740 740 740 740 741 741 741
On Opticity Control Document On Validation Window Updated Authentic DesktopControl Document	738 739 739 739 740 740 740 740 740 741 741 741 741
On Opticity Control Window Updat ed Authentic DesktopControl Document プロパティ Appearance BorderStyle Document Is Modified Path ReadOnly メノッド Exec New Open QueryStatus Poload	738 739 739 739 740 740 740 740 740 741 741 741 741 742 742
On Opticity Control Window Updat ed Authentic DesktopControl Document プロパティ Appe arance BorderStyle Document Is Modified Path ReadOnly メソッド Exec New Open QueryStatus Reload Sava	738 739 739 739 740 740 740 740 740 741 741 741 741 742 742 742 742
On Opticite Child on Window Updat e d Authentic DesktopControlDocument プロパティ Appearance BorderStyle Document Is Modified Path ReadOnly シリド Exec New Open QueryStatus Reload Save Save	738 739 739 739 740 740 740 740 740 740 741 741 741 742 742 742 742 742
On Opticity Control Window Updat ed Authentic DesktopControl Document プロパティ Appearance Border Style Document Is Modified Path ReadOnly メソッド Exec New Open QueryStatus Reload Save Save As	738 739 739 739 740 740 740 740 740 740 741 741 741 742 742 742 742 742 742
On Validation Window Update d Authentic DesktopControlDocument $\mathcal{I}\square \mathcal{I} \mathcal{I} \mathcal{F} \mathcal{I}$ Appearance Border Style Document k Modified Path ReadOnly $\mathcal{Y}\mathcal{Y} \mathcal{I}^{F}$ Exec New Open QueryStatus Reload Save Save As Events On Activate	738 739 739 739 740 740 740 740 740 740 740 741 741 741 742 742 742 742 742 743 743
On Validation Window Update d Authentic DesktopControlDocument $\mathcal{I}\square I \land \mathcal{F} \prec I$ Appearance Border Style Document Is Modified Path ReadOnly $\mathcal{Y} \lor \mathcal{Y} \vdash$ Exec New Open QueryStatus Reload Save Save As Events On Activate	738 739 739 739 740 740 740 740 740 740 741 741 741 741 742 742 742 742 743 743
$On Validat ion Window Updat e d$ Authentic Deskt opControlDocument $ \begin{array}{c} \mathcal{I} \square \mathcal{I} \overline{\mathcal{T}} \mathcal{I} \\ Appe arance \\ BorderStyle \\ Document \\ k Modified \\ Path \\ ReadOnly \\ \mathcal{Y} \mathcal{Y} \mathcal{F} \\ Exec \\ New \\ Open \\ QueryStatus \\ Reload \\ Save \\ Save As \\ Events \\ On Context Changed \\ On Document Closed \\ Decument Closed \\ Decument \\ Document \\ Documen$	738 739 739 739 740 740 740 740 740 740 741 741 741 741 742 742 742 742 743 743 743
סאט איז	738 739 739 739 740 740 740 740 740 740 740 740 740 741 741 741 742 742 742 743 743 743 743
On Validation Window Update d Authentic DesktopControlDocument $\mathcal{I}\square \mathcal{I} \mathcal{T} \mathcal{T} \mathcal{I}$ Appearance Border Style Document Is Modified Path ReadOnly $\mathcal{X}\mathcal{Y}\mathcal{Y}\mathcal{F}$ Exec New Open QueryStatus Reload Save Save As Events On Context Changed On Document Close d On Document Save As	738 739 739 739 740 740 740 740 740 740 740 740 740 741 741 741 742 742 742 743 743 743 743 744
On Validation Window Updat e d Authentic Deskt opControlDocument $\mathcal{T} \Box \mathcal{N} \overline{\tau} 1$ Appearance Border Style Document Is Modified Path ReadOnly $\mathcal{N} \mathcal{Y} \mathcal{Y} \mathcal{F}$ Exec New Open QueryStatus Reload Save Save As Events On Context Change d On Document Close d On Document Save As	738 739 739 739 740 740 740 740 740 740 740 740 741 741 741 741 742 742 742 743 743 743 744 744
On Validation WindowUpdat e dAuthentic DesktopControlDocument $\mathcal{J} \Box I \mathcal{K} \mathcal{F} 1$ AppearanceBorderStyleDocumentIs ModifiedPathReadOnly $\mathcal{X} \mathcal{Y} \mathcal{Y} \mathcal{K}$ ExecNewOpenQueryStatusReloadSaveSave AsEventsOn Context ChangedOn Document ClosedOn Document Save AsOn Document ClosedOn Document ClosedOn File Change dAlertOn Modified FlaeChanged	738 739 739 739 740 740 740 740 740 740 740 740 740 740 741 741 742 742 742 743 743 743 744 744

-	Authentic DesktopControlPlaceHolder	745
	プロパティ	745
	Labe l	745
	PlaceholderWindowID	745
	Project	745
	メソッド	746
	OpenProject	746
	Close Project	746
	イベント	746
	OnModifiedFlagChanged	746
	OnSetLabel	747
_	列挙	748
	ICActive XIntegrationLevel	748
	Authentic DesktopControlPlaceholderWindow	748

4 付録

750

4.1	技術デ	ータ	751
	4.1.1	OS とメモリ要件	752
	4.1.2	Altova XML バリデーター	753
	4.1.3	Altova XSLT とXQuery エンジン	754
	4.1.4	Unicode のサポート	755
	4.1.5	インターネットの使用	756
4.2	ライセン	ノス情報	757
	4.2.1	電子的なソフトウェアの配布	758
	4.2.2	知的所有権	759
	4.2.3	Altova Authentic エンドユーザー使用許諾契約書	760

インデックス

Chapter 1

Altova Authentic 2018 Desktop Enterprise Edition

Altova Authentic 2018 Desktop Enterprise Edition

XML ドキュメントを認証 するための画期的かつ視覚的なアプローチである Altova Authentic 2018 Desktop Enterprise Edition を使用するとエンドユーザーを XML の技術的な側面に煩わ すことなく作業することができます。Authentic Desktop は プラットフォーム更新済みの Windows 7 SP1、Windows 8、Windows 10 とプラットフォーム更新済みの Windows Server 2008 R2 SP1 また は以降で動作します。Authentic Desktop は64ビット版 ど32ビット版 がご利用になれます。



最終更新日:2018年 96月 21日

Chapter 2

ユーザーガイドとレファレンス

6

ユーザーガイドとレファレンス

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

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

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

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

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

C:\Users\ <username>\Documents</username>

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

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

メモ: XMLSpy は、プラットフォーム更新済みの Windows Server 2008 R2 SP1 または以降 でも サポートされています。

1 インターフェイスと環境

このセクションでは以下の説明を行ないます:

- アプリケーション GUI
- アプリケーション環境

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

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

1.1 グラフィカルユーザインターフェース (GUI)

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

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

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

lenu Bar			
oolbars			
Project Window	Main Window		Entry Helper Windows
	Active document	Scroll document tabs	
	Active Document View	Inactive Document Views	
	Doc1 Y Doc2)	45	1
	A		1

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

サイドバー表示の切り替え

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

サイドバーのフロートとドッキング

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

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

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

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

サイドバーを自動的に隠す

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

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

1.1.1 メイン ウィンドウ

(以下の図に表示される)メインウィンドウではドキュメントの編集を行ないます。

メニューバー	_8×
ツールバー	
アクティブなドキ 元に	ニュメントを最小化、 戻す、閉じるボタン
メイン	ウィンドウ
► 797-175×+1×2	[↓] ドキュメントタブのスクロール ↓07%ティフムヒュー
アクティブなドキュメントビュー	非アクティブなドキュメントビュー
Doc1 Doc2	40

メインウィンドウのファイル

- 開いたじ編集を行うファイルの数に制限はありません。
- 開かれた各ドキュメントは、各自ウィンドウに表示され、その名前がメインウィンドウの下部にあるタブに表示されます。ドキュメントをアクティブにするには、そのドキュメントのタブをクリックしてください。
- 複数のファイルが開かれている場合、ドキュメントタブバーに収まりきらず、一部のドキュメントタブが表示されないことがあります。目的のドキュメントタブは、(i)ドキュメントタブバーの右にあるスクロールボタンを使うことで、または(ii)ウィンドウメニューから目的のドキュメントを選択することでアクティブにすることができます。
- アクティブなドキュメントが最大化されると 「最小化」「元に戻す」、「閉じる」ボタンがメニューバーの右側に配置されます。ドキュメントが重ねて表示、または最小化されると「最大化」、「元に戻す」、「閉じる」ボタンがドキュメントウィンドウのタイトルバーに配置されます。
- 1つのファイルを最大化すると開かれている他のファイルも全て最大化されます
- <u>ウィンドウ</u> メニューのコマンドを使用することで、開かれているドキュメントを重ねて、または上下 (左右)に並べて表示することができます。
- 開かれているファイルは、「Ctrl+Tab」または「Ctrl+F6」により、開かれた順序でアクティブに切り 替えることができます。
- ドキュメントタブを右 クリックすることで表示 されるコンテキストメニューから、 印刷」や 閉じる」と いったコマンドを実行 することができます。

メインウィンドウ内 のビュー

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

利用可能なビューは編集またはブラウザビューです:

- <u>Authentic</u> ビュー: StyleVision Power Stylesheet をベースにした XML ドキュメントの編集 をグラ フィカルなインターフェイスで行ないます。
- **ブラウザビュー**: CSS ならびに XSL スタイルシートをサポートする統合 されたブラウザビューです。
- **メモ:** 各ファイル拡張子のデフォル ビューは、「<u>ツール」オプション</u>」ダイアログの、ファイルタブにある デフォル ビューペインにてカスタマイズすることができます。

1.1.2 プロジェクトウィンドウ

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



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

プロジェクトの操作

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

- プロジェクトウィントウで同時に開くことのできるプロジェクトは1つだけです。新規プロジェクトが 作成、または既存のプロジェクトが開かれると、現在プロジェクトウィントウにて開かれているプ ロジェクトは置き換えられます。
- プロジェクトに変更がなされた場合、そのプロジェクトを保存する必要があります。プロジェクトの保存は「プロジェクト| プロジェクトを保存」メニューコマンドにより行うことができます。
- プロジェクHはフォルダー、ファイル、その他のリソースなどにより構成されるツリー構造により 表示されます。リソースを追加する場所や階層に制限はありません。

- プロジェクトフォルダーは、論理的にファイルのグループ化を行う論理フォルダーです。ハードディスク上にある階層構造などに対応する必要はありません。
- フォルダーは、実際のファイルシステムに対して、直接対応するように作成することもできます。このようなフォルダーは外部フォルダーと呼ばれ、プロジェクトウィンドウでは黄色のフォルダーアイコンにより表示されます(通常のプロジェクトフォルダーは緑色のフォルダーアイコンで表示されます)。外部フォルダーは、更新コマンドにより、明示的に同期を行う必要があります。
- フォルダー内に収められるファイルの種類やその組み合わせに制限はありません。また、 フォルダーのプロパティダイアログから各フォルダーにファイルの拡張子を定義することで、共通するファイルを1つの場所に収めることができます。ファイルが親フォルダーに追加されると、そのファイル拡張子が定義されたサプフォルダーに、ファイルが自動的に追加されます。
- プロジェクドウィンドウでは、フォルダーをドラッグすることで、他のフォルダーや、同し親フォルダー内の異なる場所に、そのフォルダーを移動することができます。ファイルをドラッグすることで他のフォルダーに移動することはできますが、フォルダー内部のファイルはアルファベット順に表示されており、同じフォルダー内での移動を行うことはできません。また、Windows エクスプローラーなどからファイルやフォルダーをプロジェクドウィンドウにドラッグすることで、プロジェクトへの追加を行うこともできます。
- 各フォルダーには、フォルダーのプロパティダイアログにて定義されるプロパティがあります。これらのプロパティには、フォルダーのファイル拡張子、XML ファイルの検証を行うスキーマファイル、XML ファイルの変換を行うXSLT ファイルなどが記述されます。
- フォルダー内にあるファイルのバッチ処理を行うには、フォルダーを右クリックし、コンテキストメニューから選択することのできるコマンドを選択します(例:XMLの検証または整形式のチェック)。
- メモ: プロジェクトウィンドウの表示はウィンドウメニューから切り替えることができます。

1.1.3 情報ウィンドウ

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

情報	τ×
Element Datatype Pattern white Space	Phone string [0-9 \-]* preserve
情報 スキ	-7

情報ウィンドウの表示はウィンドウメニューから切り替えることができます。

1.1.4 入力ヘルパー

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

以下の点に留意してください:

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

1.1.5 メニューバー、ツールバー、ステータスバー

メニューバー

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

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

ツールバー

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

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

ステータスバー

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

1.2 アプリケーション環境

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

このセクションは以下のように構成されます:

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

1.2.1 設定、カスタマイズ

このセクションは、以下のように構成されます:

- <u>設定</u>
- <u>カスタマイズ</u>

設定

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

カスタマイズ

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

カスタマイズコマンドに関する詳細はユーザーレファレンスのセクションを参照ください。

1.2.2 チュートリアル、プロジェクト、サンプル

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

チュートリアル、プロジェクト、サンプルファイルの場所

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

C:\Documents and Settings**<username>\My Documents** Altova\Authentic2018\AuthenticExamples\

The My Documents Altova Authentic 2018 フォルダーは、PC 上で登録されたユーザーごとに、ユーザーの <username> フォルダー以下 にインストールされます。従って、このインストールシステムでは、各ユーザーが、それぞれの場所 に Authentic Examples フォルダーを持つことになります。

マスター Authentic Desktop フォルダーに関 する注意

Authentic Desktop がインストールされると 以下の場所に、マスター Altova\Authentic2018 フォ ルダーが次の場所に作成されます:

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

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

チュートリアル、プロジェクト、サンプルファイルの場所

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

1.2.3 Authentic Desktop 製品機能とAltova 製品

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

Authentic Desktop 機能の一覧

Altova Web サイトには <u>Authentic Desktop</u> に含まれている最新機能の一覧などが掲載されています。

Authentic Desktop ヘルプ

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

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

サポートオプション

ユーザマニュアル (本 ドキュメンテーション)には記載 されていない情報や、Altova 製品に関するお問い合わせは、Altova Web サイトの <u>サポートセンター</u> をご利用 ください。以下のような機能がご利用 になれます:

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

Altova 製品

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

2 Authentic View チュートリアル

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

Nanonull	, Inc.		
Location: 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. Sin nanoelectronic	ce 1996, Nanonull has been actively i c software technologies. It released th	nvolved i e first ver	n developing sion of its acclaimed
NanoSoft De	<i>velopment Suite</i> in February 1999	Also in 19	999, Nanonull increased

its capital base with invesment from a consortium of private investment firms. The company has been expanding rapidly ever since.

Authentic View 内 で XML ドキュメントを編集 するには2つのユーザーアクションが含まれます: (i) ドキュメントの構造の編集 (例えば、段落 とヘッドラインなどのドキュメントの一部を追加、または、削除するなど)、および (ii) データの挿入 (ドキュメントの一部のコンテンツ).

このチュートリアルでは次のステップについて説明されています:

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

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

チュートリアルの必要条件

チュートリアルで必要な全てのファイルは Altova アプリケーションの Exam ples フォルダー内にあります。 これらのファイルは、以下のとおりです:

- Nanonullorg xm 1(XML ドキュメント)
- NanonulDrg.sps (XML ドキュメントがリンクされている StyleVision Power Stylesheet)
- NanonulDrgxsd (XML ドキュメントとStyleVision Power Stylesheet がベースにされ、リンクされている XML スキーマ)
- nanonullgif and Altova right 300 gif (チュートリアルで使用 されるイメージ)

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

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

2.1 AUTH-VIEW内でXMLドキュメントを開く

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

既存の XML ドキュメントを開く

NanonulD rg xm 1というファイルを開きます。Altova アプリケーションの Exam ples フォルダー内にあり、以下の2つの方法により NanonulD rg xm 1を開くことができます:

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

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



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

SPS をベースにした新規の XML ドキュメント

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

「ファイル | 新規作成」を使用する方法

- 1. 「ファイル | 新規作成」を選択し、「新規のドキュメントの作成」 ダイアログ内で、XML を 作成 される新規のファイル型 として選択します。
- 2. 「STYLEVISION スタイルシートを選択する」 をクリックして、希望する SPS を参照します。

「Authentic | 新規のドキュメント」を使用する方法

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

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

2.2 Authentic View インターフェイス

Authentic View 編集 インターフェイスはドキュメントデータを入力 し編集 するメインウィンドウとつの入力 ヘルパーにより構成 されています。ドキュメントの編集 は簡単 です。ドキュメントのマークアップを確認 する場合、マークアップタグをオンに切り替え、ドキュメントのコンテンツの入力を開始 します。ドキュ メントの構造を変更 するには、コンテキストメニュー、または、要素入力 ヘルパーを使用 します。

XML ノードタグ (ドキュメントマークアップ)の表示 XML ドキュメントは、基本的には、ノードの階層構造です。例:

<ドコントRoot> <Person id="ABC 001"> <Nam e>A pha Beta</Nam e> <Address>Som e Address</Address> <TeD1234567</TeD </Person> </ドコントRoot>

デフォルトでは、 ノードタグ Authentic View 内では表示 されていません。 メニューアイテム

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

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

入力ヘルパー

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

Elements	μ×
Show XML tree	
() OrgChart	
Office	
Desc	
🚺 para	
() italic	
tt Italic	
tt Italic	
Attributes	ąх
italic	•
xsi:type	
Entities	ąΧ
Ent amp	&
Ent apos	·
Ent gt	>
Ent quot	с. н
terre almost	

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

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

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

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

コンテキストメニュー

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

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

2.3 ノードを操作する

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

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

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

ノードの操作の実行

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

要素の挿入

要素は次の場所に挿入することができます:

- 要素 ノード内のカーソルの場所に挿入することができます。コンテキストメニューの「挿入」 コマンドのサブメニュー内にその場所に挿入することのできる要素がリストされます。要素入 カヘルパー内では、その場所に挿入することのできる要素は ¹⁰ アイコンにより表示されま す。Nanonullorg.xml ドキュメント内では、para 要素内にカーソルを置き、コンテキストメニューと要素入力ヘルパーを使用して bold とitalic 要素を作成します。
- スキーマによい許可されている場合、選択された要素、または、その祖先の前、または、後に挿入することができます。必要とされる要素をロールアウトされるサプメニューから選択します。要素入力ヘルパー内では、選択された要素の前、または、後に挿入することのできる要素は 3 および 3 アイコンによい表示されます。要素入力ヘルパー内では、選択された要素の前後のみに要素を挿入することができます。祖先要素の前後に要素を挿入することはできません。para要素と社員をリストするテーブル内にカーソルを置いてこのコマンドを試してください。

ノードリンクの追加

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



要素のコンテンツモデルが曖昧な場合、例えば、子要素のシーケンスを指定する場合、順序に決ま リはありませんので <u>add...</u>が表示されます。ノード名が指定されていないことに注意してください。 リンクをクリックすると、有効に挿入することのできる要素のリストがポップアップします。

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

Enter キーを使用して新規要素を作成する

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

要素の適用

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

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

要素の削除

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

要素の作成

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

する要素は 🔛 アイコン(挿入ポイント選択)と 🔛 アイコン(範囲の選択)により示されます。 Nanonullorg.xml ドキュメント内で、(複合型 コンテンツを持つ)paraの bold とitalic 子要素を 使用して、クリアメカニズムを試してください。

テーブルとテーブルの構造

Authentic View テーブルには2つの型が存在します:

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

2.4 Authentic View 内にデータを挿入する

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

テキストコンテンツの追加

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

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



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

特別文字とエンティティの挿入

データの挿入する場合、次のコンテンツの型は特定の方法で扱われます:

- 特殊文字は、XML マークアップ(アンパサンド、アポストロフィ より大記号 (>)、より小記号 (<)、および引用符)のために使用されます。これらの文字は内蔵のエンティティ として使用することができ、エンティティの入力ヘルパー内の対応するエンティティをダブルクリックすることによりドキュメント内に入力することができます。(例えば、プログラムコードリスティング内では) CDATA セクション内に挿入することができます。CDATA セクションを入力するには、CDATA セクションを入力する場所を右クリックして、コンテキストメニューから「CDATA セクションの挿入」を選択します。XML プロセッサーは CDATA セクション内の全てのマークアップ文字を無視します。CDATA セクション内で特殊文字を使用する場合、エンティティの参照ではなくその文字を入力します。
- キーボードを使用して入力できない特殊文字はシステムの文字マップからコピーういてドキュメント内の必要とする場所に貼り付けます。
- 頻繁に使用される文字列は、エンティティの入力へルパー内に表示されるをエンティティとして定義することができます。エンティティは必要な場所にカーソルをポイントとして、入力へルパーとして挿入することができます。テキスト文字列の値は1つの場所に保管されるため、これは管理のために役にたちます。値を変更する場合、エンティティの定義を変更するだけで行うことができます。

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

上記のテキストコンテンツの各種類の使用法を試してください。

データ入力 デバイスを使用 してコンテンツを追加 する方法

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

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

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

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

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

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

<Address> <postreet>1190akstreet,Suite 4876</postreet> <pocity>Vereno</pocity>

>po state>DC</po state>

<po zip>29213</po zip> </Address>

コンボボックスの選択 DC は DC の値にマップされていることに注意してください。 Zip フィールドの値は、 i_{20} zp 要素のコンテンツとして直接入力されます。

2.5 属性の値を入力する

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

- 属性が値を受け入れるように作成されている場合、メインウィンドウ内のコンテンツとして入力する
- 属性入力ヘルパー内で入力する

メインウィンドウ内の属性の値

属性の値をノーマルなテキスト、または、入力フィールド内にテキストとして、または、XML値にマップ されるユーザー選択として入力することができます。これらは要素のコンテンツが入力さえる方法と同 様に入力されます:次を参照してください: <u>Authentic View内のデータの挿入</u>。このような場合、要 素のコンテンツと属性の値が StyleVision Power Stylesheet により区別され、データは適切に処理さ れます。

属性入力ヘルパー内の属性の値

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

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

Attributes ———	× ×
CompanyLogo	
href	Altova_right_300.gif
xsitype	

Nanonull ロゴが Altova ロゴにより置き換えられます。

メモ: 属性入力へルパー内にエンティティを入力することはできません。

2.6 エンティティを入力する

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

Entities	× *
Ent amp	&
Ent apos	1
Ent gt	>
Ent It	<
Ent quot	н

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

NanonulD rg zm 1内 で、Joe Martin (Marketing) のタイトルを Marketing Manager Europe & Asia に変更してください。以下を行ってください:

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

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

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

自身のエンティティの定義

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

2.7 ドキュメントを印刷する

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

NanonulDrg xm 1を印刷するには以下を行います:

- 1. まだ行っていない場合は、マークアップモードを非表示にするに切り替えます。マークアップを 印刷しない場合は、これを行う必要があります。
- 2. 「ファイル | 印刷をプレビュー」を選択して、全てのページをプレビューします。50%に縮小 された印刷のプレビューページが下に表示されています。

Organiza	tion Chart		
Location of	logo: Alteva_right_300.gif		
Nanonull, In	с.		
Location: 🗉	15 N		
Street:	119 Cakateri, Sate 4376	Phone:	+1 (221) 555 5155 0
City:	Vereno	Fax	+1 (221) 555 5155 4
	20 8 25219	E-	and the second second

ページの書式設定は Authentic View と同じであることに注意してください。 3. ファイルを印刷するには、「ファイル | 印刷」をクリックします。

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

3 Authentic View インターフェイス

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

このセクションでは以下について説明されています:

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

Authentic View 情報の追加 ソースは以下のとおりです:

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

3.1 GUI の概要

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

メニューバー		
ツールバー		
プロジェクト ウィンドウ	メインウィンドウ	入力 ヘルパー
ステータスノ	(-:現在選択されているノードへの	D XPath

メニューバー

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

ツールバー

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

プロジェクトウィンドウ

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

メインウィンドウ

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

入力ヘルパー

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

ステータスバー

ステータスバーは現在選択されているノードへの XPath を表示します。

コンテキストメニュー

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

3.2 Authentic View ツールパーアイコン

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

XML マークアップの表示 / 非表示

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



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

動的なテーブル構造の編集

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

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

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



(C) 2018 Altova GmbH



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

XML テーブルの作成 と編集

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

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

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

テキストのフォーマット アイコン

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

DB行のナビゲーションアイコン

8

このアイコンはクエリを入力することのできる「データベースクエリの変更」 ダイアログを開きます。 Authentic View はクエリされたレコードを表示します。

XML データベースの編 集

ポータブルな XML 書式 (PXF) ツールバー ボタン

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

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

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

3.3 Authentic View メインウィンドウ

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

大きなマークアップ

タグ内に要素属性名を持つ要素と属性の開始と終了タグを表示します:

Department>Name>
Enter the department
Name Person

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

Department Name Person

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

= country USA = country

小さなマークアップ

名前の無い要素属性の開始と終了タグを表示します:
Inc.		
]		
	Phone:	∞+1 (321) 555 5155 04
119 Oakstreet, Suite 4876⊄	Fax:	1 (321) 555 5155 4⊲
⊛Vereno⊲	E-mail:	office@nanonull.com
DC ▼ 4 ⊕ 29213 4		<u>_</u> .
	Inc.⊲ ■ 119 Oakstreet, Suite 4876⊲ ■ Vereno⊲ ■ DC ▼ ⊲ ■ 29213 ⊲	Inc. ⊲ Phone: Phone: Fax: E-mail: DC ▼ ⊲ © 29213 ⊲

The company was established **D** 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 **D** 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.

۵0>

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

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

Image: Summary: 4 departments, 15 employees.

複合型 マークアップ

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

すべての マークアップを非表示にする

すべての XML マークアップは各 されています。Authentic View 内 で確認 することのできる書 式 設 定は、印刷 されたドキュメントの書 式 であるため、この表 示 モードはドキュメントの WYSIWYG 表示 で

す。

コンテンツの表示

Authentic View内では、コンテンツは2つの方法で表示されています:

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

Department>Name> Enter the department (Name Person>

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

documentid documentid

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

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

任意のノード

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

add synopsis

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

3.4 Authentic View エントリヘルパー

Authentic View内には3つの入力へルパーが存在します:要素、属性、および、エンティティ。 Authentic View インターフェイスの右下のウィンドウとしてこれらは表示されています(下のスクリーンショットを参照してください)。

Elements		д	×
Show XML tree			
() OrgChart			
Office			
Desc			
🕐 para			
iiii () italic			
'E italic			_
記 italic			
×□ italic			
{) italic			
AIB bold			
A <u>l</u> B italic			
() italic			
Attributes		<u></u> д	×
italic			•
xsi:type			
T-Min-		-	~
Entities		4	_
Ent amp	&		
Ent of	>		
Ent It	<		
Ent quot	u		
1			

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

それぞれの入力ヘルパーの説明に関しては下で説明されています。

要素入力ヘルパー

要素入力ヘルパーは2つの部分から構成されています:

• 上の部分には、 XML ツリーを表示」チェックボックスを使用してオンとオフを切り替えること

のできる XML ツリーが含まれています。 XML ツリーは、現在の要素のドキュメントのルート 要素までの祖先を表示しています。 XML ツリー内の要素をクリックすると その要素に対応 する要素 (このリストの次のアイテム内で説明されているとおり) は要素入力 ヘルパーの下の 部分に表示されます。

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

入力ヘルパーからノードを使用するには、アイコンをクリックします。

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

 要素の前に挿入

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

要 素 の削 除

×□

яĮв

AB

要素とコンテンツを削除します。

要素の挿入

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

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

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

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

要素の適用

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

• 適用された要素が元の要素の子と同じ名前を持つ子要素を持つ場合、また、元の要素内でこの子要素のインスタンスが存在する場合、オリジナルの子

€**__**}

(...)

要素が新規の要素のコンテンツ内で保持されます。

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

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

要素をクリアする (範囲が選択されている場合) このアイコンは、複合型 コンテンツの要素内 のテキストが選択されると表示されます。 アイコンをクリックすると選択されたテキストの範囲の周 りから要素をクリアします。

要素をクリアする (挿入ポイントが選択されている場合) 複合型の子 コンテンツ要素である要素内にカーソルが置かれるとこのアイコンが表示されます。 アイコンをクリックするとインラインの要素 がクリアされます。

属性入力ヘルパー

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

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

Attributes	φ×
para	•
Dara	
Desc	
OrgChart	
<u> </u>	

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

Attributes	ά×
para	•
xsi:type	
Entities	Ψ×

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

nillable 要素が選択された場合属性入力へルパー内に表示されるxsi:nil属性の場合、 xsi:nil 属性の値は属性の値のためのドロップダウンリストから許可されている値(true、または、false)の1つを選択して入力することができます。

xsi:type 属性を属性の値フィールド内をクリックして変更することができ、ドロップダウンリストからリストされた値を選択することができます。 リストされた値は、Authentic View ドキュメントがベースとされる XML スキーマ内で定義 された使用することのできる 抽象型です。

エンティティの入力ヘルパー

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

Entities		д×
Ent amp	&	
Ent apos	· ·	
Ent gt	>	
Ent It	<	
Ent quot	"	

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

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

3.5 Authentic View コンテキストメニュー

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

要素の挿入

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

Insert	►		bold
Insert <u>b</u> efore	•		italic
Insert <u>a</u> fter	•	it sa	les are

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

Inser	: •	em	пона	CC	mso	nuum o.
Inser	: <u>b</u> efore 🔹 🕨		para	۲		para
Inser	: <u>a</u> fter 🕨 🕨		Office	۲	re re	estricted

ノードの挿入、置換え、(**適用**)、およびマークアップの削除 (**クリア**) コマンドはコンテキストメニュー 内、および <u>Authentic View</u> 入力へルパー</u>内で使用することができそれぞれのセクションで説明されて います。

エンティティの挿入

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

CDATA セクションの挿入

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

ノードの削除

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

クリア

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

適用

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

コピー、切り取り、貼り付け

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

貼り付け

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

- マークアップタグと共 にノード全体 が Authentic View 内 でハイライトされ、 クリップボード にコピーされます。(i) ノードはこのノードが有効な場所で XML として解析 されます。 無効 な場所では解析 されません(ii) ノードがテキストとして解析 されると (マークアップではなく)ノ ードの テキストコンテンツのみが貼 リ付けられます。 テキストを貼り付けることのできる XML ドキュメント内 にテキストコンテンツを貼り付けることができます。
- テキストフラグメント が Authentic View 内でハイライトされ、クリップボードにコピーされます。(i)フラグメントが貼 リ付けられる場所で XML ノードが有効な場合のみ、このフラグメントが XML として貼 リ付けられる場合、テキストの XML マークアップタグは、テキストフラグメント と共に明示的にコピーされず、テキストと共に貼 リ付けられます。(ii)フラグメントが テキストと して貼 リ付けられる場合、テキストが貼 リ付けられる XML ドキュメント内の場所に貼 リ付け られます。
- **メモ:** テキストは、テキストが許可されているノードにコピーされます。コピーされたテキストがドキュ メントを無効化しないように注意してください。コピーされたテキストは以下のようになります:

(i)新規の場所で構文的に有効な場合(例えば、数字のノード内の数字以外の文字が無効になります)。

(ii) 以外はノードを無効化する場合(例えば、3つの桁のみ受け入れるノード内の4つの桁

はノードを無効化します)。

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

削除

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

4 Authentic View 内で編集する

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

セクションでは以下について説明されています:

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

4.1 基本的な編集

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

Vereno Office Summary: 4 departments, 16 employees.

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

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

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

The company was established bold in Vereno in 1995 bold 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 *talic NanoSoft Development Suite talic* 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



ノードの挿入

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

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

Set xsi:type
The element Publication is defined with an abstract type PublicationType which cannot be used directly in XML documents. Therefore, the built-in xsi:type attribute must be set to one of the following allowed derived types for the XML document to be valid.
BookType
MagazineType
OK Cancel

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

<Publication xsitype="BookType"> ... </Publication> <Publication xsitype="MagazineType"> ... </Publication>

<Publication xsitype="MagazineType"> ... </Publication>

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

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

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

テキストの編集

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

XML またはテキストとしてコピーする

XML、または、テキストとしてコピーし貼り付けることができます。

- テキストが XML として解析 されると XML マークアップは、ノードのテキスト コンテンツと共に 解析 されます。XML マークアップは、ノードのコンテンツの一部 がコピーされた場合でも、貼り 付けられます。マークアップを貼り付けるには、貼り付ける場所でスキーマに従い、貼り付け が許可 されている必要 があります。
- テキストがテキストとして解析されると XML マークアップ破壊席されません。

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

更 に、 ハイライトされテキストを貼 り付 ける箇 所 にドラッグします。 テキストがドロップされるとポップアッ プが表示 され、 テキストをテキストまたは、 XML として貼 り付 けるかが問 われます。 希望 するオプション を選 択 します。

テキストのフォーマット

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

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

Authentic View 内で RichEdit を使用する

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



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

エンティティの挿入

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

Entities		τ, Χ
Ent amp	&	
Ent apos	1	
Ent gt	>	
Ent It	<	
Ent quot	"	

XML はカスタムのエンティティを作成する機会を与えます。

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

CDATA セクションの挿入

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

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

リンクの編集

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

4.2 Authentic View 内のテーブル

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

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

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

4.2.1 SPS テーブル

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

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

Nanonull,	nc.	
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

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

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



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

Admini	stratior	า						
Firet	Lact	Title	Evt	⊑Mail	Sharee	Leave		
FIISL	Lasi	The	EXL	Liviali	Shares	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	1.matise@nanonull.com	add Shares	25	7	18
Employees: 3 (20% of Office, 9% of Company) Shares: 1500 (13% of Office, 6% of Company)				6 of				
Non-Shar	eholders	: Frank Furthe	er, Loby	y Matise.				

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

4.2.2 CALS/ HTML テーブル

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

テーブルの挿入

CALS/HTML テーブルを挿入するには、以下を行います:

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

Insert Table
Select an XML table from the following list:
informattable (CALS)
informattable (HTML)
table (CALS)
table (HTML)
OK Cancel

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

Insert Table	×
Columns: 📴 🔺 Rows: 3 🔺	ОК
Use the whole available width	Cancel

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

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

書式設定テーブルとデータの挿入

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

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

テーブル、行、列、または、セルの書式設定を行うには:

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

Table Row Column	Cell	
align		UK
- Sec.		
bgcolor	✓ ⁽¹⁾	Cancel
border 1		
border-collapse		
cellpadding		
cellspacing		
frame		
height		
page-break-after		
page-break-before		
rules		
table-layout	· .	
1 • 64		

2. セルペースとセルパディング プロパティを ¹0」に設定します。テーブルは以下のようになります:

3. 最初の行にカーソルをポイントし、 20 (テーブルプロパティ) アイコンをクリックします。 1行」タ ブをクリックします。

Ta	ble Properties		×
	Table Row Column	Cell	ОК
	align	center 💌 🔺	
	bgcolor	#B2B2B2 📰 💌 🕎	Cancel
	height		
	valign	middle 💌	
	width		
-	7		

最初の行がヘッダー行になるため、この行を他の行と区別するために背景色を設定しま

す。上の図で行プロパティが設定されていることに注意してください。列ヘッダーテキストを挿入します。テーブルは以下のようになります:

Name	Telephone	Email

配置は指定されているとおり中央に揃えられています。

4. 「Telephone」列をサブ列「Office」と Home」に分割する場合、「Telephone」列の幅を 水平2つの列に分割する必要があります。最初にしかしながら、サブヘッダー行を作成する

ために ヘッダーセルの垂直に分割します。「Telephone」 セル内 にカーソルを置き、 🕮 (垂直に分割) アイコンをクリックします。 テーブルは以下 のようになります:

Name	Telephone	Email

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

Nome	Telephone		Francil
INAILIE	Office	Home	Exitiali

「Telephone」 列内の各セルの幅を水平に分割します。

テーブル編集 アイコンを使用して、列どうを追加、削除、垂直に整列することもできます。 CALS/ HTML テーブル編集 アイコンは、CALS/HTML テーブル編集 アイコン内で説明 されています。

テーブル内 のセルを移動する

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

セルにデータを挿入する

セルにデータを入力するには、セル内にカーソルを置き、データを入力します。

テキストの書式設定

他のテキストXML ドキュメント内同様、CALS/HTML テーブル内のテキストは、 XML 要素、または、

属性を使用して書式設定される必要があります。要素を追加するには、テキストをハイライトし、 要素入力ヘルパー内の必要とされる要素をダブルクリックします。属性の値を指定するには、テキス トフラグメント内にカーソルを置き、属性入力ヘルパー内に必要とされる属性の値を挿入します。書 式設定 ヘッダーテキストを太字に設定すると、テーブルは以下のようになります。

Nome		hone	Fmail
Ivane	Office	Home	Eman

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

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

4.2.3 CALS/ HTML テーブル編集アイコン

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

CALS/HTML テーブルをいつどのように使用するかについての詳細は、CALS/HTML テーブル。

テーブルの挿入

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

テーブルの削除

X

「テーブルの削除」コマンドは現在 アクティブなテーブルを削除します。

行の追加



行の追加」コマンドは現在アクティブなテーブルの最後に行を追加します。

列 の追 加

₽₽

₩

列の追加」コマンドは現在アクティブなテーブルの最後に列を追加します。

行の挿入

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

列 の追 加

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

セルを左側 にジョインする

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

セルを右側にジョインする

) Join

「セルを右側にジョインする」 コマンドは、現在のセル(現在のカーソルの場所)を右側のセル とジョインします。両方のセルのコンテンツは新規のセルに連結されます。

下のセルとジョインする



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

上のセルとジョインする

「上のセルとジョインする」 コマンドは、現在のセル(現在のカーソルの場所)を上のセルとジョ インします。 両方のセルのコンテンツは新規のセルに連結されます。

セルを左右に分割する



join

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

セルを上下に分割する

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

上揃え

₹

このコマンドはセルコンテンツをセルの上に揃えます。

上下中央揃え

¥II€

このコマンドは、セルコンテンツを上下中央に揃えます。

下揃え

	Ŷ	
Ľ	_	'

このコマンドはセルコンテンツをセルの下に揃えます。

テーブルプロパティ

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

Table Properties		×
Table Row Column	Cell	ОК
align		
bgcolor	✓ ⁽¹⁾	Cancel
border	1	
border-collapse	•	
cellpadding	•	
cellspacing	•	
frame	_	
height		
page-break-after		
page-break-before		
rules		
table-layout		
L ena		

^{*}

4.3 DB の編集

Authentic View 内 で、データベース (DB) テーブルを編集 し、データを DB に戻 すことができます。この セクションには、DB テーブルを編集 する際 に使用 することのできるインターフェイスの説 明 が含 まれて います。次 の一般的 な点 について注意 して ください:

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

Authentic View内でDBベースのStyleVision Power Stylesheetを開く方法:

- 「Authentic | データベースデータの編集」をクリックして、必要とされる StyleVision Power Stylesheet を参照します。
- **メモ:** Authentic View 内では、SQLite データベースからのデータは編集することができません。 Authentic View からsave SQLite データを保存しようと試みるとメッセージボックスにより既知の制限が通知されます。

4.3.1 DB テーブル内をナビゲートする

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



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

Go To Record	
Go to record #: 1	ОК
	Cancel

DB テーブルをナビゲートするには、必要なボタンを押してください。

XML データベース

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

4.3.2 DB クエリ

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

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

クエリを作成して送信する方法:

1. データベースクエリの変更ダイアログを開くために必要とされるテーブルのためのクエリボタン

をクリックします(スクリーンショットを参照してください)。各DBテーブルの上、または、下にこのボタンが表示されます。クエリボタンがテーブルに存在しない場合、StyleVision Power Stylesheet のデザイナーはそのテーブルに対してDBクエリ機能を有効化していません。

Edit Database Query				
'\$' as first character in value signifies parameter.				
	¢			
	₽			
	Append AND			
	Append OR			
	Delete			
	Parameters			
OK Cancel				

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

Edit Database Query '\$' as first character in value signifies parameter.				
State = CustomerNr City State Street ZIP V	Append AND Append OR Delete Parameters			
OK Cancel				

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

条件内の式

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

等しい記号
等しくない記号
より小記号
より小記号、または、等しい記号
より大記号
より大記号、または、等しい記号
発音が似ている
発音が似ていない
空である
空ではない

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

DB クエリを持 つパラメーターの使用

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

In this dialog parameter r	ames are shown without leading '\$'.		
Parameter Name	Default Value		
price	10.00		
state	CA		
date	2004-06-24		
ОК			

SPS とパラメーター内 でスタイルシートのために定義 されている **すべての** パラメーターを含むパラメーターのビュー ダイアログはスタイルシートのデザイン内 で編集 される必要 があります。

DB クエリ内の必要条件を再度並べ替える

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

Edit Database Quer y		
'\$' as first character in value signifie:	s parameter.	
State =	 CA Los Angeles 	<u>ት</u>
City • • • • • • • • • • • • • • • • • • •	= 💌 San Diego	Append AND
City City CustomerNr	= San Francisco = 25	Append OR
		Delete
		Parameters
ОК	Cancel	

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

State=CA AND (City=Los Angeles OR City=San Diego OR (City=San Francisco AND Custom enNr=25))

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

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

次の点について注意してください:

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

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

DB クエリの変更の方法

DB クエリの変更方法:

1. クエリボタン 2 をクリックします。 データベースクエリの変更 ダイアログボックスが開かれます。 式の条件内の式を編集し、新規の条件を追加し、条件を並べ替え、DB クエリ内の

条件を削除することができます。

2. 「OK」をクリックします。DBからのデータは、DBクエリへの変更が反映されるとAuthentic View に自動的に再ロードされます。

4.3.3 DB テーブルの変更

レコードを追加する方法

DB テーブルにレコードを追加する方法:

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

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

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

レコードを変更する方法

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

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

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

メモ:

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

レコードを削除する方法

レコードの削除の方法:

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

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

4.4 日付と作業する

Authentic View 内 で日 付 を編集 するには2つの方法 があります:

- 日付の選択を使用して日付が入力、または、変更されます。
- <u>値を入力して</u>日付が入力、または、変更されます。

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

日付の書式に関するメモ

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

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

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

Organization Chart				
Location of logo: nanonull.gif Last Updated: 2003-09-01				

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

Location of logo: nanonull.gif						
Last Updated: 2003-09-01	September A 2003					
	M T W T F S S					
Nanonull, Inc.	8 9 10 11 12 13 14 37 15 16 17 18 19 20 21 38 22 23 24 25 26 27 28 39 29 30 1 2 3 4 5 40					
Location: US 🔻	Today No Timezone					

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

4.4.2 テキストのエントリ

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

メモ:日付を編集する際に、書式を変えないでください。

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

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

Invoice Number: 001 2006-03-32 Customer: ERROR: Invalid value for datatype date in element Invoice Art<mark>/InvoiceDate/</mark>

日付の書式を変更しようとすると日付は赤い色で表示されエラーを通知します(スクリーンショット を参照してください。

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

4.5 エンティティを定義する

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

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

ドキュメント内で使用することのできるエンティティには2つの広い型が存在します: XML データ(テキス ト文字列、または、XML ドキュメントのフラグメント)である「解析済みのエンティティ」、または、非 -XML データ (グラフィック、サウンド、または、マルチメディアオブジェクトなどの)バイナリファイルである 「解析されていないエンティティ」。各エンティティには名前と値が存在します。解析されたエンティ ティの場合、エンティティは XML データのためのプレースホルダです。エンティティの値は XML データ 自身、または、データを含む xm1ファイルをポイントする URI です。解析されていないエンティティの 場合、エンティティの値は非 XML データファイルをポイントする URI です。

エンティティを定義する方法:

8	Name		Туре	PUBLIC	Value/Path	NDATA		OK
a	nano_dc	•	Internal		Nanonull, Inc			
a	nano_eu	•	Internal		Nanonull Europe, AG			Cancel
	nano_ma	•	Internal		Nanonull Partners, Inc			
8	website	-	Internal		http://www.nanonull.com/			Append
	branches	•	SYSTEM		branches.xml		E	
8	logo	•	SYSTEM		nanonull.gif	 GIF		Insert
								Delete

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

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

- 5. Value/Path フィールド内に、次を入力することができます:
 - エンティティの型がInternalの場合、エンティティの値として使用するテキスト文字列を入力します。エントリに引用符を入力しないでください。入力する引用符はテキスト文字列の一部として扱われます。
 - SYSTEM がエンティティの型である場合、参照ボタンを使用してリソースのURI、または、 ローカルネットワーク上のリソースを入力します。解析されたデータがリソースに含まれてい る場合、XML ファイルである必要があります(すなわち、ファイルは xm 1拡張子を持つ必 要があります)。更に、リソースはGIF ファイルなどのバイナリファイルであることができます。
 - エンティティの型が PUBLIC の場合、このフィールドにシステム識別子を追加で入力する 必要があります。
- NDATA エントリはプロセッサーにこのエンティティを解析 せず適切 なプロセッサーに送信 される ように命令 します。NDATA フィールドはエンティティが解析 されていないエンティティを示す値 を含む必要 があります。

ダイアログの機能

エンティティの定義ダイアログ内で以下の機能を使用することができます:

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

エンティティの制限

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

4.6 XML 署名

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

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

XML 署名 アクション

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

- 証明書 パスワードの選択:署名は、証明書、または、パスワードで認証されます。署名が 作成され、検証される際に認証オブジェクト(証明書、または、パスワード)が必要とされま す。Authentic XMLドキュメントが署名が有効化されているSPSを割り当てられているかに より、デフォルトの証明書、または、署名のパスワードを指定する場合があります。デフォル トの証明書、または、パスワードが指定されているかにより、Authentic Viewユーザーが自 身の証明書とパスワードを選択できるかを決定する署名を構成することができます。XML署 名ダイアログ内でこの操作を行うことができます(下のスクリーンショット)。自身の証明書及 びパスワードの選択はデフォルトの証明書とパスワードをオーバーライドします。自身の証明 書とパスワードはメモリ内に保管され、現在のセッションで使用することができます。自身の 証明書とパスワードの選択の後、ファイルまたはアプリケーションを閉じます。SPSは証明 書とパスワードのためのデフォルトの設定に戻ります。
- ドキュメントの署名: Authentic XML ドキュメントを自動的に、または、手動で署名することができます。自動署名は、SPS デザイナーにより署名の構成内で指定することができ、保存されるとAuthentic XML ドキュメントが自動的に署名されます。自動署名が有効化されていない場合、ドキュメントを手動で署名することができます。XML 署名 ツールバーアイコ

ン) すべての くまたは、「Authentic | XML 署名 コマンドをクリックすると XML 署名 ダ イアログがポップアップされるので、(*上のスクリーンショット*) ドキュメントの署名 ボタンをクリッ クします。埋め込まれた署名を使用したドキュメントの署名はスキーマにルート(ドキュメント) 要素の最後の子要素として署名要素を許可するように要求します。ドキュメントを署名す る場合、認証オブジェクトと署名の配置は署名の構成に従い決定されます。認証情報に アクセスできることを確認してください。詳細に関しては、SPS デザイナーに確認してください。

 Authentic XML ドキュメントの検証: SPS のために XML 署名が有効化されていると 検証 プロセスは Authentic View XML ドキュメントがロードされる都度、各署名に対して行われま す。パスワード、または、証明書 キー 情報が SPS と署名と保存されていない場合、 Authentic View ユーザーはパスワードを入力、および、検証のための証明書を選択するよう にプロンプトされます。XML ファイルと共に保存しない場合、埋め込まれた署名が生成され ると、XML ファイルが保存される際に XML ファイルと共に保存されます。(XML 署名ダイアロ グの **署名を削除」**ボタンを使用して)生成された署名は明示的に削除される必要があり ます 同様に、デタッチされた署名が生成されると必要では内場合、明示的に削除されま す。

4.7 Authentic View 内のイメージ

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

イメージの書式の多数は、多くの出力書式でサポートされています(下のリストを参照してください)。

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

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

相対的なパス 相対的なパスは SPS ファイルに相対して解決されます。

4.8 Authentic View 内のキーストロール

Enter +-

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

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

キーボードの使用

入力とナビゲートのためにキーボードを通常の方法で使用することができます:

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

5 Authentic スクリプト

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

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

Altova 製 品	Authentic スクリプトの作 成	Authentic スクリプトの有 効 化
StyleVision Enterprise	は、	ここ
StyleVision Professional	は、	は、
StyleVision Standard *	しいえ	ししえ
XMLSpy Enterprise	ししえ	は、
XMLSpy Professional	ししえ	は、
XMLSpy Standard	ししえ	ししえ
AuthenticDesktop Enterprise	ししえ	は、
Authentic Browser Plug-in	ししえ	は、
Enterprise Trusted **		
Authentic Browser Plug-in Enterprise Untrusted	いえ	(JL)

* Authentic ビュー無し

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

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

Authentic スクリプト警告 ダイアログ

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

Authent	ic Script Warning					
⚠	The document you are about to open in Authentic view contains scripts. These may contain harmful code. Do not enable them unless you trust the source of this document.					
	Would you like to enable scripts for documents from this folder and subfolders? C:\Documents and Settings\My Documents\Examples\					
	You can define trusted locations and change the default behaviour for SPS containing Scripts via the menu Authentic Trusted Locations					
	Yes No					

以下のオプションから選択を行うことができます:

- はいをクリックするとファイルを含んでいるフォルダーがAuthentic スクリプトの信頼された場所リストへ追加されます。今後、信頼されたフォルダー内にふくまれるファイルをAuthentic ビューで開いても、警告ダイアログは表示されなくなります。信頼された場所のリストへは 「Authentic」信頼された場所」メニューコマンドからアクセスならびに編集することができます。
- いいえをクリックすることで、ファイルが含まれているフォルダーを信頼された場所のリストへ追加することなく処理が継続されます。ファイルはスクリプトが無効になった状態で開かれます。ファイルをAuthentic ビューで開くたびにAuthentic スクリプト警告ウィンドウが表示されます。ファイルが含まれているフォルダーを信頼された場所のリストへ追加するには、 「Authentic」信頼された場所」メニューコマンドから信頼された場所のダイアログを開き、フォルダーを追加ならびに必要に応じて変更することができます。

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

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

Authentic スクリプトの使用方法

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

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

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

Authentic スクリプトのドキュメンテーション

Authentic スクリプトに関する情報は StyleVision のドキュメンテーションでご利用になれます。Altova ウェブサイトの製品ドキュメンテーションページから参照することができます。

6 ブラウザービュー

ブラウザービューは、通常以下を閲覧するために使用されます:

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

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

Microsoft Internet Explorer および XSLT に関 する注意 点

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

ブラウザービュー 機能

次の機能がブラウザービューで使用することができます。 <mark>ブラウザーメニュー</mark>、ファイルメニュー、編 集メニュー</mark>を使用してアクセスすることができます。

- 個別のウィンドウで開く: ブラウザービュー が個別のウィンドウである場合、同じドキュメントの編集ビューで横に並べて表示することができます。このコマンドは、プラウザーメニュー内にあい、個別のブラウザービューウィンドウをタブされたウィンドウとして返す切い替えコマンドです。オプションダイアログのビュータブによりブラウザービューがデフォルトで個別のウィンドウで表示されるかを設定することができます。
- 前へおよび 戻る: コマンド ブラウザー コマンド は、ブラウザービュー内 でロードされたページを ナビゲートするために使用 されます。これらのコマンドは、**ブラウザー**メニュー内 にあります。
- フォントのサイズ: ブラウザー メニューコマンド を介して、調節 することができます。
- 停止、更新、印刷:標準 ブラウザー コマンドは、プラウザー および ファイル メニュー内 にあります。
- 検索: テキスト文字列の検索を有効化します。このコマンドは、編集 メニュー内にあります。

7 Altova グローバルリソース

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



Altova グローバルリソースの使用には、2 つのステップがあります:

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

その他の Altova 製品 におけるグローバルリソース

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

7.1 グローバル リソースの定義

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

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

Default	- 🛃 🚽
---------	-------

グローバル リソース定義 ファイル

グローバル リソースに関 する情 報 はグローバル リソース定 義 ファイルと呼 ばれる XML ファイルに収 められています。 このファイルは、 グローバル リソースの管 理 ダイアログにて初 めてグローバル リソースが定義 ならびに保存 された際 に作 成 されます (下のスクリーンショットを参照)。

🛃 グローバル リソースの管理	±	×
定義ファーイル: C:\Altova\Global	Resources.xml	▶ 参照(B)
日 伯 ファイル ① Alias	*	+ 追加(A)
ーー 「ヘフォルダー/ベース URL		
() データベース		🧱 削除(<u>D</u>)
	Ψ.	🔍 Ľ1–(V)
] •	•	
	OK	キャンセル

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

C:\Users\<username>\My Documents\Altova\GlobalResources.xml

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

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

- メモ: グローバル リソース定義 ファイルの名前は自由に変更することができ、Altova アプリケーションからアクセスできる任意の場所へ配置することができます。この場合、各アプリケーションにて(定義ファイルテキストボックスから)このファイルをアプリケーションのグローバル リソース定義ファイルとして指定する必要があります。複数の Altova 製品にて同一の定義ファイルを指定することで、これらの Altova 製品にてリソースが利用できるようになります。
- メモ: 複数のグローバル リソース定義ファイルを作成することもできます。この場合、各 Altova アプ リケーションにて同時に選択できるアクティブな定義ファイルは1つだけで、そのファイルの中 に含まれている定義だけがアプリケーションでは利用できるということに注意してください。その ため、必要に応じて同じリソースを複数の定義ファイルで記述する必要が生じることもありま す。

グローバルリソースの管理:追加、編集、削除、保存

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

グローバル リソースを追 加 するには、 追 加 」 ボタンをクリックして、表示 される 「グローバル リソー ス」 ダイアログにてグローバル リソースを定義 します (このセクション以下 のサブセクションにある 「ファ イル」、「フォルダー」、「データベース」 の記述 を参照 ください)。 グローバル リソースを定義 して (グロー バル リソースの管理 ダイアログにある 「OK」 ボタンをクリックすることで) 保存 すると 選択 されたグロ ーバル リソース定義 ファイルのグローバル リソース定義 ライブラリに、そのグローバル リソースが追 加 さ れます。 グローバル リソースはエイリアスにより職別 されます。

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

グローバル リソースを削除 するには、 削除」 ボタンをクリックします。

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

構成 によりグローバル リソースをエイリアス名 へ関 連付ける

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



Altova アプリケーションではファイルの代わりにエイリアスを割り当てることができます。アプリケーションの アクティブなグローバルリソース構成を切り替えることで、各エイリアスに割り当てられたリソースを切り 替えることができます。例えば Altova XMLSpy アプリケーションにて、MyXML.xml という名前の XML ドキュメントに対して XSLT 変換を行う場合、MyXSLT という名前のエイリアスを使用される XSLT とし てグローバルリソースに割り当てることができます。XMLSpy では、アクティブな構成を変更すること で、異なる XSLT ファイルを使用することができます。構成 4 にて First.xslt が MyXSLT へマッピングさ れており、構成 4 がアクティブな構成 として選択されている場合、First.xslt が変換に使用されます。 これにより、複数の構成を使用することで、単一のエイリアスから複数のリソースへアクセスすることが できるようになります。このメカニズムはリソースのテストや比較を行う際に便利な機能 となります。更 に、グローバルリソースは複数の Altova 製品間で使用することができるため、複数の Altova 製品に てリソースのテストや比較を行うことができます。

7.1.1 ファイル

「グローバル リソースの管理」ダイアログにある「追加 | ファイル」コマンドにより、ファイルのグローバル リソース ダイアログへアクセスすることができます(以下のスクリーンショットを参照)。 グローバル リソース ダイアログでは、リソース エイリアス テキスト ボックスにて入力 されたエイリアスの構成 を定義 する ことができます。以下に記 される方法により構成のプロパティを指定して、「OK」をクリックしてエイリア スの定義を保存して ください。

エイリアスの定義を保存した後には上記の操作を(<u>ゲローバルリソースの管理」</u>ダイアログにある ⁵追加 | ファイル」コマンドを選択するところから)再度行うことで、別のエイリアスを追加することができ ます。

グローバル リソース ダイアログ

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

エイリアス名 リソース エイリアス(A):	TypeQuery			
構成	「構成 "Default"の設定			
Default Long Short	 ● ファイル(F) ● MapForce 変換の結果(M) ● StyleVision 変換の結果(S) 			
	リソースは以下のファイルを指します: C:\Altova\XMLSpy\Examples\Tutorial\AddressFir: [25]			
	<u>名前</u> パス ^			
	▲			
	~			
	OK キャンセル			

グローバル リソース ダイアログのアイコン

÷

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

ます。

- 構成をコピーして追加: 選択された構成のコピーとして追加する構成の名前を入力する ための、構成の追加ダイアログが表示されます。
- 前除: 選択 された構成を削除 します。
- 彦 🛛 🛱 🔆 グローバル リソースとして作 成 されるファイルを参 照 します。

エイリアスの定義

以下の操作により、エイリアスの名前と構成を定義することができます:

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

Outputs (use radio button to select)						
	Name	Path	*			
۲	HTML output	6				
0	RTF output					
0	PDF output	le l				
0	Word 2007+	ß				
			Ŧ			

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

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

MapForce 変換の結果

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

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

11. 15. W	→構成 "ConfigCust"の設定	
T7 103 🐡	() ファイル(F)	
Default	● MapEorce 変換の結果	(M)
Configuust		(C)
	Stylevision grading and	(12)
	リソースは以下の生成された出	出力を指します:
	C:\Altova\MapForce\M	apForceExamples\Customers_DB.mfd
	· · ·	
	入力問	
	名前	パス 🔺
	Customers	Customers.xml 🕞
	Altova_Hierarchic	Altova_Hierarchicalxml 🔗
		+
	出力(言約オポタ)を使って選択	R) (D)
	名前	1/2
	Text file	
	Oustomers	
	J	Ŧ

• .mfd (MapForce Design) ファイル:「リソースは以下の生成された出力を指します」テキス

トボックスにて、mfd ファイルを指定する必要があります。

- 入力データファイル: MFD ファイルが指定されるとそのファイルが解析され、中に含まれる入力スキーマ情報をベースにデフォルトのデータファイルが入力ペインに入力されます(上のスクリーンショットを参照)。他のファイルを指定することで、各入力スキーマに対するデフォルトファイルの選択を変更することができます。
- ・出力ファイル:MFDドキュメントに複数の出力スキーマが含まれる場合、出力ペインにそれらの出力すべてが表示されます(上のスクリーンショットを参照)。個々の出力スキーマに関連付けられた出力ファイルの場所がMFDドキュメントにて指定されている場合、その出力スキーマに対するファイルの位置が出力ペインに表示されます。上のスクリーンショットからは、Customers 出力スキーマにデフォルトのXMLデータファイル(CustomersOut.xml)が与えられており、Textファイル出力スキーマにはそのような割り当てが行われていないことが理解できます。出力ペインに表示あるデフォルトのファイルの場所を使用することができるほか、自分で指定することができます。MapForce 変換の結果が、選択された出力スキーマがある場所へ保存され、そのファイルがグローバルリソースとして使用されます。
- メモ: このオプション (MapForce の結果) により、グローバル リソースが呼び出された時に変換が行われるようになります。これで(入力 ファイルから得られた)最新のデータがグローバル リソースに含まれることが保証されます。
- **メモ:** 変換を行うためには MapForce が使用 されるため、この機能を使用 するには Altova MapForce をインストールする必要 があります。

StyleVision 変換の結果

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

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

構成	
+ + Default ConfigCust	構成 "ConfigCust" の設定 ○ ファイル(E) ○ MapForce 変換の結果(M) ④ StyleVision 変換の結果(S) リソースは以下の生成された出力を指します: C:\Altova\StyleVision\StyleVisionExamples\NanonullOrg.sps
	入力(!) 名前 パス XML C:¥Altova¥StyleVision¥StyleVisior (2)
	HTML出力 ○ HTML出力 ○ PDF出力 ○ Word 2007+出力 ○ Word 2007+出力 ○ Word 2007+出力

- .sps (SPS) ファイル: 「リソースは以下の生成された出力を指します」 テキスト ボックスに て、ファイルを指定する必要があります (上のスクリーンショットを参照)。
- 入力ファイル: 入力ファイルが SPS ファイル内部で既に指定されている場合、SPS ファイル が選択された段階で、そのファイルが入力ペインに表示されます。このエントリーは変更する ことができるほか、エントリーが指定されていない場合、追加する必要があります。
- 出力ファイル:出力ペインにて出力フォーマットを選択し、そのフォーマットに対する出力ファ イルの場所を指定します。
- メモ: このオプション (StyleVision の結果) により、グローバル リソースが呼び出された時に変換が行われるようになります。これで、(入力 ファイルから得られた) 最新のデータがグローバル リソースに含まれることが保証されます。
- **メモ:** 変換を行うためには StyleVision が使用 されるため、この機能を使用 するには Altova StyleVision をインストールする必要 があります。

7.1.2 フォルダー

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

🗿 グローバルリソース		×
I-1リアス名 リソース I-1リアス(A): Al	as1	
構成		
+ + *	- 構成 ''Test2013''の設定	
Test2013	C:\TestArea	

グローバル リソース ダイアログのアイコン

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

エイリアスの定義

以下の操作によりエイリアス(名前と構成)を定義することができます:

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

7.1.3 データペース

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

t 🗄 🕷	- 構成 "AlternativeD	'B''の設定 —			
Default AlternativeDB	🖳 データベー	🔍 データベースを選択(<u>C</u>)			
	データペース(D):				
	▽ 全般				
	接続ストリング		Data Source=C:¥Altova¥altova.mdb;Pr ovider=Microsoft.Jet.OLEDB.4.0		
	ルートオブジェクト		C:¥Altova¥altova.mdb		
	データベース種類	Ę.	MS Access		
	インポートの種類	Ę	ADO	-	
	■=== E				
	∇				
	DataSource	C:¥Alto	va¥altova.mdb		
	Catalog	altova		=	
	Provider	Microso	ft.Jet.OLEDB.4.0		
	JDBCDatabaseU	RL jdbc:od	c:;DRIVER=Microsoft Access Driver	(
	1DBCDriver	sun idh	odba IdbaOdbaDriver	Ψ.	
	•		4 11		

グローバル リソース ダイアログのアイコン

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

エイリアスの定義

以下の操作によりエイリアス(名前と構成)を定義することができます:

- 1. エイリアスに名前を付ける: リソースエイリアス テキスト ボックスにエイリアスの名前を入力します。
- 2. 構成を追加する:構成ペインには、デフォルトで(名前変更や削除を行うことができない) Default という名前の構成が表示されます。構成の追加または構成のコピーを追加アイコンをクリックして、表示されるダイアログにて構成名を入力することで、必要なだけの構成を追加することができます。構成の追加または構成のコピーを追加アイコンをクリックして、表示されるダイアログで構成の名前を入力することで、構成が追加されます。「OK」をクリックすることで新たな構成が構成ペインに表示されます。同じ作業を必要な数だけ繰り返してくださ

l I。

- 3. 構成のリソースとなるデータベースを選択する:構成ペインに表示された構成を選択し、「データベースを選択」ボタンをクリックします。グローバルリソース接続を作成ダイアログが表示されます。フォルダーにアクセスするためにセキュリティの資格条件が必要な場合、ユーザー名とパスワードフィールドを指定してください。
- 4. データベースへ接続 する: 接続 ウィザード、既存の接続、 ADO 接続、ODBC 接続、また は JDBC 接続を使用 して、データベースへの接続を作成します。
- ルートオブジェクトの選択: ルートオブジェクトを選択することのできるデータベースへ接続した場合、サーバー上にあるルートオブジェクトを選択するためのダイアログが表示されます (以下のスクリーンショットを参照)。ルートオブジェクトを選択して、「ルートオブジェクトを 設定」をクリックしてください。選択されたルートオブジェクトが、この構成が使用された時に ロードされるルートオブジェクトとなります。

ルート オブジェクトの選択	— ×
このデータベース データ ソース構成に対してルート オブジェクトを設定できます。 一度ルート オブジェクトを設定すると、変更することはできません。 設定をしないと、使用しているアプリケーションで定義しなければなりません。	
	•
ルートオブジェクトを設定(S) スキップ(K)	

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

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

7.2 グローバルリソースを使用する

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

使用するリソースの選択

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

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

7.2.1 ファイルとフォルダーの割り当て

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

Open	×
Global Resources:	
	*
CarOrders	
ManonullXML	
Contraction Contraction	
4	
, Switch to File Dialog Switch to URL Open C	ancel

グローバル リソースの管理: 「グローバル リソースの管理」ダイアログが表示 されます。

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

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

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

- <u>グローバルリソースを開く</u>
- <u>グローバルリソースとして保存</u>
- XSLT 変換
- <u>SPS の割り当て</u>

グローバルリソースを開く

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

グローバルリソースとして保存

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

Save As	×
Global Resources:	
Files MyExpReport CarOrders NanonullXML Folders Workarea Norkarea Norkarea	*
<	*
Switch to File Dialog Switch to URL Save C	Cancel

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

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

を想定して Authentic Desktop によりファイルが開かれるためです。

XSLT 変 換

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

SPS の割 リ当 て

StyleVision スタイルシートを XML ファイルに割 0当 てる際「Authentic | SPS ファイルの割 0当 て」、スタイルシートを指定するためにグローバルリソースを使用することができます。「参照」ボタン をクリックして、グローバルリソースリソースに切 1階 えボタンをクリックすることで、グローバルリソース開く ダイアログが表示 されます(セクション最上部のスクリーンショットを参照)。グローバルリソースを割当 先として選択することで、Authentic Desktopのアクティブな構成を切り替えるだけで、XML ドキュメン トの Authentic View を変更することができます。

7.2.2 構成の切り替え

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

構成の切り替えるには2つの方法があります:

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

Default	• 🛃 👳
---------	-------

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

8 Source Control

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

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

Installing and configuring the source control provider

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

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

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

licrosoft Visual SourceSafe	 Advanced
ogon ID (SourceSafe):	
IYFAVID	
Perform background status updates every	500 ms
Display output messages from plug-in	
Get everything when opening a project	
Check in everything when closing a project	
Don't show Check Out dialog box when chec	king out items
Don't show Check In dialog box when check	ing in items
Keep items checked out when checking in o	r adding items
dialogs were hidden using Don't show this agai ick Reset to view them again.	n, Reset

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

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

Some source control systems might not install the source control plug-in automatically, in which case you will need to install it separately. For further instructions, refer to the documentation of the respective source control system. A plug-in (provider) compatible with the Microsoft Source

Code Control Plug-in API is expected to be registered under the following registry entry on your operating system:

HKEY LOCAL MACHINE\SOFTWARE\SourceCodeControlProvider\InstalledSCCProviders

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

Accessing the source control commands

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

Resource / Speed issues

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

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

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

Differencing with Altova DiffDog

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

8.1 Setting Up Source Control

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

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

8.2 Supported Source Control Systems

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

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

Source Control System	Source Code Control Clients
AccuRev 4.7.0 Windows	AccuBridge for Microsoft SCC 2008.2
Bazaar 1.9 Windows	Aigenta Unified SCC 1.0.6
Borland StarTeam 2008	Borland StarTeam Cross-Platform Client 2008 R2
Codice Software Plastic SCM Professional 2.7.127.10 (Server)	Codice Software Plastic SCM Professional 2.7.127.10 (SCC Plugin)
Collabnet Subversion 1.5.4	 Aigenta Unified SCC 1.0.6 PushOK SVN SCC 1.5.1.1 PushOK SVN SCC x64 version 1.6.3.1 TamTam SVN SCC 1.2.24
ComponentSoftware CS-RCS (PRO) 5.1	ComponentSoftware CS-RCS (PRO) 5.1
Dynamsoft SourceAnywhere for VSS 5.3.2 Standard/Professional Server	Dynamsoft SourceAnywhere for VSS 5.3.2 Client
Dynamsoft SourceAnywhere Hosted	Dynamsoft SourceAnywhere Hosted Client (22252)
Dynamsoft SourceAnywhere Standalone 2.2 Server	Dynamsoft SourceAnywhere Standalone 2.2 Client
Git	PushOK GIT SCC plug-in (see <u>Source Control with</u> <u>Git</u>)
IBM Rational ClearCase 7.0.1 (LT)	IBM Rational ClearCase 7.0.1 (LT)
March-Hare CVSNT 2.5 (2.5.03.2382)	Aigenta Unified SCC 1.0.6
March-Hare CVS Suite 2008	 Jalindi Igloo 1.0.3 March-Hare CVS Suite Client 2008 (3321) PushOK CVS SCC NT 2.1.2.5 PushOK CVS SCC x64 version 2.2.0.4 TamTam CVS SCC 1.2.40
Mercurial 1.0.2 for Windows	Sergey Antonov HgSCC 1.0.1
Microsoft SourceSafe 2005 with CTP	Microsoft SourceSafe 2005 with CTP

Source Control System	Source Code Control Clients
Microsoft Visual Studio Team System 2008/2010 Team Foundation Server	Microsoft Team Foundation Server 2008/2010 MSSCCI Provider
Perforce 2008 P4S 2008.1	Perforce P4V 2008.1
PureCM Server 2008/3a	PureCM Client 2008/3a
QSC Team Coherence Server 7.2.1.35	QSC Team Coherence Client 7.2.1.35
Reliable Software Code Co-Op 5.1a	Reliable Software Code Co-Op 5.1a
Seapine Surround SCM Client/Server for Windows 2009.0.0	Seapine Surround SCM Client 2009.0.0
Serena Dimensions Express/CM 10.1.3 for Win32 Server	Serena Dimensions 10.1.3 for Win32 Client
Softimage Alienbrain Server 8.1.0.7300	Softimage Alienbrain Essentials/Advanced Client 8.1.0.7300
SourceGear Fortress 1.1.4 Server	SourceGear Fortress 1.1.4 Client
SourceGear SourceOffsite Server 4.2.0	SourceGear SourceOffsite Client 4.2.0 (Windows)
SourceGear Vault 4.1.4 Server	SourceGear Vault 4.1.4 Client
VisualSVN Server 1.6	 Aigenta Unified SCC 1.0.6 PushOK SVN SCC 1.5.1.1 PushOK SVN SCC x64 version 1.6.3.1 TamTam SVN SCC 1.2.24

8.3 Local Workspace Folder

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

Local Workspace Folder

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

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

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

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

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

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

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

8.4 Application Project

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

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



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

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

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

Local Workspace Folder	Repository
1	I
MyProject.spp	I MyProject (bound to Local Workspace)
QuickStart	MyProject.spp
QuickStart.css	QuickStart
QuickStart.xml	QuickStart.css
QuickStart.xsd	QuickStart.xml
Grouping	QuickStart.xsd
Persons	Grouping
- | | |-- Persons.xml || |-- Persons || | |-- Persons.xml
- **Note:** An application project can contain project folders (green) and external folders (yellow). Only files in (green) project folders can be placed under source control. Files in (yellow) external folders cannot be placed under source control.
- **Note:** Files from outside the local workspace folder can be added to the application project. But whether you can place such a file under source control depends upon the source control system you are using. Some source control systems could have a problem placing a file from outside the local folder into the repository. We therefore recommend that all project files you wish to place under source control be located in the local workspace folder.

8.5 Add to Source Control

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

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



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

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

Source Control - Add to Source Control	
Files C:\LocalWorkspace\Grouping\Persons\Persons.xml C:\LocalWorkspace\MyProject.spp C:\LocalWorkspace\QuickStart\QuickStart.css C:\LocalWorkspace\QuickStart\QuickStart.xml C:\LocalWorkspace\QuickStart\QuickStart.xsd	OK Cancel Select All
Keep checked out Comment	1.

The files will be added to the repository and be either <u>checked in or checked out</u> depending on whether the *Keep Checked Out* check box has been checked or not.

Configuration notes

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

Local Workspace Folder	Repository
1	I
MyProject.spp	I MyProject (bound to Local Workspace)
QuickStart	MyProject.spp
QuickStart.css	QuickStart
QuickStart.xml	QuickStart.css
QuickStart.xsd	QuickStart.xml
Grouping	QuickStart.xsd
Persons	Grouping
Persons.xml	Persons
	Persons.xml

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

C	reate or conne	t CVS module and repository	— ×
	Repository		
	CVSROOT	C:\MyRepository -	Create
	Click on create	to make a repository in the path specified	Check
	Module		
	CVS Module	MyProject +	Create
	Vendor	Altova GmbH 🗸	Connect
	Local Path is n	ot connected to CVS. Press create or connect	Check
	Local Path	C:\LocalWorkspace	
			Close

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

8.6 Working with Source Control

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

Project 📮 🗙
MyProject
🕀 🛅 XML Files
Persons.xml
QuickStart.xml
SSL Files
Cuery Files
🕀 🛅 HTML Files
QuickStart.css
ㄹ 🛅 DTD/Schemas
QuickStart.xsd
Entities

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

- Add to, Remove from Source Control
- Check Out, Check In
- Getting Files as Read-Only
- Copying and Sharing from Source Control
- <u>Changing Source Control</u>

Additional commands in the **Project | Source Control** menu are described in the <u>User Reference</u> <u>section</u> of the manual. For information specific to a particular source control system, please see the user documentation of that system.

8.6.1 Add to, Remove from Source Control

Adding

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



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

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

Source control symbols

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

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

Removing

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

8.6.2 Check Out, Check In

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

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

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

4 6 850	Checked in. Available for check-out.
	Checked out by another user. Not available for check-out.
20	Checked out locally. Can be edited and checked-in.

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



Saving and rejecting editing changes

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

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

Checking out

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

Source Control - Check Out	
Files C:\LocalWorkspace\Grouping\Persons\Persons.xml C:\LocalWorkspace\MyProject.spp C:\LocalWorkspace\QuickStart\QuickStart.css C:\LocalWorkspace\QuickStart\QuickStart.xml C:\LocalWorkspace\QuickStart\QuickStart.xsd	OK Cancel Select All Advanced
Checkout local version Comment	

Checking in

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

Source Control - Check In	
Files ✓ C:\LocalWorkspace\MyProject.spp ✓ C:\LocalWorkspace\QuickStart\QuickStart.xml ✓ C:\LocalWorkspace\QuickStart\QuickStart.xsd	OK Cancel Select All
Keep checked out Comment	Differences
	///

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

8.6.3 Getting Files as Read-Only

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

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

Source Control - Get	
Files	OK Cancel Select All Advanced
Overwrite changed files	1.

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

Changed files	
These files have been modified by an external application:	
C:\LocalWorkspace\Grouping\Persons\Persons.xml	*
	-
You can reload the file, close the document or ignore the modification	
Reload Close Cancel	

Advanced Get Options

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

Advanced Get Options		—
Replace writable:		OK Cancel
Set timestamp: Current	Make writable	Help

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

Get latest version

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

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

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

8.6.4 Copying and Sharing from Source Control

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

Create the new project as follows:

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

Create local project from SourceSafe	
Create a new project in the folder:	
C:\M20130326	Browse
SourceSafe project to download: \$/	
 Image: Signature Image: Additional Image: Image: Imag	
OK Cancel	Help

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

Sharing from source control

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

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

Project 🛛 📮 🛪	c
MyProject	1
The State St	
Persons.xml	
QuickStart.xml	
XSL Files	
📄 🖂 XQuery Files	
🕀 🛅 HTML Files	
QuickStart.css	
🕀 🛅 DTD/Schemas	
QuickStart.xsd	
Entities	

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

Share to \$/		—
File to share: QuickStart.xml QuickStart.css QuickStart.xml QuickStart.xsd	Projects: \$/QuickStart Additional Grouping Persons QuickStart	Close Share View Help
List files of type: Relevant Masks (*.*)	Branch after share	lii

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

The share folder is noted in the name of the Share to [Folder] dialog. In the screenshot above it is

the local folder (since the \$ sign is the folder in the repository to which the local folder is bound). You can see and set the share folder in the Change Source Control dialog (*screenshot below*, **Change Source Control**) by changing the local path and server binding.

Change Source Control				
Local Path:	C:\LocaWorkspace	Browse		
Scc Provider:	Microsoft Visual SourceSafe	Select		
Server Name:	C:WSSRepository	Bind		
Server Binding:	"\$/", ААААААА	Unbind		
Logon ID:	АААА			
Connected:				
	OK Cancel			

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

8.6.5 Changing Source Control

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

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

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

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

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

Change Source (Control	—
Local Path:	C:\LocaWorkspace	Browse
Scc Provider:	Microsoft Visual SourceSafe	Select
Server Name:	C:WSSRepository	Bind
Server Binding:	"\$/", ААААААА	Unbind
Logon ID:	АААА]
Connected:		
	OK Cancel	

Change source control settings as follows:

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

8.7 Source Control with Git

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

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

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



The Source Control Manager dialog box

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

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

- Enabling Git Source Control with GIT SCC Plug-in
- Adding a Project to Git Source Control
- <u>Cloning a Project from Git Source Control</u>

8.7.1 Enabling Git Source Control with GIT SCC Plug- in

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

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

Change Source C	Control	x
Local Path:	C:\Project1	Bro <u>w</u> se
Scc Provider:	PushOk GITSCC	Select
Server Name:		<u>B</u> ind
Server Binding:		<u>U</u> nbind
Logon ID:		
<u>C</u> onnected:		
	OK Cancel	

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

8.7.2 Adding a Project to Git Source Control

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

To add a project to Git source control:

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

Source Control - Add to Source Control	
Files C:MyRepo\Project.spp	OK Cancel Select All
Comment	
	1.

5. Click OK.

Please, enter the commit message	
Adding a project to a Git repository	^
	~
Do not ask for comments anymore	
Recent comments	OK Cancel

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

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



8.7.3 Cloning a Project from Git Source Control

Projects that have been previously added to Git source control (see <u>Adding a Project to Git</u> <u>Source Control</u>) can be opened from the Git repository as follows:

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

Open from Source Control Wizard	23
Specify source and destination Please specify url of GIT repository and local path where you want project to be created.	5
Source Repository:	
C:\MyRepo 🗸 🗌	-
Check	
Local Path:	
C:\GitClone	-
Sourse from SVN repository	
< <u>B</u> ack <u>N</u> ext >	Cancel

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



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

Open from Source Control Wizard	23
Copying remote repository to local folder Please wait while GIT clone your repository to local folder	5
Clone repository operation completed successfully.	
Clone the repository _ Ok.	*
	~
< <u>B</u> ack <u>N</u> ext >	Cancel

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

9 Visual Studio 内の Authentic Desktop

Authentic Desktop は Microsoft Visual Studio IDE のバージョン 2008/2010/2012/2013/2015/2017 に統合することができます。この機能により、高度な XML 編集機能を、Visual Studio の高度な開 発環境内で使用することができるようになります。

このセクションでは、以下のように構成されます:

- <u>大まかなインストールの流れ</u> とAuthentic Desktop プラグインを Visual Studio へ統合
- Visual Studio バージョンとスタンドアロンバージョンの違い

9.1 Visual Studio のための Authentic Desktop

プラグインのインストール Visual Studio で使える Authentic Desktop プラグインをインストールするには、以下の手順を踏みます:

- 1. Microsoft Visual Studio 2008/2010/2012/2013/2015/2017 をインストール
- 2. Authentic Desktop (Enterprise または Professional Edition) をインストール
- Authentic Desktop の Microsoft Visual Studio Integration Package をダウンロードして実行。パッケージはwww.altova.com にある <Authentic Desktop (Enterprise または Professional Edition)ダウンロードページにてご利用になれます。注意: Integration Package のバージョンは、お使いの Authentic Desktop のバージョンと同一のものでなければないません。最新バージョンは 2018 です。

Integration Package をインストールすると Visual Studio 環境から Authentic Desktop を使用する ことができるようになります。

プラグインを有効にする方法

インストールにてプラグインが自動的に有効化されなかった場合、以下の手順を踏んでください:

- 1. Visual Studio IDE 実行 ファイルがインストールされているディレクトリ(例:C:\Program Files\MS Visual Studio\Common7\IDE) に移動します。
- 2. コマンドプロンプトにて以下のコマンドを入力します: devenv.exe /setup.
- 3. 処理が終わるのを待って、Visual Studio内のアプリケーションの使用を開始してください。

9.2 スタンドアロンバージョンとの違い

このセクションでは Visual Studio バージョンとスタンドアロンバージョンの Authentic Desktop の違いについて説明します。

入力 ヘルパー (Visual Studio のツールウィンドウ)

Authentic Desktop の入力 ヘルパーは Visual Studio のツールウィンドウとして使用 することができます。 入力 ヘルパーに関して、以下 の点に注意して ください。 入力 ヘルパーと Authentic Desktop の GUI に関しては、<u>イントロ</u>を参照ください。

- 入力ヘルパーウィンドウは開発環境内のどのような場所にでもドラッグすることができます。
- 入力ヘルパータブを右クリックすることで、インターフェースを更にカスタマイズすることができます。入力 ヘルパーのオプションは:フローティング、ドッキング可能、タブ付きトキュメント、自動的に隠す、です。

Visual Stduio コマンドとして使える Authentic Desktop コマンド

いくつかの Authentic Desktop コマンドは Visual Studio コマンドとして Visual Studio GUI に表示 されます:

- *元に戻す、やり直し*:これらの Visual Studio コマンドは、Visual Studio 開発環境全てに 対して適用されます。
- **プロジェクト:** Authentic Desktop プロジェクトは Visual Studio プロジェクトとして扱われます。
- ツールバーのカスタマイズ、コマンドのカスタマイズ: (以下のスクリーンショットにある)ユー ザー設定ダイアログ ((ツール | ユーザー設定])内のツールバーならびにコマンドタブには Visual Studio とAuthentic Desktopのコマンドが含まれます。
- 表示:表示メニュー以下のAuthentic Desktop ツールウィンドウとAuthentic Desktop
 ビューの2つのコマンドには、入力ヘルパーや他のサイドバー、編集ビュー間の切り替え、そして特定の入力ガイドの有効 無効を切り替えるオプションが含まれています。
- Authentic Desktop ヘルプ: この Authentic Desktop メニューは Visual Studio [ヘルプ] メ ニューのサブメニューに表示 されます。

追加のメモ

追加のメモとヒントは以下にリストされるとおりです

 Authentic プラグインを使用して XML ファイルを編集するには、「ファイル | 開く」コマンドを 選択してください。そして、「ファイルを開く」ダイアログ内で、「開く」オプションを使用して Authentic プラグインを選択します。

10 Eclipse 内の XMLSpy

Eclipse はオープンソースのフレームワークで、様々なアプリケーション開発環境をプラグインという形で 統合します。

Eclipse 用の Authentic Desktop プラグインを使えば、Eclipse 4.5 / 4.6 / 4.7 プラットフォームから Authentic Desktop の機能にアクセスすることができるようになります。プラグインは Windows プラット フォームにてご利用になれます。 このセクションでは Eclipse 用の Authentic Desktop プラグインの <u>ストールの方法</u>と <u>Authentic パースペクティブ</u>の設定方法について説明します。これらの操作を行う と Authentic Desktop GUI とAuthentic Desktop メニューコマンドを Eclipse GUI から使用することが できるようになります。

10.1 Eclipse のための Authentic Desktop プラグイン

Eclipse 用の Authentic Desktop プラグインをインストールする前に、以下が既にインストールされていることを確認してください:

- Authentic Desktop Enterprise または Professional Edition。
- Eclipse に必要な、バージョン 5.0 以降の Java SE 実行環境 (JRE 5.0)。。詳細については Eclipse のウェブサイトを参照ください。Authentic Desktop のバージョン (32ビットまたは 64ビット)と等 しいバージョンの JRE(32ビットまたは64ビット)
- Eclipse プラットフォーム4.5 / 4.6 / 4.7。Authentic Desktop のバージョン (32-ビットトまたは 64-ビット)と等しいバージョンの Eclipse(32ビットまたは64ビット)をインストールしてください。

これらがインストールされた状態で、Authentic Desktop Integration Package に収められている Eclipse 用の Authentic Desktop プラグインをインストールすることができます(下記を参照)。

Eclipse インストーラーを使用して、Eclipse 4.5 をインストールした場合、Authentic Desktopの32-bit と64-bit バージョンを同じマシンで作動することができます。これは Eclipse インストーラーに搭載 されており、Eclipse (32-ビットと64-ビット)の両方のバージョンを手動 でインストールする場合には適用 されません。

Authentic Desktop Integration Package

Eclipse 用のAuthentic Desktop プラグインは Authentic Desktop Integration Package の中に含まれており、以下のようこ Authentic Desktop Integration Package のインストーラーからインストールすることができます:

- 1. Authentic Desktop (Enterprise または Professional Edition)、JRE、そして Eclipse が既に インストールされていることを確認してください(上記参照)。
- Authentic Desktop Integration Package のインストールでは、Eclipse 用の Authentic Desktop プラグインをインストールするかどうかを尋 ねられます (以下のスクリーンショットを参照)。オプションにチェックを入れて、次へをクリックしてください。

ALTOVA	Altova Authentic® 2013 Integration Package options Please configure the components to be installed
This package contains other applications.	components that allow integration of the product's functionality into
☑ Install the Ac	tiveX Control (required)
🔽 Install .NE	T assembly wrappers for the ActiveX Control
Additionally, following	integrations into widely used development platforms are available:
☑ Install the Mi	crosoft Visual Studio plug-in
V Enable th	e integration of the plugin into Visual Studio

 その次に表示される(以下のスクリーンショットの)ダイアログにて、インストールウィザードにより Eclipse へ Authentic Desktop プラグインの統合を行う(自動)か、ご自身の手で(Eclipse GUI から)Eclipse 用の Authentic Desktop プラグインを統合することができます(手動)。

ALTOWA Installation Location Select how to integrate into Edipse 3.x	
Select how to integrate into Edipse 3.x	
Integrate the plug-in into Eclipse:	
Automatically, by creating a "Link File" for an existing Eclipse installation.	
Please, enter installation location of your Eclipse 3.x :	
C:\edipse\ Browse	
Start Eclipse with -clean flag in order to refresh its plug-in related meta-data.	
Manually, using the Edipse configuration manager.	
Start Eclipse and select menu Help Install New Software, add below folders as a new local "Sites" and install the contained plug-ins.	
C:\Program Files (x86)\Altova\Common2013\edipse\UpdateSite	
C: \Program Files (x86) \Altova \Authentic2013\edipse \UpdateSite \.	
Altova	•
< Back Next > Cancel	

ここではインストールウィザードによる統合(自動)を選択することを推奨します。自動オプションのラジオボタンをクリックして、Eclipse実行ファイル(eclipse.exe)が収められているフォルダーを参照します。それらの操作を完了後、次へボタンを押下します。手動でAuthentic Desktop プラグインをインストールする場合は、(以下のスクリーンショットにあるように)手動オプションを選択してください。Eclipse内から手動で統合を行う方法については以下を参照ください。

討 Alto	va Authentic® 2	013 Integration Package - Setup	X
		Installation Location	
AL	.1004	Select how to integrate into Eclipse 3.x	
Integ	prate the plug-in in	to Edipse:	
0	Automatically, by	r creating a "Link File" for an existing Eclipse installation.	
	Please, enter i	nstallation location of your Eclipse 3.x :	
		Browse	
	Start Edipse	with -dean flag in order to refresh its ed meta-data.	
۲	Manually, using t	he Edipse configuration manager.	
	Start Edipse ar new local "Site:	nd select menu Help Install New Software, add below folders as a s" and install the contained plug-ins.	a
	C:\Program File C:\Program File	es (x86) \Altova \Common2013 \eclipse \UpdateSite \. es (x86) \Altova \Authentic2013 \eclipse \UpdateSite \.	
Altova -			
		< Back Next > Cance	

5. インストールを完了します。自動統合を選択した場合、Eclipse 用の Authentic Desktop プラグインが Eclipse 内に統合され、次に Eclipse を起動するときから利用できるようになり ます。

Eclipse 内部で Authentic Desktop プラグインを手動で統合する

以下の操作によりEclipseのAuthentic Desktop プラグインを手動で統合することができます:

- 1. Eclipse にてメニューオプションから「ヘルプ | 新規ソフトウェアのインストール」を選択します。
- 2. 表示 されるインストールダイアログにて、追加ボタンをクリックします (以下のスクリーンショット を参照)。

●インストール	<u>_ 0 ×</u>
使用可能なソフトウェア	
サイトを選択するかサイトのロケーションを入力してください。	
作業対象(11)。" 世化を入力または選択	追加(<u>A</u>)
<u>"使用可能なソフトウェア・サイト"</u> 設定で作業して、より多くのソフトウェアを検	索してください。
フィルター入力	
名前 パージョン	
□ ① 選択されたサイトがありません。	
すべて選択(S) 選択をすべて解除(D)	
	=
▶ 最新の使用可能なソフトウェア・バージョンのみ表示(L) 🔲 すでにインストールされている項目:	を隠す(<u>H</u>)
☑ 項目をカテゴリー別にグループ(L(G) <u>すでにインストールされている</u> のは?	
□ ターゲット環境へ適用可能なソフトウェアのみ表示	
☑ 必須ソフトウェアを検索するためにインストール中にすべての更新サイトに接続する(C)	
(2) < 戻る(B) 次へ(t) > 完了(F)	キャンセル

- 3. リポジトリーの追加ダイアログにて**ローカル**ボタンをクリックします(以下のスクリーンショットを参照)。
- 4. ダイアログにて c:\Program Files\Altova\Common2018\eclipse\UpdateSite のフォル ダーを選択します。サイトの名前(例: "Altova")を入力し、OK をクリックしてください。

●インストール
使用可能なソフトウェア サイトを選択するかサイトのロケーションを入力してください。
作業対象(W): ^① サイトを入力または選択 「使用可能なソフトウェア・サイト" 設定で作業して、より多くのソフトウェアを検索してください。 「フィルター入力
□ ① 選択されたサイトがありません。
☑ 最新の使用可能なソフトウェア・バージョンのみ表示(L) □ すでにインストールされている項目を隠す(H)
☑ 項目をカテゴリー別にグループ化(G) <u>すでにインストールされている</u> のは?
□ ターゲット環境へ適用可能なソフトウェアのみ表示
☑ 必須ソフトウェアを検索するためにインストール中にすべての更新サイトに接続する(©)
(ア) (京31日) 次へ(11) (京71日) キャンセル

- 5. 上記にあるステップの2から4まで c:\Program Files\Altova\Authentic Desktop2018 \eclipse\UpdateSite, というフォルダーに対して行い、別の名前(例えば Altova Authentic Desktop')を入力してください。
- インストールダイアログの作業対象コンボボックスにて -- すべての使用可能なサイト--オプションを選択します(以下のスクリーンショットを参照)。利用可能な全てのプラグインが下部のペインに表示されます。トップレベルにある Altova Category のチェックボックスにチェックを入れて、次へボタンをクリックします。

) Install			- • •	
Available S	oftware			
Check the it	ems that you wish to install.			
Work with:	All Available Sites	•	Add	
	Find more software by worki	ng with the <u>"Available Software</u>	<u>Sites"</u> preferences.	
type filter te	đ			
Name		Version	^	
🔺 📝 💷 A	ltova category		E	
	Altova Eclipse Plug-in Library	15.0.0		
	Authentic Desktop for Eclipse PI Tools Execution Environment Descriptions	10.0	-	
Select All Deselect All 4 items selected				
Details				
Show only	the latest versions of available software	Hide items that are already in	stalled	
Group iten	ns by category	What is <u>already installed</u> ?		
Show only software applicable to target environment				
Contact all update sites during install to find required software				
?	< Back	Next > Finish	Cancel	

- 7. インストール詳細画面にてインストールするアイテムを確認します。次へをクリックします。
- 表示されるライセンスのレビュー画面にて、使用条件の条項に同意しますを選択し、完 了ボタンをクリックしてインストールを行います (Authentic Desktop Enterprise または Professional Edition ライセンスに加えて Authentic Desktop プラグインのライセンスへ新たに 同意する必要はありません)。

アイコンが表示されないなどの問題が発生した場合は、elean フラグ付きで Eclipse を起動してみてください。

現在インストールされているバージョン

現在インストールされている Authentic Desktop プラグインのバージョンを確認 するには、Eclipse のメ

ニューオプションから「ヘルプ | Eclipse について」を選択します。 そこから Authentic Desktop アイコンを選択してください。

10.2 Eclipse 内の Authentic Desktop エンドリポイント

Eclipse 内 にある以下のエントリポイントを使って Authentic Desktopの機能にアクセスすることができます:

- <u>Authentic Desktop パースペクティブ</u>からAuthentic Desktop の GUI 機能に Eclipse GUI からアクセスすることができます。
- <u>Authentic Desktop メニューとソールバー</u>

Authentic Desktop パースペクティブ

Eclipse におけるパースペクティブとは、特定の機能に結び付けられた GUI ビューのことです。Eclipse 用の Authentic Desktop プラグインが Eclipse に統合 されると Authentic Desktop パースペクティブも 自動的に作成 されます。このパースペクティブは、編輯 ビュー、メニュー、入力 ヘルパー、そしてサイド バーなど、Authentic Desktop の GUI 要素を含むパースペクティブとなります。

Authentic Desktop と関連付けられている種類のファイル (例えば.xml)が開かれた時、このファイル を Authentic Desktop パースペクティブで編集することができます。同様に、他の種類のファイルも Eclipse の別のパースペクティブで開くことができます。更に、アクティブなファイルに対しては、パースペ クティブを変更することができ、異なる環境で同じファイルの編輯を行うことができます。従って、パース ペクティブには 2つの利点があります:

- 1. アクティブなファイルに対して、作業環境を素早く変更することができる。
- 2. 新たな開発環境を開くこと無くファイルの切り替えを行うことができる(パースペクティブに結び 付けられた環境も有効になります)。

Authentic Desktop パースペクティブを使った作業には以下のようなものがあります:

- Authentic Desktop パースペクティブへの切り替え
- Authentic Desktop パースペクティブの環境設定
- Authentic Desktop パースペクティブのカスタマイズ

Authentic Desktop パースペクティブへの切り替え

Eclipse にて、メニューコマンドから「ウィンドウ | パースペクティブを開く | その他」を選択します。表示 されるダイアログにて Authentic Desktop を選択し、OK をクリックします (以下のスクリーンショットを参照)。



空のドキュメント、またはアクティブなドキュメントに対して Authentic Desktop パースペクティブが開かれます。この方法でユーザーはパースペクティブの切り替えを行うことができます。目的のパースペクティ ブを**パースペクティブを開く**サブメニュー (の**その他** アイテムの上)に表示させることで、パースペクティブ により素早くアクセスすることができます。この設定はカスタマイズダイアログにて行うことができます。

ファイルが開かれた時、またはアクティブになったときにパースペクティブを自動的に切り替えることもできます。そのファイルが最初に開かれたときには、ファイルの種類に関連付けられているアプリケーションのパースペクティブが自動的に開かれ、このファイルの種類に対して、このパースペクティブをデフォルトのパースペクティブとして割り当てるか尋ねられます。この種類のファイルを開くたびに尋ねられたくない場合は、今後このメッセージを表示しないにチェックを入れOKをクリックします。

。この種類のファイルを開くたびに尋ねられたくない場合は、今後このメッセージを表示しないにチェックを入れ OK をクリックします。

Authentic Desktop パースペクティブの環境設定

パースペクティブの設定には、(i) 関連付けられた種類のファイルが開かれたときに、自動的にパースペクティブを切り替える、(ii) 各 Authentic Desktop ツールバーを含めるオプションが含まれます。 左のペインにあるリストから Authentic Desktop を選択し、必要な設定を選択します。 OK ボタンをクリックして完了します。

Authentic Desktop パースペクティブのカスタマイズ

カスタマイズオプションを使うことで、ショートカットやコマンドをパースペクティブに加えることができます。 (以下のスクリーンショットにある)パースペクティブのカスタマイズダイアログにアクセスするには、 (Authentic Desktop パースペクティブの場合は)パースペクティブをアクティブにして、メニューコマンドから「ウィンドウ」パースペクティブのカスタマイズ」を選択します。

ツール・バー可視性 タブとメニュー可視性 タブでは、表示 するツールバーとメニューを指定 することがで きます。 コマンドグループ可用性 タブでは、コマンドグループを親 メニューやツールバーへ追加 すること ができます。コマンドグループを有効 にするには、対応 するチェックボックスにチェックを入れて ください。 カスタマイズパースペクティブダイアログのショートカットタブでは、サブメニューのショートカットをセットす ることができます。サブメニューコンボボックスから目的のサブメニューを選択し、ショートカットのカテゴリ を選択し、パースペクティプに追加したいショートカットにチェックを入れます。 OK をクリックしてカスタ マイズを完了します。

Authentic Desktop メニューとツールバー

ツールバーには Authentic Desktop-に関連する2つのボタンが追加されます(以下のスクリーンショットを参照)。



これらのボタンは、(i) Authentic Desktop ヘルプを開くと(ii) 新規 Authentic Desktop ドキュメントの 作成のために使用されます。。

11 メニューコマンド

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

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

11.1 ファイルメニュー

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

<u>このセクション</u>:

- <u>アイコンとショートカット</u>
- <u>説明</u>

アイコンとショートカット

アイコン	
シ <i>ヨ</i> ートカット:	Ctrl+N

説明

このコマンドによりAuthentic View 内 で新規の XML ドキュメントテンプレートを開く事ができます。 XML ドキュメントテンプレートは、StyleVision Power Stylesheet (.sps ファイル)をベースにしています。 新規のドキュメントの作成 ダイアログ内で StyleVision Power Stylesheet (SPS ファイル)を選択 する事により開かれます (F O A O U - v > a = v + b)。 SPS を選択して、「**OK」**をクリックすると SPS ファイルのために定義 された XML ドキュメントテンプレートが Authentic View 内で開かれます。

新規ドキュメントの作成
Publishing rixml teilite xmlresume xmlspec OK daisy dita Examples NCAXML News P3P キャンセル fmm dtbook-2005 イロシーン チャンセル チャンセル

新規のドキュメントの作成 ダイアログは、一般的に使用されるDTD またはスキーマをベースにした、 XML ドキュメントテンプレートの選択を提供します。または、テンプレート XML ファイルが割 リ当 てら れているカスタムメイドの SPS ファイルを参照することができます。SPS ファイルは、Altova StyleVision を使用して作成されます。アプリケーションにより、DTD または XML スキーマをベースにし た XML ドキュメントテンプレートをデザインすることができます。StyleVision 内 で必要な SPS をデザ インすると XML ファイルは、(StyleVision 内 で) SPS への テンプレート XML ファイルとして割 リ当 て られます。XML ファイル内のデータは、Authentic Desktop の Authentic View 内 で開 かれた新規の ドキュメントテンプレートの開始データを与えます。 新規のXML ドキュメントテンプレートは、ですから、テンプレートXML ファイルとして選択された SPS とXML ファイルのデータ内で定義されたドキュメントプレゼンテーションプロパティを持つようになります。 Authentic View ユーザーは、XML ドキュメントテンプレートをグラフィカルな WYSIWYG インターフェイ スで編集することができ、XML ドキュメントとして保存することができます。

11.1.2 開く

アイコンとショートカット

アイコン	3
シ <i>ヨ</i> ートカット:	Ctrl+O

説明

開くコマンドにより、慣れ親しんだ Windows の開くダイアログが表示され、XML に関するドキュメント や、テキストドキュメントを開くことができます。開くダイアログでは、2つ以上のファイルを開くことができ ます。ファイルの種類コンボボックスを使用することで、ダイアログボックスに表示されるファイルの型セク ションを制限することができます(表示されるファイルの種類一覧は、オプションダイアログ(「ツール」 オプション」)にて設定することができます)。XML ファイルが開かれると整形式のチェックが行われま す。ファイルが整形式で無い場合、整形式に関するエラーが表示されます。エラー箇所を修正し て、メニューコマンドから XML」整形式のチェック(F7)を選択します。 ファイルを開く再に自動検 証 を行うよう選択しており、ファイルが妥当でない場合、エラーメッセージが表示されます。エラー箇 所を修正して、メニューコマンドから「XML」XML の検証(F8)」を選択して、再度検証を行いま す。

▼ URL とグローバル リソースを使用してファイルを選択または保存する

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

ファイルを開く						×
ファイルの場所(I):	🗼 Examples	•	G 🤌 📂 🛄 🗸			
Ca	名前	A		更新日時	サイズ	•
] Japanese UC	S-4.xml		4/10/2014 1:11 AM	2 KB	
最近表示した場	Japanese UT	F-16.xml		4/10/2014 1:11 AM	1 KB	
771	NanonullOrg	.xml		4/10/2014 1:11 AM	16 KB	
	NanonullOrg	.xsd		4/10/2014 1:11 AM	4 KB	
デスクトップ	OnChange.x	ml		4/10/2014 1:11 AM	1 KB	
	OnClick.xml			4/10/2014 1:11 AM	1 KB	
P	OnContextM	enu.xml		4/10/2014 1:11 AM	1 KB	
	OnFocus.xm	I		4/10/2014 1:11 AM	1 KB	
ライブラリ	OnUserAdde	dXMLNode.xml		4/10/2014 1:11 AM	1 KB	≡
	GrgChart.px	f		4/10/2014 1:11 AM	113 KB	
	PurchaseOrc	ler.zip		4/10/2014 1:11 AM	3 KB	
	Russian.xml			4/10/2014 1:11 AM	2 KB	
1767-2-	ScriptingSar	nples.css		4/10/2014 1:11 AM	1 KB	Ŧ
	•	111			1	•
	ファイル名(N):	Conditional.xml			▼ 開(0)	
ネットワーク	ファイルの種類(T):	全ての XMLSpy ファイル			▼ キャンセル	
		URL にスイッチ (U))	-パルリソースにスイッチ(G)		

URL を使用 してファイルを切り替える

URL から(開くまたは保存する)ファイルを選択するには以下を行ってください:

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

Open			×
File URL:			•
Open as:	O DTD	File load	Reload
Identification User: MyDocs	Password:	Remem between	ber password n application starts
Available files Server URL: http://viets	pstest/ harePoint® Server	•	Browse
1		New Folder	Delete
Switch to File Dialog	Switch to Global Resour	ces Open	Cancel

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

Open	
File URL: http://gd.tuwien.ac.at/vietspstest/_ca	atalogs/lt/Forms/DispForm.aspx
Open as: Auto	File load © Use cache/proxy
Identification User: TestUser Password: ••••••	Remember password between application starts
Available files Server URL: http://gd.tuwien.ac.at/ IV This is a Microsoft® SharePoint® Server	■ Browse
gd.tuwien.ac.at gd.tuwien.ac.at	
	New Folder Delete
Switch to File Dialog Switch to Global Resources	Open Cancel

ファイル URL がファイル URL フィールドに表示 されます (上のスクリーンショット参照)。 開く」または「保存」ボタンはこの時点からアクティブになります。

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

<u>以下の点に注意してください</u>:

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

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

 「利用可能なファイル」ペインに表示されるディレクトリ構造では、ファイルアイコンには ファイルのチェックイン チェックアウトの状態を表示するシンボルがあります(下のスクリーンショット)。

Open	—
File URL: http://vietspstest/Docs/Documents/flo	:/AutoCalc.sps 🔹
Open as: O Auto	File load
Identification User: MyDocs Password: ••••••	Remember password between application starts
Available files Server URL: http://vietspstest/ This is a Microsoft® SharePoint® Server	✓ Browse
⊡-@ Documents ⊡-@ chr ⊡-@ flc	
AutoCalc.sps Check Qut Bank_Java.ur Check In ⊡ im Foldertestmip Check In	R. C.
⊕ m Forms ⊕ Marketing	
	New Folder Delete
Switch to File Dialog Switch to Global Resources	Open Cancel

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

異なるファイルアイコンが以下に表示されています:

4 6 850	チェックインされています。チェックアウトできます。
	他 のユーザーによりチェックアウトされています。 チェックアウトできませ ん。
5	ローカルでチェックアウトされています。 編集 およびチェックインすること ができます。

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



- ファイルが他のユーザーによりチェックアウトされている場合、チェックアウトすることができません。
- ファイルがローカルでチェックアウトされている場合、コンテキストメニューのチェックアウト を元に戻すコマンドを使用してチェックアウトを元にもどすことができます。これによりファ

イルを変更することなくサーバーに戻すことができます。

- Altova アプリケーションでファイルをチェックアウトした場合、他の Altova アプリケーション でチェックアウトすることはできません。ファイルは既 にチェックアウト済 みと見 なされます。 この時点で使用できる Microsoft® SharePoint® Server をサポートする Altova アプリケ ーションのコマンドは、以下の通 リです:「チェックイン」および「チェックアウトを元 に 戻す」です。
- ▼ グローバルリソースを使用してファイルを選択または保存する

グローバルリソースを使用してファイルを開くまたは保存する場合、「グローバルリソースに切り替える」をクリックします。グローバルリソースを選択することのできるダイアログがボップアップします。このダイアログの詳細については、グローバルリソースを使用する. グローバルリソースの 一般的な説明に関しては、このドキュメントのグローバルリソース のセクションを参照してください。

11.1.3 再ロード

アイコン



説明

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

11.1.4 エンコード

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



XML ドキュメントにおいて、それまで使われていたものとは異なるエンコーディングを指定すると XML 宣言内のエンコーディング指定もそれに合わせて変更されます。2バイト文字ならびに4バイト文字 エンコーディング (UTF-16、UCS-2、UCS-4)では、ファイルに使用されるバイトオーダーを指定すること ができます。XML ドキュメントのエンコーディングを変更するもう1つの方法は、ドキュメントの XML 宣 言で使用されている encoding 属性を直接編集することです。既存ならびに新規 XML ドキュメン ト、または非 XML ドキュメントのデフォルトエンコーディングは、オプションダイアログのエンコードセクショ ンにて指定することができます。

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

11.1.5 閉じる、全て閉じる、非アクティブを全て閉じる

閉 じる

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

全て閉じる

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

非 アクティブを全 て閉 じる

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

11.1.6 保存、名前を付けて保存、全て保存

アイコンとショートカット

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

保存

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

名前をつけて保存

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

全て保存

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

▼ URL とグローバル リソースを使用 してファイルを選択 または保存 する

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

ファイルを開く						×
ファイルの場所(I):	👢 Examples	•	G 🤌 📂 🛄 🔻			
Ca	名前	A		更新日時	サイズ	•
] Japanese UC	S-4.xml		4/10/2014 1:11 AM	2 KB	
最近表示した場	Japanese UT	F-16.xml		4/10/2014 1:11 AM	1 KB	
771	NanonullOrg	.xml		4/10/2014 1:11 AM	16 KB	
	NanonullOrg	.xsd		4/10/2014 1:11 AM	4 KB	
デスクトップ	OnChange.x	nl		4/10/2014 1:11 AM	1 KB	
	OnClick.xml			4/10/2014 1:11 AM	1 KB	
P	OnContextM	enu.xml		4/10/2014 1:11 AM	1 KB	
	OnFocus.xm			4/10/2014 1:11 AM	1 KB	
ライブラリ	OnUserAdde	dXMLNode.xml		4/10/2014 1:11 AM	1 KB	≡
	GrgChart.pxf	:		4/10/2014 1:11 AM	113 KB	
	PurchaseOrd	er.zip		4/10/2014 1:11 AM	3 KB	
	Russian.xml			4/10/2014 1:11 AM	2 KB	
1767-8-	ScriptingSan	nples.css		4/10/2014 1:11 AM	1 KB	-
	•				1	•
	ファイル名(N):	Conditional.xml			▼ 帰 ((0)	
ネットワーク	ファイルの種類(T):	全ての XMLSpy ファイル			▼ キャンセル	
		URL にスイッチ(U)) グロー	-バルリソースにスイッチ(G)		

URL を使用 してファイルを切り替える

URL から(開くまたは保存する)ファイルを選択するには以下を行ってください:

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

Open	—
File URL:	•
Open as: O Auto XML DTD	File load
Identification User: MyDocs Password: ••••••	Remember password between application starts
Available files Server URL: http://vietspstest/	▼ Browse
✓ This is a Microsoft [®] SharePoint [®] Server	
<u> </u>	
	New Folder Delete
Switch to File Dialog Switch to Global Resources	Open Cancel

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

Open		— ×
File URL: http://gd.tuwien.ac.at/vietspstest/_cat	alogs/lt/Forms/DispForm.aspx	
Open as: Auto XML DTD	File load Use cache/proxy) Reload
Identification User: TestUser Password: ••••••	Remember pa between app	assword lication starts
Available files Server URL: http://gd.tuwien.ac.at/ Image: This is a Microsoft® SharePoint® Server	•	Browse
Image: Second		
	New Folder	Delete
Switch to File Dialog Switch to Global Resources	Open	Cancel

ファイル URL がファイル URL フィールドに表示 されます (上のスクリーンショット参照)。 「開く」または「保存」ボタンはこの時点からアクティブになります。

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

<u>以下の点に注意してください</u>:

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

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

 「利用可能なファイル」ペインに表示されるディレクトリ構造では、ファイルアイコンには ファイルのチェックイン チェックアウトの状態を表示するシンボルがあります(下のスクリーンショット)。

Open			
File URL: http://vietspstest/Docs/Documents/file	c/AutoCalc.sps 🔹		
Open as: O Auto	File load		
Identification User: MyDocs Password: ••••••	Remember password between application starts		
Available files Server URL: http://vietspstest/ Browse This is a Microsoft® SharePoint® Server			
Documents 			
AutoCalc.sps Check Out Bank_Java.ur Check In ⊡ Toldertestmip			
	New Folder Delete		
Switch to File Dialog Switch to Global Resources	Open Cancel		

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

異なるファイルアイコンが以下に表示されています:

850 850	チェックインされています。チェックアウトできます。
	他 のユーザーによりチェックアウトされています。 チェックアウトできませ ん。
5	ローカルでチェックアウトされています。 編集 およびチェックインすること ができます。

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



- ファイルが他のユーザーによりチェックアウトされている場合、チェックアウトすることができません。
- ファイルがローカルでチェックアウトされている場合、コンテキストメニューのチェックアウト を元に戻すコマンドを使用してチェックアウトを元にもどすことができます。これによりファ

イルを変更することなくサーバーに戻すことができます。

- Altova アプリケーションでファイルをチェックアウトした場合、他の Altova アプリケーション でチェックアウトすることはできません。ファイルは既 にチェックアウト済 みと見 なされます。 この時点で使用できる Microsoft® SharePoint® Server をサポートする Altova アプリケ ーションのコマンドは、以下の通 リです:「チェックイン」および「チェックアウトを元 に 戻す」です。
- ▼ グローバルリソースを使用してファイルを選択または保存する

グローバルリソースを使用してファイルを開くまたは保存する場合、「グローバルリソースに切 リ替える」をクリックします。グローバルリソースを選択することのできるダイアログがポップアップ します。このダイアログの詳細については、グローバルリソースを使用する. グローバルリソースの 一般的な説明に関しては、このドキュメントのグローバルリソース のセクションを参照してください。

11.1.7 メールで送信

アイコン

דר דר	*
-------	---

説明

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

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

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

メールで送信	?×
送信オプション 〇 ファイル全て ④ 現在の選択 ④ 添付として 「エンコーディングステートメントを追加 〇 メールコンテンジとして	OK キャンセル

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

メールで送信	?×
URL ファイルをどのように送信しますか?	OK
	キャンセル

11.1.8 印刷

アイコンとショートカット

アイコン	5
シ <i>ヨ</i> ートカット:	Ctrl+P

説明

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

11.1.9 印刷プレビュー、印刷設定

印刷 プレビュー

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

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

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

印刷設定

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

プリンタの設定	<u>? ×</u>
プリンター	
プリンタ名(N): Microsoft Office Document Image Write	æブロパティ(<u>P</u>)
状態: 準備完了	
種類: Microsoft Office Document Image Writer D	river
場所: Microsoft Document Imaging Writer Port:	
コメント	
用紙	「印刷の向き
サイズ(Z): Letter 💌	○ 縦(○)
給紙方法(S): 通常使う用紙トレイ ▼	
ネットワーク(<u>W</u>)	OK キャンセル

11.1.10 最近使用されたファイル、終了

最近使用 されたファイル

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

終了

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

11.2 編集メニュー

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

11.2.1 元に戻す、やり直し

アイコンとショートカット

コマンド	アイコン	<i>ショ、トカット</i>
元に戻す	5	Ctrl+Z
やり直し	2	Ctrl+Y

元に戻す

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

やり直し

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

11.2.2 切り取り、コピー、貼り付け、削除

アイコンとショートカット

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

切り取り

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

コピー

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

貼り付け

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

削除

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

11.2.3 全て選択

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

11.2.4 検索、次を検索

アイコンとショートカット

コマンド	アイコン	ショートカット
検索	\$	Ctrl+F
次 を検 索	¹	F3

検索

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

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

次 を検 索

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

11.2.5 置換

アイコンとショートカット

コマンド	アイコン	<i>ショ、トカット</i>
置換		Ctrl+H

説明

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

11.3 プロジェクトメニュー

Authentic Desktop では、広く使用されているツリービューにより表示される XML プロジェクトを使用 することで、複数のファイルや URL を管理することができます。ファイル とURL は拡張子などでフォ ルダー内にグループ化することができ、バッチ処理などを簡単に行うことができるようになります。

	新規プロジェクトの作成(N)
≞	プロジェクトを開く(尸)
₽. ₽	プロジェクトの再ロード(<u>R</u>)
	プロジェクトを閉じる(2)
	プロジェクトの保存(<u>S</u>)
	プロジェクトに名前を付けて保存(A)
	ソース管理(E) →
e tů	プロジェクトにファイルを追加(生)
P t 📑	プロジェクトにグローパル リソースを追加(<u>G</u>)
t.	プロジェクトに URL を追加(<u>U</u>)
P t=	プロジェクトにアクティブなファイルを追加(型)
₽ ±⊟	プロジェクトにアクティブならびに関係するファイルを追加(生)
t 🗅	プロジェクトにプロジェクト フォルダーを追加(」)…
	プロジェクトに外部フォルダーを追加(凶)
	プロジェクトに外部ウェブフォルダーを追加(Ψ)
	スクリプト設定
P) t()	プロパティ()
Ð	1 Examples

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

絶対パスと相対パス

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

```
|-Fobler1
| |
| |-Fobler2
```

⊢Fobler3 | |-Fobler4

.spp ファイルが Folder3内部にある場合、Folder1ならびに Folder2内部のファイルに対する参照は、 以下のように記述されます:

c:\Foberl\NameOffileext c:\Foberl\Fober2\NameOffileext

Folder3 ならびに Folder4 への参照は、以下のように記述されます:

.\Nam eOffie ext .\Fobler4 \Nam eOffie ext

相対 パスの使用を確実なものにするには、作業しているディスク(ドライブ)のルートディレクトリに spp ファイルを配置してください。

ドラッグアンドドロップ

プロジェクトウィンドウでは、フォルダーをドラッグすることで、別のフォルダーや、同じフォルダー内における別の位置へ、そのフォルダーを移動することができます。ファイルをドラッグすることで別のフォルダーへの移動を行うことはできますが、同一フォルダー内における移動はできません(フォルダー内部のファイルはアルファベット順に表示されます)。更に、Windowsファイルエクスプローラーからプロジェクトウィンドウへ、ファイルのドラッグを行うこともできます。

プロジェクト内の検索

ファイルの名前、またはファイル名の一部からプロジェクト内部にあるファイルやフォルダーを検索することができます。検索が行われると該当するファイルやフォルダーが1つずつハイライトされた状態で表示されます。

検索を行うには、検索を行うプロジェクトフォルダーをプロジェクトウィンドウにて選択し、「編集 | 検 索」メニューコマンドを選択するか、Ctrl+F キーを押下します。検索ダイアログが表示されるので、検 索を行う文字列を入力し、目的に従うかたちで検索オプションを指定します(以下のスクリーンショッ トを参照)。検索オプションについては以下を参照ください。

検索	×
検索対象(N): OrgChart.pxf	次を検索E
- オブション	前を検索(P)
□ 大文字・小文字を区別(C)	キャンセル
▼ フォルダー名を検索	
▼ 外部フォルダーをスキップ	

以下の検索オプションが利用できます:

単語の完全マッチ:入力された検索語とファイルまたはフォルダー名が完全に一致するかチェックされ、厳格な検索を行うことができます。ドット(".")の前後に入力された文字列も単語として認識され、例

えばファイルの拡張子だけを入力して検索を行うことができます。

- 大文字小文字を区別:検索語として入力された文字列の大文字と小文字を区別するかどうか指定する ことができます。
- フォルダー名を検索:フォルダー名を検索の対象に含めます。選択されていない場合、ファイル名だけ が検索されます。
- 外部フォルダーを検索の対象に加える、または対象から外す事ができます。 外部フォルダーはシ ステムまたはネットワーク上 に実在 しているフォルダーで、 プロジェクト内 部 で論 理 的 に作 成 された プロジェクトフォルダーとは異なるものです。

検索が正常に行われた場合、最初にマッチしたアイテムがプロジェクトウィンドウにてハイライトされます。 検索ダイアログにある次を検索ならびに前を検索ボタンをクリックすることで、その他のマッチを参照する ことができます。

プロジェクトの更新

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

コンテキストメニュー内 のグローバルリソース

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

プロジェクトならびにソース管理 プロバイダー

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

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

例 えば、上 に示 したディレクトリ構造 で、 プロジェクトファイルが Folder3 に保存 され、 ソース管理 にも 配置されると:

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

11.3.1 新規プロジェクト



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

11.3.2 プロジェクトを開く

1

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

11.3.3 プロジェクトの再ロード



プロジェクトの再 ロードコマンドは、現在 のプロジェクトをディスクから再度 ロードします。 複数 のユー ザーが作業 をしているような環境 では、他 のユーザーがプロジェクトに加えた変更 を反映するために、 プロジェクトの再 ロードが必要 になる時 があります。

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

11.3.4 プロジェクトを閉じる

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

11.3.5 プロジェクトの保存、名前を付けて保存

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

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

11.3.6 ソース管理

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

ソース管理機能の概要

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

- Authentic Desktop にて、ソース管理へ配置するファイルが含まれているアプリケーションプロジェクトフォルダーを作成します。通常アプリケーションフォルダーは、プロジェクトファイルが含まれているローカルのフォルダーに対応しています。ローカルフォルダーへのパスは、ローカルパスと呼ばれます。
- 2. (ソース管理またはレポジトリとして参照される)ソース管理システムのデータベース内では、 ソース管理下に置かれるファイルを含むフォルダーが作成されます。
- 3. 「<u>プロジェクト」ソース管理」ソース管理に追加」</u>コマンドにより、アプリケーションプロジェクトファイルがソース管理へ追加されます。
- チェックインやチェックアウト、ソース管理からファイルを削除といった、ソース管理に関するアクションは、「プロジェクト」ソース管理サブメニューコマンドを使用することで行うことができます。このセクションのサブセクションでは、このサブメニューにあるコマンドに関する記述を行います。
- ★モ: 現在使用しているソース管理プロバイダーを変更するには、(i) ソース管理オプション (ツール | オ <u>プション | ソース管理</u>)から行う方法と、(ii) ソース管理の変更ダイアログ (プロジェクト | ソー ス管理 | ソース管理の変更)にて行う方法の2つが用意されています。
- **メモ:** ソース管理プロジェクHは、アプリケーションプロジェクトとは異なるものです。ソース管理プロジェ クHはディレクトリ構造に依存しているのに対して、Authentic Desktop プロジェクHはディレクトリ 構造から独立した論理的な構造となっています。

詳細に関しては次のセクションを参照してください: ソース管理。

ソース管理から開く

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

新規プロジェクトは以下のとおり作成することができます:

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

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

ソース管 理 のシンボル

フォルダーやファイルにはシンボルが表示されており、それぞれ以下に類似した意味を持ちます。

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

ソース管理を有効にする

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

|--|

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

最新バージョンの取得

(プロジェクト | ソース管理 メニュー内の)最新バージョンの取得コマンドにより、現在作業中のディレクトルで選択されたファイルの最新版をノーズ管理から取得します。ファイルは読み取り専用として取得され、チェングアウトはされません。このコマンドは Get コマンド と同じように作動しますが、Get ダイアログは表示しません。 選択されたファイルが現在チェックアウトされている場合、ソース管理システムがどのようにこのような状況を扱うかにより取られるアクションは異なります。通常、ソース管理システムは、チェックアウトされたファイルを置き換える、マージする、またはそのままにするかを問います。

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

フォルダーの取得

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

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



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

Changed files			
These files have been modified by an external application:			
C:\LocalWorkspace\Grouping\Persons.xml	-		
	~		
You can reload the file, close the document or ignore the modification			
Reload Close Cance	*		
	///		

高度な取得オプション

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

	—X —
	ОК
	Cancel
Make writable	Help
	Make writable

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

チェックアウト チェックイン

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

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

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



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



編集の変更を保存、または、拒否する

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

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

チェックアウト

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

Source Control - Check Out	
Files C:\LocalWorkspace\Grouping\Persons\Persons.xml C:\LocalWorkspace\MyProject.spp C:\LocalWorkspace\QuickStart\QuickStart.css C:\LocalWorkspace\QuickStart\QuickStart.xml C:\LocalWorkspace\QuickStart\QuickStart.xsd	OK Cancel Select All Advanced
Checkout local version Comment	li.

チェックイン

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

Source Control - Check In	
Files C:\LocalWorkspace\MyProject.spp C:\LocalWorkspace\QuickStart\QuickStart.xml C:\LocalWorkspace\QuickStart\QuickStart.xsd	OK Cancel Select All
Comment	Differences

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

チェックアウトの取り消し

チェックアウトされたファイルが存在し、編集の変更が加えられており、変更を拒否する場合、コマンド「プロジェクト | ソース管理 | チェックアウトの取り消し」を使用してレポジトリ内で保存されて

いるドキュメントのバージョンに戻ります。

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

4 - <mark>8</mark> 859	チェックイン状態。チェックアウト可能。
	他のユーザーによりチェックアウトされた状態。チェックアウト不可能。
5	ローカルにチェックアウト済み。編集してチェックイン可能。

ソース管理へ追加

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



ソース管理にファイルが追加されると(プロジェクトフォルダー階層構造ではな ()ローカルフォルダー 階層構造がレポジトリ内でレプリケートされます。ファイルがサブフォルダー内のローカルフォルダー内 で X レベル下にあると、ファイルの親フォルダー他の祖先フォルダーは自動的にレポジトリに追加され ます。

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

ソース管 理 シンボル

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

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

💕 | ローカルにチェックアウト済み。 編集 してチェックイン可能。

ソース管理から削除

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

ソース管理から共有

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

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



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

Share to \$/		—
File to share: QuickStart.xml QuickStart.css QuickStart.css	Projects: \$/QuickStart Gravitional Gravitional Gravitional	Close Share View Help
QuickStart.xsd	QuickStart	
List files of type: Relevant Masks (*.*)	Branch after share	

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

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

Change Source (Control	×
Local Path:	C:\LocaWorkspace	Browse
Scc Provider:	Microsoft Visual SourceSafe	Select
Server Name:	C:WSSRepository	Bind
Server Binding:	"\$7", ААААААА	Unbind
Logon ID:	ΑΔΑΑ]
Connected:		
	OK Cancel	

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

履歴の表示

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

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

ļ	🔋 History of	\$/MyProject.spp)		- • •
ł	History: 3 item	IS			Close
	Version	User	Date	Action	
	3	ala	1/24/1312:54p	Checked in \$/	View
	2 1	ala ala	1/22/13 1:06p 1/16/13 3:38p	Checked in \$/ Created	Details
			·		Get
					Check Out
					Diff
					Pin
					Rollback
					Report
					Help

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

閉じる:このダイアログボックスを閉じます。

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

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

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

チェックアウト:前バージョンのファイルをチェックアウトすることができます。

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

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

ロールバック:選択されたバージョンのファイルへロールバックします。

レポート:プリンター、ファイル、またはクリップボードへ送ることができる履歴レポートを作成します。 ヘルプ:ソース管理プロバイダープラグインのオンラインヘルプを開きます。

差分の表示

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

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

Difference Op	otions				—
Compare:	\$/MyP	roject.spp	Bro	owse 🔻	ОК
To:	C:\Loc	alWorkspace (MyProje	Br	ows <u>e</u> ▼	Cancel
Format		SourceSafe projects Windows folders		orojects ders	
O Unix	June				Project
🔲 Ignore w	hite spa	ce 🛛	Igno	re case	Advanced >>

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

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

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

Differences for \$/Grouping/Persons/Persons.xml	
Pa 🗛 🏦 🖀 💷 🖷 🐙 🖏 🔂 😵 💿	
🗊 \$/Grouping/Persons/Persons.xml	C:\LocalWorkspace\Grouping\Persons\Persons.xml
1	1
2)1/XMLSchema-instance" xsi:noNamespaceSchema	2 XMLSchema-instance" xsi:noNamespaceSchema
<pre>3plepartment="Administration" grade="C"/></pre>	3 artment="Administration" grade="B"/>
4 epartment="Administration" grade="D"/>	4 "rtment="Administration" grade="D"/>
5 artment="Administration" grade="C"/>	5 ment="Administration" grade="C"/>
<pre>6 rtment="Marketing" grade="B"/></pre>	6 <pre>ient="Marketing" grade="B"/></pre>
<pre>7 rtment="Marketing" grade="C"/></pre>	7 <pre>ient="Marketing" grade="C"/></pre>
8 artment="IT & Technical Support" grade="	8 ment="IT & Technical Support" grade="
9 epartment="IT & Technical Support" grade	9 "rtment="IT & amp; Technical Support" grade:
10 iepartment="IT & Technical Support" grad	10 artment="IT & Technical Support" grad
11 ment="IT & Technical Support" grade="D"/	11 t="IT & Technical Support" grade="D"/:
12 artment="Engineering" grade="C"/>	<pre>12 ment="Engineering" grade="B"/></pre>
<pre>13 department="Engineering" grade="D"/></pre>	13 :partment="Engineering" grade="D"/>
< >	٠ F
Deleted Text Changed Text Inserted Text Ln 3, Col 79	

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



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

プロパティの表示

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

このコマンドは、単一のファイルのために有効化されていることに注意してください。

\$/Grouping/Persons/Persons.xml	x
General Check Out Status Links Paths	
Name: \$/Grouping/Persons/Persons.xml	
Type: Unicode (UTF-8)	
Size: 2150 bytes 28 lines	
Latest: Version: 6 Date: 1/25/13 3:22p	
Comment:	
Close Report Help	

このコマンドは単一のファイルに対してしか使用できないことに注意してください。

ステータスを最新の状態に更新

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

ソース管理マネジャー

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

ソース管理の変更

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

ソース管理の変更 ダイアログ内でソース内で管理システム (SCC *プロバイダー*)、ローカルフォルダー (*ローカルパス*)、およびレポジトリバインド (*サーバー名 とサーバーのバインド*).を変更することができます (*下のスクリーンショット*)。 現在のバインドのバインドを解除した後のみに設定を変更することができます。「パインドの解除」 ボタンを使用して現在のバインドを解除します。この後、全ての設定を編集することができます。

Change Source (Control	—
Local Path:	C:\LocalWorkspace	Browse
Scc Provider:	Microsoft Visual SourceSafe	Select
Server Name:	C:WSSRepository	Bind
Server Binding:	"\$/", АААААААА	Unbind
Logon ID:	АААА	
Connected:		
	OK Cancel	

ソース管理の変更設定は、以下のとおりです:

- 1. 「参照」ボタンを使用してローカルフォルダーを参照し、インストールされたソース管理シス テムから「選択」ボタンを選択します。
- 2. レポジトリデータベースにローカルフォルダーをバインドすることができます。「**バインド」**ボタン をクリックしてこれを行います。ソース管理システムの接続ダイアログが表示されます。
- 3. *ログイン ID* を入力すると ソース管理システムにパスされます。それ以外の場合、接続ダイ アログ内に詳細を入力します。
- 4. ローカルフォルダーにバインドするレポジトリ内のデータベースを選択します。 この設定 は、1 つ以上のダイアログを完成する必要がある可能性があります。
- 5. 設定が作成された後、「OK」ボタンをクリックしてください。

11.3.7 プロジェク Hこファイルを追加



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

11.3.8 プロジェク Hこグローバルリソースを追加

「クロジェクト」プロジェクトにグローバリレリノースを追加」コマンドにより、グローバリレリノースを追加ダイアログが表示され、 プロジェクトは追加するファイルまたはフォルダーのグローバリレリノースを選択することができます。ファイル型のグローバリレリー スが選択されると、<u>プロジェクトプロパティ</u>ダイアログボックスで定義されているファイル拡張子をベースに、ファイルが適切 なフォルダーは追加されます。フォルダー型のグローバリレリノースが選択されると、そのフォルダーがファイルを閉(ダイアログにて 開かれ、ファイルを選択するよう促されます。選択されたファイルは、<u>プロジェクトプロパティ</u>ダイアログボックスで定義され ているファイル拡張子をベースに、適切なフォルダーに追加されます。グローバリレノースに関する詳細は、ドキュメンテーション のグローバリレノースのセグションを参照《たさい。

11.3.9 プロジェク Hこ URL を追加



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

11.3.10 プロジェク Hこアクティブなファイルを追加



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

11.3.11 プロジェクトにアクティブならびに関係するファイルを追加



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

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

11.3.12 プロジェク Hこアクティブならびに関係するファイルを追加



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

- **メモ:** プロジェクトフォルダーをドラッグして、プロジェクトフォルダー、または、プロジェクト内の他の場所にドロップすることができます。更に、フォルダーをWindows (ファイル) Explorer からドロップして、プロジェクトフォルダーにドロップすることもできます。
- **メモ:** プロジェクトフォルダーは緑で示され、<u>外部フォルダー</u>は黄色で示されています。

11.3.13 プロジェク Hこ外部フォルダーを追加

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

メモ: 外部フォルダーは黄色で、プロジェクトフォルダーは緑で表示されます。

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

外部 フォルダーをプロジェクト追加する

外部フォルダーをプロジェクトへ追加するには、以下の操作を行ってください:

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

フォルダの参照
Altova
Examples
CoppIDEPlugin
🕀 🔂 🔁 CSharpIDEPlugin
🕀 🔂 🕀 🔁 ExampleSite
🛅 Import
🗄 🛅 IndustryStandards
MapForce
Office2007
Tutorial
🔲 🕀 🛅 XBRL Examples

選択されたフォルダーがプロジェクトウィンドウに表示されます。



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



フォルダーのコンテンツをフィルタリング

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

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



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



これで、(上の例に従うと)プロジェクトウィンドウには XML とXSD ファイルだけが表示されるようになります。

外部 フォルダーの検証

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

1. 外部フォルダーからチェックを行うファイルを選択します。

2. フォルダーをクリックして、**整形式のチェック」** または **検証」** アイコンをク リックします (又はそれぞれ **F7**)か **F8**」を押下)。フォルダー内に表示されている全てのファ イルがチェックされます。チェックされたファイルに不備があったり妥当でない場合、そのファイル がメインウィンドンで開かれ、編集を行うことができます。

プロジェクトフォルダーの更新

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

外部 フォルダーやファイルの削除

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

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

更新」を選択してください。

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

11.3.14 プロジェクトに外部ウェブフォルダーを追加

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

外部 Web フォルダーをプロジェクトに追加する

外部 Web フォルダーをプロジェクトに追加するには、以下を行います:

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

プロジェク	トに Web フォ	ルダーを追加		? <mark>×</mark>
ファイル URL(U):			
ログイン情報				
ユーザー(E):	MyDocs	パスワード(W): ••••••	アプリケーション間で パスワードを保存	
利用可能な	ファイル			
サーバー	http://viets	pstest/	▼ ₹	\$照(B)
Micros	oft(R) SharePoir	nt(R) Server		
I				
1			新規フォルダー(N)	削除(L)

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

プロジェクトに Web フォルダーを追加 ? ご
סדיוע URL (U) : http://vietspstest/News
口华心情報
ューザー(E): MyDocs パスワード(W): •••••• アブリケーション間で パスワードを保存
利用可能なファイル
サーバー URL http://vietspstest/ - 参照(B)
Microsoft(R) SharePoint(R) Server
SiteCollectionDocuments
SiteCollectionImages
WorkflowTasks
新規フォルダー(N)
ファイルダイアログにスイッチ(E) グローバルリソースにスイッチ(G) 開く キャンセル //

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





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

フォルダーのコンテンツをフィルタリング

フォルダーのコンテンツをフィルタリングするには、フォルダーを右クリックして、コンテキストメニューから 「プロパティ」を選択します。表示されるプロパティダイアログにて、ファイル拡張子フィールドをクリック し、表示させたいファイル種類の拡張子 (例えば、XML や XSD ファイル)を入力します。セミコロンに よりファイルの種類を分けることができます (例:xml; xsd; sps)。 プロジェクトウィンドウには、指定され た拡張子を持つファイルだけが表示されるようになります。

フォルダーの検証と整形式のチェック

フォルダーに収められているファイルの整形式をチェック、または検証を行うには、フォルダーを選択し

て、「整形式のチェック」 し または検証 し アイコンをクリックします (それぞれ「F7」と「F8」 に対応)。フォルダーに表示 されている全てのファイルがチェックされます。ファイルに不備があったり受 当でない場合、そのファイルがメインウィンドウで開かれ、編集を行うことができるようになります。エラ ーの修正を行い、再度チェックを行ってください。Ctrl キーを押下しながらファイルをクリックすることで、 個別にファイルを選択することができ、「F7」または「F8」を押下することで、選択されたファイルが チェックされます。

プロジェクトフォルダーのコンテンツを更新

ファイルの追加など、ウェブフォルダー内のファイルは常に更新される可能性があります。フォルダーの

ビューを更新するには、外部フォルダーを右クリックして、コンテキストメニューから最新の状態に更新オプションを選択します。

フォルダーとファイルを削除

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

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

Microsoft® SharePoint® Server 上のフォルダー

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

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



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

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

以下の点に注意してください:

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

🚰 AutoCalc.sps	Persons.xml	ExpReport.xml
	N	

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

11.3.15 スクリプトの設定

スクリプトプロジェクトは、Authentic Desktop プロジェクトに以下のとおり割り当てられます:

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

スクリプト
スクリプト プロジェクト ファイル ▼プロジェクト スクリプトをアクティブにする
<pre>{Altova¥XMLSpy2018¥Examples¥SampleScripts.asprj ▼</pre>
🥅 Examples.spp に対する相対パスにする
自動スクリプト処理 Authentic Desktop プロジェクトのロード時にオートマクロを実
OK キャンセル

- 3. *プロジェクト スクリプトの有効化* チェックボックスをチェックし、必要なスクリプトプロジェクト (asprjファイル)を選択します。Authentic Desktop プロジェクトがロードされる時に、自動マクロを実行する場合、自動マクロを実行するチェックボックスをチェックします。
- 4. 「OK」をクリックして完了します。
- **メモ:** Authentic Desktop プロジェクトのスクリプトプロジェクトを無効化(割)当てを解除)するには、プロジェクトスクリプトを有効化するチェックボックスのチェックを解除します。

11.3.16 プロパティ



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

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

プロパティ			? X
名前: Invoice	⊳EU		ОК
リアイル孤張子: xml;svg			キャンセル
使証 		•	参照 ウインドウ
×MLファイルの ×SL 変換			
🔽 使用する XSL:	C:\Invoices\reports.xslt	•	参照 ウィンドウ
- XML ファイルの XSL:FO 変	换		
🔽 使用する XSL:	C:\Invoices\reportsF0.xslt	•	参照 ウィンドウ
~XMLファイルの XQuery/Up	idate 変換		
■この XQueryを使用:		•	参照 ウインドウ
_XSL/XQuery/Update 変搏	eのための入力 XML		
📃 使用する XML:		•	参照 ウインドウ
~XSL/XQuery/Update 変技	急のファイル		
☑ 保存先フォルダー:	C:\Invoices\Reports	•	参照
□ファイル拡張子:	.html		
Authenticピュー			
🔲 設定を使用:		•	参照 ウインドウ

設定

<u>ファイル拡張子</u>

ファイル拡張子 設定は、個別のフォルダーのために有効化 されており、プロジェクトフォルダーのため には有効化 されていません。ファイルがプロジェクトに追加 されると 定義 されているファイル拡張子を 持つフォルダーに追加 されます。MyReport.xml という名前のファイルがプロジェクトに追加 されたと仮 定します。(上のスクリーンショット内で示されるとおり).xml ファイル拡張子が Invoices-EU フォルダー で設定 されている場合、MyReport.xml が Invoices-EU フォルダーに追加 されます。 XML ファイル に追加 するフォルダーが1つ以上存在する場合、個別の XML ファイルを(プロジェクトではなく)直接 フォルダーに追加します。

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

検証

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

XML ファイルの XSL 変換

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

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

<u>XSL ファイルのXSL 変換</u> XML ファイルは、フォルダー内でのXSLT ファイルのXSLT 変換に使用されます。

<u>XML ファイルの XQuery 変換</u> XQuery ファイルは、フォルダー内の XML ファイルの XQuery 実行に使用されます。

<u>XSL 変換の保存先ディレクトリ</u> XSLT 変換 ,そして、任意で結果 ドキュメントのファイル拡張子保存先のディレクトリ

Authentic ビュー

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

プロジェクトプロパティの注意点

以下の点と優先事項に注意してください:

- 変換、または、XSLT/XQuery 変換がプロジェクトフォルダー コンテキストメニューにより実行される場合、このダイアログで指定される検証または変換ファイルにXMLファイル内の割り当てよりも優先順位が与えられます。更に、個別のプロジェクトフォルダーのために指定されている設定に、祖先フォルダーのために指定されている設定よりも優先順位が与えられます。
- プロジェクトの複数フォルダー内で1つのファイルが存在する場合、異なるフォルダー内の異なる検証または変換ファイルは、異なるフォルダーに割り当てられ、プロジェクトの外部でファイルが処理される場合、割り当てを設定することができます。これらの設定を指定する方法は以下のとおりです:割り当てをしようするプロジェクトフォルダー内のファイルの場所を指定

し、プロパティを選択します。表示されるダイアログから、現在のフォルダー内で使用されて いる設定(ファイルが存在するプロジェクトフォルダーである現在のフォルダー)をデフォルトとし て使用するを選択します。オプションが無効化された場合、現在のフォルダーの設定は、既 にデフォルトの設定として使用されることを意味します。デフォルトではないプロジェクトフォル ダー内のファイルインスタンスを選択すると、オプションが有効化され、デフォルトの設定をこの フォルダーの設定に切り替えることができます。ファイルにローカルの割り当て(ファイル自身 内の割り当て)が存在する場合、ローカルの割り当てが使用され、デフォルトのフォルダー設 定は無視されます。

Properties		? 💌
File name:	C:\Examples\ipo.xml	т ок
	Use settings in current folder as default	Cancel

11.3.17 最近使用されたプロジェクト

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

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

11.4 XML × 그-

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

3	整形式のチェック(W)	F7	
3	XML 検証(V)	F8	

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

11.4.1 整形式のチェック



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

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



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

メッセージ	×
○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	1≢₩.
	/~CE

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

プロジェクトウィンドウから検証を行う

「検証」コマンドをファイル、フォルダー、ファイルのグループに適用することもできます。プロジェクトウィンドウ内で必要なファイルまたはフォルダーを(「XML | 検証」または「F8」をクリックして)選択します。プロジェクト内の有効ではないファイルが開かれ、メインウィンドウでアクティブになります。「ファイルが有効ではありません」 エラーメッセージが表示されます。

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

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

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

F8

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

ドキュメントが有効な場合、メッセージウィンドウ内に検証の成功メッセージが表示されます。

メッセージ	x
💦 😋 ファイル C:¥Altova¥XMLSpy¥Examples¥NanonullOrg.xml は妥当です。	A
(ē)	
9	
9	
5	
5	∇

そうでない場合、エラーの内容がメッセージウィンドウに表示されます(以下のスクリーンショットを参照)。リンクをクリックすることで、XML ファイル内のエラーが発見された箇所へジャンプすることができます。

メモ: :出力ウィンドウには9つのタブがあります。検証の結果はアクティブなタブに表示されます。そのため、最初のタブにて整形式のチェックを行った後、2つ目のタブに切り替えて、別のドキュメント検証を行うといったことができます。2番目のドキュメントを検証するために、チェックを実行する前に、タブ2(または、タブ3)に切り替えます。タブを切り替えない場合、タブ4(またはアクティブなタブ)が最新の検証のフォーミュラにより上書きされます。

プロジェクトウィンドウからの検証

検証 コマンドはアクティブなプロジェクト内部 にあるファイル、フォルダー、またはファイルのグループに対しても適用することができます。目的のファイルやフォルダーをプロジェクトウィンドウにて (クリックにより) 選択し、メニューオプションから XML | XML 検証」を選択するか、「F8」を押下してください。プロジェクト内の妥当ではないファイルがメインウィンドウに表示され、ファイルが妥当でない旨のメッセージが表示されます。

RaptorXML 2018 を使用して検証を自動化する

RaptorXML は、XML 検証、XSLT 変換、とXQuery 変換のための Altova のスタンドアロンアプリケーションです。Java プログラムと NET 内の COM インターフェイスを使用 して、アプリケーションコマンドラインから使用 することができます。RaptorXML を使用 すると 検証 のタスクを自動化 することもできます。例えば、ドキュメントのセット上で検証を実行 するために RaptorXML を呼び出 すバッチファイ

ルを作成し、テキストファイルに出力を送ることができます。詳細に関しては、<u>RaptorXML</u><u>ドキュメン</u> <u>テーション</u>を参照ください。

11.6 XSL/ XQuery メニュー

XSL 変換言語を使えば、どのように XML ドキュメントが他の XML ドキュメントやテキストファイルへ変換 されるかを指定 することができます。XSLT ドキュメントにより生成 される XML ドキュメントの 1つに、 PDF の生成処理を行うために使用 される FO ドキュメントがあります。XMLSpy には XSLT 1.0 なら びに XSLT 2.0 に対応 した XSLT プロセッサーが内蔵 されており、お使いのシステムにある FO プロセッ サーと連携 することで、XML ファイルから様々な形式の出力を生成 することができます。XMLSpy イン ターフェースから、直接 FO プロセッサーを使用 するには、FO プロセッサーの場所をオプションダイアロ グ (「ツール | オプション」)の XSL セクションにて指定 する必要 があります。

上記の変換を行うためのコマンドは、全てXSL/XQueryメニューに収められています。更に、このメ ニューからは Altova XSLT/XQuery デバッガーを使用するためのコマンドへもアクセスすることができま す。



11.6.1 XSL 変換

🕮 F10

XSL/XQuery | XSL 変換」コマンドにより、割り当てられた XSLT スタイルシートを使って XML ド キュメントの変換を行うことができます。変換は、内蔵されている適切な Altova XSLT エンジン (XSLT 1.0 スタイルシートに対しては Altova XSLT 1.0 エンジンを、XSLT 2.0 スタイルシートに対しては Altova XSLT 2.0 エンジンを使用)、Microsoft から提供されている MSXML モジュール、または外部の XSLT プロセッサーを使用することで行うことができます。このコマンドにより使用されるプロセッサーは、オプ ションダイアログ (「ツール | オプション」)の XSL セクションにて指定することができます。

XML ドキュメントに XSLT スタイルシートへの参照が含まれている場合、このスタイルシートが変換に 使用 されます (XML ドキュメントがプロジェクトに含まれている場合 <u>プロジェクトプロパティ</u>ダイアログに て、フォルダーごとに XSLT スタイルシートを指定することができます。変換を行うプロジェクトフォルダー やファイルを右 クリックして **XSL 変換」**を選択してください)。XSLT スタイルシートが XML ファイルに 割 13当 てられていない場合、使用する XSLT スタイルシートを指定するよう促されます。ファイルの選 択は (参照 ボタンをクリックして)グローバルリソースや URL、(ウィンドウボタンをクリックして) XMLSpy にて既に開かれているファイルからも行うことができます。

RaptorXML 2018 を使用 した自動検証

RaptorXML は、XML 検証、XSLT 変換、とXQuery 変換のための Altova のスタンドアロンアプリケーションです。Java プログラムと NET 内の COM インターフェイスを使用 して、アプリケーションコマン ドラインから使用 することができます。このため、RaptorXML を使用 して XSLT 変換 タスクを実行 する ことができます。例えば、ドキュメントのセット上で XSLT 変換を実行 するために RaptorXML を呼び 出 すバッチファイルを作成 し、テキストファイルに出力を送ることができます。詳細に関しては、 RaptorXML ドキュメンテーション を参照 ください。

ZIP ファイルへの変 換

変換の結果を.docx 拡張子を持った Open Office XML (OOXML) や ZIP ファイルに収める場合は、 出力ファイルのファイルパスに以下のような ZIP プロトコルを指定する必要があります:

filename.zip|zip/filename.xxx

filename.docx|zip/filename.xxx

メモ: 変換を行う前に、ディレクトリ構造の作成を行わなければならない場合もあります。Open Office XML アーカイブ形式で変換を行う場合、(例えば .docx のような)トップレベルの OOXML ファイル を作成するために、アーカイブファイルの ZIP 化を行う必要があります。

11.6.2 XSL-FO 変換

Ctrl+F10

FO は、印刷 することを前提 にしたドキュメントを記述 するための XML フォーマットです。Apache XML プロジェクトの FOP のような FO プロセッサーでは、FO ファイルを入力 することで PDF を出力 として 生成 することができます。そのため、XML ドキュメントから PDF を生成 するには 2つのステップを踏むこ とになります。

- XSLT (または XSL-FO)スタイルシー を使うことで XML ドキュメントを FO ドキュメントへ変換します。
- 2. FO プロセッサーにより FO ドキュメントを処理することで、 PDF (またはその他の出力)を生成します。

'XSLT/XQuery | XSL:FO 変換」コマンドによりXML ドキュメントや FO ドキュメントを PDF に変換 します。

- XSL:FO 変換コマンドをソース XML ドキュメントに対して実行すると、上にある両方のステップが順に処理されます。FO への変換に必要な XSLT (または XSL-FO)スタイルシートが XML ドキュメントから参照されていない場合、変換に使用するスタイルシートを指定する必要があります(以下のスクリーンショットを参照)。ファイルの選択は(参照 ボタンをクリックして)グローバルリソースや URL、(ウィンドウボタンをクリックして) XMLSpy にて既に開かれているファイルからも行うことができます。XML から XSL-FO への変換が、オプションダイアログ(「ツール」オプション」)の XSL タブ にて指定された XSLT プロセッサーによい行われます。デフォルトでは、Authentic Desktop XSLT プロセッサーが選択されます。結果 として生成される FO ドキュメントが、オプションダイアログ(「ツール」オプション」)の XSL タブ にて指定された FO プロセッサーによい変換されます。
- XSL:FO 変換 コマンドが FO ドキュメントに対して実行されると ドキュメントがオプションダイ アログ(「ツール|オプション」)の XSL タブにて指定された FO プロセッサーにて処理されます。

XSL:FO 変換出力

XSL:FO 変換コマンドを選択すると XSL:FO 出力を選択ダイアログが表示されます(以下のスクリ ーンショットを参照)。アクティブなドキュメントが XSLT 割 リ当 てを持たない XML ドキュメントの場合、 XSLT ファイルの選択を促されます。



FO プロセッサーの出力は FOP ビューアーを使用することで、画面上にて直接確認する他にも、以下のフォーマットにより出力ファイルを生成することができます:PDF、テキスト、XML エリアツリー、MIF PCL、または PostScript。FO プロセッサーから得られたメッセージ出力するオプションにより、(i)プロセッサーの標準出力メッセージや(ii)プロセッサーのエラーメッセージをメッセージウィンドウにて表示することができます。これら2つのオプションを有効にするには、ダイアログ下部にある対応するチェックボックスにチェックを入れてください。

メモ:

- Apache XML プロジェクトの FOP プロセッサーをインストールするオプションを解除していない限以次のフォルダーにインストールされます:C:\ProgramData\Altova \SharedBetweenVersions.インストールされるとパスは自動的にオプションダイアログ(ツール|オプション)内のXSL ダブに使用するFO プロセッサーとして入力されます。使用するFO プロセッサーにパスを設定することができます。
- XSL:変換コマンドは、メインウィンドウにてアクティブとなっているファイルだけではなく、アクティ ブなプロジェクトにて選択することのできるファイルやフォルダーに対しても使用することができ ます。右クリックで表示されるコンテキストメニューから XSL:FO 変換を選択してください。選 択されたプロジェクトフォルダーに割り当てられた XSLT スタイルシートが使用されます。

11.6.3 XSL パラメーター / XQuery 実行

「XSL/XQuery | XSL パラメーター XQuery 変数」コマンドにより、XSLT 入力パラメーター XQuery 外部変数ダイアログが表示されます(スクリーンショットを参照)。XSLT スタイルシートに渡すパラメー ターの名前や、XQuery ドキュメントへ渡す XQuery 変数の名前と対応する値を入力します。 XMLSpy においてこれらのパラメーターは以下のように使用されます:

XSL/ XQuery メニューの XSL 変換コマンドが XML ドキュメントの変換に使用される際、ダイアログ内にて保存されたパラメーター値が選択された XSLT ドキュメントへ渡され、変換に使用されます。
 XSL/ XQuery メニューの XQuery 実行コマンドが XQuery ドキュメントの処理に使用される際、ダイアログにて現在保存されている XQuery 外部変数の値が XQuery ドキュメントに渡され、実行に使用されます。

メモ: XSLT入力パラメーター XQuery外部変数ダイアログにて入力されたパラメーターや変数は、 内蔵のAltova XSLT エンジンに対してのみ使用されます。MSXML やその他の外部エンジンに対してパラメーターは渡されません。このため、MSXML、または、構成済みの外部エンジンを使用する場合、これらの値がエンジンにパスされます。

★モ: XSLT 入力パラメーター XQuery 外部変数 ダイアログ内で XSLT パラメーター、または、外部 XQuery 変数 を定義することはエラーではありませんが、XSLT/XQuery ドキュメント、または、 変換内では使用 されません。

XSLT パラメーターの使用

パラメーターに対して入力した値は引用句無しの XPath 条件式や、引用句無しのテキスト文字列 となります。 アクティブなドキュメントが XSLT ドキュメントの場合、 XSL から取得ボタンが有効になり ます。 このボタンをクリックすることで、 XSLT にて宣言されているパラメーターが、 デフォルト値とともにダ イアログに入力されます。 この操作により、宣言されたパラメーターを素早く入力し、 デフォルト値を必 要に応じて変更することができます。

(SLT 入力パラ	ラメーター/XQuery 外部変響	牧		
名前 Country		XPath		A
Country		France		
				~
削除(D)	×SLから取得(X)		ОК	キャンセル

メモ:パラメーター値がダイアログに入力されると明示的に削除、または、アプリケーションが再起動

されるまですべての変換で使用されます。ダイアログ内に入力されるパラメーターはそのセッションのア プリケーションレベルで指定され、この時点からIDEを使用して実行される各変換のためのそれぞれの XSLT ドキュメントためにパスされます。 これは以下を意味します:

- パラメーターは特定のドキュメントに割り当てられているのではありません。
- ダイアログにて入力されたパラメーターは、Authentic Desktopの終了時に削除されます。

XSLT パラメーターの使用例 国名とそれぞれの首都を含む XML ドキュメントは以下のとおりです:

<docum ent>
 <countries>
 <country nam e="USA" capital="W ashington DC"/>
 <country nam e="UK" capital="W ashington DC"/>
 <country nam e="France" capital="Paris"/>
 <country nam e="France" capital="Moscow"/>
 <country nam e="China" capital="Beijing"/>
 </countries>
</docum ent>

次のXSLTドキュメントは、XMLファイルから国名と首都を表示するXMLドキュメントを生成します。 country という名前を持つパラメーターの値として名前を入力することにより国名は選択されます(以下では黄色のハイライトにより表示されています)。

この XSLT ドキュメントが上記の XML ドキュメントで実行 されると 結果は以下のようになります:

<country><nam e>USA</nam e><capita DW ashington DC</capita D</country>

XSLT 入力 パラメーター XQuery 外部 変数 ダイアログ内 で country といろ名前のパラメーターが作成 され、値が与えられると新規のパラメーターの値は、XSLT 変換内のパラメーターにパスされます(*上のスクリーンショット参照*)。 この値は変換のために XSLT スタイルシート内のパラメーター country に パスされます。このようこして、異なるパラメーターへ異なる値をランタイムにパスすることができます。

メモ:

- XSL:FO 変換コマンド(XSL/XQuery | XSL:FO 変換」)を使用する場合、ダイアログにて入力 された値はスタイルシートへ渡されません。これらのパラメーターを PDF 出力にて使用するには、 まず XSLT 変換コマンド(XSL/XQuery | XSL 変換」)を使用して XML から FO ドキュメントへ の変換を行い、その後 XSL:FO 変換コマンド(XSL/XQuery | XSL:FO 変換)によりFO から PDF への変換を行う必要があります。
- 内蔵されている Altova XSLT エンジン以外の XSLT プロセッサーを使用する場合、ダイアログにて 入力されたパラメーターは外部プロセッサーへ渡されません。

外部 XQuery 変数の使用

外部 XQuery 変数に対して入力する値は、引用句無しの XPath 条件式か、引用句により分離さ れたテキスト文字列 となります。外部変数のデータ型は、XQuery ドキュメントの変数宣言にて指定 されます。

XSLT 入力パラメ	ペーター/XQuery 外部変	数		
名前 first		XPath 'Peter'		A
discount		doc('c:\PriceList.xml')/	PriceList/Discoun	t
				~
削除(D)	×SLから取得(X)		ОК	キャンセル

メモ:外部 XQuery 変数 がダイアログにて一度入力 されると そのエントリーが明示的 に削除 される かアプリケーションが再起動 されるまで、その後の実行でもそれらの変数 が使用 され続けます。 ダイア ログにて入力 された変数 はアプリケーションレベルで指定 されており、その後行 われる実行のたびに 関連する XQuery ドキュメントへ渡 されます。 これは以下のように言い換えることもできます:

- 変数は特定のドキュメントに割り当てられているのではありません。
- ダイアログにて入力された変数は、アプリケーション (Authentic Desktop)の終了時に削除されます。

外部 XQuery 変数の使用例

以下の例では、変数 \$first が XQuery ドキュメントにて宣言され、FLWOR ステートメントの return 句にて使用されます:

xquery version "1.0"; declare variable \$first as xs string external; let \$last = "Jones" return concat(\$first, " ", \$last)

(XSLT 入力パラメーター XQuery 外部変数ダイアログにて入力された)外部変数に Peter がセットされている場合、XQuery により Peter Jones という値が返されます。以下の点に注意してください:

- XQuery ドキュメントの変数宣言にある external キーワードにより、この変数が外部変数であるということが認識されます。
- 静的な変数の型定義はオプションです。変数の宣言時に変数のデータ型が定義されていない場合、変数の値は xs:untypedAtomic に割り当てられます。
- XQuery ドキュメントにて外部変数が宣言されているが、その変数名に対して外部変数が与えられない場合、エラーとないます。
- 外部変数が宣言され、XSLT入力パラメーター XQuery外部変数ダイアログにて入力された場合、実行されるのは XQuery ドキュメントのスコープ内となります。XQuery ドキュメントにて新たな変数が同じ名前で宣言されている場合、その変数によりスコープ内の外部変数が一時的にオーバーライドされます。例えば、以下にある XQuery ドキュメントでは、(外部変数により\$first に対して Peter という値が渡されているにも関わらず) Paul Jones という値が返されます。

xquery version "1.0"; declare variable \$first as xs string external; let \$first = "Paul" let \$last = "Jones" return concat(\$first, " ", \$last)

11.7 Authentic メニュー

Authentic View を使用することで、Altova StyleVision により作成 された StyleVision Power Stylesheet (sps ファイル)をベースにしたドキュメントの編集を行うことができます!スタイルシートには XML ファイルを Authentic View にて視覚的に表示するための情報が収められています。表示するための情報に加えて、StyleVision Power Stylesheet によりXML ファイルへデータを書き込むこともできます。データは XSLT スタイルシートに用意 されている機能により動的に処理され、その結果は Authentic View にて直ちに表示されます。

他にも StyleVision Power Stylesheet を使うことでデータベースにあるデータの編集を行うこともできます StyleVision Power Stylesheet にはデータベースへ接続するための情報が含まれており、データベースから得られたデータを Authentic View にて表示し、編集後のデータをデータベースへ書き込むことができます。

Authentic メニューコマンドには XML ドキュメントを Authentic View にて編集 するためのコマンドが収められています。のチュートリアルは、 Authentic View チュートリアル セクションを参照 ください。



11.7.1 新規ドキュメント

このコマンドにより Authentic View 内 で新規の XML ドキュメントテンプレートを開く事ができます。 XML ドキュメントテンプレートは、StyleVision Power Stylesheet (.sps ファイル)をベースにしています。 新規のドキュメントの作成 ダイアログ内で StyleVision Power Stylesheet (SPS ファイル)を選択 する事により開かれます (F O X O U - V O = V O L)。 SPS を選択して、「**OK」**をクリックすると SPS ファイルのために定義 された XML ドキュメントテンプレートが Authentic View 内で開かれます。

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

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

11.7.2 データベースの編集

「Authentic | データベースの編集 …」コマンドにより、データベースのビューを Authentic View にて開 くことができます。DB への接続方法や DB の表示方法、そして Authentic View にて許 されている編 集方法などは、StyleVision Power Stylesheet に記述 されます。データベースの編集 コマンドで開か れるのは、そのような DB をベースにした StyleVision Power Stylesheetです。このコマンドにより、DB への接続が確立され、DB のデータが (XML を介して) Authentic View にて表示 されます

データベースの編集 …コマンドにより、データベースデータの編集ダイアログが表示されます。

デ	ータベースデータの編集	×
	Publishing rixml teilite xmlresume xmlspec daisy dita Examples NCAXML News P3P	ОК \$+70-12ли
		参照

目的の SPS ファイルを選択 してください。この操作によりDB への接続が行われ、DB が編集できる 状態で Authentic View に表示 されます。Authentic View にて表示 されている DB ビューのデザイン は StyleVision Power Stylesheet に含まれています。

注意: データベースの編集 コマンドを使って DB をベースにしていない StyleVision Power Stylesheet や、StyleVision 2005 より 注前の StyleVision で作成 された DB ベースの StyleVision Power Stylesheet を開こうとした場合、エラーが表示 されます。

メモ: StyleVision Power Stylesheet は Altova StyleVision により作成 されます。

11.7.3 StyleVision スタイルシートの編集

「Authentic | StyleVision スタイルシート の編集」コマンドにより、StyleVision が起動され、直ちに tStyleVision Power Stylesheet の編集を始めることができます。このコマンドは StyleVision Power Stylesheet が XML ドキュメントに割り当てられている場合のみ有効化されている。

11.7.4 XML データの新たな行を選択し編集

***XML データの新たな行を選択し編集**」コマンドにより、IBM DB2 といた XML DB にあるテーブルから新たに行を選択することができます。新たな行は Authentic View に表示され、編集を行った後に DB への書き込みを行うことができます。

XML DB が XML データソースとして使用 される場合、XML データカラムのセルに収められている XML ドキュメントが Authentic View に表示 されます。XML データの新たな行を選択し編集 コマンドに より、XML カラム上の他のセル(または行)にある XML ドキュメントを選択することができます。このコ マンドを選択することで、XML フィールドの選択ダイアログが表示 され、XML カラムに含まれているテー ブルが表示 されます。

D •	INFO •	HISTORY	•
003	xml version="1.0" encoding="UTF-8" ? <c< td=""><td>INULLI</td><td></td></c<>	INULLI	
004	xml version="1.0" encoding="UTF-8" ? <c< td=""><td>INULLI</td><td></td></c<>	INULLI	
005	xml version="1.0" encoding="UTF-8" ? <c< td=""><td>INULLI</td><td></td></c<>	INULLI	

フィルタリングによりこのテーブルに表示 される内容を制限することができます。フィルタリングに使用されるのは SQL の WHERE 句です (例えば、CID>1002 といた WHERE 句の中に入る条件になります)。 更新ボタンをクリックして、ダイアログの表示内容を更新することができます。 上のスクリーンショットではフィルタリングされた結果を見ることができます。 目的の XML ドキュメントが表示 されているセルを選択して OK をクリックします。 選択されたセルの XML ドキュメントが Authentic View にロードされます。

11.7.5 XML 署名

関連付けられている SPS にて XML 署名が有効になっている場合、Authentic ビューで XML 署名」コマンドを使用することができるようになります。Authentic ツールバーにある XML 署名」ツール バーアイコン 💽 からも XML 署名 コマンドへアクセスすることができます。

検証と独自の証明書/パスワード

XML 署名」コマンドをクリックすると署名の検証処理が開始されます。ドキュメント内に署名が存在しない場合、その結果が XML 署名ダイアログに表示され、表示されているボタンから Authentic View ユーザーはドキュメントの署名を行うことができるようになります(以下のスクリーンショットを参照)。

🗟 XML Signature		X
☐ SXML Signature verific The file does not contained.	ation failed n any Signatures that could be verified	<u> </u>
		•
Sign Document	Remove Signature Select own Password	ОК

Authentic View にて独自の証明書 パスワードを選択することができるようにオプションが指定されている場合、 独自の署名を選択」または 独自のパスワードを選択」ボタンがダイアログにて表示されます。認証において証明書が使用されるかパスワードが使用されるかは、SPS デザイナーが署名の設定を行った際に指定されます。署名は証明書またはパスワードをベースにしたものになります。ダイアログにボタンが表示されいる場合、そのボタンをクリックすることで、Authentic View が証明書を選択、またはパスワードを入力することができるようになります。Authentic View ユーザーの選択はメモリ上に保管されており、そのセッションでのみ有効になります。証明書の選択またはパスワードの入力を行った後にドキュメントやアプリケーションが閉じられると証明書 パスワードの設定も、SPS に保存されたオリジナルの設定に戻されます。

検証ならびに認証情報

単一のドキュメントに対して検証処理が行われた場合、2つのケースが考えられます。認証情報 (署名または SPS にて)が利用できる場合、検証処理が直接実行され、その結果が表示されます (以下のスクリーンショットを参照)。

🚭 XML Signature			×
🔮 XML Signature verified	successfully		×
Sign Document	Remove Signature	Select own Password	ОК

認証情報は証明書の鍵情報または署名時に使用したパスワードとなります。XML ドキュメントが署名されたときに、証明書の鍵情報が署名の内に収められるか、またパスワードをベースにした署名の場合、パスワードが SPS の中に収められるかは、SPS デザイナーにより指定されます。 どちらの場合でも認証を行うことができ、検証処理は Authentic View ユーザーからの入力を必要とすることなく直

接実行されます。

もう1つのケースは、認証情報が署名の中に存在しない(証明書の場合)、または SPS ファイルの中に存在しない(パスワードの場合)というものです。この場合、認証情報(パスワードか証明書の場所)を入力するよう求められます(以下のスクリーンショットを参照)。

署名パスワード	×
署名の検証にはパスワードが必要です。	
[_
パスワードには最低 5 文字、最大 16 文字必要です。	
ОК +++>+	216

11.7.6 XML エンティティの定義

Authentic View にて使用するエンティティを(ドキュメントが DTD か XML スキーマをベースにしているか に関わらず)定義することができます。一度定義を行うと、これらのエンティティがエンティティ入力 ヘル パーならびにコンテキストメニューのエンティティを挿入サブメニューに表示されます。エンティティ入力 ヘルパーに表示されているエンティティをダブルクリックすると そのエンティティがカーソル位置に挿入さ れます。

テキスト文字列や XML フラグメント、その他の外部リソースを、ドキュメントの複数箇所で使用する場合、エンティティは便利な機能となります。目的のデータを表す短い名前のエンティティをエンティティの定義ダイアログにて定義します。エンティティを一度定義すると、そのエンティティをドキュメントの複数箇所で使用することができるようになります。この機能により、編集時間の短縮や、メンテナンスコストを大幅に軽減することができます。

ドキュメントで使用することのできるエンティティには、XML データ(テキスト文字列または XML ドキュメ ントのフラグメント)の解析対象実体(解析対象 エンティティ)と バイナリファイル(通常はイメージや 音楽などのマルチメディアオブジェクト)の様な非 XML データを扱う解析対象外実体(解析対象 外エンティティ)と大きく分けて2つの種類があります。各エンティティには名前と値が与えられていま す。解析対象エンティティの場合、エンティティはXML データのプレースホルダーとなります。エンティティ の値は XML データそのものか、XML データを含む xml ファイルへの URI となります。解析対象外エン ティティの場合、非 XML データファイルへの URI が値として入力されます。

エンティティを定義するには:

1. **Authentic | XML エンティティの定義** …」をクリックします。エンティティの定義ダイアログが 表示されます。

D	efir	ne Entities	;						
		[-			lun (no. l		
	6	Name		Туре	PUBLIC	Value/Path	 NDATA		UK
	8	nano_dc	▼	Internal		Nanonull, Inc			
	8	nano_eu	▼	Internal		Nanonull Europe, AG			Cancel
		nano_ma	•	Internal		Nanonull Partners, Inc			
	8	website	\bullet	Internal		http://www.nanonull.com/			Append
		branches	•	SYSTEM		branches.xml		E	lucest [
	8	logo	▼	SYSTEM		nanonull.gif	 GIF		Insert
									Delete
								_	
									///

- 2. エンティティの名前を Name フィール ドに入力します。この値がエンティティ入力ヘルパーに表示 される名前となります。
- Type フィールドのドロップダウンリストにてエンティティの種類を指定します。3つある種類から選択を行います。Internal エンティティでは、エンティティの実体であるテキストがXMLドキュメントの中に保管されます。PUBLIC や SYSTEM では、エンティティのリソースが、XML ファイルの外に配置され、それぞれ PUBLIC 識別子とSYSTEM 識別子を使用することで、リソースが特定されます。SYSTEM 識別子ではリソースの場所を表す URI が記述されます。PUBLIC 識別子は場所に依存しない識別子で、リソースの特定はプロセッサーにより行われます。PUBLIC 識別子とSYSTEM 識別子の両方を指定すると、PUBLIC 識別子により SYSTEM 識別子が使用されます。
- 4. Type に PUBLIC を選択した場合、リソースの PUBLIC 識別子を PUBLIC フィール ドに入力してく ださい。Type にて Internal または SYSTEM を選択した場合、 PUBLIC フィールドが無効になり

ます。

- 5. Value/ Path フィールドでは、以下のどれかを入力することができます:
- エンティティの種類が Internal の場合、エンティティの値となるテキスト文字列を入力してください。エントリーには引用句を入力しないでください。入力された引用句はテキスト文字列の一部として扱われます。
- エンティティの種類が SYSTEM の場合、リソースの URI を入力するか、参照ボタンを使用して、ローカルネットワークからリソースを選択してください。リソースが参照可能エンティティである場合、参照されるリソースは XML ファイル (つまり xml 拡張子を持ったファイル)である必要があります。他にも、GIFファイルのようなバイナリファイルを使用することもできます。
- エンティティの種類が PUBLIC の場合、このフィール ドこ SYSTEM 識別子を入力する必要があります。
 - NDATA エントリーにより、エンティティが解析対象エンティティではなく、適切なプロセッサーに送られるものだということがプロセッサーに伝えられます。そのため NDATA フィール ドは解析対象 外エンティティに対してのみ使用されます。

ダイアログ機能

対応するボタンをクリックすることでエンティティの追加、挿入、そして削除を行うことができます。カラムのヘッダーをクリックすることで、エンティティをアルファベット順に並び替え。1回クリックすることで昇順、 2回クリックで降順の並び替えとなります。ダイアログボックスの大きさならびにカラムの幅を変更することができます。

エンティティが 1度 XML ドキュメント内 で使用 されると、そのエンティティはロックされ、エンティティの編 集ダイアログにて編集することができなくなります。ロックされたエンティティは、最初のカラムにロックシ ンボルが表示 されます。エンティティのロックを行うことで、エンティティにおける XML ドキュメントの妥当 性が保証 されます (参照 されているエンティティが定義 されていない場合、ドキュメントは不正 となりま す)。

複製されたエンティティがある場合は、その旨が表示されます。

制約

- 他のエンティティ内に含まれているエンティティは、ダイアログ、Authentic View、XSLT出力にて解決 されず、そのようなエンティティのアンパサンドはエスケープされた形(つまり&)で表示されます。
- 外部エンティティは、そのエンティティがイメージである場合とENTITYまたはENTITIES型の属性値として入力された場合を除きAuthentic Viewにて解決されません。このようなエンティティは、SPSから生成されたXSLTにてドキュメントが処理されたときに解決されます。

11.7.7 マークアップの表示

Authentic XML ドキュメント内のマークアップをコントロールするオプションを使用するサブメニューがマ ークアップコマンドに搭載されています。これらのオプションは下で説明されています。



マークアップを隠すコマンドにより、Authentic View に表示 されているマークアップシンボルを非表示にします。



小 さなマークアップを表示 コマンドにより、小 さなマークアップシンボルを Authentic View にて表示します。

Ā

大きなマークアップを表示コマンドにより、大きなマークアップシンボルをAuthentic View にて表示します。



混合マークアップを表示 コマンドにより、異なるマークアップシンボルが Authentic View で表示 されます。 StyleVision Power Stylesheet では、デザイン時にドキュメントの各要素属性に対して、大きなマークアップ、小さなマークアップ、またはマークアップ無しというオプションを指定することができます。 混合マークアップのモードでは、カスタマイズされたマークアップが Authentic View に表示されます。

11.7.8 行の追加 /挿入 /複製 /削除

ĝ

行の追加 コマンドにより、Authentic Viewの現在 アクティブなテーブルに新たな行を追加します。



行の挿入 コマンドにより、Authentic View の現在 アクティブなテーブルに新たな行を挿入します。



行の複製 コマンドにより、Authentic View の現在 アクティブなテーブル行を複製 します。

2

行の削除 コマンドにより、Authentic View の現在 アクティブなテーブル行を削除 します。

11.7.9 上に /下に移動



行を上に移動コマンドにより、Authentic View における現在のテーブル行を、1行上に移動します。



行を下に移動コマンドにより、Authentic View における現在のテーブル行を、1行下に移動します。

11.7.10 HTML、RTF、PDF、Word 2007+ ドキュメントの生成

以下にある4つのコマンドにより、PXF ファイル内部に収められた Authentic View XML ドキュメントから 出力ドキュメントを生成することができます:

- HTML ドキュメントの生成
- RTF ドキュメントの生成
- PDF ドキュメントの生成
- Word 2007+ ドキュメントの生成

これらコマンドへは Portable XML Form (PXF) ツールバーからもアクセスすることができます (以下のスクリーンショットを参照)。



個々のコマントまたはボタンをクリックすることで、HTML、RTF、PDF、または DocX 形式の出力がそれぞれ生成されます。

これらのボタンは PXF ファイルが Authentic View にて開かれた時に有効になります。PXF ファイル内に それぞれの出力形式に対応した XSLT ファイルが含まれることで、個々のコマンドならびにボタンが有効 になります。例えば、HTML ならびに RTF を出力する XSLT スタイルシートが PXF ファイル内部に含まれ ている場合、HTML ならびに RTF 出力を行うためのコマンドならびにツールバーボタンが有効になり、 PDF と DocX (Word 2007+) 出力を行うコマンドは無効状態となります。

11.7.11 信頼された場所

信頼された場所コマンドを選択すると (以下のスクリーンショットにある)信頼された場所ダイアログが表示され、SPS内にあるスクリプトのセキュリティー設定を指定することができます。 スクリプトを含む SPS をベースにしている XML ファイルを Authentic ビューで開いた場合、このダイアログで指定された内容に従い、スクリプトの実行が許可されます (またはされません)。

S Trusted Locations
Always run Authentic scripts
C Never run Authentic scripts
Only run Authentic scripts from trusted locations
All these locations are treated as trusted sources for SPS and PXF documents. Authentic scripts will be run without any further approval. If you add a new location, make sure that this path and its subdirectories are secure.
Trusted Paths
C:\workarea\Examples\ C:\workarea\seg\
Add Remove OK Cancel

以下にある3つのオプションが指定可能です:

- Authentic ビューにてファイルが開かれた時に、Authentic スクリプトを常に動作させる。
- Authentic ビューにてファイルが開かれても Authentic スクリプトを動作させない。
- 信頼された場所にある Authentic スクリプドだけを動作させる。信頼された場所 (フォルダー)のリストが下部のペインに表示されます。追加ボタンを使用することで、フォルダーの参照とリストへの追加を行うことができます。リストからエントリーを削除するには、リストからエントリーを選択した後、削除ボタンをクリックしてください。

11.8 表示メニュー

表示 メニューでは (アのスクリーンショット)では、アクティブな メインウィンドウ の表示設定を行い、ドキュメントの表示方法を変更することができます。 このセクションでは、「表示」メニューから使用することのできるコマンドを説明します。

11.8.1 Authentic View

9

このコマンドは、現在のドキュメントをAuthentic View に切り替えます。

Authentic View を使用 するとAltova の StyleVision アプリケーションにより作成 された StyleVision Power Stylesheet テンプレートをベースに XML ドキュメントを編集 することができます。これらのテンプ レート (StyleVision スタイルシート または SPS ファイル) は、XML ドキュメントをグラフィカルな フォー マットで表示し、XML ドキュメントを(マークアップを使用下 テキストフォーマット編集 するよりも)簡単 に編集 することができます。

11.8.2 プラウザービュー

0

このコマンドにより、現在のドキュメントビューを <u>プラウザービュー</u> へ切 リ 替 えます。このビューでは XML が利用可能な プラウザーを使用 して、CSS や XSL スタイルシートによ り 得 られた情報 による XML ドキュメントのレンダリングを行います。

オプションダイアログのファイルタブ にて保存時に検証を行うよう設定している場合、ブラウザビューへの切り替えが行われる際に、ドキュメントの検証が行われます。オプションダイアログは、「ツール | オ プション」メニューコマンドにより開くことができます。詳細に関しては、このドキュメントのブラウザー ビューのセクションを参照してください。

11.9 プラウザーメニュー

「プラウザー」メニューのニマンドは、<u>プラウザービュー</u>で有効にないます。「戻る」と「進む」 コマンドはしかしながら、前回使用されたコマンドに移動することのできるスキーマビューでも有効化されることができます。



11.9.1 戻る

 \mathbf{r}

戻る コマンド (ショートカット: Alt + 左矢印) は、ブラウザービューとスキーマビューで有効化 されます。

ブラウザービューにて、「戻る」コマンドにより、その前に表示されたページが表示されます。バックスペースキーを押下することでも同様の操作を行えます。 戻るコマンドは、XML ドキュメントのリンクをクリックした後に、再度 XML ドキュメントへ戻るために使用することができます。

スキーマビューでは、**戻る**コマンドにより、その前に閲覧していたコンポーネントが表示されます。Alt + **左矢印**がショートカットキーになります。**戻る**コマンドにより、最大500回前の閲覧箇所まで戻ることができます。

11.9.2 進む

$\widehat{\Box}$

進む コマンド (ショートカット: Alt + 右矢印) は、プラウザービュー内 で有効化 されます。 「進む」コ マンドは、「戻る」コマンドを使用 した後 に有効 になります。このコマンドにより(i) プラウザビューにて 既に閲覧 したページや、(ii) スキーマビューにて既に閲覧 したスキーマコンポーネントを閲覧 することが できます。

11.9.3 中止



中止 コマンドにより、ブラウザービューで行われているドキュメントの読み取りを中止します。スピード が遅いインターネット回線などにより外部ファイルや画像などがダウンロードされており、その処理を停止したい場合にこのコマンドを使用することができます。

11.9.4 最新の状態に更新

٠.

最新の状態に更新 (F5) コマンドにより、ブラウザービューのドキュメントや、CSS、XSL スタイルシート、DTD といった関連するドキュメントが再ロードされます。

11.9.5 フォント

フォントコマンドにより、XML ドキュメント内 にあるテキストのレンダリングに使用 されるデフォルトのフォ ントサイズを選択 することができます。殆 どのブラウザーにおける文字のサイズコマンドに似た動作をし ます。

11.9.6 別のウィンドウ



別のウィンドウで開くコマンドにより、ブラウザービューが別のウィンドウにて開かれ、他のビューと隣合わせで表示することができるようになります。 ブラウ別のウィンドウでは、ブラウザービューはドキュメントの編集ビューと左右に表示することができます。

ブラウザービューが別のウィンドウで表示 されていると 編集 ビューにて F5 を押下 することで、対応す るブラウザービューの内容も更新されます。 ウィンドウを元のインターフェースに統合するには、 アクティ プなウィンドウの右上にある最大化ボタンをクリックしてください。

11.10 ツールメニュー

ツールメニューを使用して以下を行うことができます:

- XML ドキュメントのスペル チェックをチェックします。
- Authentic Desktop の <u>スクリプト環境</u> にアクセスします。フォーム、マクロ、イベントハンドラー を作成、管理、および保管することができます。
- 現在割1当てられているマクロをビューします。
- 差異をチェックするために2つのフォルダーを比較する。
- 外部アプリケーションを使用するカスタマイズされたコマンドにアクセスします。これらのコマンドは、カスタマイズダイアログのツールタブ内で作成することができます。
- グローバルリソースの定義
- XMLSpy内のグローバルリソースのためにアクティブな構成を変更する。
- Authentic Desktop のバージョンに対して独自のツールバーや、キーボードショートカット、メニューマツールバー、キーボードショートカット、メニュー、およびマクロを定義してカスタマイズすることができます。
- Authentic Desktop 全体に対してグローバルな設定 を行う

11.10.1 スペリング

内蔵の言語辞書を搭載した Authentic Desktop のスペルチェッカーは、Authentic View.

メモ: Altova ソフトウェアに内蔵されている辞書は、Altova の好みにより選択されたものではなく MPL, LGPL、または BSD ライセンスといった商用ソフトウェアにおける再配布が認められているラ イセンス下で利用可能かどうかに大きく依存しております。GPL ライセンスなどの、より強力な ライセンス下で配布されています。これらの辞書は<u>http://www.altova.com/dictionaries</u>に配 置されているインストーラーからご利用いただくことができます。辞書に対して適用されるライ センスの条件に同意するか、また辞書がお使いのコンピューターにあるソフトウェアに適したも のなのかの判断はお客様に委ねられます。

このセクションではスペルチェッカーの使用方法について記述され、以下の3つのサブセクションから構成されます:

- <u>スペルチェッカーの言語を選択</u>
- <u>スペルチェッカーを起動する</u>

スペルチェッカーの言 語 を選 択

以下の操作により、スペルチェックを行う言語をセットすることができます:

- 1. **ツール | スペルチェックのオプション**」メニューコマンドをクリックします。
- 2. スペリングコンテキストダイアログが表示されるので、「スペルチェックのオプション」ボタンをクリックします。
- 3. スペルチェックのオプションダイアログにて、辞書の言語コンボボックスのリストからインストールされた 辞書を選択します(以下のスクリーンショットを参照)。
| スペル チェックのオプション |
|--|
| スペルチェック
⑦ 常に修正を表示 ③ メイン辞書だけを使った修正を行う ◎ 大文字で書かれた単語を無視 ◎ 数値がある単語を無視 ◎ キャメルケースの単語を分離 □ ユーザー辞書(D) |
| 辞書の言語
English (US) |
| <u>http://www.altova.com/dictionaries</u> から辞書を入手可能 |
| OK キャンセル |

4. OK をクリックして完了します。

選択 された言語の辞書がスペルチェックに使用されます。スペルチェックを行う言語の辞書がまだイン ストールされていない場合、新たな辞書をダウンロードすることができます。 ダウンロードの方法 につい ては、スペルチェッカーに辞書を追加のセクションを追加を参照ください。

スペルチェッカーを実行する

「ツール | スペルチェック」コマンド (Shift + F7) により、現在 アクティブになっている XML ドキュメント内 における、に対してチェックが自動的に始まります。未知の単語が発見 されると スペリングダイアログが表示 されます (以下のスクリーンショットを参照)。未知の単語が無い場合、スペルチェックがドキュメントの最後まで行われます。

スペリング:English(US)	×
辞書にない単語:	
nanoelectronic	無視(!)
候補:	全て無視(<u>G</u>)
microelectronic optoelectronic photoelectronic	辞書に追加(人)
electronically	**ま(C)
	&£(C)
	全て変更し
ドキュメントを再チェック(広) オブション(□)	

スペリングダイアログで使用することのできるオプションを以下に記します:

辞書にない単語

このテキストボックスには、選択された言語辞書やユーザー辞書にて発見できなかった単語が表示されます。以下のオプションが利用可能です:

- ・テキストボックスに表示されている単語をキーボードを使って編集するか、候補リストから選択することができます。その後変更ボタンをクリックすることで、XMLトキュメント内にある単語が、編集された単語に置き換えられます(候補リストにある単語をダブルクリックすることで、XMLトキュメントの内容が直接置き換わります)。単語が辞書にない単語テキストボックスに表示されると、XMLトキュメント内の対応する箇所がハイライトされ、トキュメント内の単語を直接編集することもできます。全て変更をクリックすることで、置換処理がXMLトキュメント全体に対して行われます。
- 変更を行わず、現在ハイライトされている箇所のみ、またはドキュメント全体においてスペルチェッカーの 警告を無視することができます。
- 単語をユーザー辞書に追加することで、それ以降のチェックにおいてその単語が正しいものであると認識させることができます。

候補

このリストボックスには、未知の単語に近い(言語辞書ならびにユーザー辞書から得られた)単語が 表示されます。リストにある単語をダブルクリックすることで、その単語がドキュメント内へ挿入され、ス ペルチェック処理が続けられます。

無視

このコマンドにより、現在選択されている未知の単語を無視して、ドキュメントのチェックを続けることができます。同じ単語がドキュメント内で再度発見された場合、その単語が再度表示されます。

全て無視

このコマンドにより、現在の未知単語をドキュメント全体において無視することができます。

辞書に追加

このコマンドにより、未知の単語をユーザー辞書へ追加します。 ユーザー辞書へは スペルチェックの オプション ダイアログにてアクセス、編集を行うことができます。

変更

このコマンドにより、XML ドキュメント内で現在ハイライトされている単語が、(変更された)辞書にな い単語 テキストボックス内の単語に置き換えられます。

全て変更

このコマンドにより、XML ドキュメント内で現在ハイライトされている単語が、ドキュメント全体で(変更された)辞書にない単語テキストボックス内の単語に置き換えられます。

ドキュメントを再 チェック

ドキュメントを再チェックボタンにより、ドキュメントの先頭から再度チェックを行います。

オプション

現在使用されているビューにより以下のダイアログボックスが表示されます:

現在のビューが Authentic View の場合、<u>スペルチェックのオプション</u>ダイアログボックスが表示されます。

ダイアログボックスに関する詳細はスペルチェックのオプションセクションを参照ください。

閉じる

スペリングダイアログボックスが閉じられます。

11.10.2 スペルチェックのオプション

「ツール | スペルチェックのオプション」コマンドによりスペルチェックのオプション.

スペルチェックのオプション

スペルチェックのオプションダイアログではグローバルなスペルチェッカーオプションを設定することができます。

スペル チェックのオプション
スペルチェック
✓ 常に修正を表示 ■メイン辞書だけを使った修正を行う
▼ 大文字で書かれた単語を無視
■ 数値がのる半晶と無税 ▼ キャメルケースの単語を分離
ユーザー辞書(D)
辞書の言語 English (US)
<u>http://www.altova.com/dictionaries</u> から辞書を入手可能
OK キャンセル

常に修正を表示:

このオプションを選択することで、ユーザー辞書とそれ以外の辞書から得られた単語の候補が候補リストボックスに表示されます。このオプションを無効にすると候補が表示されなくなります。

メイン辞書だけを使った修正を行う

このオプションを選択することで、言語辞書 (メイン辞書) だけが使用され、スペル チェックにユーザー 辞書は使用されません。「ユーザー辞書」ボタンも無効化され、ユーザー辞書の編集を行うことが できなくなります。

大文字で書かれた単語を無視:

このオプションを選択することで、すべて大文字で記述された単語がチェックの対象から外されます。

数値を伴う単語を無視:

このオプションを選択することで、数値を含んだすべての単語が無視されます。

キャメルケースの単語を分離:

単語内に大文字の文字が含まれる単語をキャメルケースの単語といます。例えば CamelCase" という単語では、 Case"の C"が大文字で示されているため、キャメルケースの単語として認識され ます。キャメルケースの単語は通常辞書に含まれていないため、スペルチェッカーによりエラーとして 認識されます。キャメルケースの単語オプションにより、キャメルケースの単語を大文字からはじまるパ ーツに分離して、個々のパーツをチェックすることによりこの問題を回避することができます。このオプ ションはデフォルトでチェックされています。

辞書の言語:

このコンボ ボックスではスペル チェッカーにて使用 される辞書の言語が選択 されます。 デフォルトの選択は English (US) となっています。 <u>Altova Web サイト</u>.からはその他の言語の辞書を無料でダウンロードいただけます。

スペルチェッカーに辞 書を追加

辞書の言語は、.aff ファイルと.doc ファイルという2 つの Hunspell 辞書ファイルから構成されます。言語辞書は、以下の場所にある Lexicons フォルダーヘインストールされます。全ての言語の辞書は次の場所にある Lexicons フォルダー内 にインストールされます: C:\ProgramData\Altova \SharedBetweenVersions\SpellChecker\Lexicons.

Lexicons フォルダー内部には、異なる言語に対応した辞書のファイルが、<言語名>\<辞書ファイル > という構成で各フォルダーに収められます。例えば、2つの(English (British) とEnglish (US)) 英語辞書が以下のように配置されます:

C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (British)\en GB.aff

C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (British)\en_GB.dic

C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (US) \en US.dic

C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (US) \en US.dic

スペル チェッカー オプション ダイアログでは、 辞書の言語」コンボ ボックスにあるドロップダウン リスト に利用可能な辞書の言語が表示されます。Lexicons フォルダー以下にある言語サプフォルダーの名 前がリストには表示されます。例えば上にある英語辞書の場合、English (British) とEnglish (US) という名前がリストに表示されます。

インストールされた辞書はコンピューターを使っているすべてのユーザー、そして (32 ビットか 64 ビットかにかかわらず) 複数の Altova 製品 により洪有されます。

スペル チェッカーで使用 される辞書は以下に示される2種類の方法で追加することができ、どちらの 方法でもファイルをシステムに登録する必要はありません:

 Hunspell 辞書をLexicons フォルダーの新規サプフォルダーへ追加する。Hunspell 辞書は、 例えば <u>http://wiki.services.openoffice.org/wiki/Dictionaries</u>や <u>http://</u> <u>extensions.services.openoffice.org/en/dictionaries</u>. といた場所からダウンロードすることが できます (OpenOffice では OXT という拡張子の ZIP アーカイブが配布 されているため、ファイ ル拡張子を.zip へ変換して、回答した.affと.dic ファイルを ファイルを Lexicons フォルダー 以下にある言語フォルダーへコピーしてください。Hunspell 辞書は Myspell 辞書をベースに しているため、Myspell 辞書を使用することもできます)

 複数の言語辞書をコンピューターの適切な場所へインストールする <u>Altova</u> 辞書インストーラ
 <u>ー</u>を使用する。スペルチェックのオプションダイアログにあるリンクをクリックすることで、インスト ーラーをダウンロードすることができます(以下のスクリーンショットを参照)。

辞書の言語	
English (US)	-
<u>http://www.altova.com/dictionaries</u> から辞書を入手可能	

メモ: 辞書に対して適用されるライセンスの条項に同意するか、そしてお使いのコンピューターにおける辞書の使用が適切なものかの判断は、お客様へ委ねられます。

ユーザー辞書と作業

コンピューター上の各ユーザーには、ユーザーにより許可された単語を収めるためのユーザー辞書が 与えられます。スペルチェックを行うと言語辞書とユーザー辞書に含まれている単語のリストに対し てドキュメント内の単語がチェックされます。ユーザー辞書ダイアログでは、ユーザー辞書に単語を追加したり削除することができます(以下のスクリーンショットを参照)。スペルチェックのオプションダイアロ グにて「ユーザー辞書」ボタンをクリックすることで、この辞書へアクセスすることができます。

🚯 ユーザー辞書		×
単語: Nanonull		追加
辞書: xml Nanonull stylesheet		削除
	-	OK キャンセル

ユーザー辞書へ単語を追加するには、単語フィールドに単語を入力し、追加」ボタンをクリックして

ください。入力された単語がアルファベット順で辞書ペインに追加されます。辞書から単語を削除するには、目的の単語を辞書ペインにて選択し、**削除**」ボタンをクリックしてください。その単語が辞書ペインから削除されます。ユーザー辞書ダイアログにおける編集を終えた後には「OK」をクリックすることで、変更点がユーザー辞書へ保存されます。

スペル チェックの際に単語をユーザー辞書へ追加することもできます。スペル チェックを行う際に未知の単語が発見された場合、スペルチェックダイアログにて適切な選択を行うことができます。 辞書 に追加」ボタンをクリックすると未知の単語がユーザー辞書へ追加されます。

ユーザー辞書は以下の場所にあります: C:\Users\<user>\Documents\Altova\SpellChecker \Lexicons\user.dic

11.10.3 スクリプトエディター

スクリプトエディターコマンドにより、スクリプトエディターウィンドウが表示されます。スクリプトエディターの使用方法については、ドキュメンテーションのスクリプト環境 セクションを参照ください。

メモ: スクリプトエディターを動作させるには、NET Frameworkのバージョン 2.0 以降を、お使いのコ ンピューターにインストールする必要があります。

11.10.4 マクロ

マクロコマンドにマウスオーバーすることで、Authentic Desktop にて現在 アクティブなスクリプトプロジェ クトにて定義 されているマクロがサブメニューに表示 されます (以下のスクリーンショットを参照)。アク ティブなスクリプトプロジェクトは、オプションダイアログのスクリプトのセクション内で指定 されています。

マクロ ト	AddMacroMenu
	CloseAllButActiveDoc
	SearchPath

サブメニューにあるマクロをクリックすることで、マクロが開始されます(上のスクリーンショットを参照)。

11.10.5 ユーザー定義ツール

ユーザー定義ツールコマンドへカーソルを配置すると外部アプリケーションを使用するカスタムメイド のコマンドを含んだサブメニューが標示されます。<u>カスタマイズダイアログのツールタブ</u>にて、これらのコマ ンドを作成することができます。カスタムコマンドをクリックするとコマンドに関連付けられたアクションが 実行されます。

「ユーザー定義ツール | カスタマイズ」コマンドを選択すると、<u>カスタマイズダイアログのツールタブ</u>が 標示され、ユーザー定義ツールコマンドのメニューに標示されるカスタムコマンドを作成することができ ます。

11.10.6 グローバルリソース

グローバルリソースコマンドにより、以下の操作を行うことができるグローバルリソースダイアログが表示されます(以下のスクリーンショットを参照):

- グローバルリソース XML ファイルを指定。
- ファイル、フォルダー、データベースのグローバルリソース (またはエイリアス)を追加。
- 各グローバルリソース (エイリアス)に対して、各種構成を指定。各構成により、特定のリソースへの マッピングが行われます。

2. グローバル リソースの管理	×
定義ファイル: suments\Altova\GlobalResources.xml	🕞 参照(B)
 □ ファイル □ nanonull □ ExpenseReport □ ☞ フォルダー □ ☞ Work Area □ ☞ Altova □ ☞ データベース □ Customers □ AltovaDatabase 	 ・ 追加(A)

グローバルリソースの定義方法に関する詳細は、グローバルリソースの定義セクションを参照ください。

メモ: Altova グローバルリソースダイアログへは、<u>グローバルリソースツールバー</u>(「**ツール**|**カスタマ イズ | ツールバー | グローバルリソース**」)からアクセスすることができます。

11.10.7 アクティブな構成

アクティブな構成 メニューアイテムにマウスオーバーすることで、現在アクティブなグローバルリソース XML ファイルにて定義 されている全ての構成が表示 されます (以下のスクリーンショットを参照)。

2	グローバル リソース(<u>G</u>)		
	アクティブな構成・	•	Default
			new

現在アクティブな構成は、中黒(・)により表示されます。上のスクリーンショットでは、現在アクティブな構成が Default となっています。アクティブな構成を変更するには、アクティブにしたい構成を選択してください。

メモ: アクティブな構成は、グローバルリソースツールバー(「ツール | カスタマイズ | ツールバー | グローバルリソース」)から選択することができます。

11.10.8 カスタマイズ

カスタマイズコマンドにより、XMLSpy に搭載されたメニューやツールバーをカスタマイズすることができます。以下のタブが用意されています:

- <u>コマンド</u>:全てのアプリケーションやマクロコマンドをメニューバー、メニュー、ツールバーヘドラッグします。
- ツールバー: 個 々 のツールバーを有効化、無効化、リセットすることができます。
- <u>ツール</u>:外部プログラムを開くコマンドをインターフェースへ追加することができます。
- <u>キーボード</u>:個々のアプリケーションならびにマクロに対してキーボードのショートカットを作成 することができます。
- <u>メニュー</u>: カスタマイズするメニューバーならびにコンテキストメニューを選択し、有効にすることができます。コマンドタブと一緒に操作します。
- マクロ: コマンドに対してマクロを関連付けることができます。
- プラグイン: プラグインの有効化と統合を行うことができます。
- オプション: ツールバーの表示オプションをセットすることができます。

このセクションではカスタマイズダイアログが開かれ、メニューバー、メニュー、またはツールバーアイテムが右 クリックされた時に表示 される コンテキストメニュー についても記述 されます。

コマンド

コマンドタブでは、メニューやツールバーをカスタマイズすることができ、必要に応じてアプリケーションコマンドをメニューやツールバーに追加することができます。新たなアプリケーションコマンドやメニューを独自に作成することはできない点に注意してください。

カスタマイズ	×
コマンド ツールバー ツール キーボード メニュー マクロ グ: カテゴリー(G): コマンド(D): 「油集(E) 「コジェクト(P) メML(X) アキーマ設計(M) スキーマ設計(M) 一 第人作成(N) メSL/XQuery(Q) 町ロード(D) Authentic(A) 町 DB(D) 表示(V) ブラウザー(B) すべて閉じる WSDL(L) アンドウ(W) ジール(T) ・ ウィバドウ(W) ・	ラガイン オプション
	閉じる

以下の操作によりコマンドをツールバーやメニューへ追加することができます:

- 1. メニューオプションから「ツール | カスタマイズ」を選択します。カスタマイズダイアログが表示 されます。
- 2. カテゴリーリストボックスから全てのコマンドカテゴリーを選択します。利用可能なコマンドがコマンドリストボックスに表示されます。
- コマンドリストボックスに表示されたコマンドをクリックして、表示されているメニューやツールバ ーヘドラッグします。許可されている場所ヘカーソルを移動させると、マウスポインターに1というアイコンが表示されます。
- 4. コマンドを挿入する場所でマウスボタンを離してください。

以下の点に留意してください:

- コマンドをドラッグすると、マウスポインターの端に小さなボタンが表示されます。これにより、ボタンがド ラッグされていることが示されます。
- ポインター下部に表示された "x" は、現在マウスポインターがある位置にドロップできないことを表しています。
- コマンドをドロップすることができる位置(ツールバーまたはメニュー)にカーソルを移動すると X"が 消えて」がというアイコンが表示されます。
- コマンドはメニューまたはツールバーに配置することができます。
 独自のツールバーを作成した場合
 した場合、このカスタマイズ機能を使用することでコマンドを配置することができます。
- カーソルをメニュー近くに移動させると、そのメニューが開かれ、メニュー内部にコマンドを配置することができるようになります。

コンテキストメニューヘコマンドを追加

以下の操作により、コンテキストメニューにコマンドを追加することもできます:

- 1. カスタマイズダイアログにて メニュータブをクリックします。
- 2. コンテキストメニューペインにてコンボボックスから目的のコンテキストメニューを選択します。選択 されたコンテキストメニューが表示されます。
- 3. カスタマイズダイアログにてコマンドタブへ切り替えます。
- 4. コマンドリストボックスから追加するコマンドを選択し、コンテキストメニューまでドラッグします。

コマンドまたはメニューの削除

以下の操作により、メニューやコンテキストメニュー、またはツールバーからコマンドを削除する、または メニュー全体を削除することができます:

- 1. メニューオプションの「ツール | カスタマイズ」を選択してカスタマイズダイアログを開きます。
- カスタマイズダイアログが表示されている状態で、メニューまたはメニューコマンドを右クリックして、表示されるコンテキストメニューから削除を選択します。また、メニューやメニューコマンドを、***がマウスポインターに表示される場所までドラッグアンドドロップすることでもメニューまたはメニューコマンドを削除することができます。

このセクションに記述された方法により削除されたメニューコマンドは、復元することができます。削除 されたメニューを復元するには、メニューオプションから「ツール | カスタマイズ | メニュー」を選択し、 アプリケーションフレームメニューペインにあるリセットボタンをクリックしてください。その他にも、「ツール | カスタマイズ | ツールバー」を選択し、左側に表示されたリストからツールバーを選択し、リセット ボタンをクリックすることもできます。

ツールバー

ツールバータブでは、(i)特定のツールバーを有効無効にする(つまりどのツールバーがインターフェースにて表示されるかを指定する)、(ii)各ツールバーに表示されるアイコンをセットする、そして(iii)独自のツールバーを作成することができます。

ツールバーには頻繁に使用されるメニューコマンドが収められています。カーソルをアイコン上部に移動することで、そのアイコンに関する情報がツールチップとステータスバー上に表示されます。ツールバーは、フロート状態のウィンドウとして画面上の任意の場所へ配置することができます。

- メモ: ツールバーヘコマンドを追加するには、目的のコマンドをコマンドタブにあるコマンドリストボック スからツールバーヘドラッグする必要があります。ツールバーからコマンドを削除するには、カス タマイズダイアログが開かれた状態で、ツールバーからコマンドをドラッグしてください(詳細につ いてはコマンドのセクションを参照ください)。
- **メモ:** 特定のビューに対して定義されたツールバーの設定は、デフォルトでそのビューに対してだけ 有効になっています。その設定を全てのビューに対して適用するには、ダイアログ下部にあ る全てのビューに変更を適用チェックボックスにチェックを入れてください。

Customize	
Commands Toolbars Tools Keyboard Mer	nu Macros Plug-Ins Options
Toolbars: Authentic Browser CALS/HTML Table Global Resources Main Menu Bar Portable XML Form Project RichEdit Text	Reset All Rename Rename Delete Show text labels
Apply changes for all views	Close

以下の機能を利用することができます:

- ツールバーの有効化 /無効化:ツールバーリストボックス内に表示されたチェックボックスをクリックしてください。
- 全てのビューに変更を適用:ダイアログ下部に表示されているチェックボックスにチェックを入れてください。チェックが入っていない場合、アクティブなビューに対してのみ変更が適用されます。全てのビューに変更を適用チェックボックスをクリックした後の変更だけが全てのビューに対して適用される点に注意してください。
- ツールバーを新たに追加:新規ボタンをクリックして、表示されるツールバーの名前ダイアログにてツールバーの名前を入力します。作成されたツールバーには <u>コマンド</u>タブからコマンドをドラッグすること

ができます。

- 追加されたツールバーの名前を変更: 追加されたツールバーをツールバーペインにて選択し、名前の変更ボタンをクリックします。表示されるツールバーの名前ダイアログにて名前の編集を行います。
- メニューバーのリセット: ツールバーペインにてメニューバーアイテムを選択し、リセットをクリックします。この操作によりアプリケーションがインストールされた初期の状態にメニューバーがリセットされます。
- ツールバーとメニューコマンドのリセット:全てリセットボタンをクリックすることで、アプリケーションがインストールされた初期の状態に、全てのツールバーならびにメニューがリセットされます。
- ツールバーの削除: ツールバーペインにて削除するツールバーを選択し、削除ボタンをクリックします。
- 特定のツールバーにコマンドのテキストラベルを表示:目的のツールバーを選択し、テキストラベルを表示チェックボックスにチェックを入れます。テキストラベルは各ビューに対して個別に有効化する 必要がある点に注意してください。

ツール

ツールタプでは、Authentic Desktop 内部から外部アプリケーションを使用するコマンドをセットアップ することができます。これらのコマンドは、「ツール | ユーザー定義ツール」メニュー以下に表示されま す。例えば、作成されたツールメニューのコマンドを「ツール | ユーザー定義ツール」以下からクリック することで、Authentic Desktop のメインウィンドウに表示されているアクティブなファイルを、メモ帳のよ うな外部アプリケーションにて開くことができます。

<u> </u>	×
「コマンド「ツールバー」ツール キーボード メニュー マクロ フ	?ラグイン [オプション]
メニュー コンテンツ(<u>M</u>):	🗸 🕹 🐇 🖿
メモ帳で開く	
引数(<u>A</u>): \$(ActiveDocumentFilePath)	アクティブなドキュメントのファイルパス
初期ディレクトリロ:	

以下の操作により外部アプリケーションを使用するコマンドをセットアップすることができます:

 メニューコンテンツペインにて、ペインのタイトレバーにある新規作成アイコンをクリックし、作成された アイテムにメニューコマンドの名前を入力します。上のスクリーンショットでは、メモ帳で開くというメニュ ーコマンドが作成されており、アクティブなドキュメントを外部アプリケーションの (Windows に搭載され ている)メモ帳にて開くようなコマンドを作成します。新規作成アイコンをクリックすることで、更にコマン ドを追加することもできます。アイテムを上へならびにアイテムを下へアイコンを使用することで、コマ ンドの相対的な位置を変更することができます。コマンドを削除するには、削除アイコンをクリックしてく ださい。

- 2. 外部アプリケーションとコマンドを関連付けるには、メニューコンテンツペインにあるコマンドを選択し、コマンドフィールドにて外部アプリケーションの実行可能ファイルへのパスを入力するか、(入力フィールドの隣にあるボタンから)参照してください。上のスクリーンショットでは、メモ帳へのパスがコマンドフィールドへ入力されています。
- 3. 引数フィールドにあるボタンをクリックすることで、外部アプリケーションに対して利用可能なアクションが表示されます(上のスクリーンショットを参照)。これらのアクションについては以下にあるリストを参照ください。アクションを選択すると、そのアクションに対応したコード文字列が引数フィールドに入力されます。
- 4. 処理を行う際の初期ディレクトJを指定する場合、その値を初期ディレクトJフィールドに入力してください。
- 5. 閉じるボタンをクリックすることで処理を完了します。

作成されたコマンドは、「ツール | ユーザー定義ツール」メニューと、プロジェクトウィンドウにてファイル を右 クリックすることで表示されるコンテキストメニューの「ユーザー定義ツール」サブメニュー以下に表示されます。

新規作成 されたコマンドを**ツール**メニューにてクリックすると コマンドに関連付けられたアクションが実行 されます。上のスクリーンショットに示されたコマンドの例の場合、Authentic Desktopのメインウィンドウでアクティブなドキュメントがメモ帳で開かれます。外部アプリケーションコマンドへは、プロジェクトウィンドウのユーザーツールコンテキストメニューからもアクセスすることができます(プロジェクトウィンドウにあるファイルを右クリックすることでコンテキストメニューが表示されます)。プロジェクトウィンドウでは、複数のファイルを選択し、コンテキストメニューからコマンドを選択することで、複数のファイルに対して関連付けられたアクションを実行することができます。

引数

引数フィールドでは外部アプリケーションコマンドにより実行されるアクションが指定されます。以下の引数を利用することができます:

- アクティブなドキュメントのファイルバス: 「ユーザー定義ツール」メニューにあるコマンドから Authentic Desktop にてアクティブになっているドキュメントを外部アプリケーションで開きます。プロ ジェクトウィンドウのコンテキストメニューからコマンドへアクセスした場合、選択されたファイルが外部ア プリケーションで開かれます。
- プロジェクトファイルパス: Authentic Desktop のプロジェクトファイル (spp ファイル)を外部アプリケーションにて開きます。

初期 ディレクトリ

オプションとして、外部アプリケーションを実行する際にカレントディレクトリとして使用されるディレクトリ へのパスを、初期ディレクトリフィールドへ入力することができます。

キーボード

キーボードタブでは、キーボードショートカットを新たに作成したり、既存のショートカットを変更することができます。



以下の操作により、新たなショートカットをコマンドへ割り当てる、または既存のショートカットを変更 することができます:

- 1. **カテゴリー**コンボボックスから全てのコマンドを選択します。 <u>関連付けられたコマンドとしてマクロが選択された</u>場合、カテゴリーコンボボックスにマクロをセットすることもできます。
- 2. コマンドリストボックスにて、ショートカットを割 リ当 てる、またはショートカットを変更 するコマン ドを選択します。
- 3. 新規ショートカットキーテキストボックスをクリックし、そのコマンドに対して割リ当てるショートカットキーを押下します。入力されたショートカットが新規ショートカットキーテキストボックスに表示されます。そのショートカットがどのコマンドにも割リ当てられていない場合、割り当てボタンが選択可能状態になります。ショートカットが他のコマンドに既に割リ当てられている場合、そのコマンド名がテキストボックス以下に表示され、割り当てボタンが選択不可能状態になります。新規ショートカットキーテキストボックスに入力された内容をクリアするには、Ctrl、Alt、またはShiftのいずれかのキーを押下してください。
- 4. 割り当てボタンをクリックすることで、ショートカットの割り当てを行うことができます。ショートカットが現在のキーリストボックスに表示されます。複数のショートカットを同じコマンドへ割り当てることもできます。
- 5. 閉じるボタンをクリックすることで、変更を確定します。

ショートカットの削除

同 じショートカットを複数のコマンドに対して割 リ当 てることはできません。ショートカットを削除するには、現在のキーリストボックスにあるショートカットを選択し、削除ボタンをクリックし、閉じるボタンをクリックすることで削除を確定してください。

設定するアクセラレータ

現在割り当てられている機能はありません。

デフォルトのキーボードショートカット

インストール時に割り当てられているコマンドのリストを(機能ごとに)以下に示します:アプリケーションではキーボードマップダイアログ(「<u>ヘルプ」キーボードマップ</u>」)では、ショートカットとその説明とともに、コマンドのリストを確認することができます。

□ (検証と変換のためのショートカットも含む)機能キーのショートカット

F1	ヘルプメニュー
F1 + Alt	最後に開かれたファイルを開く
F3	次を検索する
F4 + CTRL	アクティブなウィンドウを閉じる
F4 + Alt	Authentic Desktop を閉じる
F5	更新
F6 + CTRL	開かれているウインドウを循環する
F7	整形式チェック
F8	検証
F10	XSL 変換
F10 + CTRL	XSL:FO 変換

□ ファイルとアプリケーションコマンド

Alt + F1	最後に開かれたファイルを開く
CTRL + O	ファイルを開く
CTRL + N	新規ファイル
CTRL + P	ファイルを印刷する
CTRL + S	ファイルを保存する
CTRL + F4	アクティブなウィンドウを閉じる
CTRL + F6	開かれているウィンドウを循環する
CTRL + TAB	開かれているドキュメントを切り替える
Alt + F4	Authentic Desktop を閉じる

□ 文字キーバッドのショートカット(ノードの展開、折りたたみ)

Num +	展開
Num *	完全に展開する
Num –	折りたたむ
CTRL + Num –	選択されていない範囲を折りたたむ

□ その他のキー

上不矢印キー	カーソルまたは選択バーを移動する
Esc	編集を破棄する、ませよ、ダイアログボックスを閉じる
Return	選択を確認する
Del	文字、または、選択範囲を削除する
Shift + Del	切り取る

□ 編集 コマンド

CTRL + A	全てを選択する
CTRL + F	検索する
CTRL + G	ライン文字に移動する
CTRL + H	置き換える
CTRL + V	貼り付ける
CTRL + X	切り取る
CTRL + Y	や直し
CTRL + Z	元に戻す

メニュー

メニュータブでは、デフォルトならびにアプリケーションメニューバーという2つのメインメニューバー、そしてアプリケーションのコンテキストメニューをカスタマイズすることができます。

カスタマイズ	X			
コマンド ツールバー ツール キーボード メニュー マクロ プラグイン オプション				
アプリケーションフレームメニュー:	コンテキスト メニュー:			
表示するメニュー(S):	コンテキスト メニューを選択(C):			
Authentic 🔹	▼			
リセット(R)	リセット(E)			
Authentic ドキュメント	ヒント: コンテキスト メニューを選択し て、'コマンド' ページに変更し、ツー ルバーのボタンをメニュー ウインドウに ドラッグ。			
✓ メニューに影をつける(H)				

デフォルトメニューバーとアプリケーションメニューバーのカスタマイズ

デフォルトのメニューバーとは、メインウィンドウにてドキュメントが開かれていない状態で表示されるメ ニューバーのことです。アプリケーションメニューバーとは、1つ以上のドキュメントがメインウィンドウにて 開かれている状態で表示されるメニューバーのことです。各メニューバーは、お互いに干渉しないかた ちで個別にカスタマイズすることができます。

メニューバーをカスタマイズするには、**、表示するメニュー**」コンボボックスから目的のメニューを選択し、 <u>カスタマイズダイアログのコマンドタブ</u>へ切り替えます。その後コマンドリストボックスに表示されたコマ ンドを、メニューバーまたはメニューへドラッグします。

コマンドの削除とメニューバーのリセット

メニュー全体 またはメニュー内部 にあるコマンドを削除 するには、そのメニューまたはメニューコマンドを 選択して、(i) 右 クリックして **削除」**を選択するか、(ii) メニューバーやメニューを別の位置 ヘドラッグし てください。

アプリケーションのコンテキストメニューをカスタマイズする

コンテキストメニューとは、アプリケーションインターフェースに表示されたオブジェクトを右クリックすること

で表示されるメニューのことです。以下の方法により、コンテキストメニューのカスタマイズを行うことがで きます:

- 1. コンテキストメニューを選択コンボボックスにて目的のコンテキストメニューを選択します。コンテキストメ ニューが表示されます。
- 2. <u>カスタマイズダイアログのコマンドタブ</u>へ切り替えます。
- コマンドリスドボックスからコンテキストメニューへコマンドをドラッグします。
 コンテキストメニューからコマンドを削除するには、コンテキストメニューに表示されたコマンドを右クリックして、削除」を選択します。コンテキストメニューからコマンドをドラッグすることでも、そのコマンドを 削除することができます。

コンテキストメニューを選択 コンボボックスにてメニューを選択し、 リセット」ボタンをクリックすることで、 コンテキストメニューをオリジナルの状態にリセットすることができます。

メニューに影を付ける

「メニューに影を付ける」チェックボックスを有効にすることで、全てのメニューに影が与えられます。

マクロ

マクロタブでは、Authentic Desktop のスクリプトエディターにより作成 されたマクロに対するアプリケー ションコマンドを作成することができます。 関連付けられたマクロを実行するこれらアプリケーションコマ ンドは、マクロタブから直接、またはカスタマイズダイアログのコマンドタブを使用することにより、メニュ ーやツールバーにて利用することができるようになります。アプリケーションコマンドとして、カスタマイズ ダイアログのキーボードタブにてショートカットを割り当てることもできます。

Authentic Desktop におけるマクロの作動

Authentic Desktop におけるマクロは以下のように動作します:

- Authentic Desktop のスクリプトエディターにより、Altova スクリプトプロジェクト (aspri ファイル) が作成されます。スクリプトプロジェクHこは、Authentic Desktop で使用 することのできるマクロ を含めることができます。
- Authentic Desktop では、(i) オプションダイアログのスクリプトタブ にて指定 されるアプリケー ションプロジェクトと (ii) スクリプト設定ダイアログ (プロジェクト スクリプト設定) にて指定することができるアクティブなのスクリプトプロジェクトの、2つのスクリプトプロジェクトを同時にア クティブにすることができます。
- これら2つのスクリプトプロジェクト内にあるマクロは プロジェクト| マクロ」メニューから利用することができ、カスタマイズダイアログのマクロタブではマクロをアプリケーションコマンドとしてセットすることができます(以下のスクリーンショットを参照)。マクロをアプリケーションコマンドとし てセットすると、メニューやツールバーにそのコマンドを配置することができるようになります。

マクロに対してアプリケーションコマンドを作成する

<u>スクリプトエディター (ツール | スクリプトエディター</u>」)にて目的のマクロを作成し、スクリプトプロジェク トとして保存します。 このファイルを (オプションダイアログのスクリプトタブ) から)アプリケーションスクリプ トプロジェクトとして、 または (アプリケーションプロジェクトの スクリプト設 定 ダイアログ (**プロジェクト**) スクリプト設定」)から)アクティブなアプリケーションプロジェクトのスクリプトプロジェクトとして指定する ことができます。 スクリプト内部 にあるマクロがマクロタブの マクロ」ペインに表示 されます (以下のスク リーンショットを参照)。

マクロに対するアプリケーションコマンドを作成するには、「マクロ」ペインにてマクロを選択し、「テキスト

を表示」テキストボックスにてコマンドのテキストを入力した後に、「コマンドを追加」ボタンをクリックします。
以下のスクリーンショットを参照)。選択されたマクロに関連付けられたコマンドが、関連付けられた コマンド」リストボックスに追加されます。

Customize								×
Commands	Toolbars	Tools	Keyb	oard	Menu	Macros	Plug-Ins	Options
Macros: AddMacro	Menu			Asso	ciated co	mmands:		
CloseAllBu SearchPa	ItActiveDoc th	:			CloseAll	ButActive	Doc	
					SearchF	Path		
Display tex	t:			Macr	o name:			
SearchPat	th			Sear	chPath]
Add Comn	nand			E	dit Icon		Remove	
								Close

関連付けられたコマンドのアイコンを編集するには、そのコマンドが選択された状態でアイコンを編集 ボタンをクリックします。削除ボタンをクリックすることで、関連付けられたコマンドを削除することができ ます。

マクロに関連付けられたコマンドをメニューやツールパーに配置する

マクロに関連付けられたコマンドをメニューやツールバーに配置するには、2つの方法があります:

- 関連付けられたコマンドリストボックスから、メニューやツールバー上にある目的の場所へコマンドをドラッグする。
- <u>カスタマイズダイアログのコマンドタブ</u>に用意されているメカニズムを使用する。

どちらの場合でも、指定された場所にコマンドが作成されます。メニューまたはツールバー上にあるコマンドをクリックすることで、関連付けられたマクロが実行されます。

メモ: 関連付けられたコマンドとしてマクロがセットされた場合、キーボードのショートカットを割り当てることができます。カスタマイズダイアログのキーボードタブにて、カテゴリーコンボボックスからマクロを選択しても思訳した状態でショートカットをセットしてください。

プラグイン

「プラグイン」タブでは、アプリケーションやツールバー動作するようプログラムされたプラグインの統合とコマンドの配置を行うことができます。プラグインタブでは、「プラグインを追加」をクリックすることで、プラグインの DLL ファイルを参照することができ、「OK」をクリックすることでプラグインが追加されます。

アプリケーションには複数のプラグインを追加することができます。

カスタマイズ	×
コマンド ツールバー ツール キーボード メニュー マクロ	プラグイン オプション
アクティブなプラグイン:	
Authentic C# Sample PlugIn	プラグインの追加
	プラグインの削除
	閉じる

プラグインの追加に成功すると、プラグインの説明がダイアログに表示され、プラグインを削除ボタンが 選択可能になります。プラグインコードによりソールバーやメニューが作成されると、それらがアプリケー ションインターフェースに表示されます。プラグインを削除するには、プラグインを削除ボタンをクリックし てください。

プラグインの作 成

サンプルプラグインのソースコードはアプリケーションの マイドキュメント フォルダー: Examples \IDEPlugin フォルダー内 に収められています。ソースコードからプラグインを作成 するには以下 を行います:

- 1. Visual Studio内でプラグインとして作成するソリューションを開きます。
- 2. ビルドメニュー内のコマンドを使用してプラグインを作成ひます。
- 3. プラグインの DLL ファイルは Bin または Debug フォルダー内 で作成 されます。この DLL ファイ ルは、プラグインとして追加 される必要 があるファイルです (上記参照)。

プラグインの作成に関する詳細については、<u>IDE</u>プラグインのセクションを参照ください。

オプション

オプション」タブでは一般的な環境設定を行うことができます。

<u> </u>	×
コマンド ツールバー ツール キーボード メニュー マクロ プラグイン オプション	
ツールバー	
▶ ツールバーに画面のヒントを表示(工)	
▶ 画面のヒントにショートカットキーを表示(K)	
🗖 大きなアイコン(L)	
	්ම

チェックボックスをクリックすることで以下のオプションを有効にすることができます:

- ツールバーに画面のヒントを表示:ツールバーのアイコン上にマウスポインターが配置されるとポップアップが表示されます。ポップアップにはアイコン機能の説明と、(ショートカットキーを表示オプションが有効になっていれば)関連付けられたショートカットが表示されます。
- 画面のヒントにショートカットキーを表示:画面のヒントにショートカットキーに関する情報を表示する か指定することができます。
- 大きなアイコン: ツールバーアイコンの大きさを通常のサイズと大きめのサイズへ切り替えることができます。

コンテキストメニューのカスタマイズ

コンテキストメニューのカスタマイズコマンドは、カスタマイズダイアログが表示されている状態で、メニュー、メニューコマンド、またはツールバーアイコンを右クリックすることにより表示されます。

既定に戻す(R) ボタン イメージをコピー(Q) 削除(D)
ボタンの外観(B)...<</p>
・イメージ(D)
テキスト(T)
イメージとテキスト(A)
グループの開始(S)

以下の機能を利用することができます:

- 規定に戻す:現在機能は与えられていません。
- ボタンイメージをコピー:右クリックしたアイコンをクリップボードにコピーします。
- 削除:選択されたメニュー、メニューコマンド、またはツールバーアイコンを削除します。削除されたアイコンを再度表示する方法については、下記を参照ください。
- ボタンの外観:ボタンの外観ダイアログが表示され、選択されたツールバーアイコンの外観を定義する ためのプロパティをセットすることができます。詳細については以下の記述を参照ください。
- イメージ、テキスト、イメージとテキスト:相互に排他的なオプションで、選択されたツールバーアイコンをアイコンだけ、テキストだけ、またはアイコンとテキストの両方で表示するかを指定します。いずれかのオプションを選択することで、必要な変更が行われます。
- グループの開始:選択されたツールバーアイコンの左側にグループを分離する垂直の線を挿入します。これにより選択されたツールバーアイコンがグループ最初のアイコンとなります。

ボタンの外 観 ダイアログ

ツールバーアイコンを右 クリックして、「**ボタンの外観」**を選択することで、ボタンの外観ダイアログが表示 されます (以下のスクリーンショットを参照)。このダイアログでは、ツールバーアイコンのイメージやテキストを選択することができます。現在このダイアログでは、マクロならびにプラグインのツールバーアイコンだけを編集することができます。

ボタンの外観 ×
 ○ イメージとテキスト(A) ○ デラオルトのイメージを使用(D): ○ オメージのみり) ○ テキストのみ(T) ○ テキストのみ(T) ○ テキストのみ(T) ○ オーザー定義イメージを使用(D): ○ ヨーザー定義イメージを使用(D): ○ 新規(D) 新規(D) 編集(E)
ポタンテキスト(B): Iンコード(&E) OK キャンセル

選択されたツールバーのアイコン (コンテキストメニューのカスタマイズにて右 クリックされたアイコン) に対して以下の編集機能を使用することができます。

- イメージのみ、テキストのみ、イメージとテキスト:目的のラジオボタンを選択して、ツールバーのア イコンが表示される形式を指定することができます。
- イメージの編集:イメージのみまたはイメージとテキストが選択された場合、イメージの編集オプションが有効になります。「新規」ボタンをクリックすることでユーザー定義の新たなイメージが作成され、イメージペインに追加されます。イメージを選択して、「編集」ボタンにより編集を行うことができます。



- イメージ選択: イメージペインから目的のイメージを選択し、OK をクリックすることにより選択されたイメ ージをツールバーで使用することができます。
- テキストの編集とセレクター: テキストのみまたはイメージとテキストが選択された場合、ボタンテキストテキストボックスが選択可能になります。 テキストを入力または編集し、OKをクリックすることで、ツールバーアイコンのテキストを変更することができます。
- メモ: ボタンの外観ダイアログでは、メニューコマンドのテキストを編集することもできます。カスタマイズ ダイアログが開かれている状態でメニューコマンドを右クリックして、「ボタンの外観」を選択し 「ボタンテキスト」テキストボックスにてメニューコマンドのテキストを編集してください。

削除 されたメニュー、メニューコマンド、ツールバーコマンドを復元

コンテキストメニューのカスタマイズにある削除コマンドにより、メニューやメニューコマンド、ツールバーア イコンが削除された場合、以下の方法により復元することができます:

- メニュー:メニューオプションから「ツール」カスタマイズ」メニュー」を選択し、アプリケーションフレームメニューペインにあるリセットボタンをクリックしてください。他にも「ツール」カスタマイズ」ツールバー」を選択し、ダイアログ左側でメニューバーを選択した状態でリセットボタンをクリックすることもできます。
- メニューコマンド:メニューオプションから「ツール」カスタマイズ」コマンド リストボックスから目的のコマンドをメニューへドラッグしてください。
- ツールバーアイコン:メニューオプションから「ツール | カスタマイズ | コマンド」を選択し、コマンドリストボックスから目的のコマンドをツールバーへドラッグしてください。

11.10.9 ツールバーとウィンドウの復元

ツールバーとウィンドウの復元コマンドにより、Authentic Desktop をデフォルト設定の状態で再起動することができます。終了する前に Authentic Desktop の終了を確認するダイアログが表示されます (以下のスクリーンショットを参照)。

Authentic Desktop
全てのツールバー、ドッキング ウィンドウ、入力へルパーなどを初期の 状態に復元しようとしています。 この処理を行うためには、Authentic Desktop を再起動させる必要があ ります。
OK キャンセル

このコマンドは、ツールバーやウィンドウのサイズ変更、移動、または非表示にするように指定した後、ツールバーやウィンドウをオリジナルの状態に戻したい場合に使用することができます。

11.10.10オプション

「ツール | オプション」コマンドによりグローバルアプリケーションの設定を定義することができます。これらの設定は、セクションにより整理されています(下のスクリーンショット内の左側のペイン参照)。 例えば、ファイルセクション(下のスクリーンショットを参照)には、Authentic Desktop によるファイルの開き方、及び、保存の方法を指定することのできるオプションが含まれています。特定のセクションのオプションを指定するには、左側のペイン内のそのセクションを選択し、必要とするプロパティの値を指定します。「OK」ボタンにより、レジストリへの変更が保存され、ダイアログが閉じられます。 用」ボタンにより現在開かれているドキュメント内の表示が変更されます。

オプション		?
ファイル ファイルの種類 エンコード 編集 ビュー フォントと色 テキスト ビュー グリッド ピュー Schema Design V WSDL デザインビ. XBRL タクソノミ比 スピードオプティマ XQuery XBRL タクソノミ パッ スクリプト ソース管理	ブアイル ヴベての行を展開 変更されたファイルを自動的に再ロード 変更されたファイルを自動的に再ロード マファイルの変更を検出 平日ードの前に確認 検証 自動的にファイルを検証: 「 関い時に ファイル サイズの 10 ▼ MB マ 保存時に 上限 DTD/2キーマ ファイルをメモリにキャッシュ XML スキーマのパージョン ③ なsschema vc.minVersion="1.1" > であれ ば い.1 それ以外の場合は vt.0 常に vt.1 常に vt.0	プロジェクト ▼ プログラムを起動時に最後のプロジェクトを開く ファイルを保存 ● 『 Edited with XMLS py" というコメントを追加 ● 『 Generated by XMLS py" という回を追加 ● Authentic: リンクをデザイン ファイルに保存 - 改行 ● 既存を保持 ● CR & LF ● CR ● LF 出力フォーマットなし: xstattribute
		OK キャンセル 適用(A)

オプションダイアログの各セクションはこのセクションのサブセクション内で詳細に説明されています。

ファイル

ファイルセクションでは、Authentic Desktop にてファイルを開くまたは保存する際の振る舞いを定義することができます。エンコードタブにも関連する設定が存在します。

変更 されたファイルを自動的に再ロード

複数のユーザーが作業を行っているような環境、または作業を行っているファイルが、サーバーにて動的に生成されるような状況では、インターフェース上で現在開かれているファイルの変更を監視することができます。開かれているドキュメントが変更されたことがAuthentic Desktop により検知されると変更されたファイルを再ロードするか確認するダイアログを表示させることができます。

検証

DTD やスキーマによりXML ドキュメントの構造を定義している場合、ドキュメントを開くまたは保存する際に検証を行うかどうか指定することができます。自動的に行われる検証は、ドキュメントの上限を定めることで、効果的な編集作業を行うことができます。ドキュメントが妥当でない場合、エラーメッセージが表示され、ドキュメントが妥当な場合は、メッセージが出力されることなく、処理が行われます。Authentic Desktop ではこれらのファイルをメモリ上にキャッシュすることで、(例えば、参照されているスキーマがURL によりアクセスされた場合など)不必要な再ロードを抑えることができます。 スキーマ位置の宣言にURLを使用している場合、「DTD/スキーマファイルをメモリにキャッシュ」オプションを無効にすることで、スキーマに対して行われた変更を、キャッシュを介さず直接保存することができます。

XML スキーマバージョン

スキーマビュー内で有効化された XSD モードは以下により異なります(i) XSD ドキュメントの / xs:schema/@vc:minVersion 属性の値の存在、または、不在。(ii) オプションダイアログのファイル セクション(ツール|オプション)内で選択される XML スキーマバージョンのオプション。

-XML スキーマのバージョン ● <xs:schema vc:minVersion="1.1" … > であれ ば v1.1 それ以外の場合は v1.0 ● 常に v1.1 ● 常に v1.0

以下のシチュエーションが可能になります。下のテーブル内 XML スキーマバージョンは、上に示される XML スキーマバージョン ペイン内の選択を参照しています。テーブル内の vc:minVersion 値は、 XML スキーマドキュメント内の xs:schema/@vc:minVersion 属性の値を参照しています。

XML スキーマパージョン	vc:minVersion 属性	XSD モード
常にv1.0	不在、または、存在していても値を持っ ていない	1.0
常にv1.1	不在、または、存在していても値を持っ ていない	1.1
@vc:minVersion の値	属性は、1.1 の値を持っている	1.1
@vc:minVersion の値	属性が不在、または、存在していても 値が1.1以外	1.0

プロジェクト

Authentic Desktop を起動すると 最後に使用されたプロジェクトが自動的に開くように指定することができます。

ファイルを保存

XML ドキュメントを保存する際、Authentic Desktop により、-- Edited with Authentic Desktop http://www.altova.com --> というコメントが、ドキュメントの上部に挿入されます。このオプションを無効にするにはユーザーのライセンス登録が必要で、拡張グリッドビューまたはスキーマデザインビューにて編集を行い、ファイルを保存した際に有効になります。

XML ファイルに StyleVision Power Stylesheet が関連付けられている場合、「Authentic:リンクをデザインファイルに保存」オプションを有効にすることで、XML とともに StyleVision Power Stylesheet を保存することができます。

改行

既存を保持が選択されていると、ファイルを開いたときに、ドキュメント内にあった改行コード文字がそのまま保持されます。他にも、改行に使用されるコードを、CR&LF (PC)、CR (Mac OS)、または LF (UNIX) から選択することができます。

出力フォーマット無し

テキストビューでは、要素をインデントすることで、要素構造における位置を表示することができます。 ユーザーは各要素に対して、このインデントをオーバーライドすることができます。出力フォーマット無 しのフィールドにて要素名を入力してください。このフィールドに入力された全ての要素以下にある子 要素は、タブなどの空白スペースが挿入されなくなります(以下のスクリーンショットを参照)。

全ての要素に対して構造インデントを適用:

11	
12	
13	<pre><rr></rr></pre>
14	-
15	- ///

出力フォーマット無しが要素 xsrestriction に対して指定される:

11	
12	<pre></pre>
13	- /// // // // // // // // // // // // /

保存して完了する

設定が完了したら、「OK」をクリックして完了します。

ファイルタイプ

ファイルタイプセクションでは、各種ファイルに対する Authentic Desktop の振る舞いを指定することができます。



ファイルタイプリストボックスからファイルの種類を選択し、その種類のファイルに対する動作をカスタマ イズすることができます。

Windows エクスプローラー設定

Windows エクスプローラーで使用されるファイルタイプの説明と MIMIE 準拠のコンテンツタイプを定義するほか、この種類のドキュメントを編集するデフォルトディターとして Authentic Desktop を指定することができます。

準 拠

Authentic Desktop では、様々な種類のファイルに対して使用することのできる数多くの機能が搭載 されています。特定のファイルタイプに対して使用されるオプションの種類は、この準拠オプションによ リセットされます。デフォルトの準拠設定では、既に数多くのファイルタイプが定義されており、新た なファイルタイプを加える場合か、特定のファイルタイプをあえて別の準拠に従うように変更する場合 を除き、これらの設定を変更するのはおすすめしません。

デフォルトビュー

このグループでは、各ファイルの種類に対して使用されるデフォルトのビューを指定することができます。上のスクリーンショットは、エディションのファイル型のセクションを表示しています。使用中のエディションが Enterprise エディションでは無い場合、スクリーンショットに表示されているよりも少ないビューの種類のみが搭載されています。

テキストビュー

このチェックボックスにより、特定の種類のファイルに対して、構文の色分けを行うよう指定することができます。

自動検証を無効化

このオプションにより、各ファイルに対して自動検証を有効にすることができます。自動検証は通常、 ファイルが開かれたとき、またはビューが切り替えられたときに行われます。

空の要素を省略して <E/> で保存する

XML ドキュメントや、XML ドキュメントにより出力 されたドキュメントを使用 するアプリケーションの中には、XML 1.0 仕様により定義 されている (<Element/> という形式の)空要素 が原因 でエラーを起こす ことがあります。このオプションを無効にすることで、Authentic Desktop は、<Element></ Element> という形式で要素の保存を行います。

新たなファイル拡張子を追加

新 しいファイルタイプを、ファイルタイプリストに追 加 します。ファイルタイプを追 加 した後 には、このタブ にあるオプションを使 用 することで、このファイルタイプの定 義 を行 う必 要 があります。

選択 されたファイル拡張子を削除

現在選択されているファイルタイプと全ての関連する設定を削除します。

保存して完了する

設定が完了したら、「OK」をクリックして完了します。

エンコード

エンコードセクションでは、ファイルのエンコーディングを指定することができます。

新規 XML ファイルのデフォルトエンコーディング

ドロップダウンリストから新規 XML ファイルで使用 されるデフォルトエンコーディングを選択 することがで きます。ここで指定 したエンコーディングが、新たなドキュメントを作成 する際、XML 宣言にて使用 さ れます。デフォルトエンコーディングに 2バイトまたは 4バイトのエンコーディング (UTF-16、UCS-2 または UCS-4)をデフォルトエンコーディングとして選択 した場合、リトルエンディアン、またはビッグエンディアン を選択 することができます。

既存の XML ファイルにおけるエンコーディングはそのまま保持されますが、「ファイル」エンコード」コマンドにより変更することができます。

XML ファイルを未知のエンコードを使用して開く

XML ファイルのエンコーディングを決定することができない、または XML ドキュメントにてエンコーディングの指定がなされていない場合、このコンボボックスで選択 したエンコーディングがファイルを開く際に 使用されます。.

XML でないファイルを開く際のエンコーディング

既存の、そして新規に作成された XML 以外のファイルは、このコンボボックスで選択したエンコーディ ングにより開かれます。 ドキュメントのエンコーディングは、 「ファイル | エンコード 」 コマンドにより変更 することができます。

BOM (バイト・オーダー・マーク)

2バイトまたは4バイトのエンコーディングによりドキュメントを保存する場合、ドキュメントの保存を、 (i) リトルエンディアンオーダー、またはリトルエンディアン BOM により行 づか (**UTF-8 でない場合、常に** BOM を作成)、(ii) 検出 されたバイトオーダーまたは検出 された BOM により行 づか (保存時に検出し た BOM を保持)指定することができます。

保存して完了する

「OK」をクリックして完了してください。

ビュー

ビューセクションでは、Authentic Desktop における XML ドキュメントの表示方法をカスタマイズする ことができます。.

整形

「編集 | XML テキストを整形して出力」を選択するか、他のビューからテキストビューへ切り替えることで、XML ドキュメントが「整形」されます。このダイアログにある「インデントを使用」オプションが有効になっている場合、整形を行うためにインデントが使用されます。=

プログラムロゴ

アプリケーションの起動時に表示されるスプラッシュ画面を無効にすることで、起動時間を短縮することができます。更に、ライセンスを購入されている場合、プログラムロゴ、著作権情報、登録に関する詳細の印刷をXMLSpy にて行わないように指定することができます。

ウィンドウタイトル

各ドキュメントウィンドウのめにウィンドウタイトルは、ファイル名、または、フルパス名になります。

保存して完了する

設定が完了したら、「OK」をクリックして完了します。

XSL

XSL セクションでは、アプリケーション内から行われる XSLT 変換 とXSL-FO 変換 におけるオプションを定義することができます



XSLT 変 換

Authentic Desktop には Altova XSLT 1.0 エンジンならびに Altova XSLT 2.0 エンジンが搭載 されてお 以 XSLT 変換に使用することができます。xsl:stylesheet または xsl:transform 要素の version 属性の値 により、適切な XSLT エンジン (1.0 または 2.0)が変換には使用 されます。

XSLT を使った XML ドキュメントの変換には、以下のどれかを使用することができます:

- 内蔵の Altova XSLT エンジン (Altova (XSLT 1.0、XSLT 2.0、および XSLT 3.0)。
- MSXML 3.0、4.0、または 6.0 パーサー:お使いのコンピューターで使用されている MSXML の バージョンが明らかな場合、選択を行うことができ、そうでない場合、アプリケーションによるバー ジョンの選択を行わせるべきでしょう(バージョンを自動的に選択オプションがデフォルトでは選択 されています)。このオプションが選択されている場合、利用できる最新バージョンが選択されま す。
- 選択した外部 XSLT プロセッサー:外部 XSLT プロセッサーを指定するコマンドライン文字列を入 力する必要があります。コマンドライン文字列を構築する際に、以下の変数を使用することができ ます:
 - 81 = 処理する XML ドキュメント
 - 82 **= 生成する出力ファイル**

83 = 使用する XSLT スタイルシート(XML ドキュメント内に、スタイルシートへの参照が含まれない場合)

例えば以下のコマンドにより、Saxon (XSLT 1.0)を使った単純な変換を行うことができます:

myxsltengine.exe -o output.xml input.xml stylesheet.xslt parametername=parameter-value

Authentic Desktop からこのコマントを実行するには、外部 XSL 変換プログラムラジオボタンを 選択し、以下の文字列をテキストボックスに入力してください。

c:\myxsltengine\myxsltengine.exe -o %2 %1 %3 parameter-name=parametervalue

適切なチェックボックスにチェックを入れることで、外部プログラムから得られた出力やエラーメッセージをAuthentic Desktopのメッセージウィンドウに表示することができます。

アウトプットウィンドウを再利用オプションを使用することで、後に続く変換の結果を、同じ出力ウィンドウにて表示することができます。XML ファイルがプロジェクトに格納されており、アウトプットウィンドウを再利用オプションが無効になっている場合、この設定はプロジェクトプロパティの出力ファイルパスにおける保存先フォルダーも**無効になっている**場合のみ適用されます。

┌─XSL 変換のファイル出力	先	
🗌 保存先フォルダー:		参照
🔲 ファイル拡張子:	.html	

XSL-FO 変換

FO ドキュメントは FO プロセッサーによい処理 され、FO プロセッサーの実行可能 ファイルへのパスは、 XSL-FO 変換 エンジンのテキストボックスにて指定する必要 があります。変換処理は、「XSL/ XQuery | XSL-FO 変換 」メニューコマンドから行うことができます。ソースファイル (IDE 内部 からコマ ンドが指定 された場合は、アクティブなドキュメント)が XSL-FO ドキュメントの場合、変換に FO プロ セッサーが呼び出 されます。ソースドキュメントが XML ドキュメントの場合、まずは XSLT 変換 を使っ て XML ドキュメントを XSL-FO ドキュメントへ変換する必要 があります。この XSLT 変換 では、アプリ ケーションのデフォルトエンジンとして指定 した XSLT エンジンのほか、デフォルト FO プロセッサーとして 指定 した FO プロセッサー内部 に組み込まれた XSLT エンジンを使用 することもできます(上を参照)。 これらオプションを選択 するには、適切 なラジオボタンをクリックしてください。

設定が完了したら、「OK」をクリックして、オプションダイアログを閉じてください。

メモ: <u>Apache XML プロジェクト</u>の FOP プロセッサをインストールするオプションを解除しない限以次のフォルダー内にインストールされます C:\ProgramData\Altova \SharedBetweenVersions。インストールされるとXSL-FO エンジン入力ボックス内にパスが自動的に入力されます。FO プロセッサへのパスを希望するように設定することができます。しかしながら、FO プロセッサを使用する他の Altova 製品によい同じパスがにより使用される場合、FO プロセッサを選択して設定をすることもできます (StyleVision とAuthentic Desktop)。

保存して完了する

設定が完了したら、「OK」をクリックして、オプションダイアログを閉じてください。
スクリプト

スクリプトタブでは、アプリケーションの起動時に、スクリプト環境を有効にすることができます。有効に するには、スクリプトをアクティブにするチェックボックスにチェックを入れてください(以下のスクリーン ショットを参照)

スクリプト	
アクティブにする	
📝 スクリプトをアクティブにする	
グローバル スクリプト プロジェクト ファイル:	
C:\\Examples\SampleScripts.asprj 🗸 🗸	参照
- 自動スクリプト処理 ☑ Authentic Desktop のスタート時にオートマクロを実行 □ イベントを処理	

Authentic Desktop に対してグローバルスクリプトプロジェクトを設定するには、スクリプトをアクティブに するチェックボックスにチェックを入れた後、必要な Altova スクリプトプロジェクト (asprj) ファイルを指 定します。その他にも、(i) Authentic Desktop の起動時にマクロを自動的に実行するか、(ii) プロジェ クト内のアプリケーションイベントハンドラーを自動的に実行するかを有効に(または無効に)すること ができます。

保存して完了する

「OK」をクリックして完了してください。グローバルリソーススクリプトプロジェクト内のマクロがマクロコマンドのサブメニューに表示されます。

ソース管理

ソース管理セクションではソース管理プロバイダーの指定を行い、デフォルトのログイン ID やユーザー 設定を各種 ソース管理プロバイダーに対して指定することができます。

Microsoft Visual SourceSafe	✓ Advanced
.ogon ID (SourceSafe):	
MYFAVID	
Perform background status updates every 50 Display output messages from plug-in Get everything when opening a project Check in everything when closing a project Don't show Check Out dialog box when checking Don't show Check In dialog box when checking	00 ms g out items in items
Don't show Check In dialog box when checking Keep items checked out when checking in or ad	in items Iding items
f dialogs were hidden using Don't show this again, slick Reset to view them again.	Reset

ソース管 理 プラグイン

現在インストールされている中から、使用するソース管理プラグインをコンボボックスのドロップダウンリ ストを使って選択することができます。目的のソース管理を選択したら、次のテキストボックスにてログ イン ID を入力します。 **詳細」**ボタンをクリックすることで、選択されたソース管理プラグインに特化し たダイアログが表示され、ソース管理プラグインに対して設定を行うことができます。これら設定の内 容はソース管理プラグインによい変化します。

ユーザー設 定

以下のようなユーザー設定を利用することができます:

- ユーザーが指定した時間間隔ごとに、バックグラウンドでステータスの更新を行うか、この機能を無効に することもできます。巨大なソース管理データベースを使用する場合、多くの CPU パワーならびにネット ワークリソースが消費されることになります。バックグラウンドのステータス更新機能を無効にするか、イ ンターバルの時間を長く取ることで、システムのスピードを図ることができます。
- プロジェクトを開くならびに閉じる時に、それぞれファイルのチェックアウトやチェックインを自動的に行うことができます。
- チェックアウトならびにチェックインを行う際に、ダイアログの表示を無効にすることができます。
- ダイアログボックスにて次回から表示しないオプションを選択した場合、リセットボタンが有効になります。リセットボタンをクリックすることで、ダイアログが再度表示されるようになります。

保存して完了

設定が完了したら、「OK」をクリックして完了します。

ネットワーク プロキシ

ネットワーク プロキシ セクションでは、カスタムのプロキシの設定を構成することができます(下のスク リーンショット)。デフォルトは、システムのプロキシの設定が使用されますが、設定はユーザーの介入 無しで作動します。代替のネットワークプロキシを設定する場合、設定を定義するオプションを使用 してください。

ネットワーク プロキシ		
◎ システムのプロキシ設定を使用	∃(U)	
◎ 自動プロキシ構成(A)		
🔽 自動検知	の設定(D)	
スクリプト URL(L)		
◎ 手動のプロキシ構成(M)		
HTTP プロキシ(H)	ポート	、 <u> </u>
このプロキョ	シサーバーをすべてのプロトコールのために使用する(P)	
SSL プロキシ(S)	ポート	、 <u> </u>
プロキシ無し(N)		
📃 プロキシの	サーバーをローカルのアドレスのために使用しない (X)	
┌現在のプロキシの設定(C)──		
URL のテスト(T)	C	
IE 自動プロキシ構成が見つか メソッド WPAD (テスト URL h PAC に対して解決された DIR プロキシを使用していません	りました。 ttp://www.example.comを使用) IECT (NO PROXY).	

システムプロキシの設 定 の使 用

システムプロキシ設定を介して構成可能なインターネットエクスプローラー (IE)設定を使用します。 netsh.exe winhttp を介して構成される設定が必要とされます。

自動 プロキシの構成

以下のオプションを使用することができます:

- 自動検知の設定: DHCP または DNS を使用して WPAD スクリプト (http://wpad.LOCALDOMAIN/wpad.dat)を検索し、プロキシセットアップのためにこのスクリプトを使用します。
- *スクリプト URL:* プロキシセットアップのために使用 されるプロキシ自動構成 (.pac) スクリプト に対する HTTP URL を指定 します。
- *再 ロード* 現在の自動プロキシ構成をリセットして再 ロードします。このアクションには Windows 8 または以降が必要とされ、30秒程の時間が必要です。

手動のプロキシの構成

ホスト名 とポートを対応 する製品 のプロキシのために手動 で指定 します。サポートされるスキームはホ スト名 に含まれている場合 があります (例: http://hostname)。プロキシがスキームをサポートする場 合、対応 するプロトコールと同じである必要 はありません。 以下のオプションを使用することができます:

- このプロキシサーバーをすべてのプロトコールのために使用する:全てのプロトコルのために HTTP プロキシのホスト名とポート使用します。
- プロキシ無し、セミコロン(;)により区別されているプロキシを使用しないホスト名、ドメイン名、または、ホストのためのIPアドレスのリスト。IPアドレスは切り捨てられず、IPv6アドレスは角かっこで囲まれる必要があります(例:[2606:2800:220:1:248:1893:25c8:1946])。 ドメイン名は、ドットと共に開始される必要があります(例:.example.com).
- プロキシのサーバーをローカルのアドレスのために使用しない:チェックされている場合、プロキシ無しリストのために <local> を追加します。このオプションが選択されている場合、次の場合、プロキシは使用されません: (i) 127.0.0.1、(ii) [::1]、(iii) (.) ドット文字を含んでいないホスト名すべて。

現 在 のプロキシの設 定

プロキシの検知の詳細なログを提供します。URL のテストフィールドの右の「更新」ボタンを使用して更新することができます(例、URL のテストを変更する場合、または、プロキシの設定が変更された場合)。

 URL のテスト: URL のテストを使用して、どのプロキシが特定のURL ために使用されている かを確認することができます。URL を使用して、I/O はされません。プロキのシ自動構成が 選択されている場合、このフィールドは空にしておく必要があります(システムプロキシの設定 の使用、または、自動プロキシの構成を使用して)。

11.11 ウィンドウメニュー

ウィンドウメニューには、殆どの Windows アプリケーションに備わっているコマンドが含まれており Authentic Desktop セッションで開かれた個々のドキュメントウィンドウを管理することができます。 開かれているドキュメントウィンドウを重ねて表示したり、上下や左右に並べて表示、最小化したド キュメントウィンドウへアクセスすることができます。各種入力へルパーを有効無効にすることができる ほか、ウィンドウダイアログから、目的のドキュメントウィンドウへ直接アクセスすることもできます。

11.11.1 重ねて表示

このコマンドにより開かれている全てのドキュメントウィンドウが重ね合わさって表示されるように配置されます。

11.11.2 上下に並べて表示

このコマンドにより、開かれている全てのドキュメントが上下に並べて配置され、複数のドキュメントを同時に閲覧することができます。

11.11.3 左右に並べて表示

このコマンドにより、開かれている全てのドキュメントが左右に並べて配置され、複数のドキュメントを同時に閲覧することができます。

11.11.4 プロジェクトウィントウ

このコマンドによりプロジェクトウィンドウの表示 非表示を切り替えることができます。

プロジェクトウィンドウはドッキングすることのできるウィンドウで、タイトルバーをドラッグすることで、オリジナルの位置から切り離してフロートウィンドウにすることができます。タイトルバーを右クリックしてウィンドウをドッキングしたり隠したりすることができます。

11.11.5 情報ウィンドウ

このコマンドにより情報ウィンドウの表示 非表示を切り替えることができます。

情報 ウィンドウはドッキングすることのできるウィンドウで、タイトルバーをドラッグすることで、オリジナル の位置から切り離してフロートウィンドウにすることができます。タイトルバーを右クリックしてウィンドウを ドッキングしたり隠したりすることができます。

11.11.6 入力ヘルパー

このコマンドにより全ての入力ヘルパーの表示非表示を切り替えることができます。

3つある入力 ヘルパーウィンドウはドッキングすることのできるウィンドウで、タイトルバーをドラッグすることで、オリジナルの位置から切り離してフロートウィンドウにすることができます。タイトルバーを右クリックしてウィンドウをドッキングしたり隠したりすることができます。

11.11.7 出力ウィンドウ

出力ウィンドウは、(検証結果のようなメッセージを表示する)メッセージウィンドウや、(XPath条件式評価の結果を表示する)XPath ウィンドウをタブ化したウィンドウです。初期状態ではメインウィンドウの下部に表示されます。 出力ウィンドウ」コマンドにより、出力ウィンドウの表示 非表示を切り替えることができます。

出力 ウィンドウのウィンドウはドッキングすることのできるウィンドウで、タイトルバーをドラッグすることで、 オリジナルの位置から切り離してフロートウィンドウにすることができます。タイトルバーを右クリックして ウィンドウをドッキングしたり隠したりすることができます。

出力ウィンドウに関する詳細は、テキストビューセクションの出力ウィンドウを参照ください。

11.11.8 プロジェクトおよび入力ヘルパー

このコマンドによりプロジェクトウィンドウならびに入力へいいーの表示非表示を選択することができます。

11.11.9 全てオン /オフ



このエアンドにより、以下にあるドッキング可能な全てのウインドウの表示 非表示を切り替えることができます:

- <u>プロジェクトウィンドウ</u>
- 情報ウィンドウ
- 3つの入力へルパーウィンドウ
- <u>出力ウィンドウ</u>

このコアンドによりドキュメンド以外の全ウインドンを素早、隠して、ドキュメントの表示を作業エリア全体に広げることができます。

11.11.10現在開かれているウィンドウ

このリストには開かれている全てのウィンドウが表示され、ドキュメント間の移動を素早ぐ行うことができます。

	全てオン/オフ(<u>A</u>)
	<u>1</u> OrgChart.xml
~	2 blocks.xml 3 navigation.xml

Ctrl-tab または Ctrl + F6 キーボードショートカットを使用 することでも、開かれているウィンドウ間の切 リ替えを行うことができます。

11.12 ヘルプメニュー

ヘルプメニューにはヘルプならびに Authentic Desktop に関する更に詳しい情報を得るために必要 なコマンドが、ウェブサーバー上で利用できる情報へのリンクや、サポートページへのリンクとともに収め られています。



ヘルプメニューからは 登録ダイアログを起動することもでき、購入した製品のライセンスキーを入力することができます。

11.12.1 目次、インデックス、検索

- ▼ 目次
 - ᆿ <u>説明</u>

ヘルプウィンドウの左側のペインに目次を表示した、Authentic Desktopの画面上のヘル プマニュアルを開きます。目次はヘルプドキュメント全体の概要を表示しています。目次の エントリをクリックしてトピックに移動することができます。

- ▼ インデックス
 - ᆿ <u>説明</u>

ヘルプウィンドウの左側のペインにキーワードインデックスを表示したAuthentic Desktopの 画面上のヘルプマニュアルを開きます。目次はヘルプドキュメント全体の概要を表示して います。インデックスはキーワードをリストし、キーワードをダブルクリックすることでトピックへ 移動することができます。キーワードが1つ以上のトピックにリンクされている場合は、トピッ クのリストが表示されます。

- ▼ 検索
 - ᆿ <u>説明</u>

ヘルプウィンドウの左側のペインに検索ダイアログを表示したAuthentic Desktopの画面上のヘルプマニュアルを開きます。単語を検索するには、入力フィールドに検索対象を入力して、「Return」を押します。ヘルプシステムは、ヘルプドキュメント全体で全文検索を行いとットしたリストを返します。アイテムを表示するためにはアイテムをダブルクリックします。

11.12.2 キーボードマップ

「ヘルプ | キーボードマップ」コマンドにより、Authentic Desktop内にある全てのコマンドが、メニュー ごとに分けられて表示されます。各メニューコマンドには、その説明とキーボードショートカットが表示されます。

? ヘルプ キーボード		
😂 🝙 カテゴリー(C):	ファイル(F)	▼ 設定するアクセラレータ(A) デフォルト ▼
	キー	説明
ファイル(F)すべて保存(L)		開いているすべてのドキュメントを保存します
ファイル(F)すべて閉じる		開いているすべてのドキュメントを閉じます
ファイル(F)エンコード(E)		現在のファイルの文字セットエンコードを設定または変更する
ファイル(F)メールで送信(M)		電子メールでドキュメントを送信します
ファイル(F)上書き保存(S)	Ctrl+S	アクティブなドキュメントを上書き保存
ファイル(F)再ロード(D)		開いているファイルを再ロードします
ファイル(F)印刷(P)	Ctrl+P	アクティブなドキュメントを印刷します
ファイル(F)印刷プレビュー(V)		印刷プレビュー
ファイル(F)印刷設定(R)		プリンターと印刷の設定を変更します
ファイル(F)名前を付けて保		アクティブドキュメントを名前を付けて保存します
ファイル(F)新規作成(N)	Ctrl+N	新しいドキュメントを作成します
ファイル(F)終了(X)		アプリケーションを終了:ドキュメントの保存を確認
ファイル(F)閉じる(C)		アクティブなドキュメントを閉じます
ファイル(F)開<(O)	Ctrl+O	既存のドキュメントを開く
ファイル(F)非アクティブをすべ		アクティブなドキュメントを除くすべてのドキュメントを閉じる

特定のメニューに収められているコマンドを確認するには、カテゴリーコンボボックスからメニュー名を選択します。 印刷 アイコンをクリックすることで、コマンドを印刷することができます。

11.12.3 ライセンス登録、注文フォーム、登録、最新情報のチェック

- ▼ ソフトウェアのライセンス認証
 - □ *説明*

Altova 製品 ソフトウェアをダウンロードすると、無料評価キーまたは購入されたライセンスキーを使用して、製品 にライセンスを供与、または、ライセンスの認証を行うことができます。

- ・ 無料評価キー:初めて製品のダウンロードとインストールを行うと、ソフトウェアア クティベーションダイアログが表示されます。ダイアログでは無料評価キーコードをリ クエストすることができます。ユーザーの名前、所属会社名、そして電子メールア ドレスを入力して、無料評価キーを取得するボタンをクリックしてください。入力さ れた電子メールアドレスに評価キーが送信されます。ソフトウェアアクティベーション のダイアログボックスにて送信されたキーコードを入力し、「OK」をクリックすること で、Altova製品を使用することができるようこなります。ソフトウェアは30日の間ア ンロックされます。
- ★モ: ソフトウェアアクティベーションダイアログにてライセンス情報を入力する際には、ライセンス電子メールに記された情報を正確に入力するようにしてください。マルチューザーライセンスの場合、Name フィールドに、各ユーザーの名前を入力してください。

ライセンスの電子メールとAltova 製品へのライセンス供与(有効化)の異なる方

Altova から受信するライセンス電子メール:

法

- ライセンスの詳細(名前、企業、電子メール、キーコード)
- .altova_licenses ファイル拡張子を持つライセンスファイルを添付 ファイルとして

Altova 製品のライセンスを認証するには、以下のうち1つ行ってください:

- Altova 製品のソフトウェアのライセンス認証ダイアログ内に電子メールで受信したライセンスの詳細を入力し、「OK」をクリックします。
- 適切な場所にライセンスファイル(.altova_licenses)を保存し、ライセンスファイルをダブルクリックし、表示されるダイアログに必要な情報を入力し、「キーの適用」をクリックして完了します。
- 適切な場所にライセンスファイル (.altova_licenses)を保存し、 Altova LicenseServer のライセンスプールにアップロードします。以下 を行うことができます: (i) 製品のソフトウェアライセンス認証ダイアログ

から Altova 製品 からライセンスを取得します。または (ii) Altova LicenseServer から製品 へのライセンスを割 リ当 てます。 LicenseServer の使用の詳細に関しては、下記のトピックを参照し てください。

ソフトウェアアクティベーションダイアログ (*下のスクリーンショット*) は、「ヘルプ | ソフトウェア アクティベーション」をクリックすることにより常にアクセスすることができます。

以下の方法によりソフトウェアをアクティブ化することができます:

- (「新しいキーコードを入力する」をクリックして)ライセンスキーの情報を入力します。または、
- (「Altova LicenseServer を使用」をクリックして)ネットワーク上の Altova LicenseServer からライセンスを取得します。Altova LicenseServer は、使用中の Altova 製品のためのライセンスをライセンスプール上に有する必要があります。ライ センスが LicenseServer プール内に存在する場合、Software Activation ダイアロ グ内に表示されます(下のスクリーンショット)。「保存」をクリックしてライセンスを 取得します。

Altov	Altova XMLSpy Enterprise Edition 2017 Software Activation					
Thar assig via A	Thank you for choosing Altova XMLSpy Enterprise Edition 2017 and welcome to the software activation process. You can view your assigned license or select an Altova LicenseServer which provides a license for you. (NOTE: To use this software you must be licensed via Altova LicenseServer or a valid license key code from Altova.)					
If yo	ou do not want	to use	e Altova LicenseServer click h	ere to enter a key code manually	Enter Key Code	
To a	ctivate vour so	ftware	e please enter or select the n	ame of the Altova LicenseServer on vol	ır network.	
Alto	va LicenseServ	er:	DOC.co			~ 5
	A license is s	alreada	v assigned to you on License	Server at DOC co		
	Name	AQA	(Concurrent 50 Users)			
	Company	Altov	a GmbH			
	User count	50				
i	License type	conc	urrent			
	Expires in	-				
	SMP	248 0	days left			
						~
	Return License Check out License Save Close					
	Connected to Altova LicenseServer at DOC.altova.com					

マシン固有のライセンスが LicenseServer からインストールされると7日間は、 LicenseServer に戻すことができません。7日過ぎると(「ライセンスを戻す」をク リックして)マシンのライセンスを LicenseServer に戻すことができ、このライセンス は、他のクライアントにより LicenseServer から取得することができます。 LicenseServer 管理者は、LicenseServer のWeb UIを使用して、取得されたラ イセンスの割り当てを解除することができます。ライセンスの返却は、マシン固有の ライセンスのみに適用され、現在使用中のライセンスには適用されないことに注意してください。

ライセンスのチェックアウト

ライセンスが製品マシン上に保管されるように、ライセンスをライセンスプールから 30日間チェックアウトすることができます。これにより、オフラインで作業することが 可能になります。これはどても役に立ちます。Altova LicenseServerにアクセスでき ない環境(例えば、旅行中にAltova製品がインストールされたラップトップコンピュ ーターで作業する)場合などが挙げられます。ライセンスはチェックアウトされていま すが、LicenseServerは、ライセンスが使用中と表示し、ライセンスは他のマシン で使用することができません。ライセンスはチェックアウトの期間が終わると自動的 にチェックインされた状態を戻します。または、チェックアウトされたライセンスはソフ トウェアのライセンスの認証ダイアログのボタンを使用して**チェックイン**することがで きます。

ライセンスをチェックアウトするには以下をおこないます:(i) ソフトウェアのライセンス の認証ダイアログで ライセンスのチェックアウト をクリックします (上のスクリーン ショット参照)。(ii) ライセンスのチェックアウトダイアログ内から、チェックアウトの期間を選択し、チェックアウトをクリックします。ライセンスがチェックアウトされます。 ソフトウェアのライセンスの認証ダイアログは、チェックアウト期間の終わりを含む チェックアウトの情報を表示します。ダイアログ内のライセンスのチェックアウト ボ タンは、チェックインボタンに変わります。チェックインボタンをクリックして、ライセン スをチェックインすることができます。ライセンスは自動的にチェックイン状態に戻さ れるため、選択したチェックアウトの期間がオフラインで作業する期間をカバーする ように確認してください。

メモ: ライセンスのチェックアウトを可能にするには、LicenseServer上で有効化される必要があります。この機能が有効化されていない場合、エラーメッセージが表示されます。この場合、LicenseServer管理者に連絡してください。

サポートコードのコピー

「サポートコードのコピー」をクリックして、ライセンスの詳細をクリップボードにコピーしてください。 オンラインサポートフォームを使用してサポートをリクエストする際に必要なデータが負ライセンスの詳細に含まれています。

Altova LicenseServer を使用することにより、IT管理者は、リアルタイムでネットワーク上の 全てのライセンスの概要、および、クライアントの割 り当 てと クライアントのライセンスの使 用状況を確認することができます。LicenseServer を使用する利点は、ですから、多数の Altova ライセンスを管理することのできる管理機能です。Altova LicenseServer は、 <u>Altova Web サイトで</u>無料で提供されています。Altova LicenseServer および Altova LicenseServer を使用したライセンスの供与に関する詳細は、<u>Altova LicenseServer</u> ドキュ メントを参照してください。

- ▼ 注文 フォーム
 - ᆿ <u>説明</u>

ソフトウェア製品のライセンスを購入する準備が整った場合、ソフトウェアアクティベーション ダイアログにて「キーコードを購入する」、ボタンを選択するか、「ヘルプ | 注文フォーム」 コマンドを選択して、Altova オンラインショップへアクセスすることができます。

▼ 登録

ᆿ <u>説明</u>

Altova 製品登録ページをブラウザーのタブに表示 します。Altova ソフトウェアを登録することとより、最新の製品の情報が得られます。

- ▼ 更新のチェック
 - □ <u>説明</u>

Altova サーバーに接続して、より新しいバージョンの製品が利用可能かどうかチェックし、その結果を表示します。

11.12.4 他のコマンド

- ▼ サポートセンター
 - ᆿ <u>説明</u>

インターネット上にある Altova サポートセンターへのリンクとなっています。サポートセンター には FAQ やディスカッションフォーラムが含まれており、問題の解決方法を探り、Altovaの 技術サポートスタッフへアクセスすることができます(現在英語のみの提供となります)。

- ▼ WEB 上の FAQ
 - ᆿ <u>説明</u>

インターネット上にある Altova の FAQ へのリンクとなっています。 FAQ データベースは Altova のサポートスタッフにより常時更新されています。

- ▼ コンポーネントのダウンロード
 - ᆿ <u>説明</u>

インターネット上 にある Altova のコンポーネントダウンロードセンターへのリンクとなっていま す。このリンク先 から様 々 なコンポーネントソフトウェアをダウンロードして、Altova 製品 ととも に使用 することができます。ソフトウェアコンポーネントは XSLT や XSL-FO プロセッサーから アプリケーションサービスプラットフォームまで、幅広く提供 されています。コンポーネントダウ ンロードセンターにてご利用 になれるソフトウェアは、通常無料でご利用になれます。

- インターネット上のAuthentic Desktop
 - □ <u>説明</u>

インターネット上 にある <u>Altova</u> ウェブサイトへのリンクとなっています。<u>Altova</u> ウェブサイトでは、Authentic Desktop や関連するテクノロジーについて確認することができます。

- ▼ Authentic Desktop について
 - ᆿ <u>説明</u>

スプラッシュ画面と製品のバージョン番号が表示されます。Authentic Desktopの64-ビット バージョンを使用している場合、アプリケーション名の後のサフィックス(x64)を示していま す。32-ビットバージョンにはサフィックスはありません。

11.13 コマンドライン

コマンドラインから Authentic Desktop に搭載 されているいくつかのコマンドを実行することができます。以下のコマンドを利用することができます:

ファイルを開く

コマンド authentic.exe file.xml 動作: Authentic Desktop にてファイル filexml を開く

複数のファイルを開く

コマンド authentic.exe file1.XML file2.xml 動作: Authentic Desktop にてファイル file1xml ならびに file2xml を開く

SPS ファイルを XML ファイルに割 り当 てて、Authentic ビューによる編集を行う

コマンド: authentic.exe myxml.xml /sps mysps.sps 動作: mysps.sps をSPS ファイルとして、ファイル myxmlxml を Authentic ビューで開く /sps フラグによ 以 あとに続くSPS ファイルが /sps フラグの前にある XML ファイルに (Authentic ビュー編集を行うた めに)割 リ当 てられます。

SPS ファイルから新 たな XML テンプレートを開く

 $\exists \forall \forall \mathcal{F}$ authentic.exe mysps.sps

動作:新規 XML ファイルを Authentic ビューで開きます。表示は SPS をベースにしたもので、SPS スキーマをベースにした新規 XML ファイルが作成 されます。 XML ファイルの保存を行う際に、ファイル の名前を指定する必要があります。

Chapter 3 プログラマーのレファレンス

プログラマーのレファレンス

Authentic Desktop は、自動化されたサーバーです:プログラムすることのできるオブジェクトを自動化 クライアントと呼ばれる他のアプリケーションに公開します。自動化クライアントは直接にオブジェクトおよび自動化サーバーにより使用することのできる機能にアクセスすることができます。ですから、 XMLSpyAuthentic Desktop の自動化クライアントは、例えば、Authentic Desktop の XML 検証機能を 使用することができます。この結果、開発者は、Authentic Desktop の既存の機能を使用してアプリケー ションを拡張することができます。

Authentic Desktop のプログラムすることのできるオブジェクHは、Authentic Desktop のアプリケーション API を使用して、自動化クライアントで使用することができます。Authentic Desktop のアプリケーション API は、以降省略してアプリケーション API と呼ばれます。アプリケーション API のオブジェクトモデル、お よび使用可能なすべてのオブジェクトの詳細はこのドキュメントで説明されています(Application API のセクションを参照してください)。

アプリケーション API は、以下の環境の内部からアクセスすることができます:

- スクリプトエディター
- <u>IDE</u> <u>プラグイン</u>
- <u>外部プログラム</u>
- <u>ActiveX</u>統合

これらの環境について下で簡単に説明がされています。

スクリプトエディター: Authentic Desktop 機能のカスタム化と変更

XMLSpy のインストールを機能を変更および追加することでカスタム化することができます。ユーザー 入力のためのフォームを作成し、新しいメニューコマンドおよびツールバーショートカットを含むユーザ ーインターフェイスを変更することもできます。これらすべての機能は、アプリケーション APIのオプジェク トとの作動するスクリプトを書くことにい達成することができます。これらのタスクを効率的に実行する ために、XMLSpy は、ビルドインのスクリプトエディターを搭載しています。機能の完全な説明と使用 方法はこのドキュメントのスクリプトエディターセクションで説明されています。サポートされるプログラミ ング言語は、JScript およびVBScript です。

IDE プラグイン: Authentic Desktop のためのプラグインの作成

Authentic Desktop により自身のプラグインを作成し、Authentic Desktop に統合することができます。 これを行うには、Authentic Desktop のプラグインのための特別なインターフェイスを使用します。プラ グイン作成に関する詳細は、Authentic Desktop IDE <u>プラグイン</u>のセクションで説明されています。

アプリケーション オブジェクトは、IDE プラグインにより実装され、アプリケーションにより呼び出される必要があるメソッドにパスされます。 IDE プラグインの実装のために使用される 典型的な言語は、**C#** および C++ です。詳細に関しては、<u>Authentic Desktop IDE</u> <u>プラグイン</u>のセクションを参照してください。

外 部 プログラム

さらに、外部スクリプトともに Authentic Desktop を操作することができます。例えば、Authentic

Desktop を指定した時刻に起動し、Authentic Desktop内のファイルを開き、ファイルを検証し、印刷するスクリプトを作成することができます。外部スクリプトは、アプリケーションAPIを使用して、これらのタスクを実行します。 アプリケーションAPIの詳細に関しては、アプリケーションAPIのセクションを参照してください。

Authentic Desktop の外部からアプリケーション APIを使用する場合は、Authentic Desktop のインス タンスが最初に開始されることが必要ですこれがどのように行われるかは、使用されるプログラミング 言語によります。個別の言語に関する情報は、<u>プログラミング言語</u>のセクションを参照してください。

基本的には、Authentic Desktop は COM 登録を使用して起動されます。そして、Authentic Desktop インスタンスと関連する Application オブジェクトを返します。 COM 設定により、既に実行 中の Authentic Desktop に関連したオブジェクトが返されます。 COM オブジェクトの作成と呼び出し をサポートするプログラミング 言語を使用することができます。 共通する項目は、下にリストされてい ます。

- JScript およびVBScript スクリプト ファイルは、シンプルな構文を持ち、COM オブジェクトにア クセスするためにデザインされています。DOS コマンド ライン から直接、または Windows Explorer をダブルクリックして実行することができます。これらのスクリプトはシンプルな 自動 化 タスクのために使用 されます。
- <u>C#</u>は、広範囲の既存の機能を搭載した完全なプログラミング言語です。Access to COM オブジェクトへのアクセスは、自動的にラップされた C#を使用します。
- C++ は、COM への直接的なコントロールを与えますが、他の言語よりも多量のコードが必要です。
- Java: Altova 製品には、アプリケーション API をラップして、Java の完全な外観と使用感を 与える、ネイティブの Java クラスが搭載 されています。
- 役に立つ代替の他のプログラミング言語は、次のとおりです:アプリケーション、Perl、および Pythonのための Visual Basic

ActiveX 統合

XMLSpy ActiveX コントロールを使用して アプリケーション API にアクセスする特別の場合、この機能 は、Authentic Desktop 統合パッケージがインストールされている場合のみ使用することができます。 すべての ActiveX コントロールには、元になる機能のための対応する COM オブジェクトを返すプロパ ティがあります。マネージャーコントロールは Application オブジェクト、ドキュメントコントロール、 Document オブジェクト、およびプレースホルダオブジェクトを与え、プロジェクトツリーを含む箇所では、 Project オブジェクトを返します。これらのオブジェクトによりサポートされるメソッドは、アプリケーション API のインターフェイスのセクションで説明されています。ActiveX コントロール統合のコンテキストで は、このメソッドを使用する必要がなくこの点を考慮する必要があります。詳細に関してはActiveX 統合を参照してください。

プログラマーのレファレンスについて

Authentic Desktop のプログラマーのレファレンスは、次のセクションから構成されています:

- <u>スクリプトエディター</u>: Authentic Desktop で使用することのできるスクリプト環境のためのユー ザー参照。
- IDE プラグイン: Authentic Desktop のための プラグイン作成の説明
- アプリケーション API: アプリケーション API のためのレファレンス
- <u>ActiveX 統合</u>: ActiveX コントロールを使用した、Authentic Desktop GUI および Authentic Desktop 機能の統合の方法の案内および参照。

1 Scripting Editor

The Scripting Editor of Authentic Desktop uses the Form Editor components of the Microsoft .NET Framework, and thus provides access to the Microsoft .NET Framework. This means that JScripts and VBScripts not only work with the Authentic Desktop API—which is a COM API and the API of Authentic Desktop—but can also access and use classes of the Microsoft .NET framework.

You can therefore create and use your own macros and forms within Authentic Desktop, and thus add to and modify the functionality of your installation of Authentic Desktop.

- **Note:** Visual Basic is **not supported** as a language in the scripting environment. Only VBScript and JavaScript are. Ensure that you use VBScript syntax and not Visual Basic syntax in the scripting environment.
- **Note:** Microsoft's **.NET Framework 2.0 or higher** is a system prerequisite for Scripting Editor, and it must be installed before Authentic Desktop is installed.

The Scripting Editor

The Scripting Editor (*screenshot below*) opens in a separate window and is accessed via the **Tools | Scripting Editor** menu command in the Authentic Desktop GUI. The programming languages that can be used in the Scripting Environment are **JScript** and **VBScript**. The scripting language can be changed by right-clicking the Project item in the Project window, selecting **Scripting Language**, and selecting the language you want.

Scripting Editor - NewProject7 -	[Registration]	x
🗅 🖻 🔂 🖬 🔛 🗠 🖓	× 🖻 🛍 🚑 🏘 🚊 🏪	
	 Country: Start Date: 9/23/2009 CheckedListBox Cancel CheckedListBox 	
Behavior AllowDrop False False	Close	• •

What you can do with the Scripting Editor

In the Scripting Editor, you can create Forms, Event Handlers, and Macros to build up a Scripting Project. A Scripting Project can then be set as the Global Scripting Project for Authentic Desktop, thus enabling scripts in the Scripting Project to be used in the application. Additionally, different Scripting Projects can be assigned to different Authentic Desktop projects, thus allowing different scripts to be used for different Authentic Desktop projects.

Every script project can define the .NET runtime version it wants to use. An application can handle multiple scripting projects with different .NET runtime versions simultaneously, but the appropriate .NET version must be installed. For example, script projects with .NET 4.0 will only run on computers having .NET 4.0 installed.

Documentation about the Scripting Editor

The documentation describing the Scripting Environment (this section) is organized into the following parts:

- <u>An overview</u>, which provides a high level description of the Scripting Editor and Scripting Projects.
- <u>A list of steps required to create a Scripting Project</u>.
- <u>An explanation of Global Declarations</u>, together with an example.
- A description of how to create Forms.
- <u>A discussion of Authentic Desktop-specific event handlers</u>.
- <u>An explanation of how to use macros</u> in the Scripting Editor and in Authentic Desktop.

1.1 Overview

The Scripting Editor provides an interface in which you can: (i) graphically design Forms while assigning scripts for components in the Form; (ii) create Event Handlers, and (iii) create Macros.

These Forms, Event Handlers, and Macros are organized into scripting projects, which are then assigned to Authentic Desktop application projects and can be used in the application.

Variables and functions can be defined in a Global Declarations script, which is always executed before Macro or Event Handler scripts.

This section gives an overview of the Scripting Editor and Scripting Projects. It is organized into the following sections:

- <u>Scripting Projects in Authentic Desktop</u>, which describes how the scripting projects you create with the Scripting Editor will be used in Authentic Desktop.
- <u>The Scripting Editor GUI</u>, which provides a detailed look at the different parts of the Scripting Editor GUI and how they are to be used.
- <u>Components of a Scripting Project</u>, which explains the different components that go to make up a scripting project.

The details about the creation of the various components (<u>Global Declarations</u>, <u>Forms</u>, <u>Event</u> <u>Handlers</u>, and <u>Macros</u>) are described in their respective sections.

.NET assemblies

Every scripting project can have references to .NET assemblies—in addition to the default references. .NET assemblies can be added for the whole scripting project or for individual macros (by using the new CLR.LoadAssembly command in the source code; see <u>Built-in Commands</u>). Assemblies can be added, for example, from the Global Assembly Cache.

To add an assembly, right-click the project or macro, and, from the context menu that pops up, select Add .NET Assembly | Assembly from Global Cache (GAC).

This works in the same way as with Visual Studio and allows access not only to the complete Microsoft .NET Framework but also to any user-defined assembly.

1.1.1 Scripting Projects in Authentic Desktop

All scripts and scripting information created in the Scripting Editor are stored in **Altova Scripting Projects** (asprjfiles).

You can create any number of Altova Scripting Projects. After a scripting project has been created, it can be used in the following ways:

- It can be set as the global scripting project for Authentic Desktop. Scripts in the global scripting project can then be called from within the application, and macros of the Global Scripting Project can be used for all Authentic Desktop projects.
- It can be assigned to an Authentic Desktop project (as an application project). When an Authentic Desktop project is open in Authentic Desktop, scripts in the associated scripting project can be called.

Your Authentic Desktop package contains a sample scripting project called SampleScripts.asprj. This file contains global declarations for a few standard tasks and is located in the folder: C:\Users\<username>\Documents\Altova\Authentic Desktop2018
\AuthenticExamples.

Setting the global scripting project of an application

The global scripting project of an application is set in the Scripting tab of the Options dialog of Authentic Desktop (*screenshot below*, **Tools | Options**).

スクリプト	
アクティブにする	
▼ スクリプトをアクティブにする	
グローバル スクリプト プロジェクト ファイル:	
C:\\Examples\SampleScripts.asprj	参照
自動スクリプト処理 Authentic Desktop のスタート時にオートマクロを実行	

To set a global scripting project for Authentic Desktop, check the *Activate Scripting* check box and then browse for the Altova Scripting Project (.asprj) file you want. You can also specify: (i) whether Auto-Macros in the scripting project should be automatically executed when Authentic Desktop starts, and (ii) whether application event handler scripts in the project should be automatically executed or not; check or uncheck the respective check boxes accordingly.

Note: Nested script execution is possible, i.e. Macros can call other macros, and events are received during macro, or event, execution.

Assigning a scripting project to an Authentic Desktop project

A scripting project is assigned to an Authentic Desktop project as follows:

- 1. In the Authentic Desktop GUI, open the required application project.
- 2. Select the menu command **Project | Script Settings**. The Scripting dialog (*screenshot below*) opens.

スクリプト
スクリプト プロジェクト ファイル ▼ プロジェクト スクリプトをアクティブにする (Altova¥XMLSpy2018¥Examples¥SampleScripts.asprj ▼) ■ Examples.spp に対する相対パスにする
自動スクリプト処理 V Authentic Desktop プロジェクトのロード時にオートマクロを実
ок <i>キ</i> ャンセル

- 3. Check the *Activate Project Scripts* check box and select the required scripting project (asprj file). If you wish to run Auto-Macros when the Authentic Desktop project is loaded, check the *Run Auto-Macros* check box.
- 4. Click **OK** to finish.
- **Note:** To deactivate (that is, unassign) the scripting project of an Authentic Desktop project, uncheck the *Activate Project Scripts* check box.

1.1.2 The Scripting Editor GUI

The Scripting Editor GUI is shown below. It has the following parts:

- A toolbar
- A <u>Scripting Project Tree pane</u> (top left-hand side)
- A <u>Properties and Events pane</u> (bottom left)
- A Main Window with Design and Source tabs
- A Form Object Palette (right-hand side)

Scripting Editor - NewProject7 -	Registration]	X			
□ ☞ 🗟 🖬 🚰 🗠 ∾ 🐰 🗶 🛍 🖓 榊 🏯 🏪					
Forms Registration Forms Events Macros	Image: Second and the second and th	•			
Macro1 Text TextAlign TopLeft UseMnemonic UseWaitCursor False	Country: Start Date: 9/23/2009 OK Cancel Cancel Cancel Concel				
AllowDrop False	ComboBox ListView TreeView TabControl Close				

Scripting Editor toolbar

The Scripting Editor toolbar contains icons for:

- Standard file commands such as **New**, **Open**, **Save**, and **Print**. These commands are used to create new scripting projects, open existing scripting projects, and save and print scripting projects.
- Standard editing commands such as **Copy**, **Paste**, **Undo**, **Redo**, **Find**, and **Replace**. Note that the **Find** and **Replace** commands are applied to code in the Source tab of the Scripting Editor.

Scripting Project Tree

The Scripting Project Tree (*screenshot below*) shows the various components of the scripting project, structured along four main branches: (i) Global Declarations, (ii) Forms, (iii) Events, and (iv) Macros.



The Scripting Project Tree provides access to each component of the scripting project. For example, in order to display and edit a particular Form, expand the Forms folder in the tree (*see screenshot above*), right-click the Form you wish to display or edit, and click **Open** from the context menu that pops up.

A quicker way to open a Form, Event, macro, or the Global Declarations script, is to double-click the respective icon, or text. To delete a Form or Macro from the scripting project, right-click the component and select the **Delete** command from the context menu.

The Scripting Project Tree pane contains a toolbar with icons (screenshot below).



The icons, from left to right, are for: (i) <u>creating a new macro</u>, (ii) <u>creating a new form</u>, (iii) <u>running a macro</u>, and (iv) <u>debugging a macro</u>. These commands are also available in the context menu that appears when you right-click any component in the Scripting Project Tree.

Properties and Events

The Properties and Events pane (screenshot below) displays the following:

- Form properties, when the Form is selected
- Object properties, when an object in a Form is selected. (The screenshot below shows, at left, the properties of the object selected in the Form at right.)
- Form events, when a Form is selected
- Object events, when an object in a Form is selected

	∄ ≵↓ 💷 💉				<u>_ </u>
	Text	Start Date: 📃 🔺	I м Г		
	TextAlign	TopLeft	I INAME: I		
	UseMnemonic	True	Email:		
	UseWaitCursor	False			
Œ	Behavior				
Œ	Data		Country:		•
ΙE	Design				
	(Name)	Label1 📃	OOO	9/23/2009	
	Locked	False		372372003	
Ð	Focus				
	Layout	<u> </u>		ηκ	Cancel
() Ir tł	Name) ndicates the name us ne object.	ed in code to identify			

To switch between the properties and events of the selected component, click, respectively, the

Properties icon (third from left in the Properties and Events toolbar, *see screenshot above*) and the **Events** icon (fourth from left).

The first and second icons from left in the toolbar are, respectively, the **Categorized** and **Alphabetical** icons. These display the properties or events either organized by category or organized in ascending alphabetical order.

When a property or event is selected, a short description of it is displayed at the bottom of the Properties and Events pane.

Main Window

The Main Window displays one component at a time and has one or two tabs depending on what is being displayed. If a Global Declarations script, an Event, or a Macro is being displayed, then a single tab, the Source tab, displays the source code of the selected component.

The Source tab supports:

- syntax coloring
- source code folding
- setting/deleting bookmarks using CTRL+F2
- autocompletion entry helper with parameter info
- Goto Brace, Goto Brace Extend
- Zoom In / Zoom Out
- full method/property signature shown next to the autocompletion entry helper
- brace highlighting during code entry
 - if (x == y.GetName(a, b, c()))
- mouse over popups; placing the mouse over a known method or property, displays its signature (and documentation if available)

If a **Form** is being displayed, then the Main Window has two tabs: a Design tab showing and enabling the layout of the Form, and a Source tab containing the source code for the Form. Content in both the Design tab and Source tab can be edited.

Note: Since JScript and VB Script are untyped languages, entry helpers and auto-completion is supported only in cases of "fully qualified constructs" and "predefined" names.

If names start with <code>objDocument</code>, <code>objProject</code>, <code>objXMLData</code>, **or** <code>objAuthenticRange</code>, members of the corresponding interface will be shown. Auto-completion entry helper and parameter info are shown during editing, but can also be obtained on demand by pressing **Ctrl+Space**.

Form Object Palette

The Form Object Palette contains all the objects that are available for designing Forms and looks something like the screenshot below. Registered ActiveX controls can be added to the Form Object Palette by right-clicking the pane and selecting the **Add ActiveX Control** command


To insert an object from the Form Object Palette click the object you want in the palette, then click at the location in the Form where you wish to insert the object. The object will be placed at this location. In many cases you will need to supply some properties of the object via the Properties and Events pane. You can drag the object to other locations as well as resize it. Further, a number of editing commands, such as centering and stacking objects, can be accessed via the context menu of the selected Form object.

Some Form objects, such as Timer, are not added to the Form but are created as Tray Components in a tray at the bottom of the Main Window. You can select the object in the tray and set properties and event handlers for the object via the Properties and Events pane. For an example of how Tray Components are handled, see <u>Form usage and commands</u>.

1.1.3 Components of a Scripting Project

An Altova Scripting Project consists of the following four major components:

- *Global Declarations*, a component which contains definitions of variables and functions that are available to, and can be used by, all Forms, Macros, and Event Handler scripts in the scripting project.
- Forms, a component which contains all the Forms defined in the scripting project.
- *Events*, a component which contains Event Handler scripts for all application-based—as opposed to Form-based—events.
- Macros, a component which contains all the Macros defined in the scripting project.

These components are displayed in and accessed via the Scripting Project Tree of the Scripting Editor (*screenshot below*).

📮 🗁 Project (JScript)
🌆 (GlobalDeclarations)
🕀 💼 Forms
- 🕀 💼 Events
🗄 🕀 🛅 Macros

Given below is a brief description of each of these components.

Global Declarations

The Global Declarations component is a script that contains variables and functions that can be used by Forms, Event Handlers, and Macros. The functions make use of the XMLSpy API to access Authentic Desktop functionality. Creating a variable or function in the Global Declarations module enables it to be accessed from all the Forms, Event Handlers and Macros in the scripting project.

To add a variable or function, open the Global Declarations component (by right-clicking it in the Scripting Project Tree and selecting **Open**) and edit the Global Declarations script in the Main Window. In this script, add the required variable or function.

Forms

In the Scripting Editor, you can build a Form graphically using a palette of Form objects such as text input fields and buttons. For example, you can create a Form to accept the input of an element name and to then remove all occurrences of that element from the active XML document.

For such a Form, a function script can be associated with a text box so as to take an input variable, and an Event Handler can be associated with a button to start execution of the delete functionality, which is available in the XMLSpy API. A Form is invoked by a call to it either within a function (in the Global Declarations script) or directly in a Macro. For details of how to create and edit Forms, see the Forms section.

Event handling

Event Handler scripts can be associated with a variety of available events. You can control events that occur both within Forms (Form events) and within the general application interface (application events). The script associated with an event is executed immediately upon the triggering of that event.

Most events have parameters which provide detailed information about the event. The return value from the script typically instructs the application about how to continue its processing (for example, the application may not allow editing).

An Event Handler runs when the relevant event occurs in the Form or in Authentic Desktop. For details about how to create event handlers, see Event Handlers.

Macros

Macros are used to implement complex or repetitive tasks. Macros do not use either parameters or return values.

In a Macro, it is possible to access all variables and functions declared in the Global Declarations and to display Forms for user input.

For a simple example of creating a Macro, see <u>Writing a Macro</u>. Also see <u>Running Macros</u> for a description of the ways in which a Macro can be called. A Macro is run from within the Authentic Desktop interface by clicking **Tools | Macros |** *[MacroName]*

1.2 Creating a Scripting Project

The broad steps for creating a Scripting Project are as follows:

- 1. Open the Scripting Editor by clicking the command **Tools | Scripting Editor**.
- In the Scripting Editor, open a new scripting project by clicking the New icon in the Scripting Editor toolbar. The Project Settings dialog (*screenshot below*) pops up. You can also access this dialog by right-clicking a project in the Scripting Project Tree pane (in the top left part of the Scripting Editor window) and clicking the command Project Settings.

Project Settings						
Scripting language: JScript						
Target framework: .NET Framework 2.0						
Automatically use higher .NET Framework when specified Target framework is not available on target computer						
For macro and event execution, the target computer will need the preset Target Framework version. You should not change the Target Framework version unless you need new features of a new .NET Framework version.						
Changing the Target Framework does not automatically change the referenced assemblies. To use additional assemblies, you can right-click the project window and use "Add .NET Assembly".						
OK Cancel						

Select either JScript or VBScript in the first combo box and the .NET Framework in the second combo box. To enable higher .NET Frameworks (such as .NET Framework 4.5 on Windows 8), check the *Automatically use higher .NET Framework* check box. Then click **OK**. The new Scripting Project is created.

- 3. Click the **Save** icon in the Scripting Editor toolbar to save the Scripting Project as a .asprj file.
- 4. A Scripting Project can be considered to be made up of several components that work together. These components will typically be a combination of: Global Declarations, Forms, Events, and Macros. They can be created in any order, but you should clearly understand how they work together. The way each type of component is called and executed is <u>described below</u>. How to create each type of component is described in the respective sections about the component type.
- 5. After you have finished creating all the required components, save the Scripting Project (by clicking the **Save** icon in the Scripting Editor toolbar).
- 6. Close the Scripting Editor.
- **Note:** Right-clicking the Project folder and selecting **Project Settings** lets you change the scripting language at any time.

How Forms, Event Handlers, and Macros are called and executed

Forms, Event Handlers, and Macros are all created in the Scripting Editor. However, the way they are called and executed is different for each and has a bearing on how you create your scripting

projects.

- A Form is invoked by a call to it either within a function in the Global Declarations script or directly in a Macro.
- An Event Handler runs when the relevant event occurs in Authentic Desktop. If an Event Handler for a single event is defined in both the Global Scripting Project and the Authentic Desktop-project-specific Scripting Project, then the event handler for the project-specific Scripting Project is executed first and that for the Global Scripting Project immediately afterwards.
- A Macro is executed from within the Authentic Desktop interface by clicking Tools | Macros | [MacroName]. In a Macro, it is possible to access all variables and functions declared in the Global Declarations and to display Forms for user input.

1.3 Global Declarations

The Global Declarations component is present by default in every Scripting Project (see *screenshot below*), and therefore does not have to be created. In order to add variables and functions to the Global Declarations script of a Scripting Project, you need to open the Global Declarations script and add the code fragment to the Global Declarations script. See <u>Components</u> of a Scripting Project and Creating a Scripting Project for more information.

8 🗖 8 8
📮 🗁 Project (JScript)
🎯 (GlobalDeclarations)
🕀 🛅 Forms
🕀 🛅 Events
🗄 🕀 🛅 Macros

To open the Global Declarations script of a Scripting Project, right-click the *Global Declarations* item in the Scripting Project Tree (*screenshot above*), and select **Open**. The Global Declarations script opens in the Main Window.

Note: Every time a macro is executed or an event handler is called, global declarations are re-initialized.

Given below is an example function. Remember that creating a variable or function in the Global Declarations script makes this variable or function accessible to all Forms, Event Handlers, and Macros.

Example function

A function called RemoveAlNamespaces would have code like this:

```
function RemoveA 1Namespaces (objXM LD ata)
{
         if (objXM LD ata == null)
     return;
  if (ob XM LD ata H as C hildren)
                                    {
     var objChild;
     // spyXMLDataElement = 4
     objChild = objXMLDataGetFirstChild(4);
     while (objChild)
                           {
     Rem oveAlNam espaces (objChild);
             trv {
                  varnPos,txtName;
                  txtName = objChildName;
                  if((nPos = txtNam e indexOf("!)) >= 0)
                      objChildName = txtNamesubstring(nPos+1);
                  }
     objChild = objXM LData GetNextChild ();
             }
             catch (Em)
                                              {
                  objChild = null;
             }
```

```
}
}
}
```

Note:

• It is possible to define local variables and helper functions within macros and event handlers. Example:

```
//return value: true allows editing
//return value: false disallows editing
var txtLocal;
function Helper()
{
    txtM essage = txtLocal;
    Application Show Form ("M sgBox");
}
function On_BeforeStartEditing(ob)XM LD ata)
{
    txtLocal = "On_BeforeStartEditing()";
    Helper();
}
```

• Recursive functions are supported.

1.4 Forms

Creating and editing Forms in the Scripting Editor consists of the following steps:

- 1. <u>Creating a New Form</u>. The new Form is created and named, and has properties defined for it.
- 2. <u>Designing the Form</u>. A Form is designed by adding Form Objects to it and assigning values for the different Form Objects.
- 3. <u>Scripting Form Events</u>. Scripts are assigned to Form-related events.

1.4.1 Creating a New Form

Creating a new Form in the Scripting Editor involves the following steps:

- 1. Creating a new Form and naming it
- 2. Specifying the properties of the Form

Creating a new Form and naming it

To add a new Form to a scripting project, click the **Add Form** icon (*highlighted in screenshot below*) in the toolbar of the Project Overview pane. Enter the name of the new Form.



A new Form is added to the project. It appears in the Main Window and an entry for it is created in the Scripting Project Tree pane, under the Forms heading. Press the F2 function key to rename the form, or right click the form name and select Rename from the context menu. In the screenshot below, we have named the new Form *Registration*.



Form properties

The properties of the Form, such as its size, background color, and font properties, can be set in the Properties pane. The screenshot below shows the size and background-color property values in bold, in the *Layout* and *Appearance* categories, respectively.

	Appearance	<u>ـ</u>			
	BackColor	ActiveBorder			
	BackgroundImage	(none)			
	BackgroundImageLayout	Tile			
	Cursor	Default			
Ð	Font	Microsoft Sans Serif, 8.25pt			
	ForeColor	ControlText			
	FormBorderStyle	Sizable			
	RightToLeft	No			
	RightToLeftLayout	False			
	Text				
	UseWaitCursor	False			
Ð	Behavior				
Œ	Data				
Ð	Design				
Ð	Focus				
	Layout				
	AutoScaleMode	Inherit			
	AutoScroll	False 📃			
Ð	AutoScrollMargin	0, 0			
Ð	AutoScrollMinSize	0, 0			
	AutoSize	False			
	AutoSizeMode	GrowOnly			
Œ	Location	0, 0			
Ð	MaximumSize	0, 0			
Ð	MinimumSize	0, 0			
Ð	Padding	0, 0, 0, 0			
	Size	387, 500			
	Width	387			
	Height	500 🔹			

Testing a Form

You can test a form in the Scripting Editor by right-clicking it in the Project Overview pane and selecting the **Test Form** Command.

1.4.2 Form Design and Form Objects

Designing a Form consists of the following steps:

- Placing an object from the Form Object Palette in the Form design.
- Assigning values for the properties of individual Form Objects.
- Assigning scripts for Form-based events.

The Form Object Palette

The Form Object Palette contains all the objects that are available for designing Forms and looks something like the screenshot below. Registered ActiveX controls can be added to the Form Object Palette by right-clicking the pane and selecting the **Add ActiveX Control** command



To insert an object from the Form Object Palette click the object you want in the palette, then click at the location in the Form where you wish to insert the object. The object will be placed at this location. In many cases you will need to supply some properties of the object via the Properties and Events pane. You can drag the object to other locations as well as resize it. Further, a number of editing commands, such as centering and stacking objects, can be accessed via the context menu of the selected Form object.

Some Form objects, such as Timer, are not added to the Form but are created as Tray Components in a tray at the bottom of the Main Window. You can select the object in the tray and set properties and event handlers for the object via the Properties and Events pane. For an example of how Tray Components are handled, see Form usage and commands.

Some of the most commonly used objects are described below:



ab

Label: Adds text fields such as captions or field descriptions.

Button: Adds a button. It is possible to assign bitmaps as background images for these buttons.

×	Check Box: Adds a check box,	which enables	Yes/No type selections.
---	------------------------------	---------------	-------------------------

- **Combo Box:** Adds a combo box, which allows the user to select an option from a dropdown menu.
- **List Box:** Adds a list box, which displays a list of items for selection.
- **TextBox:** Enables the user to enter a single line of text.
- **Rich TextBox:** Enables the user to enter multiple lines of text.

Creating objects and setting their properties

To create an object in the Form, first select the required object in the Form Object Palette and then click the location in the Form where you want to insert it. After the object has been inserted, you can resize it as well as drag it to another location in the Form.

When an object is selected in the design, you can specify its properties in the Properties and Events pane. In the toolbar of the Properties and Events pane, click the Properties icon to display a list of the object's properties.

For example, in the screenshot below, the Label object with the text *Start Date* has been selected in the design. In the Properties and Events pane, the name of the object (which is the name that is to be used to identify the object in code, Label1 in the screenshot below) is given in the *Design* category of properties; in this case, the name of the object is Label1.

The text of the label (which is what appears in the Form) must be entered as the value of the *Text* property in the *Appearance* category of properties.

	<u>₽</u> 2↓ 🔲 🛩	6	
	Text	Start Date:	News
	TextAlign	TopLeft	iname:
	UseMnemonic	True	Email:
	UseWaitCursor	False	
Ð	Behavior		
Ð	Data		Country:
	Design		
	(Name)	Label1 📃	OO
	Locked	False	00
Ð	Focus		
🗆 Layout 📃 💌		<u> </u>	OK Cancel
(Name) Indicates the name used in code to identify the object.		ed in code to identify	

To assign other object properties, enter values for them in the Properties and Events pane.

Testing a Form

You can test a form in the Scripting Editor by right-clicking it in the Project Overview pane and selecting the **Test Form** Command.

1.4.3 Form Events

When an object is selected in the design, clicking on the Events icon in the toolbar of the Properties and Events pane (*fourth icon from left*), displays all the events available for that object (*see screenshot below*). These can be displayed either by category (*screenshot below*) or alphabetically.

	2↓ 🗉 🗲 🖻		
Ξ	Action	▲	
	Click	-	
	DoubleClick		
	MouseCaptureChanged		
	MouseClick		
	MouseDoubleClick		
	Appearance		
	Paint		
	Behavior		
	ChangeUICues		
	ControlAdded		
	ControlRemoved		
	HelpRequested		
	QueryAccessibilityHelp		
	StyleChanged	•	
Click Occurs when the component is clicked.			

For each event, you can enter the name of an existing event handler or function. Alternatively:

- you can double click on an event to create: (i) an empty function script in the *Source* tab of the Main Window, and (ii) an association of the newly created function with the selected event.
- double click a button in the design tab, to directly generate the handler stub in the code window.

The screenshot below was taken after the *Click* event was double-clicked. Notice that an empty event handler function called <code>FormExample_Label1_Click</code> has been created in the Main Window and that, in the Properties and Events pane, this function has been associated with the *Click* event.

. 2↓ 🗉 🗡		
Action	×	
Click	FormExample_Label1_Click	//Occurs when the component is clicked.
DoubleClick		<pre>function FormExample_Labell_Click(objSender,e_</pre>
MouseCaptureCha		₽ (
MouseClick		
MouseDoubleClick		L }

Enter the required scripting code and save the project.

Writing the required scripts

After the visual design of the form is complete, form objects will typically be associated with suitable scripts. The example below is a script that adds colors when a button is clicked. The

script is inserted as an event handler for the Click event of the button Button1 (the event is available in the Properties and Events pane when the button is selected in the design):

```
function FormExample_Button1_Click( objSender, e_EventArgs )
{
     // Sets the ForeColor (red) of the button.
     objSender.ForeColor = CLR.Static( "System.Drawing.Color" ).Red;
     // Sets the BackColor (blue) of the button.
     objSender.BackColor = CLR.Static( "System.Drawing.Color" ).Blue;
     // Sets the form BackColor (green).
     objSender.FindForm().BackColor =
CLR.Static( "System.Drawing.Color" ).Green;
}
```

1.5 Events

The Events folder of the scripting project (*see screenshot below*) contains folders for the following type of events:

- Application Events
- Document Events
- Authentic View Events
- Grid View Events
- Text View Events

Note that these events are Authentic Desktop-specific, as opposed to Form-based events. Each of the folders listed above contains a set of events for which Event Handler scripts can be written.

Application Events, for example, are shown in the screenshot below.

To access the event handler script of any of these events, right-click the event and select **Open** from the context menu. The script will be displayed in the Main Window (*see screenshot below*) and can be edited there. After you have finished editing the script, save changes by clicking the **Save** command in the toolbar of the Scripting Editor.



Note the following points:

- Event Handlers need function headers with the correct spelling of the event name. Otherwise the Event Handler will not be called.
- It is possible to define local variables and helper functions within Macros and Event Handlers. Example:

```
//return value: true allows editing
//return value: false disallows editing
var txtLocal;
function Helper()
{
    txtM essage = txtLocal;
    Application Show Form ("M sgB ox");
}
```

function On_BeforeStartEditing(ob)XMLData)

```
{
  txtLocal= "On_BeforeStartEditing()";
  Helper();
}
```

- In order for events to be processed, the Process Events options must be toggled on in the Scriptings options of Authentic Desktop. See <u>Scripting Projects in Authentic Desktop</u> for details.
- Also see <u>Programming Points</u>.

Application Events

OnInitialize

The OnInitialize event is raised after the main window becomes visible but before any project is loaded. This event is not raised if the application can't be loaded at all.

OnRunning

If the application is completely loaded and after the OnInitialize event occurs, the OnRunning event is raised.

OnShutdown

The event is raised after any open project and all documents have been closed on shutdown of the application. The main window is no longer visible.

Example

The following script is an Event Handler for the On_BeforeOpenProject event. It allows you to add a script that will be executed each time before Authentic Desktop opens a project. The example script below sequentially opens all XML files located in the XML folder of the project and validates them. If the validation fails, the script shows the validation error and stops. If a file passes the validity test, it will be closed and the next file will be opened.

Enter the following script for the On_BeforeOpenProject() event, and then save the scripting project.

```
function On BeforeOpenProject()
var bOK;
var nIndexnCount;
var objlem sobjXM LFolder = null;
 objlem s = Application CurrentProjectRootlem s;
nCount = objlem sCount;
 // search for XML folder
 for(nIndex = 1nIndex <= nCountnIndex++)</pre>
                                                  {
  var txtExtensions;
  txtExtensions = objlem s.lem (nIndex) FileExtensions;
 if(txtExtensions_indexOf("xml") >= 0) {
  ob%MLFolder = objlem s.ltem (nIndex);
break;
}
}
// does XML fobler exist?
if(ob<sup>+</sup>XM LFolder) {
varobjChildpbjDoc;
```

nCount = objXM LFolderC hild Item s Count;

```
// step through associated xm l files
for(nIndex = 1nIndex <= nCountnIndex++)</pre>
                                               {
  objChild = objXMLFolderChildItem s.Item (nIndex);
  try {
   objDoc = objChildOpen();
   // use JScript method to access out-parameters
   varstrEmor = new Amay(1);
   varnEmorPos = new Amay(1);
   var ob BadData = new A may (1);
   bOK = objDoc IsValid (strE mornE morPos objB adD ata);
   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 txtM essage variable
     // txtM essage = Position:" + nEmorPos[0] + "\n" + // strEmor[0];
     // txtMessage += "\n\nXML\n" + obBadData[0]Name + "," +
    //
          objBadData[0]TextValue;
     11
     // Application Show Form ("MsgBox");
    break;
    }
   obDocCbse(true);
   obDoc = null;
  }
  catch (Em) {
   // displaying the error description here is a good idea
   // txtMessage = EmDescription;
   // Application Show Form ("MsgBox");
   break:
   }
  }
 1
}
```

Testing the Event Handler

Switch to Authentic Desktop, and open a project to see how the BeforeOpenProject event is handled.

1.6 Macros

Macros automate repetitive or complex tasks. In the Scripting Environment, you can create a script that calls application functions as well as custom functions that you have defined. This flexibility provides you with a powerful method of automating tasks within Authentic Desktop. This section about macros is organized as follows:

- <u>Creating and Editing a Macro</u> describes how to create a new macro and edit an existing one.
- <u>Running a Macro</u> explains how a macro can be run from the Scripting Editor and from the broader Authentic Desktop environment as well.
- <u>Debugging</u> describes how macros can be debugged.

Key points about macros

Given below is a summary of important points about macros.

- Any number of macros can be added to the active scripting project. These macros are saved in the Altova Scripting Project file (.asprj file).
- Functions that are used in a macro can be saved as a Global Declaration. All Global Declarations are also saved in the Altova scripting project file (.asprj file).
- The macro can be tested by running it from within the Scripting Editor, and it can be debugged from within the Scripting Editor.
- Authentic Desktop can have one global Scripting Project, and a second scripting project, assigned to the currently loaded project, active at any one time; the macros are available to both of them. See <u>Running a Macro</u> for details.

1.6.1 Creating and Editing a Macro

The following operations enable you to create a new macro and edit an existing macro.

Creating a new macro

Right-click the Macro folder in the Scripting Projects tree and select **Add Macro** from the context menu. (The **Add Macro** command can also be selected from the context menu of any item in the Scripting Projects tree.) Alternatively, click the **New Macro** icon in the toolbar of the Scripting Projects tree.

The newly created (and empty) macro document is displayed in the Main Window, and the name of the macro is displayed in the title bar of the Scripting Editor (*screenshot below*).



Naming or renaming a macro

To name or rename a macro, click the macro name in the Scripting Project tree and press the **F2** function key, or right click the name and select **Rename** from the context menu.

Opening a macro

To open a macro, right-click the macro in the Macros folder of the Scripting Project tree (see screenshot above), and select the **Open** command. The macro is displayed in the Main Window and its name is displayed in the title bar of the Scripting Editor (screenshot below). Alternatively, double-clicking a macro in the Scripting Project tree opens it in the Main Window.

Scripting Editor - NewProject17 - [Macro2]

Editing the macro

To edit a macro, enter or edit its code in the Main Window. For example, the following code creates the Form named Form1 in memory and then shows it. Form1 must already have been created (using the Scripting Editor's Form creation) before this macros is run.

```
obForm = CreateForm (Form 1');
obForm ShowDiabg();
```

The following macro uses the RemoveAllNamespaces function to remove all namespaces in the active XML document.

The RemoveAllNamespaces function itself will have to be defined in the Global Declarations script. After the RemoveAllNamespaces function has been defined, the macro is complete and can be run.

Note: Macros do not support parameters or return values.

Setting a macro as an Auto-Macro

When a macro is set as an Auto-Macro it can be run automatically when: (i) Authentic Desktop is started, or (ii) an Altova Authentic Desktop project is loaded in Authentic Desktop. To specify whether Auto-Macros should be run in each of these two events, check the *Run Auto-Macros* option in the Automatic Script Processing pane of the relevant dialogs:

- When Authentic Desktop is started: the Scripting tab of the Authentic Desktop Options dialog (Tools | Options menu command).
- When an Authentic Desktop project is loaded into Authentic Desktop: the Scripting dialog (screenshot below, **Project | Scripting Settings** menu command).

スクリプト					
スクリプト プロジェクト ファイル ▼ プロジェクト スクリプトをアクティブにする (Altova¥XMLSpy2018¥Examples¥SampleScripts.asprj ▼)					
 Examples.spp に対する相対パスにする 自動スクリプト処理 Authentic Desktop プロジェクトのロード時にオートマクロを実 					
OK キャンセル					

To set a macro as an Auto-Macro, right-click the macro in the Scripting Project tree and select the command **Set as Auto-Macro**. This is a toggle command; so to remove the Auto-Macro setting of a macro, select the command again.

1.6.2 Running a Macro

To run a macro in the Scripting Editor, right-click the macro in the Scripting Project tree and select the command **Run Macro**.

There are different ways to run a macro from Authentic Desktop:

- Via the Tools | Macros menu of Authentic Desktop.
- By creating and using a toolbar button for a macro.
- By creating and using a menu item for a macro.

Note that only one macro can be run at a time. After a macro (or event) is executed, the script is closed and global variables lose their values.

The Authentic Desktop command to run Macros

The **Tools | Macros** menu command (*screenshot below*) opens a submenu containing the macros defined in the Scripting Project that is currently active in Authentic Desktop. The active Scripting Projects are specified in the Scripting tab of the Options dialog, or in the Scripting tab of the project settings.

マクロ ト	AddMacroMenu
	CloseAllButActiveDoc
	SearchPath

From the submenu of available macros, select the macro to run. The macro will be executed.

Toolbar icon

You can create an icon in the toolbar or a menu item that runs a selected macro. To do this, click **Tools | Customize | Macros**. This causes the Customize dialog to be displayed (*screenshot below*).

Customize	×
Commands Toolbars Keyboard N	Menu Macros Plug-Ins Options
Macros: AddMacroMenu CloseAllButActiveDoc SearchPath	Associated commands: S CloseAllButActiveDoc
Display text: Close All Docs Except Active Doc	Macro name: CloseAllButActiveDoc
Add Command	Edit Icon Remove
	Close

Now do the following:

- 1. In the Macros tab of the Customize dialog, select the required macro from the Macros pane. The macros in the Macros pane are those in the active Scripting Project (which is specified in the Scripting tab of the Options dialog).
- 2. In the *Display Text* input field enter the name of the icon. This name will appear when the cursor is placed over the icon when it is in the toolbar.
- 3. Click **Add Command** to add it to the list of commands.
- 4. Select the command and click Edit Icon to create a new icon.
- 5. Drag the finished icon from the *Associated Commands* pane and drop it on to the toolbar or menu when the cursor changes from an arrow to an I-beam or line.
- 6. Macros can even be assigned their own shortcuts in the Keyboard tab of the Customize dialog (see screenshot above).

To remove the toolbar icon, open the Macros tab of the Customize dialog and drag the icon out of the toolbar and into the *Associated Commands* pane. Select the command in the *Associated Commands* pane and click **Remove** to remove the command from the pane.

Item in the Tools menu

The XMLSpy API includes a function, AddMacroMenuItem(), to add macros as menu items to the **Tools** menu. This function can be used to add one or more macros to the **Tools** | **Macros** list of macros. Typically, you should do this as follows:

1. Add the macro menu item by calling the XMLSpy API function, AddMacroMenuItem().

Application.AddMacroMenuItem("DeleteElements","Delete Elements

Dialog");

• The function's first parameter (DeleteElements in the example listing above) is the

name of the macro. If you run the macro and there is an open project having scripts associated with it, Authentic Desktop searches for the macro in the project scripts first.

If there are no project scripts, or if Authentic Desktop cannot find the macro, then it looks for the macro in the global scripts.

- The second parameter (Delete Elements Dialog) is the display text for the menu item.
- 2. Reset the **Tools** menu by calling ClearMacroMenu(). This removes all previously added menu items

The best way to call these two functions is with the Autorun macro of the global scripting project or the On OpenProject event.

1.6.3 Debugging a Macro

You can debug a macro using an installed debugger. To do this, right-click the macro in the Scripting Project tree and select the command **Debug Macro**.

This pops up the Just-In-Time Debugging dialog (*screenshot below*), which lists the debuggers available on the machine. Select the debugger you wish to use and click **Yes**.

Just-In-Time Debugging	×
An exception 'Runtime Error' has occurred in Script.	
Possible Debuggers:	
New instance of Microsoft Script Editor	
☐ Set the currently selected debugger as the default.	
Do you want to debug using the selected debugger?	

The selected debugger starts.

1.7 Programming Points

The following programming points should be noted:

 All namespaces and types of the following .NET assemblies can be accessed in the Microsoft .NET Framework per default:

```
System
System.Data
System.Design
System.Drawing
System.Windows.Forms
System.XML
```

Additional assemblies can be added to the scripting project via the project's context menu, or dynamically (at runtime) in the source code by using <u>CLR.LoadAssembly</u>.

• Out-parameters from methods of the XMLSpy API require special variables in JScript. Given below are some examples.

```
// 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);END
```

• Out-parameters from methods of the.NET Framework require special variables in JScript. For example:

 .NET Methods that require integer arguments should not be called directly with JScript Number Objects which are Floating Point Values.

For example, instead of:

```
var objCustomColor = CLR.Static( "System.Drawing.Color" ). FromArgb( 128,
128, 128 );
```

use:

```
var objCustomColor = CLR.Static( "System.Drawing.Color" ).
FromArgb( Math.floor( 128 ), Math.floor( 128 ), Math.floor( 128 ) );
```

• To iterate .NET collections the JScript Enumerator as well as the .NET iterator

technologies can be used:

For example:

• .NET templates can be instantiated as shown below:

var coll = CLR C reate ("System C ollections Generic List<System String>");

or

```
CLR In port("System");
CLR In port("System C ollections Generic");
var dictionary = CLR C reate ("Dictionary< String, Dictionary< String, String > >");
```

.NET Enum values are accessed as shown below:

```
var enumValStretch =
CLR.Static( "System.Windows.Forms.ImageLayout" ).Stretch;
```

• Enumeration literals, as defined in the Altova type libraries, can now be used instead of numerical values.

```
objExportXMIFileDlg.XMIType = eXMI21ForUML23;
```

1.7.1 Built - in Commands

This section lists:

- Built-in commands
 <u>alert
 conform
 doevents
 CreateForm
 lastform
 prompt
 ShowForm
 watchdog
 </u>
- <u>.NET interoperability</u> commands <u>CLR.Create</u> <u>CLR.Import</u> <u>CLR.LoadAssembly</u> <u>CLR.ShowImports</u> <u>CLR.ShowLoadedAssemblies</u> <u>CLR.Static</u>

Built-in commands

The following built-in commands are available.

ShowForm(strFormName : String)

Instantiates a New Form object from the given form name and immediately shows it as Dialog. *Return Value:* A Number that represents the generated DialogResult (System WindowsFormsDialogResult).

Example:

var dialogResult = Show Form ("Form Nam e");

Shows Form "FormName" as Dialog:

Form Name	×
First name: Last name:	
	OK Cancel

The DialogResult can be evaluated e.g. by:

<mark>if (</mark> diab	ogResult — CLR Static ("System Windows FormsDialogResult") OK)
	alert("ok");	
else		
	alert("cancel");	

CreateForm (strFormName : String) Instantiates a New Form object from the given Form name. *Return Value:* The Form object (System WindowsFormsForm) of the given name, or null if no Form with such name exists.

```
Example:
    varmyForm = CreateForm ("FormName");
    if (myForm != null)
    {
        myForm textboxFirstNameText = "Daniel";
        myForm textboxLastNameText = "Smith";
        vardialogResult = myForm ShowDialog();
```

Shows Form "FormName" as Dialog - TextBoxes are initialized:

Form Name	×
First name:	Daniel
Last name:	Smith
	OK Cancel

The DialogResult can be evaluated e.g. by:

lastform

This global field can be used to conveniently access the last form object that was created. *Return Value:* Returns a reference to the last form object (System WindowsFormsForm) that was successfully instantiated via CreateForm() or ShowForm().

```
Example:
```

Shows Form "FormName" as Dialog - TextBoxes are initialized (similar to the CreateForm example above):

Form Name	×
First name:	Daniel
Last name:	Smith
	OK Cancel

doevents()

Processes all Windows messages currently in the message queue. *Return Value:* None

Example:

```
for (i=0; i < nLongLastingProcess;++i)
{
     // do long lasting process
     doevents();// process windows messages; give UIa chance to update
}</pre>
```

watchdog(bEnable : boolean)

Long running CPU-intensive scripts cause the watchdog to ask the user if the script should be terminated. The watchdog() method is used to disable or enable this behavior. Per default the watchdog is enabled. *Return Value:* None

Example:

```
watchdog(false) // disable watchdog - we know the next statement is CPU intensive but it will
term inate for sure
doCPUIntensiveScript();
watchdog(true) // re-enable watchdog
```

Usage tip:

Calling watchdog(true) can also be used to reset the watchdog. This can be useful before executing long running (CPU intensive) tasks to ensure they have the maximum allowed script processing quota.

```
alert(strMessage : String) or MsgBox(strMessage : String)
An alert box is used to show a given message. The user will have to click "OK" to proceed.
Return Value: None
```

Example: alert("Hello World");



confirm(strMessage : String)

Opens a dialog that shows the given confirm message.

A confirm box is often used to verify or accept something. The user will have to click either "OK" or "Cancel" to proceed.

Return Value: A Boolean that represents the users answer. If the user clicks "OK", the dialog returns true, if the user clicks "Cancel", the dialog returns false.

Example:



```
prompt(strMessage : String, strDefault : String)
```

Opens a dialog that shows the given prompt message and a TextBox control with a default answer.

A prompt box is often used to input a simple string value.

Return Value: A String that contains the TextBox value or null if the user selected "Cancel".

Example:

	×
	Please enter your name
	Daniel Smith
	OK Cancel
Ľ	

.NET interoperability commands

To allow further interoperability with the .NET Framework additional functions are provided under CLR.

```
CLR.Import(strNamespaceCLR : String)
```

This is the scripting equivalent to the C# using / VB.Net imports keyword. This allows to leave out the given namespaces in successive calls like CLR.Create() and CLR.Static(). Return Value: None

```
Example:
```

Instead of always having to use full qualified names:

```
if (Show Form ("Form Name") == CLR Static ("System W indows Form sDialogResult")OK)
{
     var sName = lastform textboxFirstNameText + " " + lastform textboxLastNameText;
     CLR Static ("System W indows Form sMessageBox")Show ("Hello " + sName);
}
```

one can import namespaces and use the short form:

```
CLR Im port("System W indows Forms");
if (Show Form ("Form Name") == CLR Static("DialogResult")OK)
{
    var sNam e = lastform textboxFirstNam e Text + " " + lastform textboxLastNam e Text;
    CLR Static("MessageBox")Show("Hello " + sName);
}
```

Please note:

Importing a namespace does not add or load the corresponding assembly to the scripting project! Assemblies can be added to the scripting project dynamically (at runtime) in the source code by using <u>CLR.LoadAssembly</u>.

```
CLR. ShowImports ()
Opens a MessageBox dialog that shows the currently imported namespaces. The user will have
to click "OK" to proceed.
Return Value: None
```

```
Example:
```

CLR.ShowImports();



CLR.LoadAssembly(strAssemblyNameCLR : String)

Loads the .NET assembly with the given long assembly name or file path. *Return Value:* A Boolean value. True if the assembly could be loaded, false otherwise.

```
Example:
```

CLR.ShowLoadedAssemblies()

Opens a MessageBox dialog that shows the currently loaded assemblies. The user will have to click "OK" to proceed. *Return Value:* None

Example:

CLR Show LoadedAssem blies ();



CLR.Create(strTypeNameCLR : String, constructor arguments ...)

Creates a new .NET object instance for the given typename. If more than one argument is passed the successive arguments are interpreted as the arguments for the constructor of the .NET object. *Return Value:* A reference to the created .NET object

Examples:

```
var objA may = CLR C meate ("System C ollections A mayList");
var new Item = CLR C meate ( "System W indows Form s ListV iew Item ", "New Item Text" );
var coll = CLR C meate ( "System C ollections Generic List<System String>" );
CLR In port( "System " );
CLR In port( "System C ollections Generic" );
var dictionary = CLR C meate ( "D ictionary< String, D ictionary< String, String > >" );
```

CLR.Static(strTypeNameCLR : String)

Gives access to .NET types that have no instances and contain only static members. *Return Value:* A reference to the static .NET object

```
Examples:
    var enumValStretch =
    CLR.Static("System.Windows.Forms.ImageLayout").Stretch
    varclipboard = CLR Static("System WindowsClipboard");
    clipboardSetText("HelloClipboard");
    if (ShowForm("FormName") == CLR Static("System WindowsFormsDialogResult")OK)
        alert("ok");
```

else

alert("cancel");

Form usage and commands

Form usage is as follows:

With Form objects, the Form Component Tree can be accessed naturally via field access:

For example, suppose there is a Form designed as follows:

```
MyForm
ButtonPanel
OkButton
CancelButton
TextEditor
AxMediaPlayer1
TrayComponents:
MyTimer
```

The Form can then be instantiated from script as:

var objForm = CreateForm("MyForm");

To access one its components the field access can be used:

objForm.ButtonPanel.OkButton.Enabled = false;

or

objForm.TextEditor.Text = "Hello World";

To access Tray Components use the following method on the Form object:

var objTrayComponent = <A form object>.GetTrayComponent(strComponentName :
String);

In our example to get a reference to the Timer Component to enable it use the following:

```
var objTimer = objForm.GetTrayComponent("MyTimer");
objTimer.Enabled = true;
```

For ActiveX Controls the underlying COM object can be accessed via the OCX property:

```
var ocx = lastform.AxMediaPlayer1.OCX; // get underlying COM object
ocx.enableContextMenu = true;
ocx.URL = "mms://apasf.apa.at/fm4_live_worldwide";
```

1.8 Migrating to Scripting Editor 2010 and Later

The Scripting Editor in Authentic Desktop from version 2010 onwards uses a different underlying technology than earlier versions used. Consequently, scripting projects that were created with versions of Authentic Desktop prior to version 2010 might need to be modified. The following points need to be noted.

- If a previous Scripting Projects (.prj file) is opened with the new Scripting Editor (version 2010 and later), the visual layout of Forms will be migrated as faithfully as possible and scripts will be copied as they are in the .prj file. You will then need to modify the scripts to be in accordance with the new technology used by the Scripting Editor, and which is described in this documentation.
- TheView object: The old Scripting Environment provided an artificial property named TheView that was only accessible from inside event handlers. It was used to access the Form that trigged the event (either directly or from one of its child controls). The new Scripting IDE does **not** provide this artificial property but instead provides the same functionality, and much more, with orthogonal <u>built-in scripting helper functions</u> combined with the power of the .NET framework.
- Since all event handlers in the new Scripting Environment get a sender object as a first parameter, the source that triggered the event is always available. By calling the .NET function FindForm() on the sender object one can access the Form object easily. Alternatively (if only one Form is involved) the built-in property lastform can be used. Note that the use of lastform is not constrained to event handlers (as was the case with TheView). It can be used everywhere in script code.

Given below is a list of methods and properties of the TheView object, each accompanied by an alternative mechanism offered by the new Scripting Environment.

Methods

The following methods were provided by the TheView object and must be migrated as explained:

Cancel()

In the new scripting environment the same can be achieved with: lastform.Close(); // Use
.NET Form.Close()

IsFormOpen(Name as String) as Boolean

Since for .NET Forms there is a distinction between showing a Form and instantiating a Form, the previous concept does not directly translate. Instead the user can ask if a certain Form is currently shown. For example:

FormFind(Name as String) as Object
The new Scripting Environment allows you to instantiate more Forms of the same kind. In the old Scripting Environment each Form could only exist once (as a Singleton). Thus there is no equivalent of FormFind(). In the new Scripting Environment.

OpenDoc(File as String)

The same can be achieved with: Application.OpenDocument(File as String)

PumpData()

This corresponds to the built-in function doevents() which processes all Windows messages currently in the message queue.

RunClick(), RunInitialize(), RunTerminate()

There is no direct replacement for these methods. Call the corresponding handlers directly instead.

Properties

The following properties were provided by the TheView object and must be migrated as explained:

ToolTipText as String

To use tooltips in the new scripting environment, the .NET infrastructure can be used. This allows fine-grained control of tooltip behaviour (adjusting delays, when to show, etc). For example, to provide tooltips for a Form with two controls, the following code could to be added to the Form's Load event handler:

```
//Occurs whenever the user loads the form.
function MyForm Load( objSender, e EventArgs )
{
       // Create the ToolTip and associate with the Form container.
      var toolTip = CLR.Create("System.Windows.Forms.ToolTip");
       // Set up the delays for the ToolTip.
       toolTip.AutoPopDelay = 3000;
       toolTip.InitialDelay = 1000;
       toolTip.ReshowDelay = 500;
      // Force the ToolTip text to be displayed whether or not
       // the form is active.
       toolTip.ShowAlways = true;
       // Set up the ToolTip text for several Controls.
       toolTip.SetToolTip(objSender.ProgressBar1,
                    "Shows the progress of the operation");
       toolTip.SetToolTip(objSender.Button1,
                    "Click Button to start the processing");
}
```

Color as Long

Since all Form/controls in the new Scripting Environment are .NET controls from the System.Windows.Forms namespace, the possibilities to modify colors, background image, fonts, and all other visual aspects are numerous. For example, every Visual Component has the properties BackColor and ForeColor to modify the visual appearance. The following handler could be used to change the color of a button at runtime:

Please refer to the .NET documentation to find out more about this topic: <u>http://msdn.microsoft.com/en-us/library/system.windows.forms.aspx</u>

2 IDE Plugins

Authentic Desktop allows you to create your own IDE plug-ins and integrate them into Authentic Desktop.

Use plug-ins to:

- Configure your version of Authentic Desktop, add commands through menus, icons, buttons etc.
- React to events from Authentic Desktop.
- Run your specific code within Authentic Desktop with access to the complete Authentic Desktop API

Authentic Desktop expects your plug-in to implement the <u>IXMLSpyPlugIn</u> interface. VB.NET, C# and C++ examples are included with your installation package and are located in the Authentic2018\AuthenticExamples\IDEPlugin folder of your Authentic Desktop installation.

See <u>ATL sample files</u> for an example using C++.

2.1 Registration of IDE PlugIns

Authentic Desktop maintains a specific key in the Registry where it stores all registered IDE plugins:

HKEY_CURRENT_USER\Software\Altova\XML Spy\PlugIns

All values of this key are treated as references to registered plug-ins and must conform to the following format:

Value name:	ProgID of the plug-in
Value type:	must be REG_SZ
Value data:	CLSID of the component

Each time the application starts the values of the "PlugIns" key is scanned, and the registered plug-ins are loaded.

Register plug-in manually

To register a plug-in manually, use the "Customize" dialog box of the Authentic Desktop "Tools" menu. Use the "Add Plug-In..." button to specify the DLL that implements your plug-in. Authentic Desktop registers the DLL as a COM server and adds the corresponding entry in its "PlugIns" key.

If you experience problems with manual registration you can check if the CLSID of your plug-in is correctly registered in the "PlugIns" key. If this is not the case, the name of your plug-in DLL was probably not sufficiently unique. Use a different name or perform direct registration.

Register plug-in directly

A plug-in can be directly registered as an IDE plug-in by first registering the DLL and then adding the appropriate value to the "PlugIns" key of Authentic Desktop during plug-in setup for example. The new plug-in will be activated the next time Authentic Desktop is launched.

Creating plug-ins

Source code for sample plug-ins has been provided in the application's (My) Documents folder: Examples\IDEPlugin folder. To build a plug-in from such source code, do the following:

- 1. Open the solution you want to build as a plug-in in Visual Studio.
- 2. Build the plug-in with the command in the Build menu.
- 3. The plug-in's DLL file that will be created in the Bin or Debug folder. This DLL file is the file that must be added as a plug-in (see above).

2.2 ActiveX Controls

ActiveX controls are supported. Any IDE PlugIn which is also an ActiveX control will be displayed in a Dialog Control Bar. A sample PlugIn that is also an ActiveX control is included in the XMLSpyPlugInActiveX folder in the Examples folder of your application folder.

2.3 Configuration XML

The IDE plug-in allows you to change the user interface (UI) of Authentic Desktop. This is done by describing each separate modification using an XML data stream. The XML configuration is passed to Authentic Desktop using the <u>GetUIModifications</u> method of the IXMLSpyPlugIn interface.

The XML file containing the UI modifications for the IDE PlugIn, must have the following structure:

```
<C onfigurationD ata>
<Im ageFile>path To im age file</Im ageFile>
<M odifications>
<M odification>
...
</M odification>
...
</M odifications>
</C onfigurationD ata>
```

You can define icons or toolbar buttons for the new menu items which are added to the UI of Authentic Desktop by the plug-in. The path to the file containing the images is set using the ImageFile element. Each image must be 16 x 16 pixels using max. 256 colors. The image references must be arranged from left to right in a single (<ImageFile>...) line. The rightmost image index value, is zero.

The Modifications element can have any number of Modification child elements. Each Modification element defines a specific change to the standard UI of Authentic Desktop. Starting with version 4.3, it is also possible to remove UI elements from Authentic Desktop.

Structure of Modification elements

All Modification elements consist of the following two child elements:

```
<M odification>
  <A ction>Type of action</A ction>
  <U E lem ent Type="type of U I e lem ent">
  </U E lem ent>
  </M odification>
```

Valid values for the Action element are:

Add - to add the following UI element to Authentic Desktop
 Hide - to hide the following UI element in Authentic Desktop
 Remove - to remove the UI element from the "Commands" list box, in the customize dialog

You can combine values of the Action element e.g. "Hide Remove"

The UlEement element describes any new, or existing UI element for Authentic Desktop. Possible elements are currently: new toolbars, buttons, menus or menu items. The **type** attribute, defines which UI element is described by the XML element.

Common UIElement children

The ID and Name elements are valid for all different types of XML UIElement fragments. It is however possible, to ignore one of the values for a specific type of UIElement e.g. Name is ignored for a separator.

 $< \mathbb{D} > < / \mathbb{D} >$

<N am e></N am e>

If UlEement describes an existing element of the UI, the value of the ID element is predefined by Authentic Desktop. Normally these ID values are not known to the public. If the XML fragment describes a new part of the UI, then the ID is arbitrary and the value should be less than 1000. The Name element sets the textual value. Existing UI elements can be identified just by name, for e.g. menus and menu items with associated sub menus. For new UI elements, the Name element sets the caption e.g. the title of a toolbar, or text for a menu item.

Toolbars and Menus

To define a toolbar its necessary to specify the ID and/or the name of the toolbar. An existing toolbar can be specified using only the name, or by the ID if it is known. To create a **new** toolbar both values must be set. The **type** attribute must be equal to "ToolBar".

```
<U E lem ent Type="ToolBar">
<D>1</D>
<N am e>TestPlugh</N am e>
</U E lem ent>
```

To specify an Authentic Desktop menu you need two parameters:

- The ID of the menu bar which contains the menu. If no XML documents are open in the main window, the menu bar ID is 128. If one or more XML documents are open, the menu bar ID is 129.
- The menu name. Menus do not have an associated ID value. The following example defines the "Edit" menu of the menu bar which is active, when at least one XML document is open:

```
<U E lem ent Type="M enu">
<D>129</D>
<N am e>Edit</N am e>
</U E lem ent>
```

An additional element is used if you want to create a new menu. The Place element defines the position of the new menu in the menu bar:

```
<U E lem ent Type="M enu">
<D>129</D>
<N am e>Plugh M enu</N am e>
<Place>12</Place>
</U E lem ent>
```

A value of -1 for the Place element sets the new button or menu item at the end of the menu or toolbar.

Commands

If you add a new command, through a toolbar button or a menu item, the UIEement fragment can contain any of these sub elements:

```
<MacroName></MacroName>
<hfo></hfo>
<fmageD></fmageD>
```

If MacroName is specified, Authentic Desktop searches for a macro with the same name in the scripting environment and executes it each time this command is processed. The Info element contains a short description string which is displayed in the status bar, when the mouse pointer is over the associated command (button or menu item). ImageID defines the index of the icon the external image file. Please note that all icons are stored in one image file.

To define a toolbar button create an UIEement with this structure:

```
<U E lem ent Type="ToolB arftem ">
    <!--don t reuse local Ds even the comm ands do the sam e-->
    <D>5</D>
    </wd>

    <D>5</D>
    </wd>

    <Nam e>Open file from repository..</Nam e>
    <!--Set Place To -1 If this is the first button To be inserted-->
    </place>-1</place>
    </wd>

    <Im ageD>0</m ageD>
    </wd>

    <ToolB arD>1

    <!--instead of the toolbar D the toolbar nam e could be used-->
    </wd>

    <ToolB arNam e>TestPlugh</toolB arNam e>
  <//ur>

    </ur>
```

Additional elements to declare a toolbar button are Place, ToolBarID and ToolBarName. ToolBarID and ToolBarName are used to identify the toolbar which contains the new or existing button. The textual value of ToolBarName is case sensitive. The (UIElement) **type** attribute must equal "ToolBarItem".

To define a menu item, the elements MenuID, Place and Parent are available in addition to the standard elements used to declare a command. MenuID can be either 128 or 129. Please see "Toolbars and Menus" for more information on these values.

The **Parent** element is used to identify the **menu** where the new menu entry should be inserted. As sub menu items have no unique Windows ID, we need some other way to identify the parent of the menu item.

The value of the Parent element is a path to the menu item.

The text value of the Parent element, must equal the **parent menu name** of the submenu, where the submenu name is separated by a colon. If the menu has no parent, because its not a submenu, add a colon to the beginning of the name. The **type** attribute must be set to "MenuItem". Example for an UIEement defining a menu item:

```
<U E lem ent Type="M enultem ">
    <!--the following element is a Local comm and D-->
    <D>3</D>
    </name>open file from repository..</name>
    <Place>-1</Place>
    </manuelement>
    </manuelement>
```

Authentic Desktop makes it possible to add toolbar separators and menus if the value of the ID element is set to 0.

2.4 ATL sample files

The following pages show how to create a simple Authentic Desktop IDE plug-in DLL using ATL. To build the DLL it is necessary to know about ATL, the wizards that generate new ATL objects, as well as MS VisualStudio.

To access the API the implementation imports the Type Library of Authentic Desktop. The code reads various properties and calls methods using the smart pointers provided by the #import statement.

In addition, the sample code uses the MFC class CString and the ATL conversion macros such as W2T.

At a glance the steps to create an ATL DLL are as follows:

- 1. Open VisualStudio and select "New..." from the "File" menu.
- 2. Select the "Projects" tab.
- 3. Select "ATL COM AppWizard" and type in a project name.
- 4. Select "Support for MFC" if you want to use MFC classes, or if you want to create a project for the sample code.

Having created the project files you can add an ATL object to implement the IXMLSpyPlugIn interface:

- 1. Select "New ATL Object..." from the "Insert" menu.
- 2. Select "Simple Object" from the wizard and click "Next".
- 3. Type in a name for the object.
- 4. On the "Attributes" tab, select "Custom" for the type of interface, and disable Aggregation.

These steps produce the skeleton code for the implementation of the IDE plug-in interface. Please see the following pages on how to modify the code and achieve some basic functionality.

2.4.1 Interface description (IDL)

The IDL of the newly created ATL object contains a declaration for one COM interface.

- This interface declaration must be replaced by the declaration of IXMLSpyPlugIn as shown below.
- The IDL must also contain the definition of the SPYUpdateAction enumeration.
- Replace the generated default interface name, (created by the wizard) with "IXMLSpyPlugIn" in the coclass declaration. The IDL should then look something like the example code below:

Having created the ATL object, you then need to implement the IDE plug-in interface of Authentic Desktop.

```
import "oaidlidl";
import "ocidlidl";
// ----- please insert the following block into your DL file -----
 typedefenum {
  spyEnable = 1,
  spyD isable = 2,
  spyCheck = 4,
  spyUncheck = 8
 }SPYUpdateAction;
// ----- end insert block ----
// ----- Eg. Interface entry autom atically generated by the ATL wizard -----
11
  [
11
      object,
11
      uuid (AB7CD86A-8145-429A-A1F3-270692EO8AFC),
      helpstring ("IXM LSpyPlugIn Interface")
11
11
      pointer default (unique)
//
11
   interface IXMLSpyPlugIn : Unknown
//
// };
// ----- end autom atically generated Interface Entry
```

// ----- **replace** the **Interface Entry** (shown above) generated for you by the ATL wizard, with the following block -----

```
[
    odl,
    uuid (88F2A 622-4B 7E-42C D-8D 04-3C 0E 5389D D 85),
    helpstring ("KM LSpyPlugh Interface")
]
interface KM LSpyPlugh : Unknown
{
    HRESULT _stdcall OnC omm and ([in] long nD, [in] D ispatch* pXM LSpy);
    HRESULT _stdcall OnUpdateC omm and ([in] long nD, [in] D ispatch* pXM LSpy, [out, retval]
SPYUpdateA ction* pA ction);
    HRESULT _stdcall OnEvent([in] long nEventD, [in] SAFEARRAY (VAR ANT)* arrayParam eters, [in]
D ispatch* pXM LSpy, [out, retval] VAR ANT* pReturnValue);
    HRESULT _stdcall GetU M odifications ([out, retval] B STR* pM odificationsXM L);
```

HRESULT stdcallGetDescription ([out, retval] BSTR * pDescription);

};

```
// ----- end replace block -----
```

// ----- The code below is automatically generated by the ATL wizard and will bok slightly different in your case -----

```
[
uuid (24FE0D1B-3FC0-494E-B36E-1D4CE412B014),
 version (10),
 helpstring ("XMLSpyDEPlugInDLL 10 Type Library")
library XM LSPY DEPLUG NDLLLib
 {
 im portlib ("stdole32 tlb");
 im portlib ("stdole2 tb");
 [
  uuid (3800E791-7F6B-4ACD-9E32-2AC184444501),
  helpstring ("XMLSpyIDEPlugIn Class")
 ]
 coclass XM LSpy DEPlugIn
 {
   [default] interface KMLSpyPlugh; // ----- define KMLSpyPlugh as the default interface -----
 };
};
```

2.4.2 Class definition

In the class definition of the ATL object, several changes must be made. The class has to derive from IXMLSpyPlugIn, the "Interface Map" needs an entry for IXMLSpyPlugIn, and the methods of the IDE plug-in interface must be declared:

```
#ifndef_XMLSPYDEPLUGN_H_
#define_XMLSPYDEPLUGN_H_
```

#include "resource h" // main symbols

```
// CXMLSpyDEPlugh
class ATL NO_VTABLE CXMLSpyDEPlugh :
    public CComObjectRootEx<CComSingleThreadModeD,
    public CComCoClass<CXMLSpyDEPlugh, &CLSD_XMLSpyDEPlugh>,
    public KMLSpyPlugh
{
```

```
public:
CXMLSpyDEPlugIn()
{
}
```

DECLARE REGISTRY RESOURCED (DR XMLSPYDEPLUGN) DECLARE NOT AGGREGATABLE (CXMLSpyDEPlugh)

DECLARE_PROTECT_FINAL_CONSTRUCT()

```
BEG N_COM_MAP (CXM LSpyDEP lugin)
COM_NTERFACE_ENTRY (XM LSpyP lugin)
END_COM_MAP ()
```

```
// XMLSpyDEPlugh
public:
virtualHRESULT stdcallOnCommand(long nD,Dispatch* pXMLSpy);
```

virtual HRESULT stdcall OnUpdateComm and (long nD, D ispatch* pXM LSpy, SPYU pdateAction* pAction);

virtual HRESULT _stdcall OnEvent(long nEventD, SAFEARRAY **anayParam eters, D ispatch* pXMLSpy, VAR ANT* pReturnValue);

virtual HRESULT stdcallGetUM odifications(BSTR* pM odificationsXML);

virtualHRESULT _stdcallGetDescription(BSTR* pDescription);
);

#endif // XMLSPYIDEPLUGIN H

2.4.3 Implementation

The code below shows a simple implementation of an Authentic Desktop IDE plug-in. It adds a menu item and a separator (available with Authentic Desktop) to the Tools menu. Inside the OnUpdateCommand() method, the new command is only enabled when the active document is displayed using the Grid View. The command searches for the XML element which has the current focus, and opens any URL starting with "http://", from the textual value of the element.

```
#im port "XM LSpy tb"
using namespace XMLSpyLib;
HRESULT CXMLSpyDEPlugh OnComm and (long nD, Dispatch* pXMLSpy)
 USES CONVERSION;
 if(nD = 1)  {
  ApplicationPtripSpyApp;
  if (pXM LSpy) {
    if (SUCCEEDED (pXM LSpy->QueryInterface ( uuidof (Application), (void **)& pSpyApp))) {
     Docum entPtr ipDocPtr = ipSpyApp->ActiveDocum ent;
     // we assume that grid view is active
     if(ioDocPtr) {
      IG ridV iew Ptr ipG ridPtr = ipD ocPtr->G ridV iew;
      if (inGridPtr) {
        IXM LDataPtr ipXM LData = ipG ridPtr->CurrentFocus;
        CString strValue = W2T (ipXMLData->TextValue);
        if (stValue JSEm pty () \& \& (stValue Left(7) = T ("http://")))
          ShelExecute (NULL, T ("open") W 2T (pXMLData->TextValue) NULLNULLSW SHOWNORMAL);
     }
    }
  }
 }
 return SOK;
HRESULT CXMLSpyDEPlugh DnUpdateComm and (long nD, Dispatch* pXMLSpy, SPYUpdateAction*
pAction)
 *pAction = spyD isable;
 if(n\mathbb{D} = 1) 
  ApplicationPtripSpyApp;
  if (pXM LSpy) {
    if (SUCCEEDED (pXM LSpy->QueryInterface ( uuidof (Application), (void **)& pSpyApp))) {
     Docum entPtr ipDocPtr = ipSpyApp->ActiveDocum ent;
     // only enable if grid view is active
     if((ipDocPtr ⊨ NULL) & & (ipDocPtr->CumentViewMode == spyViewGrid))
       *pAction = spyEnable;
    }
  }
```

```
return SOK;
}
HRESULT CXMLSpyDEPlugin OnEvent(long nEventD, SAFEARRAY **anayParam eters, D ispatch*
pXMLSpy, VAR ANT* pReturnValue)
 return SOK;
}
HRESULT CXM LSpy DEPlugIn SetU M odifications (BSTR * pM odificationsXM L)
{
 CComBSTR bstm Mods = T (" \setminus
      <C onfigurationData>
       <M odifications> ");
 // add "Open URL..." to Tools menu
 bstmM ods Append (T (" \setminus
         <Action>Add</Action> \
          < \mathbb{D} > 1 < / \mathbb{D} > \setminus
           <Name>OpenURL...</Name> \
            <Place>0</Place> \
           \leq M enu \mathbb{D} > 129 < /M enu \mathbb{D} > \setminus
           <Parent>Tools</Parent> \
          </U E lem ent> \
         </M odification> "));
 // add Seperator to Tools menu
 bstmM ods Append (T (" \
         < M \text{ odification} > \setminus
          <Action>Add</Action> \
          <U \mathbbm{E} lem ent type=\"M enultem \"> \
            < \mathbb{D} > 0 < / \mathbb{D} > \setminus
           <Place>1</Place> \
           \leq menu D > 129 < /Menu D > \
           <Parent>Tools</Parent> \
          \langle U \mathbb{E} \text{lement} \rangle
        </Modification>"));
 // finish modification description
 bstmM ods Append (T (" \
       </M odifications> \
      </ConfigurationData>"));
 metum bstm ods C opyTo (pM odificationsXM L);
}
HRESULT CXMLSpyIDEPlugIn:GetDescription (BSTR* pDescription)
{
 CComBSTR bstDescr= T ("ATL C++ XM LSpy DE Plugin fis Plugin demonstrates the implementation of a
simple ATL DLL as a DE PlugIn for XMLSpy!');
 return bstrDescrCopyTo (pDescription);
}
```

2.5 IXMLSpyPlugIn

See also

Methods

OnCommand OnUpdateCommand OnEvent GetUModifications GetDescription

Description

If a DLL is added to Authentic Desktop as an IDE plug- in, it is necessary that it registers a COM component that answers to an IXMLSpyPlugIn interface with the reserved uuid(88F2A622-4B7E-42CD-8D04-3C0E5389DD85), for it to be recognized as a plug- in.

2.5.1 OnCommand

See also

Declaration: OnComm and (n D as long, pXM LSpy as Dispatch)

Description

The OnCommand method of the interface implementation, is called each time a command added by the the IDE plug-in (menu item or toolbar button) is processed. nID stores the command ID defined by the D element of the respective UElement

pXMLSpy holds a reference to the dispatch interface of the Application object of Authentic Desktop.

Example

```
Public Sub KM LSpyPlugIn OnCommand (ByValnD As Long, ByValpXM LSpy As Object)
   If (Not (pXMLSpy Is Nothing)) Then
      Dim objDlg
      Dim objoc As XM LSpyLib Docum ent
      Dim objSpy As XM LSpyLib Application
      Set ob Spy = pXM LSpy
      fn\mathbb{D} = 3 \text{ Orn}\mathbb{D} = 5 \text{ Then}
         objDgFilter = "XML Files (* xm l) |* xm lAll Files (* *) |* * ||"
         ob^{-}D by FilterIndex = 1
         objD lg Show Open
         If Len (ob - D \ln F = N am e) > 0 Then
            Set objOcc = objSpyDocumentsOpenFile(objDgFileName,False)
            Set ob Doc = Nothing
         End Tf
      End If
      \mathbb{I} \mathbb{E} n \mathbb{D} = 4 \text{ Orn} \mathbb{D} = 6 \text{ Then}
         Set ob D lg = C reateObject ("M SC om D lg C om m onD jabg")
         obDgFilter = "AllFiles (**) |** ||"
         objD lg F lags = cdlD FN PathM ustExist
         ob D b Show Save
         If Len (ob D by FileN am e) > 0 Then
            Set objDoc = objSpyActiveDocument
            If Not (ob Doc Is Nothing) Then
               objDocSetPathName objDlgFileName
               ob Doc Save
               Set ob Doc = Nothing
            End If
         End If
      End If
      Set objSpy = Nothing
  End If
End Sub
```

2.5.2 OnUpdateCommand

See also

Declaration: OnUpdateComm and (nD as long, pXM LSpy as Dispatch) as SPYUpdateAction

Description

The <code>OnUpdateComm</code> and <code>()</code> method is called each time the visible state of a button or menu item needs to be set. nD stores the command ID defined by the D element of the respective UElement.

pXMLSpy holds a reference to the dispatch interface of the Application object.

Possible return values to set the update state are:

spyEnable = 1 spyDisable = 2 spyCheck = 4 spyUncheck = 8

Example

```
Public Function XMLSpyPlugIn_OnUpdateComm and (ByValnD As Long, ByValpXMLSpy As Object) As
SPYUpdateAction
XMLSpyPlugIn_OnUpdateComm and = spyDisable
```

```
If (Not (pXM LSpy Is Nothing)) Then
D in objSpy As XM LSpyLib Application
Set objSpy = pXM LSpy
If nD = 3 OrnD = 5 Then
```

```
XM LSpyPlugh_OnUpdateComm and = spyEnable
End If
If nD = 4 Or nD = 6 Then
If ob'SpyDocumentsCount > 0 Then
XM LSpyPlugh_OnUpdateComm and = spyEnable
Else
XM LSpyPlugh_OnUpdateComm and = spyDisable
End If
End If
End If
End If
End If
```

2.5.3 OnEvent

See also

Declaration: OnEvent(*nEvent*) as long, arrayParameters as SAFEARRAY (VARANT), pXMLSpy as Dispatch) as VARIANT

Description

OnEvent() is called each time an event is raised from Authentic Desktop.

Possible values for nEventD are:

On_BeforeStartEditing	= 1
On_EditingFinished	= 2
On_FocusChanged	= 3
On_Beforedrag	= 4
On_BeforeDrop	= 5
On_OpenProject	= 6
On_OpenDocument	= 7
On_CloseDocument	= 8
On_SaveDocument	= 9

Events available since Authentic Desktop 4r4:

On_DocEditDragOver	= 10
On_DocEditDrop	= 11
On_DocEditKeyDown	= 12
On_DocEditKeyUp	= 13
On_DocEditKeyPressed	= 14
On_DocEditMouseMove	= 15
On_DocEditButtonUp	= 16
On_DocEditButtonDown	= 17
On_DocEditContextMenu	= 18
On_DocEditPaste	= 19
On_DocEditCut	= 20
On_DocEditCopy	= 21
On_DocEditClear	= 22
On_DocEditSelectionChanged	= 23

Events available since Authentic Desktop 2004:

On_DocEditDragOver	= 10
--------------------	------

Events available since Authentic Desktop 2004r4 (type library version 1.4):

On_BeforeOpenProject	= 25
On_BeforeOpenDocument	= 26
On_BeforeSaveDocument	= 27
On_BeforeCloseDocument	= 28
On_ViewActivation	= 29
On_DocEditKeyboardEvent	= 30
On_DocEditMouseEvent	= 31

Events available since Authentic Desktop 2006 SP1 (type library version 1.5):

On BeforeValidate = 32

Events available since Authentic Desktop 2007 (type library version 1.6):

On_BeforeShowSuggestions	= 33
On_ProjectOpened	= 34
On Char	= 35

Events available since Authentic Desktop 2009 (type library version 2.2):

On_Initialize	= 36
On Running	= 37
On_Shutdown	= 38

Events available since Authentic Desktop 2012 (type library version 2.8):

On_AuthenticBeforeSave	= 39
On_AuthenticContextMenuActivated	= 40
On AuthenticLoad	= 41
On_AuthenticToolbarButtonClicked	= 42
On_AuthenticToolbarButtonExecuted	= 43
On_AuthenticUserAddedXMLNode	= 44

The names of the events are the same as they appear in the Scripting Environment of Authentic Desktop. For IDE plug-ins the names used are immaterial. The events are identified using the \mathbb{D} value.

arrayParameters is an array which is filled with the parameters of the currently raised event. Order, type and meaning of the single parameters are available through the scripting environment of Authentic Desktop. The events module of a scripting project, contains predefined functions for all events prior to version 4.4. The parameters passed to the predefined functions are identical to the array elements of the arrayParameters parameter.

Events raised from the Authentic View of Authentic Desktop do not pass any parameters directly. An "event" object is used instead. The event object can be accessed through the Document object of the active document.

pXM LS py holds a reference to the dispatch interface of the Application object of Authentic Desktop.

If the return value of OnEvent() is set, then neither the IDE plug-in, nor an event handler inside of the scripting environment will get this event afterwards. Please note that all IDE plug-ins get/ process the event before the Scripting Environment does.

2.5.4 GetUIModifications

See also

Declaration: GetUM odifications () as String

Description

The GetUM odifications () method is called during initialization of the plug-in, to get the configuration XML data that defines the changes to the UI of Authentic Desktop. The method is called when the plug-in is loaded for the first time, and at every start of Authentic Desktop.

See also Configuration XML for a detailed description how to change the UI.

Example

```
Public Function IXM LSpyPlugIn GetU Modifications () As String
   'GetU Modifications () 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 = AppPath
   If Len (strPath) > 0 Then
      Dim fso As New FileSystemObject
      Dim file As file
      Set file = fsoGetFile (strPath & "\config xm l")
      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
         strM ods = stream ReadAll
         stm ods = Replace (stm ods, "**path**", stm ath)
         KMLSpyPlugIn GetU Modifications = strM ods
      Else
         XM LSpyP lugIn_GetU M odifications = ""
      End If
  End If
End Function
```

2.5.5 Get Description

See also

Declaration: GetDescription () as String

Description

GetDescription () is used to define the description string for the plug-in entries visible in the Customize dialog box.

Example

```
Public Function IXMLSpyPlugIn GetDescription () As String
IXMLSpyPlugIn GetDescription = "Sample Plug-in for XMLSpyFhis Plug-in demonstrates the
implementation of a simple V isualBasic DLL as a Plug-in for XMLSpy."
End Function
```

3 Application API

The COM-based API of Authentic Desktop (also called the Application API from now on) enables other applications to use the functionality of Authentic Desktop. As a result, it is possible to automate a wide range of tasks, from validating an XML file to modifying complex XML content (with the <u>XMLData</u> interface).

Authentic Desktop and its Application API follow the common specifications for automation servers set out by Microsoft. It is possible to access the methods and properties of the Application API from common development environments, such as those using C, C++, VisualBasic, and Delphi, and with scripting languages like JScript and VBScript.

Execution environments for the Application API

The Application API can be accessed from the following execution environments:

- External programs (described <u>below</u> and in the <u>Overview</u> part of this section)
- From within the built-in Scripting Editor of Authentic Desktop. For a description of the scripting environment, see the section, <u>Scripting Editor</u>.
- Authentic Desktop allows you to create and integrate your own plug- ins into the application using a special interface for plug- ins. A description of how to create plug- ins is given in the section IDE Plug- ins.
- Via an ActiveX Control, which is available if the <u>integration package</u> is installed. For more information, see the section ActiveX Integration.

External programs

In the <u>Overview</u> part of this section, we describe how the functionality of Authentic Desktop can be accessed and automated from external programs.

Using the Application API from outside Authentic Desktop requires an instance of Authentic Desktop to be started first. How this is done depends on the programming language used. See the section, Programming Languages, for information about individual languages.

Essentially, Authentic Desktop will be started via its COM registration. Then the Application object associated with the Authentic Desktop instance is returned. Depending on the COM settings, an object associated with an already running Authentic Desktop can be returned. Any programming language that supports creation and invocation of COM objects can be used. The most common of these are listed below.

- JScript and <u>VBScript</u> script files have a simple syntax and are designed to access COM objects. They can be run directly from a DOS command line or with a double click on Windows Explorer. They are best used for simple automation tasks.
- <u>C#</u> is a a full-fledged programming language that has a wide range of existing functionality. Access to COM objects can be automatically wrapped using C#..
- C++ provides direct control over COM access but requires relatively larger amounts of code than the other languages.
- <u>Java</u>: Altova products come with native Java classes that wrap the Application API and provide a full Java look-and-feel.
- Other programming languages that make useful alternatives are: Visual Basic for Applications, Perl, and Python.

Programming points

The following limitations must be considered in your client code:

- Be aware that if your client code crashes, instances of Authentic Desktop may still remain in the system.
- Don't hold references to objects in memory longer than you need them, especially those from the XM LData interface. If the user interacts between two calls of your client, then there is no guarantee that these references are still valid.
- Don't forget to disable dialogs if the user interface is not visible.
- See <u>Error handling in JScript</u> (and in <u>C#</u> and <u>Java</u>) for details of how to avoid annoying error messages.
- Free references explicitly if you are using C or C++.

This documentation

This documentation section about the Application API is broadly divided into two parts.

- The first part consists of an <u>Overview</u>, which describes the object model for the API and explains how the API is accessed via various <u>programming languages</u>.
- The second part is a reference section (<u>Interfaces</u> and <u>Enumerations</u>) that contains descriptions of the interface objects of the Application API.

3.1 Overview

This overview of the Application API is organized as follows:

- The Object Model describes the relationships between the objects of the Application API.
- <u>Programming Languages</u> explains how the most commonly used programming languages (JScript, VBScript, C#, and Java) can be used to access the functionality of the Application API. Code listings from the example files supplied with your application package are used to describe basic mechanisms.

3.1.1 Object Model

The starting point for every application which uses the Application API is the <u>Application</u> object. This object contains general methods like import/export support and references to the open documents and any open project.

The Application object is created differently in various programming languages. In scripting languages such as JScript or VBScript, this involves calling a function which initializes the application's COM object. For examples, see the Programming Languages section.

XMLSpy.Application or AuthenticDesktop.Application

Authentic Desktop installs a TypeLibrary containing the XMLSpyLib. If this TypeLibrary has been added to the development environment (VB development environment, for example) then an object of the Application type can be created with:

Set objSpy = New XMLSpyLib.Application

If only Authentic Desktop is installed (and not XMLSpy), then

```
Set objSpy = GetObject("", "XMLSpy.Application")
```

does not work, because there won't be any object registered in the Registry with a ProgID of XMLSpy.Application. In this case, the registered object is AuthenticDesktop.Application.

The code listings in this documentation assume that both Authentic Desktop and XMLSpy have been installed. If, however, only Authentic Desktop has been installed, then please modify code fragments to take account of this difference.

The application object consists of the following parts:

- 1. Document collection and reference to the active document.
- 2. Reference to current project and methods for creating and opening projects.
- 3. Methods to support the export to and import from databases, text files, and Word documents.
- 4. URL management.
- 5. Methods for macro menu items.

Once you have created an Application object you can start using the functionality of Authentic Desktop. In most cases, you either open a project and access the documents from there or you directly open a document via the Documents interface.

3.1.2 Programming Languages

Programming languages differ in the way they support COM access. A few examples for the most frequently used languages (*links below*) will help you get started. The code listings in this section show how basic functionality can be accessed. This basic functionality is included in the files in the API Examples folder and can be tested straight away. The path to the API Examples folder is given below:

Windows 7、Windows 8、	C:\Users\ <username>\Documents\</username>
Windows 10	Altova\Authentic2018\Authentic

JScript

The JScript listings demonstrate the following basic functionality:

- <u>Start application or attach to a running instance</u>
- Simple document access
- <u>Iteration</u>
- Error handling
- <u>Events</u>

VBScript

VBScript is different than JScript only syntactically; otherwise it works in the same way. The listings below describe is an example of how VBScript can be used. For more information, refer to the <u>JScript examples</u>.

• Events: Shows how events are handled using VBScript.

C#

C# can be used to access the Application API functionality. The code listings show how to access the API for certain basic functionality.

- <u>Start Authentic Desktop</u>: Starts Authentic Desktop, which is registered as an automation server, or activates the application if it is already running.
- <u>Open OrgChart.pxf</u>: Locates one of the example documents installed with Authentic Desktop and opens it. If this document is already open it becomes the active document.
- <u>OnDocumentOpened Event On/Off</u>: Shows how to listen to Authentic Desktop events. When turned on, a message box will pop up after a document has been opened.
- Open ExpReport.xml: Opens another example document.
- <u>Toggle View Mode</u>: Changes the view of all open documents between Browser View and Authentic View. The code shows how to iterate through open documents.
- <u>Validate</u>: Validates the active document and shows the result in a message box. The code shows how to handle errors and COM output parameters.
- <u>Shutdown Authentic Desktop</u>: Stops Authentic Desktop.

Java

The Authentic Desktop API can be accessed from Java code. <u>The Java sub-section of this section</u> explains how some basic Authentic Desktop functionality can be accessed from Java code. It is organized into the following sub-sections:

- Mapping Rules for the Java Wrapper
- Example Java Project
- Application Startup and Shutdown
- Simple Document Access
- Iterations
- Use of Out-Parameters
- Event Handlers

JScript

This section contains listings of JScript code that demonstrate the following basic functionality:

- <u>Start application or attach to a running instance</u>
- Simple document access
- <u>Iteration</u>
- Error handling
- Events

Example files

The code listings in this section are available in example files that you can test as is or modify to suit your needs. The JScript example files are located in the <code>JScript</code> folder of the API Examples folder:

Windows 7、Windows 8、	C:\Users\ <username>\Documents\</username>
Windows 10	Altova\Authentic2018\Authentic

The example files can be run in one of two ways:

- *From the command line:* Open a command prompt window, change the directory to the path above, and type the name of one of the example scripts (for example, Start.js).
- From Windows Explorer: In Windows Explorer, browse for the JScript file and double-click it.

The script is executed by Windows Script Host that is packaged with Windows operating system. For more information about Windows Script Host, refer to MSDN documentation (<u>https://msdn.microsoft.com</u>).

Start Application

The JScript below starts the application and shuts it down. If the COM object of the 32-bit Authentic Desktop cannot be found, the code attempts to get the COM object of the 64-bit application; otherwise, an error is thrown. If an instance of the application is already running, the running instance will be returned.

Note: For 32-bit Authentic Desktop, the registered name, or programmatic identifier (Progld) of the COM object is AuthenticDesktop.Application. For 64-bit Authentic Desktop, the name is AuthenticDesktop x64.Application.

```
// Initialize application's COM object. This will start a new instance of the
application and
// return its main COM object. Depending on COM settings, the main COM object
```

```
of an already
// running application might be returned.
try { objAuthentic = WScript.GetObject("",
"AuthenticDesktop.Application"); }
catch(err) {}
if( typeof( objAuthentic ) == "undefined" )
  try { objAuthentic = WScript.GetObject("",
"AuthenticDesktop x64.Application") }
  catch(err)
  {
     WScript.Echo( "Can't access or create AuthenticDesktop.Application" );
     WScript.Quit();
  }
}
// if newly started, the application will start without its UI visible. Set it
to visible.
objAuthentic.Visible = true;
WScript.Echo(objAuthentic.Edition + " has successfully started. ");
objAuthentic.Visible = false; // will shutdown application if it has no more
COM connections
//objAuthentic.Visible = true; // will keep application running with UI
visible
```

The JScript code listed above is available in the sample file Start.js (see Example Files).

Simple Document Access

After you have started the application as shown in <u>Start Application</u>, you will most likely want to programmatically open a document in order to work with it. The JScript code listing below illustrates how to open two documents from the Authentic Desktop Examples folder and set one of them as the active document.

```
// Locate examples via USERPROFILE shell variable. The path needs to be
adapted to major release versions.
objWshShell = WScript.CreateObject("WScript.Shell");
majorVersionYear = objAuthentic.MajorVersion + 1998
strExampleFolder = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\
\My Documents\\Altova\\Authentic" + majorVersionYear + "\\AuthenticExamples\
\";
// Tell Authentic to open two documents. No dialogs
objDoc1 = objAuthentic.Documents.OpenFile(strExampleFolder + "OrgChart.pxf",
false);
objAuthentic.Documents.OpenFile(strExampleFolder + "ExpReport.xml", false);
// The document currently active can be easily located.
objDoc2 = objAuthentic.ActiveDocument;
// Let us make sure that the document is shown in Authentic view.
objDoc2.SwitchViewMode(5); // SPYViewModes.spyViewAuthentic = 5
```

```
// Now switch back to the document opened first
objDoc1.SetActiveDocument();
```

The JScript code listed above is available in the sample file DocumentAccess.js (see Example Files).

Iteration

The JScript listing below shows how to iterate through the open documents. It is assumed that you have already started the application and opened some documents as shown in the previous sections.

```
// go through all open documents using a JScript Enumerator
bRequiresSaving = false;
for (var iterDocs = new Enumerator(objAuthentic.Documents); !iterDocs.atEnd();
iterDocs.moveNext())
   if (iterDocs.item().IsModified)
      bRequiresSaving = true;
   var strErrorText = new Array(1);
   var nErrorNumber = new Array(1);
   var errorData = new Array(1);
   if (!iterDocs.item().IsValid(strErrorText, nErrorNumber, errorData))
      var text = strErrorText;
      // access that XMLData object only if filled in
      if (errorData[0] != null)
         text += "(" + errorData[0].Name + "/" + errorData[0].TextValue + ")";
     WScript.Echo("Document \"" + iterDocs.item().Name +"\" validation
error[" + nErrorNumber + "]: " + text);
  }
  else
      // The COM call succeeded and the document is valid.
     WScript.Echo("Document \"" + iterDocs.item().Name + "\" is valid.");
   }
}
// go through all open documents using index-based access to the document
collection
for (i = objAuthentic.Documents.Count; i > 0; i--)
   objAuthentic.Documents.Item(i).Close(false);
```

The JScript code listed above is available in the sample file DocumentAccess.js (see Example Files).

Error Handling

The Application API returns errors in two different ways:

- The HRESULT returned by every API method
- The IErrorInfo interface of the Application API

Every API method returns an <code>HRESULT</code>. This return value gives the caller information about errors during execution of the method. If the call was successful, the return value is <code>s_ok</code>. The <code>HRESULT</code> option is commonly used in C/C++ programs.

However, programming languages such as VisualBasic and scripting languages (and other highlevel development environments) don't give the programmer access to the HRESULT return of a COM call. Such languages use the IErrorInfo interface, which is also supported by the Application API. If an error occurs, the Application API creates a new object that implements the IErrorInfo interface. The information provided by the IErrorInfo interface is imported by the development environment into its own error-handling mechanism.

For example, the JScript code listing below causes an error to be thrown by incorrectly declaring an array. Additional information about the error object is provided by its properties number and description.

```
try {
    var arr = new Array(-1);
}
catch (err) {
    WScript.Echo("Error : (" + (err.number & 0xfff) + ")" + err.description);
}
```

Events

COM specifies that a client must register itself at a server for callbacks using the connection point mechanism. The automation interface for XMLSpy defines the necessary event interfaces. The way to connect to those events depends on the programming language you use in your client. The following code listing shows how this is done using JScript.

The method WScript.ConnectObject is used to receive events.

```
// The event-handler function
function DocEvent OnBeforeCloseDocument(objDocument)
{
   WScript.Echo("Received event - before closing document");
}
// Create or connect to XMLSpy (or Authentic Desktop)
try
{
  // Create the environment and XMLSpy (or Authentic Desktop)
  objWshShell = WScript.CreateObject("WScript.Shell");
  objFSO = WScript.CreateObject("Scripting.FileSystemObject");
  objSpy = WScript.GetObject("", "XMLSpy.Application");
// If only Authentic Desktop is installed (and XMLSpy is not installed) use:
// objSpy = WScript.GetObject("", "AuthenticDesktop.Application")
}
catch(err)
   { WScript.Echo ("Can't create WScript.Shell object or XMLSpy"); }
// Create document object and connect to its events
objSpy.Visible = true;
```

```
majorVersionYear = objSpy.MajorVersion + 1998
docPath = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\\Documents
\\Altova\\XMLSpy" + majorVersionYear + "\\Examples\\ExpReport.xml";
objDoc = objSpy.Documents.OpenFile (docPath, false);
WScript.ConnectObject(objDoc, "DocEvent_");
// Keep running while waiting for the event
// In the meanwhile close this document in XMLSpy (or Authentic Desktop)
manually
WScript.Echo ("Sleeping for 10 seconds ...");
WScript.Echo ("Stopped listening for event");
objDoc = null;
WScript.Echo ("Stopped listening for event");
```

VBScript

VBScript is syntactically different than JScript but works in the same way. This section contains a listing showing how events are used with VBScript and an example.

For information about other functionality, refer to the JScript examples listed below:

- Start application or attach to a running instance
- Simple document access
- Iteration
- Error handling

Events

COM specifies that a client must register itself at a server for callbacks using the connection point mechanism. The automation interface for XMLSpy defines the necessary event interfaces. The way to connect to those events depends on the programming language you use in your client. The following code listing shows how this is done using VBScript.

The method WScript.ConnectObject is used to receive events.

To run this code, paste it into a file with .vbs extension, and either double-click in Windows Explorer, or run it from a command prompt.

```
' the event handler function
Function DocEvent_OnBeforeCloseDocument(objDocument)
    Call WScript.Echo("received event - before closing document")
End Function
' create or connect to XmlSpy
Set objWshShell = WScript.CreateObject("WScript.Shell")
Set objFSO = WScript.CreateObject("Scripting.FileSystemObject")
Set objSpy = WScript.GetObject(", "XMLSpy.Application")
' If only Authentic is installed (and XMLSpy is not installed) use:
' Set objSpy = WScript.GetObject(", "AuthenticDesktop.Application")
' If only XMLSpy 64-bit is intalled, use:
' Set objSpy = WScript.GetObject(", "XMLSpy_x64.Application")
' create document object and connect to its events
```

```
objSpy.Visible = True
```

```
' Find out user's personal folder and locate one of the installed examples.
personalFolder = objWshShell.ExpandEnvironmentStrings("%UserProfile%")
majorVersionYear = objSpy.MajorVersion + 1998
xmlspyExamplesFolder = personalFolder & "\Documents\Altova\XMLSpy" &
majorVersionYear & "\Examples\"
docPath = xmlspyExamplesFolder & "ExpReport.xml"
' open a document
Set objDoc = objSpy.Documents.OpenFile (docPath, False)
Call WScript.ConnectObject(objDoc, "DocEvent ")
' keep running while waiting on the event
' in the meantime close the document in XMLSPY manually
Call WScript.Echo ("sleeping for 10 seconds ...")
Call WScript.Sleep (10000)
Set objDoc = Nothing
Call WScript.Echo ("stopped listening for event")
Call objSpy.Quit
```

Note: For 32-bit Authentic Desktop, the registered name, or programmatic identifier (Progld) of the COM object is AuthenticDesktop.Application. For 64-bit Authentic Desktop, the name is AuthenticDesktop_x64.Application.

Example: Using Events

Authentic View supports event connection on a per-object basis. Implementation of this feature is based on COM connection points and is available in environments that support this mechanism.

The following example is a VBScript code example that shows how to use events from within a VBScript project.

```
' VBScript example that demonstrates how to use events.
' Event handler for OnSelectionChanged event of AuthenticView
Function AuthenticViewEvent OnSelectionChanged (objAuthenticRange)
   If objAuthenticRange.FirstTextPosition <>
objAuthenticRange.LastTextPosition Then
       Call WScript.Echo("Selection: " & objAuthenticRange.Text & vbNewLine &
vbNewLine & "Close this dialog.")
   Else
       Call WScript.Echo("Cursor position: " &
objAuthenticRange.FirstTextPosition & vbNewLine & vbNewLine & "Close this
dialog.")
   End If
End Function
' Start/access XMLSpy and connect to its automation interface.
Set WshShell = WScript.CreateObject("WScript.Shell")
Set objSpy = GetObject("", "XMLSpy.Application")
' Make the UI of XMLSpy visible.
objSpy.Visible = True
```

```
' Find out user's personal folder and locate one of the installed XMLSpy
examples.
personalFolder = WshShell.ExpandEnvironmentStrings("%UserProfile%")
majorVersionYear = objSpy.MajorVersion + 1998
xmlspyExamplesFolder = personalFolder & "\Documents\Altova\XMLSpy" &
majorVersionYear & "\Examples\"
docPath = xmlspyExamplesFolder & "ExpReport.xml"
' Create object to access windows file system and test if the our document
exists.
Set fso = CreateObject("Scripting.FileSystemObject")
If fso.FileExists(docPath) Then
    ' open the document
   Call objSpy.Documents.OpenFile(docPath, False)
   set objDoc = objSpy.ActiveDocument
    ' switch active document to authentic view
   objDoc.SwitchViewMode 4 ' spyViewAuthentic
    ' Register for connection point events on the authentic view of the active
document.
    ' Any function with a valid event name prefixed with "AuthenticViewEvent "
will
    ' be called when the corresponding event gets triggered on the specified
object.
   set objView = objDoc.AuthenticView
   Call WScript.ConnectObject(objView, "AuthenticViewEvent ")
   Call WScript.Echo("Events are connected." & vbNewLine & vbNewLine & "Now
set or move the cursor in XMLSpy." & vbNewLine & vbNewLine & "Close this
dialog to shut down XMLSpy.")
    ' To disconnect from the events delete the reference to the object.
   set objView = Nothing
Else
   Call WScript.Echo("The file " & docPath & " does not exist.")
End If
' shut down XMLSpy when this script ends
objSpy.Visible = False
```

C#

The C# programming language can be used to access the Application API functionality. You could use Visual Studio 2008 or Visual Studio 2010 to create the C# code, saving it in a Visual Studio project. Create the project as follows:

- 1. In Microsoft Visual Studio, add a new project using File | New | Project.
- Add a reference to the Authentic Desktop Type Library by clicking Project | Add Reference. The Add Reference dialog appears. Browse for the Authentic Desktop Type Library component, which is located in the Authentic Desktop application folder, and add it.
- 3. Enter the code you want.
- 4. Compile the code and run it.

Example C# project

Your Authentic Desktop package contains an example C# project, which is located in the C#

folder of the API Examples folder:

Windows 7、Windows 8、	C:\Users\ <username>\Documents\</username>
Windows 10	Altova\Authentic2018\Authentic

You can compile and run the project from within Visual Studio 2008 or Visual Studio 2010.

The code listing below shows how basic application functionality can be used. This code is similar to the example C# project in the API Examples folder of your application package, but might differ slightly.

Platform configuration

If you have a 64-bit operating system and are using a 32-bit installation of Authentic Desktop, you must add the x86 platform in the solution's Configuration Manager and build the sample using this configuration.

A new x86 platform (for the active solution in Visual Studio) can be created in the New Solution Platform dialog (**Build | Configuration Manager | Active solution platform | <New...>**).

What the code listing below does

The example code listing below creates a simple user interface (*screenshot below*) with buttons that invoke basic Authentic Desktop operations:



- <u>Start Authentic Desktop</u>: Starts Authentic Desktop, which is registered as an automation server, or activates the application if it is already running.
- <u>Open OrgChart.pxf</u>: Locates one of the example documents installed with Authentic Desktop and opens it. If this document is already open it becomes the active document.
- Open ExpReport.xml: Opens another example document.
- <u>Toggle View Mode</u>: Changes the view of all open documents between Text View and Authentic View. The code shows how to iterate through open documents.
- <u>Validate</u>: Validates the active document and shows the result in a message box. The code shows how to handle errors and COM output parameters.

• Shut down Authentic Desktop: Stops Authentic Desktop.

You can modify the code (of the code listing below or of the example C# project in the API Examples folder) in any way you like and run it.

Compiling and running the example

In the API Examples folder, double-click the file AutomateAuthenticDesktop_VS2008.sln (to open it in Visual Studio 2008) or the file AutomateAuthenticDesktop_VS2010.sln (to open it in Visual Studio 2010). Alternatively the file can be opened from within Visual Studio (with File | Open | Project/Solution). To compile and run the example, select Debug | Start Debugging or Debug | Start Without Debugging.

Code listing of the example

Given below is the C# code listing of the basic functionality of the form (Form1.cs) created in the AutomateAuthenticDesktop example. Note that the code listed below might differ slightly from the code in the API Examples form. The listing below is commented for ease of understanding. Parts of the code are also presented separately in the sub-sections of this section, according to the Application API functionality they access.

The code essentially consists of a series of handlers for the buttons in the user interface shown in the screenshot above.

```
namespace WindowsFormsApplication2
  public partial class Form 1 : Form
    public Form 10
       InitializeComponent();
     // An instance of Authentic Desktop is accessed via its automation interface.
    XMLSpyLbApplication AuthenticDesktop;
     // Location of examples installed with Authentic Desktop
    String strExam plesFolder;
    private void Form 1 Load (object sender, EventArgs e)
       // Locate examples installed with Authentic Desktop.
       // REMARK: You might need to adapt this if you have a different maprices on of the product.
       stExam pbsFober = EnvironmentGetEnvironmentVariable ("USERPROFILE") + "\\MyDocuments\\Albova\
\Authentic2012\\AuthenticExamples\\";
    }
    // Handler for the "Start Authentic Desktop" button
    private void StartAuthenticDesktop Click (bject sender, EventArgs e)
       if (AuthenticDesktop == nul)
       {
         CursorCurrent = Cursors WaltCursor;
          // If there is no AuthenticDesktop instance, create one and make it visible.
         AuthenticDesktop = new XMLSpyLbApplication ();
         AuthenticDesktopVisble = true;
         CursorCurrent = Cursors Defaul;
       ebe
```
```
{
     // If an Authentic Desktop instance is already running, make sure it's visible.
     if (AuthenticDesktopVisble)
       AuthenticDesktopVisble = true;
  }
}
// Handler for the "Open OrgChartpxf" button
private void openO rgChart Click (object sender, EventArgs e)
   // Make sure there's a running Authentic Desktop instance, and that it's visible
  StartAuthenticDesktopClick(nul,nul);
   // Open a sample file installed with the product.
  AuthenticDesktopDocumentsOpenFie (strExamplesFolder + "OrgChartpxf", fabe);
}
// Handler for the "Open ExpReport xm l' button
private void openExpReport Click (bject sender, EventArgs e)
   // Make sure there's a running Authentic Desktop instance, and that it's visible
  StartAuthenticDesktopClick(nul,nul);
   // Open a sample file installed with the product.
  AuthenticDesktopDocumentsOpenFie (strExamplesFolder + "ExpReportxm I", fabe);
}
// Handler for the "Toggle View Mode" button
private void toggeView_Click(object sender, EventArgs e)
   // Make sure there's a running Authentic Desktop instance, and that it's visible
  StartAuthenticDesktopClick(hul,nul);
  // Iterate through allopen documents and toggle the current view between TextView and Authentic View.
  foreach (MLSpyLbDocument doc in AuthenticDesktopDocuments)
     if doc CurrentV iewMode == XMLSpyLbSPYV iewModes spyV iewAuthentic)
       doc SwitchViewMode (XMLSpyLibSPYViewModesspyViewBrowser);
    eke
       docSwitchViewMode &MLSpyLbSPYViewModesspyViewAuthentic);
}
// Handler for the "Shut down Authentic Desktop" button
// Shut down application instance by explicitely releasing the COM object.
private void shutdownAuthenticDesktopClick(bject sender, EventArgs e)
  if AuthenticDesktop ⊨ nul)
  {
     // Albw shut down of Authentic Desktop by releasing the UI
    AuthenticDesktopVisble = false;
     // Explicitly release the COM object
    try
     {
       while $ystem Runtime InteropServicesMarshalReleaseComObjectAuthenticDesktop)>0);
     finally
     {
        // Avoid subsequent access to this object.
       AuthenticDesktop = nul;
    }
  }
}
// Handler for the "Validate" button
private void validate_Click (bject sender, EventArgs e)
  // COM errors get returned to C# as exceptions.Use a try/catch block to handle them .
  try
  {
     // Method 'IsValid' is one of the few functions that use output parameters.
     // Use bbject'type for these parameters.
    object strEmorText = "";
    object nErrorNum ber = 0;
    object errorData = null;
```

```
f (AuthenticDesktopActiveDocumentIsVald (refstrEnorText, refnEmorNumber, referrorData))
            // The COM callsucceeds but the document is not valid.
            // A detailed description of the problem is returned in strEncorText, nErrorNumber and errorData.
            listBoxMessages_Item sAdd("Document" + AuthenticDesktopActiveDocumentName + " is not valid.");
            lstBoxMessagesItemsAdd("\tEmorText :" + strEmorText);
            listBoxMessages.ltem sAdd ("\tErrorNumber:" + nErrorNumber);
            lstBoxMessagesItemsAdd("\tElement :" + (enorData = null? (KMLSpyLbXMLData)enorData)TextValue
:"null));
         ebe
         {
            // The COM callsucceeds and the document is valid.
            listBoxMessagesItemsAdd("Document" + AuthenticDesktopActiveDocumentName + " is valid.");
         }
       }
       catch (Exception ex)
       {
          // The COM callwas not successful.
          // Probably no application instance has been started or no document is open.
         listBoxMessages.ltem sAdd ("Enorvalidating active document:" + exMessage);
       }
     }
     de legate void addListBoxItem de legate (string sText);
     // Called from the UIthread
     private void addListBoxItem (string sText)
       listBoxMessages.ltemsAdd(sText);
     //Wrappermethod to callUI controlmethods from a worker thread
     void syncWithUIthread Controlctrl, addListBoxItem delegate methodToInvoke,String sText)
     {
       // ControlInvoke:Executes on the UI thread, but calling thread waits for completion before continuing.
       // ControlBeginInvoke: Executes on the UI thread, and calling thread doesn't wait for completion.
       if (ctrl.InvokeRequired)
         ctrlBeginInvoke (methodToInvoke, new Object[] { sText });
     }
     // Eventhandler for On DocumentOpened event
    private void handleOnDocumentOpened (KMLSpyLbDocument ipDocument)
       String sText = "";
       if (ipDocumentNameLength > 0)
         sText = "Document" + ipDocumentName + " was opened!";
       ebe
         sText = "An empty docum entwas created.";
       // Synchronize the calling thread with the UI thread because
       // COM events are triggered from a working thread
       addListBoxItem delegate methodToInvoke = new addListBoxItem delegate (addListBoxItem);
       // CallsyncW thU Thread with the following arguments:
       // 1 - listBoxMessages - list box control to display messages from COM events
       // 2 - methodToInvoke - a C# delegate which points to the method which will be called from the UI thread
       // 3 - sText
                         - the text to be displayed in the list box
       syncWithUIthread (istBoxMessages,methodToInvoke,sText);
     }
    private void checkBoxEventOnOffCheckedChanged(bject sender, EventArgs e)
       if AuthenticDesktop ⊨ nul)
          f (checkBoxEventOnOffChecked)
            AuthenticDesktopOnDocumentOpened += new
XMLSpyLb. ApplicationEvents OnDocumentOpenedEventHandlerhandleOnDocumentOpened);
         ebe
            AuthenticDesktopOnDocumentOpened -= new
XMLSpyLb_ApplicationEventsOnDocumentOpenedEventHandlerhandleOnDocumentOpened);
       }
  }
```

Add Reference to Authentic Desktop API

Add the application's type library as a reference in a .NET project as follows: With the .NET project open, click **Project | Add Reference**. Then browse for the type library, which is called Authentic Desktop.tlb, and is located in the Authentic Desktop application folder.

Then declare a variable to access the Authentic Desktop API:

```
// An instance ofAuthentic Desktop is accessed via its automation interface.
XMLSpyLbApplication Authentic Desktop;
```

Application Startup and Shutdown

In the code snippets below, the methods StartAuthenticDesktop_Click and ShutdownAuthenticDesktop_Click are those assigned to buttons in the AutomateAuthenticDesktop example that, respectively, start up and shut down the application. This example is located in the C# folder of the API Examples folder (see the file Form1.cs):

Windows 7、Windows 8、	C:\Users\ <username>\Documents\</username>
Windows 10	Altova\Authentic2018\Authentic

You can compile and run the project from within Visual Studio 2008 or Visual Studio 2010.

Starting Authentic Desktop

The following code snippet from the <u>AutomateAuthenticDesktop example</u> shows how to start up the application.

```
// Handler for the "Start AuthenticDesktop" button
private void StartAuthenticDesktop_Click (bject sender, EventArgs e)
{
    f AuthenticDesktop == null)
    {
        CursorCurrent = Cursors WaltCursor;
        // If there is no AuthenticDesktop instance, create one and make ivisble.
        AuthenticDesktop = new XMLSpyLbApplication ();
        AuthenticDesktopVisble = true;
        CursorCurrent = Cursors Defaul;
    }
    else
    {
        // If an instance of Authentic Desktop is already running, make sure its visble
        AuthenticDesktopVisble = true;
    }
}
```

Shutting down Authentic Desktop

The following code snippet from the <u>AutomateAuthenticDesktop example</u> shows how to shut down the application.

```
// Handler for the "Shut down AuthenticDesktop" button
// Shut down application instance by explicitely releasing the COM object.
private void shutdownAuthenticDesktop_Click(bject sender, EventArgs e)
```

```
{
    ff AuthenticDesktop = null)
    {
        // Albw shut down of AuthenticDesktop by melasing the UI
        AuthenticDesktopVisble = false;
        // Explicitly melase the COM object
        try
        {
            while & ystem Runtime InteropServicesMarshalReleaseComObjectAuthenticDesktop) > 0);
        }
        finally
        {
            // Avoid subsequent access to this object.
            AuthenticDesktop = null;
        }
    }
}
```

Opening Documents

The code snippets below (from the <u>AutomateAuthenticDesktop example</u>) show how two files are opened via two separate methods assigned to two buttons in the user interface. Both methods use the same Application API access mechanism: <u>DocumentsOpenFile(string, boolean)</u>.

The <u>AutomateAuthenticDesktop example</u> (see the file Form1.cs) is located in the C# folder of the API Examples folder:

Windows 7、Windows 8、	C:\Users\ <username>\Documents\</username>
Windows 10	Altova\Authentic2018\Authentic

You can compile and run the project from within Visual Studio 2008 or Visual Studio 2010.

Code snippet

```
// Handler for the "Open OrgChartpxf" button
private voil openOrgChart_Click(bject sender, EventArgs e)
{
    // Make sure there is a running Authentic Desktop instance, and that its visible
    StartAuthenticDesktop_Click(nul,nul);
    // Open a sample file installed with the product.
    AuthenticDesktopDocumentsOpenFile(strExamplesFolder + "OrgChartpxf", false);
}
// Handler for the "Open ExpReport_m I' button
private voil openExpReport_Click(bject sender, EventArgs e)
{
    // Make sure there is a running Authentic Desktop instance, and that its visible
    StartAuthenticDesktop_Click(nul,nul);
    // Open a sample file installed with the product.
    AuthenticDesktop_Click(nul,nul);
    // Open a sample file installed with the product.
    AuthenticDesktop_Click(nul,nul);
    // Open a sample file installed with the product.
    AuthenticDesktopDocumentsOpenFile(strExamplesFolder + "ExpReport_xml", false);
}
```

The file opened last will be the active file.

Iterating through Open Documents

The code snippet below (from the <u>AutomateAuthenticDesktop example</u>; see the file Form1.cs) shows how to iterate through open documents. A condition is then tested within the iteration loop,

and the document view is switched between Text View and Authentic View.

```
// Handler for the "Toggle V isw Mode" button
private voil toggleV isw_Click(bbject sender, EventArgs e)
{
    // Make sure there is a nunning Authentic Desktop instance, and that it's visible
    StartAuthenticDesktop_Click(null,null);

    // Iterate through allopen documents and toggle the current view between TextView and Authentic View.
    foreach (KMLSpyLbDocument doc in AuthenticDesktopDocuments)
    if (boc CurrentViewMode == XMLSpyLbSPYViewModes.spyViewBrowser);
    else
        doc SwitchViewMode (KMLSpyLbSPYViewModes.spyViewAuthentic);
}
```

The <u>AutomateAuthenticDesktop example</u> example is located in the C# folder of the API Examples folder:

Windows 7、Windows 8、	C:\Users\ <username>\Documents\</username>
Windows 10	Altova\Authentic2018\Authentic

You can compile and run the project from within Visual Studio 2008 or Visual Studio 2010.

Errors and COM Output Parameters

The code snippet below (from the <u>AutomateAuthenticDesktop example</u>) shows how to handle errors and COM output parameters. The method <u>AuthenticDesktopActiveDocument_SValid (ref</u> <u>stEnorText, refnEnorNumber, referrorData</u>) uses output parameters that are used, in the code snippet below, to generate an error-message text.

The <u>AutomateAuthenticDesktop example</u> (see the file Form1.cs) is located in the C# folder of the API Examples folder:

Windows 7、Windows 8、	C:\Users\ <username>\Documents\</username>
Windows 10	Altova\Authentic2018\Authentic

You can compile and run the project from within Visual Studio 2008 or Visual Studio 2010.

Code snippet

```
// Handler for the "Validate" button
private void validate_Click (bbject sender, EventArgs e)
{
    // COM emors get returned to C # as exceptions.Use a try/catch bbck to handle them.
    try
    {
        // Method "sValid" is one of the few functions that use output parameters.
        // Use bbject"type for these parameters.
        object stdEmorText = "";
        object nEmorNumber = 0;
        object emorData = null;
        if (AuthenticDesktopActiveDocum ent_IsValid (refstdEmorText, refnEmorNumber, referrorData)))
        {
            // The COM callsucceeds but the document is not valid.
            // A detailed description of the problem is returned in stdEmorText, nEmorNumber and emorData.
            // A detailed description of the problem is returned in stdEmorText, nEmorNumber and emorData.
            // A detailed description of the problem is returned in stdEmorText, nEmorNumber and emorData.
            // A detailed description of the problem is returned in stdEmorText, nEmorNumber and emorData.
            // A detailed description of the problem is returned in stdEmorText, nEmorNumber and emorData.
            // A detailed description of the problem is returned in stdEmorText, nEmorNumber and emorData.
            // A detailed description of the problem is returned in stdEmorText, nEmorNumber and emorData.
            // A detailed description of the problem is returned in stdEmorText, nEmorNumber and emorData.
            // A detailed description of the problem is returned in stdEmorText, nEmorNumber and emorData.
            // A detailed description of the problem is returned in stdEmorText, nEmorNumber and emorData.
            // A detailed description of the problem is returned in stdEmorText, nEmorNumber and emorData.
            // A detailed description of the problem is returned in stdEmorText, nEmorNumber and emorData.
            // A detailed description of the problem is returned in stdEmorText, nEmorNumber and emorData.
            // A detailed description of the
```

```
listBoxMessagesItem sAdd("\tErrorText :" + strEnorText);
listBoxMessagesItem sAdd("\tErrorNumber:" + nErrorNumber);
listBoxMessagesItem sAdd("\tElement :" + @rorData \= null? (&MLSpyLbXMLData)errorData)TextValue
:"null"));
}
else
{
// The COM callsucceeds and the document is valid.
listBoxMessagesItem sAdd("Document " + AuthenticDesktopActiveDocumentName + " is valid.");
}
catch @cception ex)
{
// The COM callwas not successful.
// The COM callwas not successful.
// Probably no application instance has been started or no document is open.
listBoxMessagesItem sAdd("Error validating active document:" + exMessage);
}
```

Events

The code snippet below (from the <u>AutomateAuthenticDesktop example</u>) lists the code for two event handlers. The <u>AutomateAuthenticDesktop example</u> (*see the file Form1.cs*) is located in the C# folder of the API Examples folder:

Windows 7、Windows 8、	C:\Users\ <username>\Documents\</username>
Windows 10	Altova\Authentic2018\Authentic

```
You can compile and run the project from within Visual Studio 2008 or Visual Studio 2010.
```

```
delegate void addListBoxItem delegate (string sText);
// Called from the UIthread
private void addListBoxItem (string sText)
  listBoxMessages_ltemsAdd(sText);
//Wrappermethod to callUI controlmethods from a worker thread
void syncW ihu Ihread (Controlctr), addListBoxItem de legate methodTo Invoke, String sText)
{
  // ControlInvoke: Executes on the UI thread, but calling thread waits for completion before continuing.
  // ControlBeginInvoke:Executes on the UI thread, and calling thread doesn twait for completion.
  if (ctrl.InvokeRequired)
    ctrlBeginInvoke (methodToInvoke, new Object[] { sText });
     // Eventhandler for On DocumentOpened event
private void handleOnDocum entOpened (KMLSpyLbDocum ent ipDocum ent)
  String sText = "";
  if (ipDocumentNameLength > 0)
     sText = "Document" + ipDocumentName + " was opened!";
  ebe
    sText = "An empty document was created.";
  // Synchronize the calling thread with the UI thread because
  // COM events are triggered from a working thread
  addListBoxItem delegate methodToInvoke = new addListBoxItem delegate (addListBoxItem);
  // CallsyncW thU Thread with the following arguments:
  // 1 - listBoxMessages - list box control to display messages from COM events
  // 2 - methodToInvoke - a C# delegate which points to the method which willbe called from the UI thread
  // 3 - sText
                     - the text to be displayed in the list box
  syncWithUIthread (istBoxMessages,methodToInvoke,sText);
}
private void checkBoxEventOnOffCheckedChanged(bject sender, EventArgs e)
```

```
if AuthenticDesktop ⊨ nul)
```



Java

The Application API can be accessed from Java code. To allow accessing the Authentic Desktop automation server directly from Java code, the libraries listed below must reside in the classpath. They are installed in the folder: JavaAPI in the Authentic Desktop application folder.

- AltovaAutomation.dll: a JNI wrapper for Altova automation servers (AltovaAutomation_x64.dll in the case of 64-bit versions)
- AltovaAutomation.jar: Java classes to access Altova automation servers
- AuthenticAPI.jar: Java classes that wrap the Authentic Desktop automation interface
- AuthenticAPI_JavaDoc.zip: a Javadoc file containing help documentation for the Java API

Note: In order to use the Java API, the DLL and Jar files must be on the Java Classpath.

Example Java project

An example Java project is supplied with your product installation. You can test the Java project and modify and use it as you like. For more details of the example Java project, see the section, Example Java Project.

Rules for mapping the Application API names to Java

The rules for mapping between the Application API and the Java wrapper are as follows:

• Classes and class names

For every interface of the Authentic Desktop automation interface a Java class exists with the name of the interface.

Method names

Method names on the Java interface are the same as used on the COM interfaces but start with a small letter to conform to Java naming conventions. To access COM properties, Java methods that prefix the property name with get and set can be used. If a property does not support write-access, no setter method is available. Example: For the Name property of the Document interface, the Java methods getName and setName are available.

• Enumerations

For every enumeration defined in the automation interface, a Java enumeration is defined with the same name and values.

• Events and event handlers

For every interface in the automation interface that supports events, a Java interface with

the same name plus 'Event' is available. To simplify the overloading of single events, a Java class with default implementations for all events is provided. The name of this Java class is the name of the event interface plus 'DefaultHandler'. For example: Application: Java class to access the application ApplicationEvents: Events interface for the Application

ApplicationEventsDefaultHandler: Default handler for ApplicationEvents

Exceptions to mapping rules

There are some exceptions to the rules listed above. These are listed below:

Interface	Java name
Document, method SetEncoding	setFileEncoding
AuthenticView, method Goto	gotoElement
AuthenticRange, method Goto	gotoElement
AuthenticRange, method Clone	cloneRange

This section

This section explains how some basic Authentic Desktop functionality can be accessed from Java code. It is organized into the following sub-sections:

- Example Java Project
- Application Startup and Shutdown
- Simple Document Access
- Iterations
- Use of Out-Parameters
- Event Handlers

Example Java Project

The Authentic Desktop installation package contains an example Java project, located in the Java folder of the API Examples folder:

Windows 7、Windows 8、	C:\Users\ <username>\Documents\</username>
Windows 10	Altova\Authentic2018\Authentic

This folder contains Java examples for the Authentic Desktop API. You can test it directly from the command line using the batch file BuildAndRun.bat, or you can compile and run the example project from within Eclipse. See below for instructions on how to use these procedures.

File list

The Java examples folder contains all the files required to run the example project. These files are listed below. If you are using a 64-bit version of the application, some filenames contain $_x64$ in the name. These filenames are indicated with $(_x64)$.

```
AltovaAutomation (_x64). Java-COM bridge: DLL part
```

dll	
AltovaAutomation.jar	Java-COM bridge: Java library part
AuthenticAPI.jar	Java classes of the Authentic Desktop API
RunAuthenticDesktop.jav	Java example source code
a	
BuildAndRun.bat	Batch file to compile and run example code from the command line prompt. Expects folder where Java Virtual Machine resides as parameter.
.classpath	Eclipse project helper file
.project	Eclipse project file
Authentic_JavaDoc.zip	Javadoc file containing help documentation for the Java API

What the example does

The example starts up Authentic Desktop and performs a few operations, including opening and closing documents. When done, Authentic Desktop stays open. You must close it manually.

- <u>Start Authentic Desktop</u>: Starts Authentic Desktop, which is registered as an automation server, or activates Authentic Desktop if it is already running.
- <u>Open example files</u>: Locates example documents installed with Authentic Desktop and opens them.
- <u>Iteration and Changing the View Mode</u>: Changes the view of all open documents to Text View. The code also shows how to iterate through open documents.
- <u>Iteration, validation, output parameters</u>: Validates the active document and shows the result in a message box. The code shows how to use output parameters.
- Event Handling: Shows how to handle Authentic Desktop events.
- <u>Shut down Authentic Desktop</u>: Shuts down Authentic Desktop.

You can modify the example in any way you like and run it.

Running the example from the command line

To run the example from the command line, open a command prompt window, go to the Java folder of the API Examples folder (*see above for location*), and then type:

buildAndRun.bat "<Path-to-the-Java-bin-folder>"

The Java binary folder must be that of a JDK 1.5 or later installation on your computer.

Press the **Return** key. The Java source in RunAuthenticDesktop.java will be compiled and then executed.

Loading the example in Eclipse

Open Eclipse and use the **Import | Existing Projects into Workspace** command to add the Eclipse project file (.project) located in the Java folder of the API Examples folder (*see above for location*). The project RunAuthenticDesktop will then appear in your Package Explorer or Navigator.

Select the project and then the command Run as | Java Application to execute the example.

Note: You can select a class name or method of the Java API and press F1 to get help for that class or method.

Java source code listing

The Java source code in the example file RunAuthenticDesktop.java is listed below with comments.

```
01 // Access generalJAVA-COM bridge classes
02 in port com altova autom ation .lbs.*;
03
04 // Access AuthenticDesktop Java-COM bridge
05 in port com altova autom ation Authentic Desktop*;
06 in port com altova autom ation Authentic Desktop Enum sSPYV iew Modes;
07
08 /*
09 * A sinple example that starts AuthenticDesktop COM server and performs a view operations on 1.
10 * Feelfree to extend.
11 *.
12 public class RunAuthenticDesktop
13 {
14 public static void main (String[] args)
15
16
      // An instance of the application.
17
     Application authenticDesktop = null;
18
19
     // Instead of COM error-handling, use Java exception mechanism.
20
     try
21
     {
22
      // Start Authentic Desktop as COM server.
23
      authenticDesktop = new Application ();
24
      // COM servers start up invisible so we make it visible
25
      authenticDesktopsetVisble (true);
26
27
       // Locate samples installed with the product.
      String strExamplesFolder = System getenv ("USERPROFILE") + "\\MyDocuments\\Alova\\Authentic2012\
28
\AuthenticExamples\\";
29
30
       // Open two files from the product samples.
31
      authenticDesktopgetDocuments (openFile (strExamplesFobler + "OrgChartpxf", false);
32
      authenticDesktopgetDocuments (openFile (strExamplesFober + "ExpReport xm I", false);
33
34
       // Iterate through allopen documents and set the View Mode to Text!
35
      for (Document doc authentic Desktopget Documents ())
36
       if (doc getCumentV iewMode () ⊨ SPYV iewModes spyV iewText)
37
         doc switchViewMode (SPYViewModes spyViewText);
38
39
       // An alternative iteration mode is index-based.COM indices are typically zero-based.
40
      Documents documents = authenticDesktopgetDocuments();
41
      for (int i= 1; i<= documents getCount(); i+)
42
        Document doc = documents getItem ();
43
44
45
        // Validation is one of the few methods that have output parameters.
46
        // The class JVariant is the correct type for parameters in these cases.
47
        // To get values back mark them with the by-reference flag.
        JVarant valbatonEmorText = new JVarantJStringVarant(""); valbatonEmorTextsetByRefFlag(;
48
49
        JVariant validation Error Count = new JVariant JIntVariant();
                                                                      valdationErrorCountsetByRefFlag();
50
        JVariant validationEmorXMLData = new JVariantJID ispatchVariant(); validationEmorXMLData setByRefFlag(;
51
        if (boc isValid (validationEmorText, validationEmorCount, validationEmorXMLData))
System outprinth ("Document" + doc getName () + " is not wellformed - " +
52
validationErrorTextgetStringValue ();
53
        else
54
         System outprinth ('Document " + doc getName () + " is wellform ed.");
55
       }
56
```

```
57
      // The following lines attach to the document events using a default in plementation
58
      // for the events and override one of its methods.
59
      // If you want to overribe all document events i is better to derive your listener class
60
       // from DocumentEvents and in plement all methods of this interface.
61
      Document doc = authenticDesktopgetActiveDocument();
62
      docaddListener new DocumentEventsDefaulHander)
63
64
        00 verride
65
       public bookan onBeforeCbseDocument(DocumentipDoc) throws AutomationException
66
67
         System outprinth ('Document " + ipDoc getName () + " requested cbsing.");
68
69
70
71
72
73
74
75
76
77
          // Albw cbsing of document
         return true;
      });
      doc c b se (true);
      doc = null:
      System outprinth ("Watch AuthenticDesktop!");
78
     catch (AutomationException e)
79
     -{
80
       // e printStackTrace ();
81
82
     finally
83
     {
84
       // Make sure that Authentic Desktop can shut down properly.
85
      if authenticDesktop = null)
86
       authenticDesktopdispose ();
87
88
      // Since the COM server was made visible and still is visible, it will keep running
89
       // and needs to be cbsed manually.
90
      System outprinth ('Now cbse AuthenticDesktop!');
91
     }
92
    }
93 }
```

Application Startup and Shutdown

The code listings below show how the application can be started up and shut down.

Application startup

Before starting up the application, the appropriate classes must be imported (see below).

```
01 // Access generalJAVA-COM bridge classes
02 in port com altova autom ation .lbs.*;
03
04 // Access Authentic Desktop Java-COM bridge
05 in port com altova autom ation AuthenticDesktop.*;
06 in port com altova autom ation AuthenticDesktopEnum sSPYV iewModes;
07
08 /**
09 * A simple example that starts AuthenticDesktop COM server and performs a view operations on it.
10 * Feelfree to extend.
11 *
12 public class RunAuthenticDesktop
13 {
14 public static void main (String[] args)
15
   - {
16
      // An instance of the application.
17
     Application authenticDesktop = null;
18
19
     // Instead of COM error-handling, use Java exception mechanism.
20
     try
21
22
     // Start AuthenticDesktop as COM server.
23
      authenticDesktop = new Application ();
24
      // COM servers start up invisible so we make t visible
25
      authenticDesktopsetVisble (true);
26
```

```
27 ...
28 }
29 }
30 }
```

Application shutdown

The application can be shut down as shown below.

```
1 {
2 // Make sure that AuthenticDesktop can shut down properly.
3 if (authenticDesktop = null)
4 authenticDesktopdispose ();
5
6 // Shee the COM server was made visible and still is visible, it willkeep running
7 // and needs to be closed manually.
8 System outprinth ("Now close AuthenticDesktop!");
9 }
```

Simple Document Access

The code listing below shows how to open a document.

```
1 // Locate samples installed with the product.
2 String strExamplesFolder = System getenv("USERPROFILE") + "\\MyDocuments\\Alova\\Authentic2012\
\AuthenticExamples\\";
3
4 // Open two files from the product samples.
5 authenticDesktopgetDocuments(openFile(strExamplesFolder + "OrgChartpxf", false);
6 authenticDesktopgetDocuments(openFile(strExamplesFolder + "ExpReportxm I', false);
```

Iterations

The listing below shows how to iterate through open documents.

```
01 // Iterate through allopen documents and set the View mode to Text!
02 for (Document doc authenticDesktopgetDocuments ())
03 if (doc getCurrentViewMode () = SPYViewModes spyViewText)
04 doc switchViewMode (SPYViewModes spyViewText);
05
06 // An alternative iteration mode is index-based.COM indices are typically zero-based.
07 Documents documents = authenticDesktopgetDocuments ();
08 for (int i= 1; i<= documentsgetCount(); i+)
09 {
10 Document doc = documentsgetItem ();
11 ...
12 }
```

Use of Out-Parameters

The code listing below iterates through open documents and validates each of them. For each validation, a message is generated using the output parameters of the Validation method.

^{01 //} Iterate through allopen documents and set the View mode to Text!

⁰² for (Document doc authentic Desktopget Documents ()

⁰³ **if** (doc getCumentV iewMode () ⊨ SPYV iewModes spyV iewText)

⁰⁴ doc switchViewMode (SPYViewModes spyViewText);

```
05
06 // An alternative iteration mode is index-based.COM indices are typically zero-based.
07 Documents documents = authenticDesktopgetDocuments ();
08 for (int i= 1; i<= documentsgetCount(); i+)
09 {
10 Document doc = documents getItem ();
11
12 // Validation is one of the few methods that have output parameters.
13 // The class JV ariant is the correct type for parameters in these cases.
14 // To get values back, mark them with the by-reference flag.
15 JVariant validationEmorText = new JVariantJStringVariant(""); validationEmorTextsetByRefFlag();
16 JVariant validationErrorCount=new JVariantJIntVariant(); validationErrorCountsetByRefFlag(;
17 JVariant validationEmorXMLData = new JVariantJD ispatchVariant(); validationEmorXMLData setByRefFlag(;
18 if (bloc isValid (validationEmorText, validationEmorCount, validationEmorXMLData))
     System outprinth ('Document' + doc getName () + " is not we life med - " + valibation Error TextgetS tringValue ();
19
20 else
21
     System outprinth ("Document" + doc getName () + " is wellform ed.");
22 }
```

Event Handlers

The listing below shows how to listen for and use events.

```
01 // The following lines attach to the document events using a default in plementation
02 // for the events and override one of its methods.
03 // If you want to override all document events, t is better to derive your listener class
04 // from DocumentEvents and in plement all methods of this interface.
05 Document doc = authenticDesktopgetActiveDocument();
06 doc addListener new DocumentEventsDefaulHandler)
07 {
08 @Override
09
   public boolean onBeforeCbseDocument(DocumentipDoc) throws AutomationException
10
     System outprinth ("Document" + ipDoc getName () + " requested cbsing.");
11
12
13
     // allow cbsing of document
14
    return true;
15
   }
16 });
17 doc.cbse (true);
18 \text{ doc} = \text{null}
```

3.2 Interfaces

Object Hierarchy

```
Application
   SpyProject
     SpyProjectItem s
       SpyProjectItem
  Docum ents
     Document
        GridView
        Authentic View
           AuthenticRange
           AuthenticDataTransfer (previously DocEditDataTransfer)
           AuthenticDataTransfer (previously DocEditDataTransfer)
         TextView
        XM LData
   Diabqs
    CodeGeneratorDb
    FibSelectionDlg
     SchemaDocumentationDb
     GenerateSam pleXM LD b
    DTDSchemaGeneratorDb
    FindhFilesDb
  DatabaseConnection
  ExportSettings
  TextIm portExportSettings
  ElementList
     Elem entListItem
```

Enum erations

Description

This chapter contains the reference of the Authentic Desktop 1.5 Type Library.

Most of the given examples are written in VisualBasic. These code snippets assume that there is a variable defined and set, called **objSpy of type Application**. There are also some code samples written in JavaScript.

3.2.1 Application

See also

Methods

GetDatabaseIm portElem entList GetDatabaseSettings GetDatabaseTables Im portFrom Database C reateXM LS chem aFrom DBS tructure

GetTextIm portElem entList GetTextIm portExportSettings Im portFrom Text

In portFrom W ord

In portFrom Schema

GetExportSettings

NewProject OpenProject

AddMacroMenultem ClearMacroMenu

ShowForm

ShowApplication

URLDelete URLMakeDirectory

AddXSLT XQ Param eter GetXSLT XQ Param eterCount GetXSLT XQ Param eterNam e GetXSLT XQ Param eterXPath Rem oveXSLT XQ Param eter

FindhFiles

Quit

Properties

Application Parent

ActiveDocum ent Docum ents

CurrentProject

Diabgs

WamingNumber WamingText

Status MajorVersion MinorVersion Edition <u>ISAP Supported</u> ServicePackVersion

Description

Application is the root for all other objects. It is the only object you can create by CreateObject (VisualBasic) or other similar COM related functions.

Example

```
D in objSpy As Application
Set objSpy = C reateObject("XM LSpy Application")
```

Events

OnBeforeOpenDocument

See also

Event: OnBeforeOpenDocum ent (ob Diabg as FileSelectionDb)

Description

This event gets fired whenever a document gets opened via the OpenFile or OpenURL menu command. It is sent after a document file has been selected but before the document gets opened. The file selection dialog object is initialized with the name of the selected document file. You can modify this selection. To continue the opening of the document leave the FileSelectionDgDialogAction property of *io_objDialog* at its default value spyDialogOK. To abort the opening of the document set this property to spyDialogCancel

Examples

Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:

```
Function On BeforeOpenDocum ent (obp abg)
End Function
```

XMLSpy scripting environment - JScript:

function On BeforeOpenDocum ent(obplabg)
{
}

XMLSpy IDE Plugin: KMLSpyPluginOnEvent (26, ...)

// nEventId = 26

OnBeforeOpenProject

See also

Event: OnBeforeOpenProject(obj/iabg as FileSelectionDlg)

Description

This event gets fired after a project file has been selected but before the project gets opened. The file selection dialog object is initialized with the name of the selected project file. You can modify this selection. To continue the opening of the project leave the FileSelectionDlgDialogAction property of *io_objDialog* at its default value spyDiabgOK. To abort the opening of the project set

this property to spyDiabgCancel

Examples

Given below are examples of how this event can be scripted.

 XMLSpy scripting environment - VBScript:

 Function On BeforeOpenProject(objDibbg)

 End Function

```
XMLSpy scripting environment - JScript:
function On BeforeOpenProject(obj) hbg)
```

XMLSpy IDE Plugin:

KMLSpyPluginOnEvent(25, ...)
// nEventId = 25

OnDocumentOpened

See also

Event: OnDocum entOpened (ob Docum ent as Docum ent)

Description

This event gets fired whenever a document opens in Authentic Desktop. This can happen due to opening a file with the OpenFile or OpenURL dialog, creating a new file or dropping a file onto Authentic Desktop. The new document gets passed as parameter. The operation cannot be canceled.

Examples

Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:

Function On OpenDocum ent (obpDocum ent) End Function

XMLSpy scripting environment - JScript:

function On_OpenDocum ent(obpocum ent)
{
}

XMLSpy IDE Plugin:

KMLSpyPlughOnEvent(7, ...) // nEventId = 7

OnProjectOpened

See also

Event: OnProjectOpened (objProject as SpyProject)

Description

This event gets fired whenever a project gets opened in Authentic Desktop. The new project gets passed as parameter.

Examples

Given below are examples of how this event can be scripted.

```
XMLSpy scripting environment - VBScript:
```

Function On OpenProject(objProject) End Function

XMLSpy scripting environment - JScript:

```
function On_OpenProject(objProject)
{
}
```

XMLSpy IDE Plugin:

IMM LSpyP lugIn OnEvent (6, ...) // nEventId = 6

ActiveDocument

See also

Property: ActiveDocum ent as Docum ent

Description

Reference to the active document. If no document is open, ActiveDocum ent is null (nothing).

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

AddMacroMenuItem

See also

Method: AddM acroM enultem (strM acro as String, strD isplayText as String)

Description

Adds a menu item to the **Tools** menu. This new menu item invokes the macro defined by strM acro. See also "Calling macros from Authentic Desktop".

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1108 Number of macro items is limited to 16 items.

AddXSLT_XQParameter

Method: AddXSLT XQParam eter(nam e as String, XPath as String)

Description

Adds an XSLT or XQuery parameter. The parameter's name and value are the two arguments of the method.

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.
- 1124 The XPath expression is not set.
- 1125 Not a QName.
- 1126 The specified XPath is not valid. Reason for invalidity appended.
- 1127 A parameter with the submitted name already exists.

Application

See also

Property: Application as Application (read-only)

Description

Accesses the Authentic Desktop application object.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

ClearMacroMenu

See also

Method: C barM acroM enu ()

Return Value None

Description

Removes all menu items from the Tools menu. See also Running macros.

Errors

1111 The application object is no longer valid.

CreateXMLSchemaFromDBStructure

See also

Method: C reateXM LS chem aFrom DBS tructure (pIm portSettings as DatabaseConnection, pT ables as Elem entList)

Description

C reateXMLS chem aFrom DBS tructure creates from a database specified in pIm portSettings for the defined tables in pTables new XML Schema document(s) describing the database tables structure.

The parameter pTables specifies which table structures the XML Schema document should contain. This parameter can be NULL, specifying that all table structures will be exported.

See also Get Dat aBaseTables.

- 1112 Invalid dat abase specified.
- 1120 Database import failed.

CurrentProject

See also

Property: CurrentProject as SpyProject

Description

Reference to the active document. If no project is open, CurrentProject is null (nothing).

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

Dialogs

See also

Property: Diabgs as Diabgs (read-only)

Description

Access the built- in dialogs of Authentic Desktop.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

Documents

See also

Property: Docum ents as Docum ents

Description

Collection of all open documents.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

Edition

See also

Property: Edition as String

Description

Returns the edition of the application, for example Altova Authentic Desktop Enterprise Edition for the Enterprise edition.

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

FindInFiles

See also

Method: FindInFiles(pSettings as FindInFilesDlg) as FindInFilesResults

Description

Returns a <u>FindInFilesResults</u> object containing information about the files that matched the specified settings.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

Get Dat abaseImport Element List

See also

Method: GetDatabase In portElem entList (pIn portSettings as DatabaseConnection) as Elem entList

Description

The function returns a collection of ElementListTeem s where the properties <u>ElementListTeem Name</u> contain the names of the fields that can be selected for import and the properties <u>ElementListTeem ElementKind</u> are initialized either to *spyXMLDataAttr* or *spyXMLDataElement*, depending on the value passed in <u>DatabaseConnection AsAttributes</u>. This list serves as a filter to what finally gets imported by a future call to <u>ImportFrom Database</u>. Use <u>ElementListRem oveElement</u> to exclude fields from import.

Properties mandatory to be filled out for the database connection are one of DatabaseConnection File, DatabaseConnection ADOConnection and DatabaseConnection ODBCConnection, as well as DatabaseConnection SQLSelect. Use the property DatabaseConnection AsAttributes to initialize ElementListTem ElementKind of the resulting element list to either spyXMLDataAttr or spyXMLDataElement, respectively.

Example

See example at Im portFrom Database.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1107 Import from database failed.
- 1112 Invalid database specified.
- 1114 Select statement is missing.
- 1119 database element list import failed.

GetDatabaseSettings

See also

Method: GetDatabaseSettings() as DatabaseConnection

Description

GetDatabaseSettings creates a new object of database settings. The object is used to specify database connection parameters for the methods GetDatabaseTables, GetDatabaseImportElementList, ImportFromDatabase, ImportFromSchema and ExportToDatabase.

Example

See example of In portFrom Database.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

GetDatabaseTables

See also

Method: GetDatabaseTables (pIn portSettings as DatabaseConnection) as ElementList

Description

GetDatabaseTables reads the table names from the database specified in *plmportSettings*. Properties mandatory to be filled out for the database connection are one of DatabaseConnection File, DatabaseConnection ADOConnection and DatabaseConnection ODBCConnection. All other properties are ignored.

The function returns a collection of Elem entListItem s where the properties Elem entListItem Name contain the names of tables stored in the specified database. The remaining properties of Elem entListItem are unused.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1112 Invalid database specified.
- 1113 Error while reading database table information.
- 1118 Database table query failed.

Example

Dim objim pSettings As DatabaseConnection Set objim pSettings = objSpyGetDatabaseSettings objim pSettings ADOConnection = TxtADO Text

store table nam es in list box ListTablesClear

Dim objlist As ElementList Dim objlem As ElementListlem On Emor GoTo EmorHandler Set objlist = objpy GetDatabaseTables (objlin pSettings)

For Each objlem In objlist ListTablesAddlem objlem Name Next

Get Export Settings

See also

Method: GetExportSettings (as ExportSettings (read-only)

Description

GetExportSettings creates a new object of common export settings. This object is used to pass the parameters to the export functions and defines the behaviour of the export calls. See also the export functions from Document.

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

GetTextImportElementList

See also

Method: GetTextIn portElem entList (pIn portSettings as TextIn portExportSettings) as Elem entList

Description

GetTextIm portElem entList retrieves importing information about the text-file as specified in pIm portSettings. The function returns a collection of Elem entListItem s where the properties Elem entListItem Name contain the names of the fields found in the file. The values of remaining properties are undefined.

If the text-file does not contain a column header, set plmportSettings.<u>HeaderRow</u> to false. The resulting element list will contain general column names like 'Field1' and so on.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1107 Import from database failed.
- 1115 Error during text element list import. Cannot create parser for import file.
- 1116 Error during text element list import.

Example

'-----'VBA client code fizom ent - in port selected fields from text file

D in objin pSettings As Textin portExportSettings Set objin pSettings = obj5pyGetTextIn portExportSettings

objīn pSettings In portFile = "C \In portMetxt" objīn pSettings HeaderRow = False

Dim objList As ElementList Set objList = objSpyGetTextIn portElementList(objIn pSettings)

exclude first colum n objListRem ovellem 1

D in objin pDoc As Docum ent On EnorResum e Next Set objin pDoc = objispy In portFrom Text(objin pSettings, objist) CheckForEnor

GetTextImportExportSettings

See also

Method: GetTextIn portExportSettings() as TextIn portExportSettings (read-only)

Description

GetTextIn portExportSettings creates a new object of common import and export settings for text files. See also the example for Application GetTextIn portElementList.

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

GetXSLT_XQParameterCount

Method: GetXSLT XQParameterCount() as Long

Description

Returns the number of XSLT and XQuery parameters.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

GetXSLT_XQParameterName

Method: GetXSLT XQParam eterNam e (index as Long) as String

Description

Returns the name of the XSLT or XQuery parameter identified by the supplied index.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

GetXSLT_XQParameterXPath

Method: GetXSLT XQParam eterXPath (index as Long) as String

Description

Returns the XPath expression of the XSLT or XQuery parameter identified by the supplied index.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

Import From Database

See also

Method: In portFrom Database (pIn portSettings as <u>DatabaseConnection</u> pElem entList as Elem entList) as Docum ent

Return Value

Creates a new document containing the data imported from the database.

Description

In portFrom Database imports data from a database as specified in pIm portSettings and creates a new document containing the data imported from the database. Properties mandatory to be filled out are one of DatabaseConnection File, DatabaseConnection ADOConnection or DatabaseConnection ODBCConnection and DatabaseConnection SQLSelect. Additionally, you can

USE DatabaseConnection AsAttributes, DatabaseConnection ExcludeKeys, DatabaseConnection_hcludeEmptyElem ents and <u>NumberDateTimeFormat</u> to further parameterize import.

The parameter $p \in lem entList$ specifies which fields of the selected data gets written into the newly created document, and which are created as elements and which as attributes. This parameter can be NULL, specifying that all selected fields will be imported as XML elements.

See GetDatabaseSettings and GetDatabaseIn portElementList for necessary steps preceding any import of data from a database.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1107 Import from database failed.
- 1112 Invalid dat abase specified.
- 1114 Select statement is missing.
- 1117 Transformation to XML failed.
- 1120 Database import failed.

Example

D in objin pSettings As DatabaseConnection Set objin pSettings = objpyGetDatabaseSettings

objīm pSettings ADOC onnection = stADOC onnection objīm pSettings SQLSelect = "SELECT * FROM MyTable"

Dim objOcc As Document On EnorResume Next Set objOcc = objSpyImportFrom Database (objImpSettings, objSpyGetDatabaseImportElementList(objImpSettings)) 'CheckForEnorhere

Import FromSchema

See also

Method: In portFrom Schem a (pin portSettings as DatabaseConnection, strTable as String, pSchem aDoc as Document) as Document

Return Value

Creates a new document filled with data from the specified database as specified by the schema definition in *pSchemaDoc*.

Description

In portFrom Schema imports data from a database specified in pIn portSettings. Properties mandatory to be filled out are one of DatabaseConnection File, DatabaseConnection ADOConnection OF DatabaseConnection ODBCConnection. Additionally, you can use DatabaseConnection AsAttributes, DatabaseConnection ExcludeKeys and Num berDateTim eForm at to further parameterize import. All other properties get ignored.

In portFrom Schema does not use and explicit SQL statement to select the data. Instead, it expects a structure definition of the document to create in form of an XML schema document in *pSchemaDoc*. From this definition the database select statement is automatically deduced. Specify in *strTable* the table name of the import root that will become the root node in the new document.

See GetDatabaseSettings and GetDatabaseTables for necessary steps preceding an import from a database based on a schema definition. To create the schema definition file use command 'create database schema' from the 'convert' menu of Authentic Desktop.

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1107 Import from database failed.
- 1112 Invalid dat abase specified.
- 1120 Database import failed.
- 1121 Could not create validator for the specified schema.
- 1122 Failed parsing schema for database import.

Import FromText

See also

Method: In portFrom Text (pin portSettings as <u>TextIn portExportSettings</u> <u>pE hem entList</u> as ElementList) as Document

Description

In portFrom Text imports the text file as specified in pIn portSettings. The parameter pElementList can be used as import filter. Either pass the list returned by a previous call to

<u>GetTextImportElementList</u> or null to import all columns. To avoid import of unnecessary columns use <u>ElementList.RemoveElement</u> to remove the corresponding field names from pElementList before calling ImportFromText.

The method returns the newly created document containing the imported data. This document is the same as the active document of Authentic Desktop.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1107 Import from text file failed.
- 1117 Transformation to XML failed.

Example

'VBA client code fragment - import from text file

D in objin pSettings As Textin portExportSettings Set objin pSettings = objpy GetTextIn portExportSettings

objm pSettings Im portFile = strFileName objm pSettings HeaderRow = False

```
Dim objīm pDoc As Docum ent
On Enor Resum e Next
Set objīm pDoc = objāpy in portFrom Text(objīm pSettings,
objāpy GetTextīm portE km entList(objīm pSettings))
```

CheckForEnor

Import FromWord

See also

Method: In portFrom W ord (strFile as String) as Document

Description

In portFrom W ord imports the MS-Word Document strFile into a new XML document.

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified. Import from document failed.

IsAPISupported

See also

Property: ISAP Supported as Boolean

Description

Returns whether the API is supported in this version or not.

Errors

1111 The application object is no longer valid.

1100 Invalid address for the return parameter was specified.

MajorVersion

See also

Property: MajorVersion as Integer

Description

Returns the application version's major number, for example 15 for 2013 versions, and 16 for 2014 versions.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

MinorVersion

See also

Property: M inorVersion as Integer

Description

Returns the application version's minor number.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

NewProject

See also

Method: NewProject(strPath as String, bD iscardCurrent as Boolean)

Description

NewProject creates a new project.

If there is already a project open that has been modified and bDiscardCurrent is false, then NewProject() fails.

- 1111 The application object is no longer valid.
- 1102 A project is already open but *bDiscardCurrent* is *true*.
- 1103 Creation of new project failed.

OpenProject

See also

Method: OpenProject(strPath as String,bDiscardCurrent as Boolean,bDiabg as Boolean)

Parameters

strPath Path and file name of the project to open. Can be empty if bDiabg is true.

bDiscardCurrent Discard currently open project and possible lose changes.

bDiabg Show dialogs for user input.

Return Value None

Description

OpenProject opens an existing project. If there is already a project open that has been modified and bDiscardCurrent is false, then OpenProject() fails.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.
- 1101 Cannot open specified project.
- 1102 A project is already open but *bDiscardCurrent* is *true*.

Parent

See also

Property: Parent as Application (read-only)

Description

Accesses the Authentic Desktop application object.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

Quit

See also

Method: Quit()

Return Value

None

Description

This method terminates Authentic Desktop. All modified documents will be closed without saving the changes. This is also true for an open project.

If Authentic Desktop was automatically started as an automation server by a client program, the application will not shut down automatically when your client program shuts down if a project or any document is still open. Use the Quit method to ensure automatic shut-down.

Errors

1111 The application object is no longer valid.

ReloadSettings

See also

Method: RebadSettings

Return Value

Description

The application settings are reloaded from the registry.

Available with TypeLibrary version 1.5

Errors

1111 The application object is no longer valid.

RemoveXSLT_XQParameter

Method: Rem oveXSLT XQParam eter(index as Long)

Description

Removes the XSLT or XQuery parameter identified by the supplied index.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

RunMacro

See also

Method: RunMacro (strMacro as String)

Return Value

Description

Calls the specified macro either from the project scripts (if present) or from the global scripts.

Available with TypeLibrary version 1.5

1111 The application object is no longer valid.

ScriptingEnvironment

See also

Property: ScriptingEnvironment as IUnknown (read-only)

Description

Reference to any active scripting environment. This property makes it possible to access the TypeLibrary of the XMLSpyFormEditor.exe application which is used as the current scripting environment.

Available with TypeLibrary version 1.5

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

ServicePackVersion

See also

Property: ServicePackVersion as Long

Description

Returns the Service Pack version number of the application. Eg: 1 for 2010 R2 SP1

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

ShowApplication

See also

Method: ShowApplication (bShow as Boolean)

Return Value None

Description

The method shows (bShow = True) or hides (bShow = False) Authentic Desktop.

Errors

1110 The application object is no longer valid.

ShowFindInFiles

See also

Method: ShowFindInFiles (pSettings as FindInFilesDlg) as Boolean

Return Value

Returns false if the user pressed the Cancel button, true otherwise.

Description

Displays the FindInFiles dialog preset with the given settings. The user modifications of the settings are stored in the passed dialog object.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.

ShowForm

See also

Method: ShowForm (strForm Name as String) as Long

Return Value

Returns zero if the user pressed a Cancel button or the form calls TheView Cancel).

Description

Displays the form strForm Name.

Forms, event handlers and macros can be created with the Scripting Environment. Select "Switch to scripting environment" from the **Tools** menu to invoke the Scripting Environment.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter or invalid address for the return parameter was specified.

Status

See also

Property: Status as ENUMApplicationStatus

Description

Returns the current status of the running application.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

URLDelete

See also

Method: URLDelete (*strURL* as String, *strUser* as String, *strPassword* as String)

Return Value None

Description

The method deletes the file at the URL strURL.

Errors

1111 The application object is no longer valid.

1109 Error deleting file at specified URL.

URLMakeDirectory

See also

Method: URLM akeDirectory (strURL as String, strUser as String, strPassword as String)

Return Value None

Description

The method creates a new directory at the URL strURL.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid parameter specified.

Visible

See also

Property: Visible as VARIANT BOOL

Description

Sets or gets the visibility attribute of Authentic Desktop. This standard automation property makes usage of ShowApplication obsolete.

Errors

- 1110 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

WarningNumber

See also

Property: WamingNumber as integer

Description

Some methods fill the property WarningNumber with additional information if an error occurs.

Currently just Docum ents OpenFile fills this property.

Errors

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

WarningText

See also

Property: WamingText as String

Description

Some methods fill the property W amingText with additional information if an error occurs.

Currently just Docum ents OpenFile fills this property.

- 1111 The application object is no longer valid.
- 1100 Invalid address for the return parameter was specified.

3.2.2 AuthenticContextMenu

The context menu interface provides the mean for the user to customize the context menus shown in Authentic. The interface has the methods listed in this section.

CountItems

Method: CountItem s() nltems as long

Return Value Returns the number of menu items.

Errors

2501 Invalid object.

Deleteltem

Method: Deleteltem (IndexPosition as long)

Return Value

Deletes the menu item that has the index position submitted in the first parameter.

Errors

2501 Invalid object 2502 Invalid index

GetItemText

Method: GetItem Text(IndexPosition as long) MenultemName as string

Return Value

Gets the name of the menu item located at the index position submitted in the first parameter.

Errors

2501 Invalid object2502 Invalid index

InsertItem

Method: InsertLem (IndexPosition as long, MenuItemName as string, MacroName as string)

Return Value

Inserts a user-defined menu item at the position in the menu specified in the first parameter and having the name submitted in the second parameter. The menu item will start a macro, so a valid macro name must be submitted.

Errors

2501 Invalid object2502 Invalid index2503 No such macro

2504 Internal error

SetItemText

Method: SetItem Text(IndexPosition as long, MenuItemName as string)

Return Value

Sets the name of the menu item located at the index position submitted in the first parameter.

Errors

2501 Invalid object2502 Invalid index

3.2.3 AuthenticDataTransfer

Renamed from DocEditDataTransfer to AuthenticDataTransfer

The DocEditView object is renamed to O bAuthenticView. DocEditSelection is renamed to AuthenticSelection. DocEditEvent is renamed to AuthenticEvent. DocEditDataTransfer is renamed to AuthenticDataTransfer.

Their usage—except for AuthenticDataTransfer—is no longer recommended. We will continue to support existing functionality for a yet undefined period of time but no new features will be added to these interfaces.

For examples on migrating from DocEdit to Authentic see the description of the different methods and properties of the different DocEdit objects.

See also

Methods

getData

Properties

dropEffect ownDrag type

Description

The events OnDragOver and OnBeforeDrop provide information about the object being dragged with an instance of type AuthenticDataTransfer. It contains a description of the dragged object and its content. The latter is available either as string or a pointer to a COM object supporting the Unkown interface.

dropEffect

See also

Property: dropEffect as long

Description

The property stores the drop effect from the default event handler. You can set the drop effect if you change this value and return TRUE for the event handler.

Errors

2101 Invalid address for the return parameter was specified.

getData

See also

Method: getData () as Variant

Description

Retrieve the data associated with the dragged object. Depending on <u>AuthenticDataTransfer.type</u>, that data is either a string or a COM interface pointer of type <u>Unknown</u>.
Errors

2101 Invalid address for the return parameter was specified.

ownDrag

See also

Property: ownDrag as Boolean (read-only)

Description

The property is TRUE if the current dragging source comes from inside Authentic View.

Errors

2101 Invalid address for the return parameter was specified.

type

See also

Property: type as String (read-only)

Description

Holds the type of data you get with the DocEditDataTransfergetData method.

Currently supported data types are:

OWN	data from Authentic View itself
TEXT	plain text
UNICODETEXT	plain text as UNICODE

Errors

2101 Invalid address for the return parameter was specified.

3.2.4 AuthenticEventContext

The EventContext interface gives access to many properties of the context in which a macro is executed.

EvaluateXPath

Method: EvaluateXPath (strExpression as string) as strValue as string

Return Value

The method evaluates the XPath expression in the context of the node within which the event was triggered and returns a string.

Description

EvaluateXPath() executes an XPath expressions with the given event context. The result is returned as string, in the case of a sequence it is a space-separated string.

Errors

2201	Invalid object.
2202	No context.
2209	Invalid parameter.
2210	Internal error.
2211	XPath error.

Get Event Context Type

Method: GetEventContextType () Type as AuthenticEventContextType enumeration

Return Value

Returns the context node type.

Description

GetEventContextType allows the user to determine whether the macro is in an XML node or in an XPath atomic item context. The enumeration AuthenticEventContextType is defined as follows:

authenticEventContextXML, authenticEventContextAtomicItem, authenticEventContextOther

If the context is a normal XML node, the GetXMLNode() function gives access to it (returns NULL if not).

Errors

2201 Invalid object.2202 No context.2209 Invalid parameter.

Get NormalizedText Value

Method: GetNorm alizedTextValue() strValue as string

Return Value

Returns the value of the current node as string

Errors

2201	Invalid object.	
	•• • • •	

2202 No context.

2203 Invalid context

2209 Invalid parameter.

GetVariableValue

Method: GetVariableValue(strName as string) strValue as string

Return Value

Gets the value of the variable submitted as the parameter.

Description

GetVariableValue gets the variable's value in the scope of the context.

```
nZoom = parseInt(AuthenticView EventContextGetVariableValue(Zoom'));
if (nZoom > 1)
{
    AuthenticView EventContextSetVariableValue(Zoom',nZoom - 1);
}
```

Errors

- 2201 Invalid object.
- 2202 No context.
- 2204 No such variable in scope
- 2205 Variable cannot be evaluated
- 2206 Variable returns sequence
- 2209 Invalid parameter

GetXMLNode

Method: GetXM LNode () Node as XMLData object

Return Value

Returns the context XML node or NULL

Errors

- 2201 Invalid object.
- 2202 No context.
- 2203 Invalid context
- 2209 Invalid parameter.

IsAvailable

Method: IsAvailable() as Boolean

Return Value

Returns true if EventContext is set, false otherwise.

Errors

2201 Invalid object.

Set Variable Value

Method: SetVariableValue(strName as string, strValue as string)

Return Value

Sets the value (second parameter) of the variable submitted in the first parameter.

Description

SetVariableValue sets the variable's value in the scope of the context.

```
nZoom = parseInt(AuthenticView EventContextGetVariableValue(Zoom'));
if (nZoom > 1)
{
    AuthenticView EventContextSetVariableValue(Zoom', nZoom - 1);
```

- 2201 Invalid object.
- 2202 No context.
- 2204 No such variable in scope
- 2205 Variable cannot be evaluated
- 2206 Variable returns sequence
- 2207 Variable read-only
- 2208 No modification allowed

3.2.5 AuthenticRange

See also

The first table lists the properties and methods of AuthenticRange that can be used to navigate through the document and select specific portions.

Properties

- Application FirstTextPosition FirstXM LData FirstXM LDataO ffset LastTextPosition LastXM LData LastXM LData Parent
- C bne C ollapsToBegin C ollapsToEnd ExpandTo G oto G otoNext G otoPrevious IsEm pty IsEqual

Methods

MoveBegin MoveEnd NextCursorPosition PreviousCursorPosition Select SelectNext SelectPrevious SetFrom Range

The following table lists the content modification methods, most of which can be found on the right/button mouse menu.

Properties	Edit operations	Dynamic table operations
Text	Сору	AppendRow
	Cut	DeleteRow
	Delete	DuplicateRow
	IsC opyEnabled	InsertRow
	IsC utEnabled	IsFirstRow
	IsDeleteEnabled	IsInDynam icTable
	IsPasteEnabled	IsLastRow
	Paste	MoveRowDown
		MoveRowUp

The following methods provide the functionality of the Authentic entry helper windows for range objects.

Operations of the entry helper windows

Elements	Attributes	Entities
CanPerform ActionWith	GetElementAttributeValue	GetEntityNames
CanPerform Action	GetElementAttributeNames	InsertEntity
PerformAction	GetElementHierarchy	
	HasElementAttribute	
	IsTextS tateApplied	
	SetElementAttributeValue	

Description

AuthenticRange objects are the 'cursor' selections of the automation interface. You can use them to point to any cursor position in the Authentic view, or select a portion of the document. The operations available for AuthenticRange objects then work on this selection in the same way, as the corresponding operations of the user interface do with the current user interface selection. The main difference is that you can use an arbitrary number of AuthenticRange objects at the same time, whereas there is exactly one cursor selection in the user interface.

To get to an initial range object use <u>AuthenticView Selection</u>, to obtain a range corresponding with the current cursor selection in the user interface. Alternatively, some trivial ranges are accessible via the read/only properties <u>AuthenticView DocumentBegin</u>, <u>AuthenticView DocumentEnd</u>, and <u>AuthenticView WholeDocument</u>. The most flexible method is <u>AuthenticView Goto</u>, which allows navigation to a specific portion of the document within one call. For more complex selections, combine the above, with the various navigation methods on range objects listed in the first table on this page.

Another method to select a portion of the document is to use the position properties of the range object. Two positioning systems are available and can be combined arbitrarily:

- Absolute text cursor positions, starting with position 0 at the document beginning, can be set and retrieved for the beginning and end of a range. For more information see <u>FirstTextPosition</u> and <u>LastTextPosition</u>. This method requires complex internal calculations and should be used with care.
- The **XMLData** element and a text position inside this element, can be set and retrieved for the beginning and end of a range. For more information see <u>FirstXM LData</u>, <u>FirstXM LDataO ffset</u>, <u>LastXM LData</u>, and <u>LastXM LDataO ffset</u>. This method is very efficient but requires knowledge on the underlying document structure. It can be used to locate XMLData objects and perform operations on them otherwise not accessible through the user interface.

Modifications to the document content can be achieved by various methods:

- The <u>Text</u> property allows you to retrieve the document text selected by the range object. If set, the selected document text gets replaced with the new text.
- The standard document edit functions Cut, Copy, Paste and Delete.
- Table operations for tables that can grow dynamically.
- Methods that map the functionality of the Authentic entry helper windows.
- Access to the XMLData objects of the underlying document to modify them directly.

AppendRow

See also

Method: AppendRow ()as Boolean

Description

If the beginning of the range is inside a dynamic table, this method inserts a new row at the end of the selected table. The selection of the range is modified to point to the beginning of the new row. The function returns *true* if the append operation was successful, otherwise *false*.

Errors

- 2001 The authentic range object or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

Examples

```
Altova Authentic 2018 Desktop Enterprise Edition
```

Application

See also

Property: Application as Application (read-only)

Description

Accesses the Authentic Desktop application object.

Errors

2001 The authentic range object or its related view object is no longer valid.2005 Invalid address for the return parameter was specified.

CanPerformAction

See also

Method: CanPerform Action (*eAction* as <u>SPYAuthenticActions</u>, *strElem entNam e* as String) as Boolean

Description

CanPerform Action and its related methods enable access to the entry-helper functions of Authentic. This function allows easy and consistent modification of the document content, without having to know exactly where the modification will take place. The beginning of the range object is used to locate the next valid location where the specified action can be performed. If the location can be found, the method returns *True*, otherwise it returns *False*.

HINT: To find out all valid element names for a given action, use CanPerform ActionWith.

Errors

- 2001 The authentic range object or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.
- 2007 Invalid action was specified.

Examples

See Perform Action.

CanPerformActionWith

See also

Method: CanPerform ActionW ith (*eAction* as <u>SPYAuthenticActions</u>, *out_arrElem entNam es* as Variant)

Description

Perform ActionW ith and its related methods, enable access to the entry-helper functions of Authentic. These function allows easy and consistent modification of the document content without having to know exactly where the modification will take place.

This method returns an array of those element names that the specified action can be performed with.

HINT: To apply the action use CanPerform ActionW ith.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.
- 2007 Invalid action was specified.

Examples

See Perform Action.

Clone

See also

Method: C bne () as AuthenticRange

Description

Returns a copy of the range object.

Errors

2001 The authentic range object, or its related view object is no longer valid. 2005 Invalid address for the return parameter was specified.

CollapsToBegin

See also

Method: CollapsToBegin () as AuthenticRange

Description

Sets the end of the range object to its begin. The method returns the modified range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

CollapsToEnd

See also

Method: CollapsToEnd() as AuthenticRange

Description

Sets the beginning of the range object to its end. The method returns the modified range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

Сору

See also

Method: Copy() as Boolean

Description

Returns *False* if the range contains no portions of the document that may be copied. Returns *True* if text, and in case of fully selected XML elements the elements as well, has been copied to the copy/ paste buffer.

Errors

- 2001 The authentic range object or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

Cut

See also

Method: Cut() as Boolean

Description

Returns *False* if the range contains portions of the document that may not be deleted. Returns *True* after text, and in case of fully selected XML elements the elements as well, has been deleted from the document and saved in the copy/ paste buffer.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

Delete

See also

Method: Delete () as Boolean

Description

Returns *False* if the range contains portions of the document that may not be deleted. Returns *True* after text, and in case of fully selected XML elements the elements as well, has been deleted from the document.

Errors

- 2001 The authentic range object or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

DeleteRow

See also

Method: DeleteRow () as Boolean

Description

If the beginning of the range is inside a dynamic table, this method deletes the selected row. The selection of the range gets modified to point to the next element after the deleted row. The function returns *true*, if the delete operation was successful, otherwise *false*.

Errors

2001	The authentic range object, or its related view object is no longer valid.
2005	Invalid address for the return parameter was specified.

Examples

```
'Scripting environment - VBScript
'Delete selected row from dynamically growing table
'_______
Dim obRange
'we assume that the active document is open in authentic view mode
Set obRange = Application ActiveDocumentAuthenticView Selection
'check if we are in a table
```

```
If obRange IsInD ynam icTable Then
obRange DeleteRow
End If
```

DuplicateRow

See also

Method: DuplicateRow () as Boolean

Description

If the beginning of the range is inside a dynamic table, this method inserts a duplicate of the current row after the selected one. The selection of the range gets modified to point to the beginning of the new row. The function returns *true* if the duplicate operation was successful, otherwise *false*.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

Examples

```
'Scripting environment - VBScript
'duplicate row in current dynamically growable table
'______
Dim obRange
'we assume that the active document is open in authentic view mode
Set obRange = Application ActiveDocumentAuthenticView Selection
'check if we can insert something
If obRange IsinDynamicTable Then
```

```
ob Range Select
```

End If

EvaluateXPath

Method: EvaluateXPath (strExpression as string) strValue as string

Return Value

The method returns a string

ob Range DuplicateRow

'ob Range points to beginning of new row

Description

EvaluateXPath() executes an XPath expressions with the context node being the beginning of the range selection. The result is returned as string, in the case of a sequence it is a space-separated string. If XML context node is irrelevant, the user may provide any node, like AuthenticView.XMLDataRoot.

Errors

2001 Invalid object

- 2005 Invalid parameter
- 2008 Internal error
- 2202 Missing context node
- 2211 XPath error

ExpandTo

See also

Method: ExpandTo (eKind as SPYAuthenticElem entKind), as AuthenticRange

Description

Selects the whole element of type eKind, that starts at, or contains, the first cursor position of the range. The method returns the modified range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2003 Range expansion would be beyond end of document.
- 2005 Invalid address for the return parameter was specified.

FirstTextPosition

See also

Property: FirstTextPosition as Long

Description

Set or get the left-most text position index of the range object. This index is always less or equal to LastTextPosition. Indexing starts with 0 at document beginning, and increments with every different position that the text cursor can occupy. Incrementing the test position by 1, has the same effect as the cursor-right key. Decrementing the test position by 1 has the same effect as the cursor-left key.

If you set FirstTextPosition to a value greater than the current LastTextPosition, LastTextPosition gets set to the new FirstTextPosition.

HINT: Use text cursor positions with care, since this is a costly operation compared to XM LData based cursor positioning.

Errors

- 2001 The authentic range object, or its related view object is not valid.
- 2005 Invalid address for the return parameter was specified.
- 2006 A text position outside the document was specified.

Examples

```
'Scripting environm ent – VBScript
```

```
Dim objAuthenticView
```

'we assume that the active document is open in authentic view mode Set objAuthenticView = Application ActiveDocumentAuthenticView

nDocStartPosition = objAuthenticView Docum entBegin FirstTextPosition nDocEndPosition = objAuthenticView Docum entEndFirstTextPosition

```
'lets create a range that selects the whole document
'in an inefficient way
```

End If

FirstXMLData

See also

Property: FirstXM LData as XM LData

Description

Set or get the first XM LData element in the underlying document that is partially, or completely selected by the range. The exact beginning of the selection is defined by the FirstXM LDataO ffset attribute.

Whenever you set FirstXM LData to a new data object, FirstXM LDataO ffset gets set to the first cursor position inside this element. Only XMLData objects that have a cursor position may be used. If you set FirstXM LData / FirstXM LDataO ffset selects a position greater then the current LastXM LData / LastXM LData / fiset, the latter gets moved to the new start position.

HINT: You can use the <u>FirstXMLData</u> and <u>LastXMLData</u> properties, to directly access and manipulate the underlying XML document in those cases where the methods available with the <u>AuthenticRange</u> object are not sufficient.

Errors

- 2001 The authentic range object, or its related view object is not valid.
- 2005 Invalid address for the return parameter was specified.
- 2008 Internal error
- 2009 The XMLData object cannot be accessed.

Examples

```
'Scripting environment - VBScript
'show name of currently selected XMLData element
```

```
Dim objAuthenticView
```

'we assume that the active document is open in authentic view mode Set objAuthenticView = ApplicationActiveDocumentAuthenticView

```
Dim objkm Data
Set objKM LData = objAuthenticView Selection FirstXM LData
'authentic view adds a text'child element to elements
'of the document which have content. So we have to go one
'element up.
Set objKM LData = objKM LData Parent
M sgBox "Current selection selects element " & objKM LData Name
```

FirstXMLDataOffset

See also

Property: FirstXM LDataO ffset as Long

Description

Set or get the cursor position offset inside <u>FirstXM LData</u> element for the beginning of the range. Offset positions are based on the characters returned by the <u>Text</u> property, and start with 0. When setting a new offset, use -1 to set the offset to the last possible position in the element. The following cases require specific attention:

- The textual form of entries in Combo Boxes, Check Boxes and similar controls can be different from what you see on screen. Although the data offset is based on this text, there only two valid offset positions, one at the beginning and one at the end of the entry. An attempt to set the offset to somewhere in the middle of the entry, will result in the offset being set to the end.
- The textual form of XML Entities might differ in length from their representation on the screen. The offset is based on this textual form.

If FirstXM LData / FirstXM LDataO ffset selects a position after the current LastXM LData / LastXM LDataO ffset, the latter gets moved to the new start position.

Errors

The authentic range object, or its related view object is not valid.
 Invalid offset was specified.
 Invalid address for the return parameter was specified.

Examples

```
'Scripting environment - VBScript
'Select the complete text of an XM LD ata element
'using XM LD ata based selection and ExpandTo
Dim ob Authentic View
'we assume that the active document is open in authentic view mode
Set ob AuthenticV iew = Application ActiveD ocum entAuthenticV iew
'first we use the XM LD ata based range properties
'to select all text of the first XM LD ata element
'in the current selection
Dim ob Range
Set ob Range = ob Authentic View Selection
ob Range FirstXM LD ataO ffset = 0 'start at beginning of element text
obRangeLastXMLData = obRangeFirstXMLData 'select only one element
ob Range LastXM LD ataO ffset = -1 'select till its end
'the same can be achieved with the ExpandTo method
Dim ob Range2
Set ob Range2 = ob AuthenticV iew Selection ExpandTo (spyAuthenticTag)
'were we successful?
If ob Range IsEqual(ob Range2) Then
```

```
obstangeSelect()
Else
MsgBox "Oops"
```

End If

Get Element AttributeNames

See also

Method: GetElementAttributeNames (*strElementName* as String, *out_arrAttributeNames* as Variant)

Description

Retrieve the names of all attributes for the enclosing element with the specified name. Use the element/attribute pairs, to set or get the attribute value with the methods GetElementAttributeValue and SetElementAttributeValue.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid element name was specified.
 - Invalid address for the return parameter was specified.

Examples

See SetElem entAttributeValue.

Get Element AttributeValue

See also

Method: GetElementAttributeValue (*strElementName* as String, *strAttributeName* as String) as String

Description

Retrieve the value of the attribute specified in strAttributeName, for the element identified with strElementName. If the attribute is supported but has no value assigned, the empty string is returned. To find out the names of attributes supported by an element, use GetElementAttributeNames, or HasElementAttribute.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid element name was specified. Invalid attribute name was specified. Invalid address for the return parameter was specified.

Examples

See SetElem entAttributeValue.

Get Element Hierarchy

See also

Method: GetElem entHierarchy (*out arrElem entNam es* as Variant)

Description

Retrieve the names of all XML elements that are parents of the current selection. Inner elements get listed before enclosing elements. An empty list is returned whenever the current selection is not inside a single XM LData element.

The names of the element hierarchy, together with the range object uniquely identify XM LData elements in the document. The attributes of these elements can be directly accessed by GetElementAttributeNames, and related methods.

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

Get EntityNames

See also

Method: GetEntityNames (*out arrEntityNames* as Variant)

Description

Retrieve the names of all defined entities. The list of retrieved entities is independent of the current selection, or location. Use one of these names with the InsertEntity function.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

Examples

See: GetElementHierarchy and InsertEntity.

Get Variable Value

Method: GetVariableValue(strName as string) strVal as string

Return Value

Gets the value of the variable named as the method's parameter.

Errors

- 2001 Invalid object.
- No context.
- 2204 No such variable in scope
- 2205 Variable cannot be evaluated
- 2206 Variable returns sequence
- 2209 Invalid parameter

Goto

See also

Method: Goto (*eKind* as <u>SPYAuthenticElem</u> entKind, *nCount* as Long, *eFrom* as SPYAuthenticDocum entPosition) as AuthenticRange

Description

Sets the range to point to the beginning of the nCount element of type eKind. The start position is defined by the parameter eFrom.

Use positive values for nCount to navigate to the document end. Use negative values to navigate to the beginning of the document. The method returns the modified range object.

- 2001 The authentic range object, or its related view object is no longer valid.
- 2003 Target lies after end of document.
- 2004 Target lies before begin of document.
- 2005 Invalid element kind specified.
 - Invalid start position specified.

Invalid address for the return parameter was specified.

GotoNext

See also

Method: GotoNext (eKind as SPYAuthenticElem entKind) as AuthenticRange

Description

Sets the range to the beginning of the next element of type eKind. The method returns the modified range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2003 Target lies after end of document.
- 2005 Invalid element kind specified. Invalid address for the return parameter was specified.

Examples

'Scripting environment - VBScript
'Scan through the whole document word-by-word
'-----D in objAuthenticView
'we assume that the active document is open in authentic view mode
Set objAuthenticView = ApplicationActiveDocumentAuthenticView

```
Dim obRange
Set obRange = obAuthenticView Docum entBegin
Dim bEndOfDocum ent
bEndOfDocum ent = False
```

```
On EmorResume Next

While NotbEndOfDocument

obRangeGotoNext(spyAuthenticWord)Select

If (Emnumber - vbObjectemon) = 2003)Then

bEndOfDocument = True

Emclear

ElseIf (Emnumber <> 0)Then

EmRaise 'forward emor

End If

Wend
```

Got oNext Cursor Position

See also

Method: GotoNextCursorPosition () as AuthenticRange

Description

Sets the range to the next cursor position after its current end position. Returns the modified object.

2001	The authentic range object, or its related view object is no longer valid.
2003	Target lies after end of document.
2005	Invalid address for the return parameter was specified.

GotoPrevious

See also

Method: GotoPrevious (eKind as SPYAuthenticElem entKind) as AuthenticRange

Description

Sets the range to the beginning of the element of type eKind which is before the beginning of the current range. The method returns the modified range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2004 Target lies before beginning of document.
- 2005 Invalid element kind specified. Invalid address for the return parameter was specified.

Examples

Dim objAuthenticView 'we assume that the active document is open in authentic view mode Set objAuthenticView = ApplicationActiveDocumentAuthenticView

```
Dim obRange
Set obRange = obAuthenticView DocumentEnd
Dim bEndOfDocument
bBeginOfDocument = False
```

```
On EmorResume Next

While NotbBeginOfDocument

objRangeGotoPrevious(spyAuthenticTag)Select

If (Ernumber - vbObjectenor) = 2004)Then

bBeginOfDocument = True

Emclear

Elself (Ernumber <> 0)Then

ErrRaise 'forward emor

End If

W end
```

Wena

GotoPreviousCursorPosition

See also

Method: GotoPreviousCursorPosition () as AuthenticRange

Description

Set the range to the cursor position immediately before the current position. Returns the modified object.

- 2001 The authentic range object, or its related view object is no longer valid.
- 2004 Target lies before begin of document.
- 2005 Invalid address for the return parameter was specified.

HasElementAttribute

See also

Method: HasElem entAttribute (*strElem entNam e* as String, *strAttributeNam e* as String) as Boolean

Description

Tests if the enclosing element with name strElem entName, supports the attribute specified in strAttributeName.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid element name was specified.
 - Invalid address for the return parameter was specified.

Insert Entity

See also

Method: InsertEntity (*strEntityName* as String)

Description

Replace the ranges selection with the specified entity. The specified entity must be one of the entity names returned by GetEntityNames.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Unknown entry name was specified.

Examples

```
'Scripting environment - VBScript
'Insert the first entity in the list of available entities
'
Dim obRange
'we assume that the active document is open in authentic view mode
Set obRange = Application ActiveDocumentAuthenticView Selection
```

```
'first we get the names of all available entities as they
'are shown in the entry helper of XM LSpy
Dim arEntities
ob Range GetEntityNames arEntities
```

Insert Row

See also

Method: InsertRow () as Boolean

Description

If the beginning of the range is inside a dynamic table, this method inserts a new row before the

current one. The selection of the range, gets modified to point to the beginning of the newly inserted row. The function returns *true* if the insert operation was successful, otherwise *false*.

Errors

2001 The authentic range object, or its related view object is no longer valid.

2005 Invalid address for the return parameter was specified.

Examples

End If

IsCopyEnabled

See also

Property: IsC opyEnabled as Boolean (read-only)

Description

Checks if the copy operation is supported for this range.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

IsCut Enabled

See also

Property: IsCutEnabled as Boolean (read-only)

Description

Checks if the cut operation is supported for this range.

Errors

2001 The authentic range object, or its related view object is no longer valid. 2005 Invalid address for the return parameter was specified.

IsDeleteEnabled

See also

Property: IsDeleteEnabled as Boolean (read-only)

Description

Checks if the delete operation is supported for this range.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

IsEmpt y

See also

Method: IsEm pty () as Boolean

Description

Tests if the first and last position of the range are equal.

Errors

The authentic range object, or its related view object is no longer valid.Invalid address for the return parameter was specified.

IsEqual

See also

Method: IsEqual (objCmpRange as AuthenticRange) as Boolean

Description

Tests if the start and end of both ranges are the same.

Errors

2001	One of the two range objects being compared, is invalid.
2005	Invalid address for a return parameter was specified.

IsFirst Row

See also

Property: IsFirstRow as Boolean (read-only)

Description

Test if the range is in the first row of a table. Which table is taken into consideration depends on the extend of the range. If the selection exceeds a single row of a table, the check is if this table is the first element in an embedding table. See the entry helpers of the user manual for more information.

Errors

2001 The authentic range object, or its related view object is no longer valid.2005 Invalid address for the return parameter was specified.

IsInDynamicTable

See also

Method: IsInDynam icTable () as Boolean

Description

Test if the whole range is inside a table that supports the different row operations like 'insert',

'append', duplicate, etc.

Errors

2001 The authentic range object, or its related view object is no longer valid. 2005 Invalid address for the return parameter was specified.

IsLast Row

See also

Property: IsLastRow as Boolean (read-only)

Description

Test if the range is in the last row of a table. Which table is taken into consideration depends on the extend of the range. If the selection exceeds a single row of a table, the check is if this table is the last element in an embedding table. See the entry helpers of the user manual for more information.

Errors

The authentic range object, or its related view object is no longer valid.Invalid address for the return parameter was specified.

IsPasteEnabled

See also

Property: IsPasteEnabled as Boolean (read-only)

Description

Checks if the paste operation is supported for this range.

Errors

2001 The authentic range object, or its related view object is no longer valid.

2005 Invalid address for the return parameter was specified.

IsSelected

Property: IsSelected as Boolean

Description

Returns true() if selection is present. The selection range still can be empty: that happens when e.g. only the cursor is set.

IsTextStateApplied

See also

Method: **INTEXTS** tateApplied (*istrE lem entNam e* as String) as Boolean

Description

Checks if all the selected text is embedded into an XML Element with name istrElementName. Common examples for the parameter istrElementName are "strong", "bold" or "italic".

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

Last Text Position

See also

Property: LastTextPosition as Long

Description

Set or get the rightmost text position index of the range object. This index is always greater or equal to FirstTextPosition. Indexing starts with 0 at the document beginning, and increments with every different position that the text cursor can occupy. Incrementing the test position by 1, has the same effect as the cursor-right key. Decreasing the test position by 1 has the same effect as the cursor-left key.

If you set LastTextPosition to a value less then the current <u>FirstTextPosition</u>, <u>FirstTextP</u>

HINT: Use text cursor positions with care, since this is a costly operation compared to XMLData based cursor positioning.

Errors

- 2001 The authentic range object, or its related view object is not valid.
- 2005 Invalid address for the return parameter was specified.
- 2006 A text position outside the document was specified.

Examples

'_____ 'Scripting environment - VBScript '_____ Dim objAuthenticView 'we assume that the active document is open in authentic view mode Set objAuthenticView = Application ActiveDocumentAuthenticView

nDocStartPosition = objAuthenticView Docum entBegin FirstTextPosition nDocEndPosition = objAuthenticView Docum entEndFirstTextPosition

```
'lets create a range that selects the whole document
'in an inefficient way
Dim obRange
'we need to get a (any) range object first
Set obRange = obAuthenticView DocumentBegin
obRange FirstTextPosition = nDocStartPosition
obRange LastTextPosition = nDocEndPosition
```

LastXMLData

See also

Property: LastXM LData as XM LData

Description

Set or get the last XM LData element in the underlying document that is partially or completely selected by the range. The exact end of the selection is defined by the LastXM LDataO ffset attribute.

Whenever you set LastXM LData to a new data object, LastXM LDataO ffset gets set to the last cursor position inside this element. Only XM LData objects that have a cursor position may be used. If you set LastXM LData / LastXM LDataO ffset, select a position less then the current FirstXM LData / FirstXM LDataO ffset, the latter gets moved to the new end position.

HINT: You can use the FirstXM LData and LastXM LData properties to directly access and manipulate the underlying XML document in those cases, where the methods available with the AuthenticRange object are not sufficient.

Errors

- 2001 The authentic range object, or its related view object is not valid.
- 2005 Invalid address for the return parameter was specified.
- 2008 Internal error
- 2009 The XM LData object cannot be accessed.

LastXMLDataOffset

See also

Property: LastXM LDataO ffset as Long

Description

Set or get the cursor position inside LastXM LData element for the end of the range.

Offset positions are based on the characters returned by the $\underline{\text{Text}}$ property and start with 0. When setting a new offset, use -1 to set the offset to the last possible position in the element. The following cases require specific attention:

- The textual form of entries in Combo Boxes, Check Boxes and similar controls can be different from what you see on the screen. Although, the data offset is based on this text, there only two valid offset positions, one at the beginning and one at the end of the entry. An attempt to set the offset to somewhere in the middle of the entry, will result in the offset being set to the end.
- The textual form of XML Entities might differ in length from their representation on the screen. The offset is based on this textual form.

If LastXM LData / LastXM LDataO ffset selects a position before FirstXM LData / FirstXM LDataO ffset, the latter gets moved to the new end position.

Errors

- 2001 The authentic range object, or its related view object is not valid.
- 2005 Invalid offset was specified.
 - Invalid address for the return parameter was specified.

Examples

^{&#}x27;Scripting environment - VBScript

^{&#}x27;Select the complete text of an XM LD ata element

^{&#}x27;using XM LD ata based selection and ExpandTo

```
D in ob Authentic View
'we assume that the active document is open in authentic view mode
Set objAuthenticView = Application ActiveDocum entAuthenticView
'first we use the XM LD ata based range properties
'to select all text of the first XM LD ata element
' in the current selection
Dim ob Range
Set objRange = objAuthenticView Selection
ob Range FirstXM LD ataO ffset = 0 'start at beginning of element text
obRangeLastXMLData = obRangeFirstXMLData 'select only one element
obRangeLastXMLDataOffset = -1 'select till its end
'the same can be achieved with the ExpandTo method
Dim ob Range2
Set ob Range2 = ob Authentic View Selection ExpandTo (spyAuthenticTag)
'were we successful?
If ob Range IsEqual(ob Range2) Then
      ob;RangeSelect()
Else
      MsgBox "Ooops"
End If
```

MoveBegin

See also

Method: MoveBegin (ekind as SPYAuthenticElem entKind, nCount as Long) as AuthenticRange

Description

Move the beginning of the range to the beginning of the nCount element of type eKind. Counting starts at the current beginning of the range object.

Use positive numbers for nCount to move towards the document end, use negative numbers to move towards document beginning. The end of the range stays unmoved, unless the new beginning would be larger than it. In this case, the end is moved to the new beginning. The method returns the modified range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2003 Target lies after end of document.
- 2004 Target lies before beginning of document.
- 2005 Invalid element kind specified.
 - Invalid address for the return parameter was specified.

MoveEnd

See also

Method: MoveEnd (eKind as SPYAuthenticElem entKind, nCount as Long) as AuthenticRange

Description

Move the end of the range to the begin of the nCount element of type eKind. Counting starts at the current end of the range object.

Use positive numbers for nCount to move towards the document end, use negative numbers to move towards document beginning. The beginning of the range stays unmoved, unless the new end would be less than it. In this case, the beginning gets moved to the new end. The method

returns the modified range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2003 Target lies after end of document.
- 2004 Target lies before begin of document.
- 2005 Invalid element kind specified. Invalid address for the return parameter was specified.

MoveRowDown

See also

Method: MoveRowDown () as Boolean

Description

If the beginning of the range is inside a dynamic table and selects a row which is not the last row in this table, this method swaps this row with the row immediately below. The selection of the range moves with the row, but does not otherwise change. The function returns *true* if the move operation was successful, otherwise *false*.

Errors

- 2001 The authentic range object or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

MoveRowUp

See also

Method: MoveRowUp() as Boolean

Description

If the beginning of the range is inside a dynamic table and selects a row which is not the first row in this table, this method swaps this row with the row above. The selection of the range moves with the row, but does not change otherwise. The function returns *true* if the move operation was successful, otherwise *false*.

Errors

The authentic range object, or its related view object is no longer valid.Invalid address for the return parameter was specified.

Parent

See also

Property: Parent as AuthenticView (read-only)

Description

Access the view that owns this range object.

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

Paste

See also

Method: Paste () as Boolean

Description

Returns *False* if the copy/ paste buffer is empty, or its content cannot replace the current selection.

Otherwise, deletes the current selection, inserts the content of the copy/ paste buffer, and returns *True*.

Errors

2001 The authentic range object, or its related view object is no longer valid.

2005 Invalid address for the return parameter was specified.

PerformAction

See also

Method: Perform Action (*eAction* as <u>SPYAuthenticActions</u>, *strE.lem entNam e* as String) as Boolean

Description

Perform Action and its related methods, give access to the entry-helper functions of Authentic. This function allows easy and consistent modification of the document content without a need to know exactly where the modification will take place. The beginning of the range object is used to locate the next valid location where the specified action can be performed. If no such location can be found, the method returns *False*. Otherwise, the document gets modified and the range points to the beginning of the modification.

HINT: To find out element names that can be passed as the second parameter use CanPerform ActionWith.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.
- 2007 Invalid action was specified.

Examples

objRangeExpandTo (spyAuthenticTag)CollapsToEnd()Select

```
Else
M sgB ox "Cant insert any elements at current position"
End If
```

Select

See also

Method: Select()

Description

Makes this range the current user interface selection. You can achieve the same result using: 'objRange.Parent.Selection = objRange'

Errors

2001 The authentic range object or its related view object is no longer valid.

Examples

```
'Scripting environment - VBScript
'-----
Dim objAuthenticView
'we assume that the active document is open in authentic view mode
Set objAuthenticView = ApplicationActiveDocumentAuthenticView
```

```
'set current selection to end of docum ent
ob AuthenticView Docum entEnd Select()
```

Select Next

See also

Method: SelectNext (eKind as SPYAuthenticElem entKind) as AuthenticRange

Description

Selects the element of type eKind after the current end of the range. The method returns the modified range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2003 Target lies after end of document.
- 2005 Invalid element kind specified. Invalid address for the return parameter was specified.

Examples

```
'Scripting environment - VBScript
'Scan through the whole document word-by-word
'_______
Dim objAuthenticView
'we assume that the active document is open in authentic view mode
Set objAuthenticView = ApplicationActiveDocumentAuthenticView
```

```
Dim obRange
Set obRange = obAuthenticView Docum entBegin
Dim bEndOfDocum ent
bEndOfDocum ent = False
```

```
On Error Resum e Next
While Not bEndOfDocum ent
```

```
obRange SelectNext(spyAuthenticWord) Select

If (Errnumber - vbObjectenor) = 2003) Then

bEndOfDocument = True

ErrC lear

ElseIf (Errnumber <> 0) Then

ErrRaise 'forward error

End If

Wend
```

Select Previous

See also

Method: GotoPrevious (eKind as SPYAuthenticElem entKind) as AuthenticRange

Description

Selects the element of type eKind before the current beginning of the range. The method returns the modified range object.

Errors

- 2001 The authentic range object, or its related view object is no longer valid.
- 2004 Target lies before begin of document.
- 2005 Invalid element kind specified.
 - Invalid address for the return parameter was specified.

Examples

```
'Scripting environm ent - VBScript
'Scan through the whole docum ent tag-by-tag
```

Dim objAuthenticView

'we assume that the active document is open in authentic view mode Set objAuthenticView = ApplicationActiveDocumentAuthenticView

```
Dim obRange
Set obRange = obAuthenticView Docum entEnd
Dim bEndOfDocum ent
bBeginOfDocum ent = False
```

```
On Error Resume Next
While Not bB eginOfD ocument
obRange SelectPrevious (spyAuthenticTag) Select
If (Errnumber - vbObjecterror) = 2004) Then
bB eginOfD ocument = True
ErrClear
ElseIf (Errnumber <> 0) Then
ErrRaise 'forward error
End If
```

```
Wend
```

Set Element AttributeValue

See also

Method: SetElem entAttributeValue (*strElem entNam e* as String, *strAttributeNam e* as String, *strAttributeValue* as String)

Description

Set the value of the attribute specified in strAttributeName for the element identified with

strElem entName. If the attribute is supported but has no value assigned, the empty string is returned. To find out the names of attributes supported by an element, use GetElem entAttributeNames, Or HasElem entAttribute.

Errors

2001 The authentic range object or its related view object is no longer valid.

2005 Invalid element name was specified. Invalid attribute name was specified. Invalid attribute value was specified.

Examples

'Scripting env 'Get and set	vironment – VBScript element attributes
Dim objRange 'we assume t Set objRange	hat the active docum ent is open in authentic view mode = Application ActiveDocum entAuthenticView Selection
'finstwe find Dim annEleme objRangeGetf	out all the elements below the beginning of the range nts I lementH ierarchy and lements
If ISA may (ame If UB or	<pre>Slem ents) Then und (an Elem ents) >= 0 Then 'we use the top level elem ent and find out its valid attributes D in an Attrs 0 ob Range GetElem ent AttributeN am es an Elem ents (0), an Attrs</pre>
stAttīVal	<pre>If UB ound (anA ttrs) >= 0 Then 'we retrieve the current value of the first valid attribute D in stAttrVal stAttrVal= obRangeGetElementAttributeValue (anElements(0), anAttrs(0)) m sgbox "current value of " & anElements(0) & "//" & anAttrs(0) & " is:" &</pre>
	'we change this value and read it again stAtuVal = "Helb" obRangeSetElementAturbuteValue an Elements(0), an Atus(0), stAtuVal stAtuVal = obRangeGetElementAturbuteValue (an Elements(0), an Atus(0)) m sgbox "new value of " & an Elements(0) & "//" & an Atus(0) & " is: " &
strAttrVal End If End If	End If

SetFromRange

See also

Method: SetFrom Range (objSrcRange as AuthenticRange)

Description

Sets the range object to the same beginning and end positions as objarcRange.

2001	One of the two range objects, is invalid.
2005	Null object was specified as source object.

Set Variable Value

Method: SetVariableValue(strName as string, strValue as string)

Return Value

Sets the value (second parameter) of the variable named in the first parameter.

Errors

- 2201 Invalid object.
- 2202 No context.
- 2204 No such variable in scope
- 2205 Variable cannot be evaluated
- 2206 Variable returns sequence
- 2207 Variable read-only
- 2208 No modification allowed

Text

See also

Property: Text as String

Description

Set or get the textual content selected by the range object.

The number of characters retrieved are not necessarily identical, as there are text cursor positions between the beginning and end of the selected range. Most document elements support an end cursor position different to the beginning cursor position of the following element. Drop- down lists maintain only one cursor position, but can select strings of any length. In the case of radio buttons and check boxes, the text property value holds the string of the corresponding XML element.

If the range selects more then one element, the text is the concatenation of the single texts. XML entities are expanded so that '&' is expected as '&'.

Setting the text to the empty string, does not delete any XML elements. Use <u>Cut</u>, <u>Delete</u> or <u>Perform Action</u> instead.

- 2001 The authentic range object or its related view object is no longer valid.
- 2005 Invalid address for a return parameter was specified.

3.2.6 AuthenticView

See also

Properties
Application
A SXM LS tring
Docum entBegin
Docum entEnd
Event
MarkupV isibility
Parent
Selection
XM LDataRoot
WholeDocument

Methods Goto IsRedoEnabled IsUndoEnabled Print Redo Undo UpdateXMLInstanceEntities

Events

OnBeforeCopy OnBeforeCut OnBeforeDelete OnBeforeDrop OnBeforePaste OnDragOver OnKeyBoardEvent OnMouseEvent OnSelectionChanged

Description

AuthenticView and its child objects <u>AuthenticRange</u> and <u>AuthenticDataTransfer</u> provide you with an interface for Authentic View, which allow easy and consistent modification of document contents. These interfaces replace the following interfaces which are marked now as **obsolete**:

O blauthenticView (old name was DocEditView) AuthenticSelection (old name was DocEditSelection, superseded by <u>AuthenticRange</u>) AuthenticEvent (old name was DocEditEvent)

AuthenticView gives you easy access to specific features such as printing, the multi-level undo buffer, and the current cursor selection, or position.

AuthenticView uses objects of type <u>AuthenticRange</u> to make navigation inside the document straight-forward, and to allow for the flexible selection of logical text elements. Use the properties <u>DocumentBegin</u>, <u>DocumentEnd</u>, or <u>WholeDocument</u> for simple selections, while using the <u>Goto</u> method for more complex selections. To navigate relative to a given document range, see the methods and properties of the <u>AuthenticRange</u> object.

Events

OnBeforeCopy

See also

Event: OnBeforeCopy() as Boolean

Scripting environment - VBScript:

```
Function On AuthenticBeforeCopy()
'On AuthenticBeforeCopy = False 'to disable operation
End Function
```

Scripting environment - JScript:

IDE Plugin:

KM LSpyP ligh OnEvent (21, ...) // nEventId = 21

Description

This event gets triggered before a copy operation gets performed on the document. Return *True* (or nothing) to allow copy operation. Return *False* to disable copying.

OnBeforeCut

See also

Event: OnBeforeCut() as Boolean

Scripting environment - VBScript: Function On AuthenticBeforeCut()

```
'On AuthenticBeforeCut = False 'to disable operation
End Function
```

Scripting environment - JScript:

```
function On AuthenticBeforeCut()
{
     // return false; /* to disable operation */
}
```

IDE Plugin:

KM LSpyP lugIn OnEvent (20, ...)// nEventId = 20

Description

This event gets triggered before a cut operation gets performed on the document. Return *True* (or nothing) to allow cut operation. Return *False* to disable operation.

OnBeforeDelete

See also

Event: OnBeforeDelete () as Boolean

Scripting environment - VBScript:

```
Function On AuthenticBeforeDelete ()
'On AuthenticBeforeDelete = False 'to disable operation
End Function
```

Scripting environment - JScript:

```
function On_AuthenticBeforeDelete ()
{
     // return false; /* to disable operation */
}
```

IDE Plugin:

KM LSpyP lugh OnEvent (22, ...)// nEventId = 22

Description

This event gets triggered before a delete operation gets performed on the document. Return *True* (or nothing) to allow delete operation. Return *False* to disable operation.

OnBeforeDrop

See also

Event: OnBeforeDrop (<u>inXPos</u> as Long, <u>inYPos</u> as Long, <u>ipRange</u> as <u>AuthenticRange</u>, <u>ipData</u> as cance Boolean

Scripting environment - VBScript:

```
Function On AuthenticBeforeDrop (nXPos, nYPos, obRange, obData)
'On AuthenticBeforeDrop = False 'to disable operation
End Function
```

Scripting environment - JScript:

```
function On_AuthenticBeforeDrop(nXPos, nYPos, obfange, obf)ata)
{
     // return false; /* to disable operation */
}
```

IDE Plugin:

KM LSpyP lugIn OnEvent (11, ...)// nEventId = 11

Description

This event gets triggered whenever a previously dragged object gets dropped inside the application window. All event related information gets passed as parameters.

The first two parameters specify the mouse position at the time when the event occurred. The parameter ob_{Range} passes a range object that selects the XML element below the mouse position. The value of this parameter might be *NULL*. Be sure to check before you access the range object. The parameter ob_{Pata} allows to access information about the object being dragged.

Return False to cancle the drop operation. Return True (or nothing) to continue normal operation.

OnBeforePaste

See also

Event: OnBeforePaste (*obpata* as Variant, *strType* as String) as Boolean

Scripting environment - VBScript:

```
Function On AuthenticBeforePaste (obpata, strType)
'On AuthenticBeforePaste = False 'to disable operation
End Function
```

Scripting environment - JScript:

```
function On AuthenticBeforePaste (objData, strType)
{
     // return false; /* to disable operation */
}
```

IDE Plugin:

KMLSpyPlughOnEvent(19, ...) // nEventId = 19

Description

This event gets triggered before a paste operation gets performed on the document. The parameter strType is one of "TEXT", "UNICODETEXT" or "IUNKNOWN". In the first two cases obpata contains a string representation of the object that will be pasted. In the later case, obpata contains a pointer to an IUnknown COM interface.

Return True (or nothing) to allow paste operation. Return False to disable operation.

OnBeforeSave

Event: OnBeforeSave (SaveAs flag) as Boolean

Description: OnBeforeSave gives the opportunity to e.g. warn the user about overwriting the existing XML document, or to make the document read-only when specific circumstances are not met. The event will be fired before the file dialog is shown.

OnDragOver

See also

Event: OnDragOver (*nXPos* as Long, *nYPos* as Long, *eMouseEvent* as <u>SPYMouseEvent</u>, *obRange* as AuthenticBataTransfer) as Boolean

Scripting environment - VBScript:

```
Function On AuthenticDragOver(nXPos, nYPos, eMouseEvent, obRange, obData)
'On AuthenticDragOver = False 'to disable operation
End Function
```

Scripting environment - JScript:

```
function On_AuthenticDragOver(nXPos, nYPos, eMouseEvent, obRange, obData)
{
    // return false; /* to disable operation */
}
```

IDE Plugin:

KM LSpyP lugh OnEvent (10, ...)// nEventId = 10

Description

This event gets triggered whenever an object from within or outside of Authentic View gets dragged with the mouse over the application window. All event related information gets passed as parameters.

The first three parameters specify the mouse position, the mouse button status and the status of the virtual keys at the time when the event occurred. The parameter *objRange* passes a range object that selects the XML element below the mouse position. The value of this parameter might be *NULL*. Be sure to check before you access the range object. The parameter *objData* allows to access information about the object being dragged.

Return False to cancel the drag operation. Return True (or nothing) to continue normal operation.

OnKeyboardEvent

See also

Event: OnKeyboardEvent (*eKeyEvent* as <u>SPYKeyEvent</u>, *nKeyCode* as Long, *nVirtualKeyStatus* as Long) as Boolean

Scripting environment - VBScript:

```
Function On AuthenticKeyboardEvent(eKeyEvent, nKeyCode, nVirtualKeyStatus)
'On AuthenticKeyboardEvent = True 'to cancelbubbling of event
End Function
```

Scripting environment - JScript:

```
function On_AuthenticKeyboardEvent(eKeyEvent, nKeyCode, nVirtualKeyStatus)
{
     // return true; /* to cancelbubbling of event */
}
```

IDE Plugin:

KM LSpyP ligh OnEvent (30, ...)// nEventId = 30

Description

This event gets triggered for WM_KEYDOWN, WM_KEYUP and WM_CHAR Windows messages.

The actual message type is available in the *eKeyEvent* parameter. The status of virtual keys is combined in the parameter *nVirtualKeyStatus*. Use the bit-masks defined in the enumeration datatype <u>SPYVirtualKeyMask</u>, to test for the different keys or their combinations.

OnLoad

Event: OnLoad()

Description: OnLoad can be used e.g. to restrict some AuthenticView functionality, as shown in the example below:

```
function On AuthenticLoad()
{
    // We are disabling all entry helpers in order to prevent user from manipulating XML tree
    AuthenticV iew D isableElem entEntryHelper();
    AuthenticV iew D isableA tributeEntryHelper();
    // We are also disabling the markup buttons for the same purpose
    AuthenticV iew SetToolbarB uttonState ( AuthenticM arkupSm all, authenticToolbarB uttonD isabled );
    AuthenticV iew SetToolbarB uttonState ( AuthenticM arkupLarge , authenticToolbarB uttonD isabled );
    AuthenticV iew SetToolbarB uttonState ( AuthenticM arkupLarge , authenticToolbarB uttonD isabled );
    AuthenticV iew SetToolbarB uttonState ( AuthenticM arkupM ixed , authenticToolbarB uttonD isabled );
}
```

In the example the status of the Markup Small, Markup Large, Markup Mixed toolbar buttons are manipulated with the help of button identifiers. See complete list.

OnMouseEvent

See also

Event: OnMouseEvent (*nXPos* as Long, *nYPos* as Long, *eMouseEvent* as <u>SPYMouseEvent</u>, *obgRange* as AuthenticRange) as Boolean

Scripting environment - VBScript:

Scripting environment - JScript:

```
function On AuthenticM ouseEvent(nXPos, nYPos, eM ouseEvent, obfRange)
{
     // return true; /* to cancelbubbling of event */
}
```

IDE Plugin:

KM LSpyP ligh O nEvent (31, ...)// nEventId = 31

Description

This event gets triggered for every mouse movement and mouse button Windows message.

The actual message type and the mouse buttons status, is available in the *eMouseEvent* parameter. Use the bit-masks defined in the enumeration datatype <u>SPYMouseEvent</u> to test for the

different messages, button status, and their combinations.

The parameter *objRange* identifies the part of the document found at the current mouse cursor position. The range objects always selects a complete tag of the document. (This might change in future versions, when a more precise positioning mechanism becomes available). If no selectable part of the document is found at the current position, the range object is *null*.

OnSelectionChanged

See also

Event: OnSelectionChanged (obNewSelection as AuthenticRange)

Scripting environment - VBScript:

Function On AuthenticSelectionChanged (objNewSelection) End Function

Scripting environment - JScript:

```
function On_AuthenticSelectionChanged (obNewSelection)
{
}
```

IDE Plugin:

KM LSpyP ligh O nEvent (23, ...)// nEventId = 23

Description

This event gets triggered whenever the selection in the user interface changes.

OnToolbarButtonClicked

Event: OnToolbarButtonClicked (Button identifier)

Description: OnToolbarButtonClicked is fired when a toolbar button was clicked by user. The parameter button identifier helps to determine which button was clicked. The list of predefined button identifiers is below:

- AuthenticPrint
- AuthenticPrintPreview
- AuthenticUndo
- AuthenticRedo
- AuthenticCut
- AuthenticCopy
- AuthenticPaste
- AuthenticC lear
- AuthenticMarkupHide
- AuthenticMarkupLarge
- AuthenticMarkupMixed
- AuthenticMarkupSmall
- AuthenticValidate
- AuthenticChangeWorkingDBXMLCell
- AuthenticSave
- AuthenticSaveAs
- AuthenticRebad
- AuthenticTableInsertRow
- AuthenticTableAppendRow
- AuthenticTableDeleteRow
- AuthenticTableInsertCol
- AuthenticTableAppendCol
- AuthenticTableDeleteCol
- AuthenticTableJoinCelRight
- AuthenticTableJoinCellLeft
- AuthenticTableJoinCellAbove
- AuthenticTableJoinCelBebw
- AuthenticTableSplitCelHorizontally
- AuthenticTableSplitCellVertically
- AuthenticTableAlignCellContentTop
- AuthenticTableCenterCelVertically
- AuthenticTableAlignCellContentBottom
- AuthenticTableAlignCellContentLeft
- AuthenticTableCenterCellContent
- AuthenticTableAlignCellContentRight
- AuthenticTableJustifyCellContent
- AuthenticTableInsertTable
- AuthenticTableDeleteTable
- AuthenticTableProperties
- AuthenticAppendRow
- Authentic InsertRow
- AuthenticDuplicateRow
- AuthenticMoveRowUp
- AuthenticMoveRowDown
- AuthenticDeleteRow
- AuthenticDefineEntities
- AuthenticXMLSignature

For custom buttons the user might add his own identifiers. Please, note that the user must take care, as the identifiers are not checked for uniqueness. The same identifiers can be used to identify buttons in the <code>Set/GetToolbarState()</code> COM API calls. By adding code for different buttons, the user is in the position to completely redefine the AuthenticView toolbar behavior, adding own methods for table manipulation, etc.

OnToolbarButtonExecuted

Event: OnToolbarButtonExecuted (Button identifier)

Description: OnToolbarButtonClicked is fired when a toolbar button was clicked by user. The parameter button identifier helps to determine which button was clicked. See the list of predefined button identifiers.

OnToolbarButtonExecuted is fired after the toolbar action was executed. It is useful e.g. to add update code, as shown in the example below:

```
//event fired when a toolbarbutton action was executed
function On AuthenticToolbarButtonExecuted(varBtnBentifier)
{
    // After whatever comm and user has executed - make sure to update toolbarbutton states
    UpdateOwnToolbarButtonStates();
}
```

In this case <code>UpdateOwnToolbarButtonStates</code> is a user function defined in the Global Declarations.

OnUserAddedXMLNode

Event: OnUserAddedXMLNode (XML node)

Description: OnUserAddedXMLNode will be fired when the user adds an XML node as a primary action. This happens in the situations, where the user clicks on

- auto-add hyperlinks (see example OnUserAddedXMLNode.sps)
- the Insert..., Insert After..., Insert Before... context menu items
- Append row, Insert row toolbar buttons
- Insert After..., Insert Before... actions in element entry helper (outside StyleVision)

The event doesn't get fired on Duplicate row, or when the node was added externally (e.g. via COM API), or on Apply (e.g. Text State Icons), or when in XML table operations or in DB operations.

The event parameter is the XML node object, which was added giving the user an opportunity to manipulate the XML node added. An elaborate example for an event handler can be found in the <code>OnUserAddedXMLNode.sps</code> file.

Application

See also

Property: Application as Application (read-only)

Description

Accesses the Authentic Desktop application object.

Errors

- 2000 The authentic view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

AsXMLString

See also

Property: AsXMLString as String

Description

Returns or sets the document content as an XML string. Setting the content to a new value does not change the schema file or sps file in use. If the new XMLString does not match the actual schema file error 2011 gets returned.

Errors

- 2000 The authentic view object is no longer valid.
- 2011 AsXMLString was set to a value which is no valid XML for the current schema file.

Context Menu

Property: ContextMenu () as ContextMenu

Description

The property ContextMenu gives access to customize the context menu. The best place to do it is in the event handler OnContextMenuActivated.

Errors

2000 Invalid object.2005 Invalid parameter.

CreateXMLNode

Method: CreateXM LNode (nKind as SPYXM LDataKind) as XM LData

Return Value

The method returns the new XM LData object.

Description

To create a new XM LData object use the C reateXM LNode () method.

Errors

2000 Invalid object.2012 Cannot create XML node.

DisableAttributeEntryHelper

Method: DisableAttributeEntryHelper()

Description

DisableAttributeEntryHelper() disables the attribute entry helper in XMLSpy, Authentic Desktop and Authentic Browser plug-in.

Errors

2000 Invalid object.

DisableElementEntryHelper

Method: DisableElementEntryHelper()

Description

DisableElementEntryHelper() disables the element entry helper in XMLSpy, Authentic Desktop and Authentic Browser plug-in.

Errors

2000 Invalid object.

DisableEntityEntryHelper

Method: DisableEntityEntryHelper()

Description

DisableEntityEntryHelper() disables the entity entry helper in XMLSpy, Authentic Desktop and Authentic Browser plug-in.

2000 Invalid object.

Document Begin

See also

Property: Docum entBegin as AuthenticRange (read-only)

Description

Retrieve a range object that points to the beginning of the document.

Errors

- 2000 The authentic view object is no longer valid.2005 Invalid address for the return parameter was specified.
- **Document End**

See also

Property: Docum entEnd as AuthenticRange (read-only)

Description

Retrieve a range object that points to the end of the document.

Errors

2000 The authentic view object is no longer valid.2005 Invalid address for the return parameter was specified.

DoNot PerformStandardAction

Method: DoNotPerform StandardAction ()

Description

DoNotPerformStandardAction() serves as cancel bubble for macros, and stops further execution after macro has finished.

Errors

2000 Invalid object.

EvaluateXPath

Method: EvaluateXPath (XMLData as XMLData, strExpression as string) strValue as string

Return Value

The method returns a string

Description

EvaluateXPath() executes an XPath expressions with the given XML context node. The result is returned as string, in the case of a sequence it is a space-separated string.

Errors

2000 Invalid object.2005 Invalid parameter.2008 Internal error.

2013 XPath error.

Event

See also

Property: Event as AuthenticEvent (read- only)

Description

This property gives access to parameters of the last event in the same way as O bAuthenticView event does. Since all events for the scripting environment and external clients are now available with parameters this Event property should only be used from within IDE-Plugins.

Errors

2000 The authentic view object is no longer valid.2005 Invalid address for the return parameter was specified.

Event Context

Property: EventContext) as EventContext

Description

EventContext property gives access to the running macros context. See the <u>EventContext</u> interface description for more details.

Errors

2000 Invalid object.

GetToolbarButtonState

Method: GetToolbarButtonState (ButtonIdentifier as string) as AuthenticToolbarButtonState

Return Value

The method returns AuthenticToolbarButtonState

Description

Get/SetToolbarButtonState queries the status of a toolbar button, and lets the user disable or enable the button, identified via its button identifier (see list above). One usage is to disable toolbar buttons permanently. Another usage is to put SetToolbarButtonState in the OnSelectionChanged event handler, as toolbar buttons are updated regularly when the selection changes in the document.

Toolbar button states are given by the listed enumerations.

The default state means that the enable/disable of the button is governed by AuthenticView. When the user sets the button state to enable or disable, the button remains in that state as long as the user does not change it.

Errors 2000 Invalid object.

- 2005 Invalid parameter.
- 2008 Internal error.
- 2014 Invalid button identifier.

Goto

See also

Method: Goto (eKind as <u>SPYAuthenticElementKind</u>, nCount as Long, eFrom as SPYAuthenticDocumentPosition) as AuthenticRange

Description

Retrieve a range object that points to the beginning of the nCount element of type eKind The start position is defined by the parameter eFrom. Use positive values for nCount to navigate to the document end. Use negative values to navigate towards the beginning of the document.

Errors

- 2000 The authentic view object is no longer valid.
- 2003 Target lies after end of document.
- 2004 Target lies before beginning of document.
- 2005 Invalid element kind specified. The document position to start from is not one of *spyAuthenticDocumentBegin* or *spyAuthenticDocumentEnd*. Invalid address for the return parameter was specified.

Examples

'Scripting environment - VBScript

Dim objAuthenticView 'we assume that the active document is open in authentic view mode Set objAuthenticView = ApplicationActiveDocumentAuthenticView

IsRedoEnabled

See also

Property: BRedoEnabled as Boolean (read-only)

Description

True if redo steps are available and Redo is possible.

Errors

2000 The authentic view object is no longer valid.2005 Invalid address for the return parameter was specified.

IsUndoEnabled

See also

Property: IsUndoEnabled as Boolean (read-only)

Description

True if undo steps are available and Undo is possible.

Errors

2000 The authentic view object is no longer valid.2005 Invalid address for the return parameter was specified.

MarkupVisibility

See also

Property: MarkupV isibility as SPYAuthenticM arkupV isibility

Description

Set or get current visibility of markup.

Errors

2000	The authentic view object is no longer valid.
2005	Invalid enumeration value was specified.
	Invalid address for the return parameter was specified.

Parent

See also

Property: Parent as Document (read-only)

Description

Access the document shown in this view.

Errors

2000	The authentic view object is no longer valid.
2005	Invalid address for the return parameter was specified.

Print

See also

Method: Print (bW ith Preview as Boolean, bProm ptUser as Boolean)

Description

Print the document shown in this view. If *bWthPreview* is set to *True*, the print preview dialog pops up. If *bPromptUser* is set to *True*, the print dialog pops up. If both parameters are set to *False*, the document gets printed without further user interaction.

Errors

2000 The authentic view object is no longer valid.

Redo

See also

Method: Redo () as Boolean

Description

Redo the modification undone by the last undo command.

Errors

2000	The authentic view object is no longer valid.
2005	Invalid address for the return parameter was specified.

Selection

See also

Property: Selection as AuthenticRange

Description

Set or get current text selection in user interface.

Errors

- 2000 The authentic view object is no longer valid.
- 2002 No cursor selection is active.
- 2005 Invalid address for the return parameter was specified.

Examples

```
'Scripting environm ent - VBScript
```

```
Dim ob Authentic View
```

'we assume that the active document is open in authentic view mode Set objAuthenticView = Application ActiveDocumentAuthenticView

'if we are the end of the docum ent, re-start at the beginning

```
If (objAuthenticView Selection JsEqual(objAuthenticView DocumentEnd)) Then
```

objAuthenticView Selection = objAuthenticView Docum entBegin

```
Else
```

```
'objAuthenticView Selection = objAuthenticView Selection GotoNextCursorPosition ()
'or shorter:
```

objAuthenticView SelectionGotoNextCursorPosition()Select

```
{\rm End}\ {\rm I}\!{\rm f}
```

SetToolbarButtonState

Method: SetToobarButtonState (ButtonIdentifier as string, AuthenticToolbarButtonState state)

Description

Get/SetToolbarButtonState queries the status of a toolbar button, and lets the user disable or enable the button, identified via its button identifier (see list above). One usage is to disable toolbar buttons permanently. Another usage is to put SetToolbarButtonState in the OnSelectionChanged event handler, as toolbar buttons are updated regularly when the selection changes in the document.

Toolbar button states are given by the listed enumerations.

The default state means that the enable/disable of the button is governed by AuthenticView. When the user sets the button state to enable or disable, the button remains in that state as long as the user does not change it.

Errors

2000 Invalid object.2008 Internal error.2014 Invalid button identifier.

Undo

See also

Method: Undo () as Boolean

Description

Undo the last modification of the document from within this view.

Errors

2000	The authentic view object is no longer valid.
2005	Invalid address for the return parameter was specified.

UpdateXMLInstanceEntities

See also

Method: UpdateXM LInstanceEntities ()

Description

Updates the internal representation of the declared entities, and refills the entry helper. In addition, the validator is reloaded, allowing the XML file to validate correctly. Please note that this may also cause schema files to be reloaded.

Errors

The method never returns an error.

Example

```
// ------
// Scripting environm ent - JavaScript
// ------
if(Application A ctiveD ocum ent & & (Application A ctiveD ocum entC unentV iew M ode == 4))
{
    var objOcType;
    objOcType = Application A ctiveD ocum entD ocEditV iew XM LR ootG etF irstC hild (10);
    if(objOcType)
    {
        var objEntity = Application A ctiveD ocum entC reateC hild (14);
        objEntityNam e = "child";
        objEntityTextValue = "SYSTEM \"childxm l\"";
        objOcType AppendC hild (objEntity);
    }
}
```

Application ActiveDocum entAuthenticView UpdateXMLInstanceEntities ();

}

WholeDocument

See also

Property: WholeDocument as AuthenticRange (read-only)

Description

Retrieve a range object that selects the whole document.

Errors

- 2000 The authentic view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

XMLDataRoot

See also

Property: XM LDataRoot as XM LData (read-only)

Description

Returns or sets the top-level XMLData element of the current document. This element typically describes the document structure and would be of kind spyXMLDataXMLDocStruct, spyXMLDataXMLEntityDocStruct or spyXMLDataDTDDocStruct..

- 2000 The authentic view object is no longer valid.
- 2005 Invalid address for the return parameter was specified.

3.2.7 CodeGeneratorDlg

See also

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Properties and Methods

Standard automation properties Application Parent

Programming language selection properties ProgrammingLanguage TemplateFileName

Settings for C++ code

CPPSettings DOM Type CPPSettings LibraryType CPPSettings UseM FC CPPSettings GenerateVC 6ProjectFile CPPSettings GenerateVSProjectFile

Settings for C# code CSharpSettings ProjectType

Dialog handling for above code generation properties PropertySheetDialogAction

Output path selection properties OutputPath OutputPathDiabgAction

Presentation of result OutputResultDiabgAction

Description

Use this object to configure the generation of program code for schema files. The method <u>GenerateProgram Code</u> expects a CodeGeneratorDlg as parameter to configure code generation as well as the associated user interactions.

Application

See also

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Property: Application as Application (read-only)

Description

Access the Authentic Desktop application object.

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

CPPSettings_DOMType

Property: CPPSettings DOM Type as SPYDOM Type

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Defines one of the settings that configure generation of C++ code.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

CPPSettings_GenerateVC6ProjectFile

Property: CPPSettings GenerateVC 6ProjectFile as Boolean

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Defines one of the settings that configure generation of C++ code.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

CPPSettings_GenerateGCCMakefile

Property: CPPSettings GenerateGCCM akefile as Boolean

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Creates makefiles to compile the generated code under Linux with GCC.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

CPPSettings_GenerateVSProjectFile

Property: CShapSettings_GenerateVSProjectFile as <u>SPYProjectType</u> Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Defines one of the settings that configure generation of C++ code. Only

```
spyVisualStudio2005Project (=4) and spyVisualStudio2008Project (=5) and
spyVisualStudio2010Project (=6) are valid project types.
```

Errors

2200 The object is no longer valid.

2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

CPPSettings_LibraryType

Property: CPPSettings LibraryType as SPYLibType

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Defines one of the settings that configure generation of C++ code.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

CPPSettings_UseMFC

Property: CPPSettings UseMFC as Boolean

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Defines one of the settings that configure generation of C++ code.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

CSharpSettings_ProjectType

Property: CShapSettings ProjectType as SPYProjectType

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Defines the only setting to configure generation of C# code.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

Output Path

Property: OutputPath as String

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other

version.

Description

Selects the base directory for all generated code.

Errors

2200 The object is no longer valid.

2201 Invalid address for the return parameter was specified.

Output PathDialogAction

Property: OutputPathDiabgAction as SPYDiabgAction

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Defines how the sub-dialog for selecting the code generation output path gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog with the current value of the <u>OutputPath</u> property as default. Use *spyDialogOK(0)* to hide the dialog from the user.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

Output Result DialogAction

Property: OutputResultDiabgAction as SPYDiabgAction

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Defines how the sub-dialog that asks to show the result of the code generation process gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog. Use *spyDialogOK(0)* to hide the dialog from the user.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

Parent

See also

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Property: Parent as Diabgs (read-only)

Description

Access the parent of the object.

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

ProgrammingLanguage

Property: Program m ingLanguage as Program m ingLanguage

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Selects the output language for the code to be generated.

CAUTION: Setting this property to one of C++, C# or Java, changes the property <u>TemplateFileName</u> to the appropriate template file delivered with Authentic Desktop as well. If you want to generate C ++, C# or Java code based on your own templates, set first the programming language and then select your template file.

Errors

2200 The object is no longer valid.

2201 Invalid address for the return parameter was specified.

PropertySheetDialogAction

Property: PropertySheetDiabgAction as SPYDiabgAction

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Defines how the sub-dialog that configures the code generation process gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog with the current values as defaults. Use *spyDialogOK(0)* to hide the dialog from the user.

Errors

2200 The object is no longer valid.

2201 Invalid action passed as parameter or an invalid address was specified for the return parameter.

TemplateFileName

Property: Tem plateFileName as String

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Selects the code generation template file. Authentic Desktop comes with template files for C++, C# or Java in the SPL folder of your installation directory.

Setting this property to one of the code generation template files of your Authentic Desktop installation automatically sets the <u>ProgrammingLanguage</u> property to its appropriate value.

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

3.2.8 DatabaseConnection

See also

Properties for import and export

File or ADOConnection or ODBCConnection

Properties for import only

DatabaseKind SQLSelect AsAttributes ExcludeKeys hcludeEmptyElements NumberDateTimeFormat NulReplacement Commenthcluded

Properties for export only

C reateM issingTables C reateNew TextFiebLen DatabaseSchem a

Properties for XML Schema from DB Structure generation

Prim aryKeys ForeignKeys UniqueKeys Schem aExtensionType Schem aForm at In portColum nsType

Description

DatabaseConnection specifies the parameters for the database connection.

Please note that the properties of the DatabaseConnection interface are referring to the settings of the import and export dialogs of Authentic Desktop.

ADOConnection

See also

Property: ADOConnection as String

Description

The property ADOConnection contains a connection string. Ether use this property or ODBCConnection or File to refer to a database.

Errors

No error codes are returned.

Example

Dim objSpyConn As DatabaseConnection Set objSpyConn = objSpyGetDatabaseSettings

Dim objADO As DataLinks Set objADO = CreateObject("DataLinks")

```
If Not (obADO Is Nothing) Then
Dim objConn As Connection
Set objConn = objADO PromptNew
objBpyConn ADOConnection = objConn ConnectionString
End If
```

AsAttributes

See also

Property: AsAttributes as Boolean

Description

Set AsAttributes to true if you want to initialize all import fields to be imported as attributes. Default is false and will initialize all fields to be imported as elements. This property is used only in calls to Application GetDatabaseImportElementList.

Errors

No error codes are returned.

CommentIncluded

See also

Property: Comment Included as Boolean

Description

This property tells whether additional comments are added to the generated XML. Default is true. This property is used only when importing from databases.

Errors

No error codes are returned.

CreateMissingTables

See also

Property: C reateM issingTables as Boolean

Description

If CreateM issingTables is true, tables which are not already defined in the export database will be created during export. Default is true. This property is used only when exporting to databases.

Errors

No error codes are returned.

CreateNew

See also

Property: C reateNew as Boolean

Description

Set CreateNew true if you want to create a new database on export. Any existing database will be overwritten. See also <u>DatabaseConnection File</u>. Default is false. This property is used only when exporting to databases.

Errors

No error codes are returned.

DatabaseKind

See also

Property: DatabaseKind as SPYDatabaseKind

Description

Select the kind of database that gets access. The default value is spyDB Unspecified(7) and is sufficient in most cases. This property is used only when importing from databases.

Errors

No error codes are returned.

DatabaseSchema

See also

Property: DatabaseSchema as String

Description

This property specifies the Schema used for export in Schema aware databases. Default is "". This property is used only when exporting to databases.

Errors

No error codes are returned.

ExcludeKeys

See also

Property: ExcludeKeys as Boolean

Description

Set ExcludeKeys to true if you want to exclude all key columns from the import data. Default is false. This property is used only when importing from databases.

Errors

No error codes are returned.

File

See also

Property: File as String

Description

The property File sets the path for the database during export or import. This property can only be used in conjunction with a Microsoft Access database. Either use this property or <u>ODBCConnection</u> or <u>ADOConnection</u> to refer to the database.

Errors

No error codes are returned.

ForeignKeys

See also

Property: ForeignKeys as Boolean

Description

Specifies whether the Foreign Keys constraint is created or not. Default is true. This property is used only when creating a XML Schema from a DB structure.

Errors

No error codes are returned.

Import ColumnsType

See also

Property: In portColumnsType as <u>SPYImportColumnsType</u>

Description

Defines if column information from the DB is saved as element or attribute in the XML Schema. Default is as element. This property is used only when creating a XML Schema from a DB structure.

Errors

No error codes are returned.

IncludeEmptyElements

See also

Property: IncludeEm ptyElem ents as Boolean

Description

Set IncludeEmptyElements to false if you want to exclude all empty elements. Default is true. This property is used only when importing from databases.

Errors

No error codes are returned.

NullReplacement

See also

Property: NullReplacement as String

Description

This property contains the text value that is used during import for empty elements (null values). Default is "". This property is used only when importing from databases.

Errors

No error codes are returned.

NumberDateTimeFormat

See also

Property: Num berDateT in eForm at as SPYNum berDateT in eForm at

Description

The property NumberDateTimeFormat sets the format of numbers and date- and time-values. Default is spySystem Locale. This property is used only when importing from databases.

Errors

No error codes are returned.

ODBCConnection

See also

Property: ODBCConnection as String

Description

The property ODBCConnection contains a ODBC connection string. Either use this property or ADOC onnection or File to refer to a database.

Errors

No error codes are returned.

PrimaryKeys

See also

Property: PrimaryKeys as Boolean

Description

Specifies whether the Primary Keys constraint is created or not. Default is true. This property is used only when creating a XML Schema from a DB structure.

Errors

No error codes are returned.

SchemaExtensionType

See also

Property: SchemaExtensionType as <u>SPYSchemaExtensionType</u>

Description

Defines the Schema extension type used during the Schema generation. This property is used only when creating a XML Schema from a DB structure.

Errors

No error codes are returned.

SchemaFormat

See also

Property: SchemaFormat as SPYSchemaFormat

Description

Defines the Schema format used during the Schema generation. This property is used only when creating a XML Schema from a DB structure.

Errors

No error codes are returned.

SQLSelect

See also

Property: SQLSelect as String

Description

The SQL query for the import is stored in the property SQLSelect. This property is used only when importing from databases.

Errors

No error codes are returned.

Text FieldLen

See also

Property: TextFieldLen as long

Description

The property TextFieldLen sets the length for created text fields during the export. Default is 255. This property is used only when exporting to databases.

Errors

No error codes are returned.

UniqueKeys

See also

Property: UniqueKeys as Boolean

Description

Specifies whether the Unique Keys constraint is created or not. Default is true. This property is used only when creating a XML Schema from a DB structure.

Errors

No error codes are returned.

3.2.9 Dialogs

See also

Properties and Methods

Standard automation properties Application Parent

Various dialog objects

CodeGeneratorDb FileSelectionDb Schem aDocum entationDb GenerateSam pleXMLDb DTDSchem aGeneratorDb FindhFilesDb

Description

The Dialogs object provides access to different built-in dialogs of Authentic Desktop. These dialog objects allow to initialize the fields of user dialogs before they get presented to the user or allow to simulate complete user input by your program.

Application

See also

Property: Application as Application (read-only)

Description

Access the Authentic Desktop application object.

Errors

2300 The object is no longer valid.

2301 Invalid address for the return parameter was specified.

CodeGeneratorDlg

See also

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Property: CodeGeneratorD b as CodeGeneratorD b (read-only)

Description

Get a new instance of a code generation dialog object. You will need this object to pass the necessary parameters to the code generation methods. Initial values are taken from last usage of the code generation dialog.

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

FileSelectionDlg

See also

Property: FileSelectionDlg as FileSelectionDlg (read-only)

Description

Get a new instance of a file selection dialog object.

File selection dialog objects are passed to you with the some events that signal opening or saving of documents and projects.

Errors

The Dialogs object or one of its parents is no longer valid.Invalid address for the return parameter was specified.

Parent

See also

Property: Parent as Application (read-only)

Description

Access the Authentic Desktop application object.

Errors

2300 The object is no longer valid.2301 Invalid address for the return parameter was specified.

SchemaDocumentationDlg

See also

Property: Schem aDocum entationD b as Schem aDocum entationD b (read-only)

Description

Get a new instance of a dialog object that parameterizes generation of schema documentation. See Document.GenerateSchemaDocumentation for its usage.

Errors

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

GenerateSampleXMLDIg

See also

Property: GenerateSam pleXMLDb as GenerateSam pleXMLDb (read-only)

Description

Get a new instance of a dialog object that parameterizes generation of a sample XML based on a W3C schema or DTD. See GenerateSampleXML for its usage.

Errors

2300 The Dialogs object or one of its parents is no longer valid.

2301 Invalid address for the return parameter was specified.

DTDSchemaGeneratorDlg

See also

Property: DTDSchem aGeneratorD b as DTDSchem aGeneratorD b (read-only)

Description

Get a new instance of a dialog object that parameterizes generation of a schema or DTD. See Document.GenerateDTDOrSchemaEx for its usage.

Errors

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

FindInFilesDlg

See also

Property: FindlnFilesDlg as <u>FindlnFilesDlg</u> (read-only)

Description

Get a new instance of a dialog object that parameterizes the search (or replacement) of strings in files. See <u>Application.FindInFiles</u> for its usage.

Errors

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

WSDLDocumentationDlg

See also

Property: W SDLDocum entationD b as W SDLDocum entationD b (read-only)

Description

Get a new instance of a dialog object that parameterizes generation of WSDL documentation. See <u>Document.GenerateWSDLDocumentation</u> for its usage.

Errors

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

WSDL20DocumentationDlg

See also

Property: W SDL20Docum entationD b as W SDL20Docum entationD b (read-only)

Description

Get a new instance of a dialog object that parameterizes generation of WSDL 2.0 documentation. See <u>Document.GenerateWSD20LDocumentation</u> for its usage.

Errors

2300 The Dialogs object or one of its parents is no longer valid.

2301 Invalid address for the return parameter was specified.

XBRLDocumentationDlg

See also

Property: XBRL20Docum entationD b as XBRL20Docum entationD b (read-only)

Description

Get a new instance of a dialog object that parameterizes generation of WSDL 2.0 documentation. See <u>Document.GenerateXBRLDocumentation</u> for its usage.

- 2300 The Dialogs object or one of its parents is no longer valid.
- 2301 Invalid address for the return parameter was specified.

3.2.10 Document

See also

Properties and Methods

Standard automation properties Application Parent

Various document properties and methods SetActiveDocument Encoding SetEncoding (obsolete) Suggestions

XML validation BValid SetExtemalsValid

Document conversion and transformation

AssignDTD AssignSchema AssignXSL AssignXSLFO ConvertDTDO rSchema ConvertDTDOrSchemaEx GenerateDTD0rSchema GenerateDTD0rSchemaEx C reateS chem aD iagram ExecuteXQuery Transform XSL Transform XSLEx Transform XSLFO GenerateProgram Code (Enterprise Edition only) GenerateSchem aDocum entation GenerateSampleXML ConvertToWSDL20

Document export

GetExportElementList ExportToText ExportToDatabase CreateDBStructureFromXMLSchema GetDBStructureList

File saving and naming

FulName Name Path GetPathName (obsolete) SetPathName (obsolete) Tite IsM odified Saved SaveAs Save Save DS tring SaveToURL C bse

View access CurrentViewMode SwitchViewMode AuthenticView GridView DocEditView (obsolete)

Access to XMLData

RootElement DataRoot CreateChild UpdateViews StartChanges EndChanges UpdateXMLData

Description

Document objects represent XML documents opened in Authentic Desktop.

Use one of the following properties to access documents that are already open Authentic Desktop: Application ActiveDocument Application Documents

Use one of the following methods to open a new document in Authentic Desktop:

Docum ents OpenFile Docum ents OpenURL Docum ents OpenURLD abg Docum ents NewFile Docum ents NewFile From Text SpyProjectItem Open Application In portFrom Database Application In portFrom Schem a Application In portFrom Text Application In portFrom Word Docum ent ConvertDTDOrS chem a Docum ent GenerateDTDOrS chem a

Events

OnBeforeSaveDocument

See also

Event: OnBeforeSaveDocument(obpocument as Document, obpabg as FileSelectionDb)

XMLSpy scripting environment - VBScript:

Function On BeforeSaveDocum ent(obpocum ent, obp abg) End Function

```
'old handler - now obsolete
'return string to save to new file name
'return empty string to cancel save operation
'return nothing to save to original name
```

Function On SaveDocum ent (ob Docum ent, strFilePath) End Function

XMLSpy scripting environment - JScript:

```
function On BeforeSaveDocum ent(obpDocum ent, obp is bg)
{
}
```

```
// old handler - now obsolete
// return string to save to new file name
```

```
(C) 2018 Altova GmbH
```

```
// return empty string to cancel save operation
// return nothing to save to original nam e
function On_SaveDocument(obp)ocument, strFilePath)
{
}
```

XMLSpy IDE Plugin:

KMLSpyPlughOnEvent(27, ...) // nEventId = 27

Description

This event gets fired on any attempt to save a document. The file selection dialog object is initialized with the name chosen for the document file. You can modify this selection. To continue saving the document leave the FileSelectionDigDialogAction property of *io_objDialog* at its default value spyDiabgOK. To abort saving of the document set this property to spyDiabgCancel

OnBeforeCloseDocument

See also

Event: OnBeforeC bseDocum ent(ob) Docum ent as Docum ent) as Boolean

XMLSpy scripting environment - VBScript:

XMLSpy scripting environment - JScript:

XMLSpy IDE Plugin:

KMLSpyPlughOnEvent(28, ...) // nEventId = 28

Description

This event gets fired on any attempt to close a document. To prevent the document from being closed return false.

OnBeforeValidate

See also

Event: OnBeforeValidate (obf)ocument as Document, bOnLoading as Boolean, bOnComm and as Boolean) as Boolean

XMLSpy scripting environment - VBScript:

 Function On BeforeValidate (obpocum ent, bOnLoading, bOnComm and)

 On BeforeValidate = bC anceDefaultValidation

 'set by the script if necessary

 End Function

XMLSpy scripting environment - JScript:

```
function On_BeforeValidate (obf)ocum ent, bOnLoading, bOnComm and)
{
    return bCanceDefaultValidation //set by the script if necessary
}
```

```
XMLSpy IDE Plugin:
```

XM LSpyP ligh OnEvent (32, ...)// nEventId = 32

Description

This event gets fired before the document is validated. It is possible to suppress the default validation by returning false from the event handler. In this case the script should also set the validation result using the SetExternalIsValid method.

bOnLoading is true if the event is raised on the initial validation on loading the document.

b0 nC om m and is true whenever the user selected the Validate command from the Toolbar or menu.

Available with TypeLibrary version 1.5

OnCloseDocument

See also

Event: OnC bseDocum ent(obpDocum ent as Docum ent)

XMLSpy scripting environment - VBScript: Function On C bse Docum ent(obp)ocum ent)

End Function

XMLSpy scripting environment - JScript:

function On_C bse Docum ent(obpocum ent)
{
}

XMLSpy IDE Plugin:

KMLSpyPlighOnEvent(8, ...) // nEventId = 8

Description

This event gets fired as a result of closing a document. Do not modify the document from within this event.

OnViewActivation

See also

Event: OnViewActivation (obj/ocum ent as Docum ent, eViewMode as SPYViewModes, bActivated as Boolean)

XMLSpy scripting environment - VBScript:

Function On ViewActivation (ob Document, eViewMode, bActivated) End Function

XMLSpy scripting environment - JScript:

function On_ViewActivation (ob_Document, eViewMode, bActivated)
{
}

XMLSpy IDE Plugin:

KM LSpyP lugIn O nEvent (29, ...)// nEventId = 29

Description

This event gets fired whenever a view of a document becomes visible (i.e. becomes the active view) or invisible (i.e. another view becomes the active view or the document gets closed).

However, the first view activation event after a document gets opened cannot be received, since there is no document object to get the event from. Use the <u>Application OnDocumentOpened</u> event instead.

Application

See also

Property: Application as Application (read-only)

Description

Accesses the Authentic Desktop application object.

Errors

1400 The object is no longer valid.1407 Invalid address for the return parameter was specified.

AssignDTD

See also

Method: AssignDTD (strDTDFile as String, bDiabg as Boolean)

Description

The method places a reference to the DTD file "strDTDFile" into the document. Note that no error occurs if the file does not exist, or is not accessible. If bDiabg is true Authentic Desktop presents a dialog to set the file.

Errors

- 1400 The object is no longer valid.
- 1409 You are not allowed to assign a DTD to the document.

AssignSchema

See also

Method: AssignSchema (strSchemaFile as String, bDiabg as Boolean)

Description

The method places a reference to the schema file "strSchemaFile" into the document. Note that no error occurs if the file does not exist or is not accessible. If bDiabg is true Authentic Desktop presents a dialog to set the file.

Errors

- 1400 The object is no longer valid.
- 1409 You are not allowed to assign a schema file to the document.

AssignXSL

See also

Method: AssignXSL (strXSLFile as String, bDiabg as Boolean)

Description

The method places a reference to the XSL file "strXSLFile" into the document. Note that no error occurs if the file does not exist or is not accessible. If bDiabg is true Authentic Desktop presents a dialog to set the file.

Errors

1400 The object is no longer valid.

1409 You are not allowed to assign an XSL file to the document.

AssignXSLFO

See also

Method: AssignXSLFO (*strXSLFO File* as String, *bDiabg* as Boolean)

Description

The method places a reference to the XSLFO file "strXSLFile" into the document. Note that no error occurs if the file does not exist or is not accessible. If bDiabg is true Authentic Desktop presents a dialog to set the file.

Errors

1400 The object is no longer valid.

1409 You are not allowed to assign an XSL file to the document.

AsXMLString

See also

Property: AsXMLString as String

Description

This property can be used to get or set the document content.

Errors

- 1400 The document object is no longer valid.
- 1404 Cannot create XMLData object.
- 1407 View mode cannot be switched.

AuthenticView

See also

Method: AuthenticView as AuthenticView (read-only)

Description

Returns an object that gives access to properties and methods specific to Authentic view. The object returned is only valid if the current document is opened in Authentic view mode. The lifetime of an object ends with the next view switch. Any attempt to access objects or any of its children afterwards will result in an error indicating that the object is invalid.

Errors

- 1400 The object is no longer valid.
- 1417 Document needs to be open in authentic view mode.

Examples

'XMLSpy scripting environment - VBScript 'secure access to authentic view object

Dim objocum ent Set objocum ent = Application ActiveDocum ent If Not objocum ent Is Nothing) Then

'we have an active document, now check for view mode

If (objDocum entCumentViewMode <> spyViewAuthentic) Then

If NotobpocumentSwitchViewMode (spyViewAuthentic)) Then

MsgBox "Active document does not support authentic view mode"

Else

'now it is safe to access the authentic view object D in objAuthenticView Set objAuthenticView = objDocumentAuthenticView

'now use the authentic view object

End If

End If

Else

MsgBox "No document is open"

End If

Close

See also

Method: C bse (*bD iscardC hanges* as Boolean)

Description

To close the document call this method. If bD iscardChanges is true and the document is modified, the document will be closed but not saved.

Errors

1400 The object is no longer valid.1401 Document needs to be saved first.

Convert DTDOrSchema

See also

Method: ConvertDTDO1Schem a (*nForm at* **as** <u>SPYDTDSchem aForm at</u>, *nFrequentE lem ents* **as** <u>SPYFrequentE lem ents</u>)

Parameters

nFormat Sets the schema output format to DTD or W3C.

nFrequentElements

Create complex elements as elements or complex types.

Description

ConvertDTDOrSchem a takes an existing schema format and converts it into a different format. For a finer tuning of DTD/ XSD conversion, use <u>ConvertDTDOrSchemaEx</u>.

Errors

- 1400 The object is no longer valid.
- 1412 Error during conversion. In the case of DTD to DTD or XSD to XSD conversion, the following errors are returned: *DTD to DTD conversion is not supported. Please use function FlattenDTDOrSchema instead* and *Schema to schema conversion is not supported. Please use function FlattenDTDOrSchema instead.*

Convert DTDOrSchemaEx

See also

Method: ConvertDTDOrSchem aEx (*nForm at* as <u>SPYDTDSchem aForm at</u>, *nFrequentElem ents* as <u>SPYFrequentElem ents</u>, sOutputPath as String, nOutputPathDialogAction as <u>SPYDialogAction</u>)

Parameters

nForm at Sets the schema output format to DTD, or W3C.

nFrequentElements Create complex elements as elements or complex types.

sOutputPath The file path for the newly generated file.

nOutputPathDialogAction Defines the dialog interaction for this call.

Description

ConvertDTDDrSchemaEx takes an existing schema format and converts it into a different format.

Errors

- 1400 The object is no longer valid.
- 1412 Error during conversion. In the case of DTD to DTD or XSD to XSD conversion, the following errors are returned: *DTD to DTD conversion is not supported. Please use function FlattenDTDOrSchema instead* and *Schema to schema conversion is not supported. Please use function FlattenDTDOrSchema instead.*

ConvertToWSDL20

Method: ConvertToW SDL20 (*sFitePath* as String, *bShowDialogs* as Boolean)

Parameters

sFilePath

This specifies the file name of the converted WSDL. In case the source WSDL includes files which also must be converted, then only the directory part of the given path is used and the file names are generated automatically.

bShowDiabgs

Defines whether file/ folder selection dialogs are shown.

Description

Converts the WSDL 1.1 document to a WSDL 2.0 file. It will also convert any referenced WSDL files that are referenced from within this document. Note that this functionality is limited to WSDL View only. See DocumentCurrentViewMode. and SPYViewModes.

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid parameters have been passed or an empty file name has been specified as output target.
- 1417 The document is not opened in WSDL view, maybe it is not an '.wsdl' file.
- 1421 Feature is not available in this edition.
- 1433 WSDL 1.1 to WSDL 2.0 conversion failed.

CreateChild

See also

Method: CreateChild (nKind as SPYXMLDataKind) as XMLData

Return Value

The method returns the new XM LData object.

Description

To create a new XM LData object use the C reateC hild () method.

Errors

- 1400 The object is no longer valid.
- 1404 Cannot create XMLData object.
- 1407 Invalid address for the return parameter was specified.

CreateDBStructureFromXMLSchema

See also

Method: C reateDBS tructure From XM LS chem a (*pDatabase* as <u>DatabaseConnection</u>, *pTables* as Elem entList, *bD ropTableW ithExistingName* as Boolean) as String

Description

C reateDBStructureFrom XMLSchem a exports the given tables to the specified database. The function returns the SQL statements that were necessary to perform the changes.

See also Get DBStructureList.

- 1429 Database selection missing.
- 1430 Document export failed.

CreateSchemaDiagram

See also

Method: CreateSchem aDiagram (*nKind* as <u>SPYSchem aDefKind</u>, *strNam e* as String, *strFile* as String)

Return Value None.

Description

The method creates a diagram of the schema type strName of kind nKind and saves the output file into strFile. Note that this functionality is limited to Schema View only. See DocumentCurrentViewMode. and SPYViewModes.

Errors

- 1400 The object is no longer valid.
- 1414 Failed to save diagram.
- 1415 Invalid schema definition type specified.

Current ViewMode

See also

Method: CurrentViewMode as SPYViewModes

Description

The property holds the current view mode of the document. See also DocumentSwitchViewMode.

Errors

1400 The object is no longer valid.

1407 Invalid address for the return parameter was specified.

DataRoot

See also

Property: DataRoot as XM LData (read-only)

Description

This property provides access to the document's first XMLData object of type *spyXMLData* ==ment. This is typically the root element for all document content data. See <u>XMLSpyDocument.RootElement</u> to get the root element of the whole document including XML prolog data. If the <u>CurrentViewMode</u> is not *spyViewGrid* or *spyViewAuthentic* an <u>UpdateXMLData</u> may be necessary to get access to the latest <u>XMLData</u>.

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

DocEditView

See also

Method: DocEditView as DocEditView

Description

Holds a reference to the current Authentic View object.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid address for the return parameter was specified.
- 1417 Document needs to be open in authentic view mode.

Encoding

See also

Property: Encoding as String

Description

This property provides access to the document's encoding value. However, this property can only be accessed when the document is opened in *spyViewGrid*, *spyViewText* or *spyViewAuthentic*. See <u>CurrentViewMode</u> on how to detect that a document's actual view mode.

This property makes the method SetEncoding obsolete.

Possible values are, for example:

8859-1, 8859-2, ASCII, ISO-646, 850, 1252, 1255, SHIFT-JIS, MS-KANJI, BIG5, FIVE, UTF-7, UTF-8, UTF-16

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.
- 1416 Operation not supported in current view mode.

EndChanges

See also

Method: EndChanges()

Description

Use the method EndChanges to display all changes since the call to DocumentStartChanges.

Errors

1400 The object is no longer valid.
ExecuteXQuery

See also

Method: ExecuteXQuery (*strXMLFileNam e* as String)

Description

Execute the XQuery statements contained in the document of the document object. Either an XQuery execution or an XQuery Update is performed depending on the file extension of the document. Use the XML file specified in the argument as the XML target document that the XQuery document processes.

- If the document has an XQuery file extension as defined in the Options dialog of Authentic Desktop, then an XQuery execution is performed. By default: .xq, .xql, and .xquery are set as XQuery file extensions in Authentic Desktop.
- If the document has an XQuery Update file extension as defined in the Options dialog of Authentic Desktop, then an XQuery Update action is performed. By default: .xqu is set as an XQuery Update file extension in Authentic Desktop.

If your XQuery script does not use an XML source, set the parameter strXMLFileName to an empty string.

Errors

- 1400 The document object is no longer valid.
- 1423 XQuery transformation error.
- 1424 Not all files required for operation could be loaded. Most likely, the file specified in strXMLFileName does not exist or is not valid.

ExportToDatabase

See also

Method: ExportToDatabase (*pFrom Child* as <u>XMLData</u>, *pExportSettings* as <u>ExportSettings</u>, *pDatabase* as DatabaseConnection)

Description

ExportToDatabase exports the XML document starting with the element pFrom Child. The parameter pExportSettings defines the behaviour of the export (see <u>Application SetExportSettings</u>). The parameter pDatabase specifies the destination of the export (see

Application GetDatabaseSettings). UpdateXMLData() might be indirectly needed as you have to pass the XMLData as parameter to this function.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.
- 1416 Error during export.
- 1429 Database selection missing.
- 1430 Document export failed.

Example

Dim objDoc As Document Set objDoc = objSpyActiveDocument

set the behaviour of the export with ExportSettings

```
D in objexpSettings As ExportSettings
Set objexpSettings = objexpGetExportSettings
```

```
set the destination with DatabaseConnection
Dim obDB As DatabaseConnection
Set obDB = obBpyGetDatabaseSettings
```

objDBC reateM issingTables = True objDBC reateNew = True objDBFile = "C \Exportmdb"

```
objOcExportToDatabase objOcRootElement,objExpSettings,objDB
If ErnNumber <> 0 Then
  a = MsgBox("Enor:" & (ErnNumber - vbObjectEnor) & Chr(13) &
      "Description:" & ErnDescription)
End If
```

ExportToText

See also

Method: ExportToText (*pFrom Child* as XMLData, *pExportSettings* as ExportSettings, *pTextSettings* as TextIn portExportSettings)

Description

ExportToText exports tabular information from the document starting at pFrom Child into one or many text files. Columns of the resulting tables are generated in alphabetical order of the column header names. Use <u>GetExportElementList</u> to learn about the data that will be exported. The parameter pExportSettings defines the specifics for the export. Set the property <u>ExportSettings ElementList</u> to the - possibly modified - list returned by <u>GetExportElementList</u> to avoid exporting all contained tables. The parameter pTextSettings defines the options specific to text export and import. You need to set the property <u>TextIn portExportSettings DestinationFober</u> before you call ExportToText <u>UpdateXMLData()</u> might be indirectly needed as you have to pass the <u>XMLData</u> as parameter to this function.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.
- 1416 Error during export.
- 1430 Document export failed.

Example

۰.

'VBA client code fragment - export document to text files

```
Din objDoc As Document
Set objDoc = objBpyActiveDocument
```

Dim objTextExp As TextIm portExportSettings Set objTextExp = objSpyGetTextExportSettings objTextExp HeaderRow = True objTextExp DestinationFolder = "C \Exports"

On Enor Resume Next objocExportToText objocRootElement, objExpSettings, objTextExp If ErrNumber <> 0 Then
 a = M sgBox ("Error:" & (ErrNumber - vbObjectError) & Chr(13) & "Description:"
 & ErrDescription)
End If

FlattenDTDOrSchema

Method: FlattenDTDO1Schema (sOutputPath **as** String, nOutputPathDialogAction **as** <u>SPYDialogAction</u>)

Parameters

sOutputPath The file path for the newly generated file.

nOutputPathDialogAction Defines the dialog interaction for this call.

Description

FlattenDTDOrSchema takes an existing DTD or schema, generates a flattened file, and saves the generated file at the specified location. In the case of DTDs, flattening removes parameter entities and produces a single DTD from a collection of modules; sections marked IGNORE are suppressed and unused parameter entities are deleted. When an XML Schema is flattened, (i) the components of all included schemas are added as global components of the active schema, and (ii) included schemas are deleted.

Errors

- 1400 The object is no longer valid.
- 1412 Error during conversion.

FullName

See also

Property: FullName as String

Description

This property can be used to get or set the full file name - including the path - to where the document gets saved. The validity of the name is not verified before the next save operation.

This property makes the methods GetPathName and SetPathName obsolete.

Errors

- 1400 The document object is no longer valid.
- 1402 Empty string has been specified as full file name.

GenerateDTDOrSchema

See also

Method: GenerateDTDOrSchem a (*nForm at* as SPYDTDSchem aForm at, *nValuesList* as integer, *nDetection* as SPYTypeDetection, *nFrequentElem ents* as SPYFrequentElem ents)

Parameters

nForm at Sets the schema output format to DTD, or W3C.

nValuesList

Generate not more than this amount of enumeration-facets per type. Set to - 1 for unlimited.

nDetection

Specifies granularity of simple type detection.

nFrequentElem ents

Shall the types for all elements be defined as global? Use that value *spyGlobalComplexType* to define them on global scope. Otherwise, use the value *spyGlobalElements*.

Description

Use this method to automatically generate a DTD or schema for the current XML document. For a finer tuning of DTD / schema generation, use <u>GenerateDTDOrSchemaEx</u>. Note that this functionality is not available in ZIP View only. See <u>DocumentCurrentViewMode</u>. and <u>SPYViewModes</u>.

Errors

1400 The object is no longer valid.

1407 Invalid parameter or invalid address for the return parameter was specified.

GenerateDTDOrSchemaEx

See also

Method: GenerateDTDOrSchemaEx (objDlg as DTDSchemaGeneratorDlg) as Document

Description

Use this method to automatically generate a DTD or schema for the current XML document. A <u>DTDSchemaGeneratorDlg</u> object is used to pass information to the schema/ DTD generator. The generation process can be configured to allow user interaction or run without further user input. Note that this functionality is not available in ZIP View only. See <u>DocumentCurrentViewMode</u>. and <u>SPYViewModes</u>.

Errors

1400 The object is no longer valid.

1407 Invalid parameter or invalid address for the return parameter was specified.

GenerateProgramCode

Method: GenerateProgram Code (obj) as CodeGeneratorD b)

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

Description

Generate Java, C++ or C# class files from the XML Schema definitions in your document. A codeGeneratorDb object is used to pass information to the code generator. The generation process can be configured to allow user interaction or run without further user input.

- 1400 The document object is no longer valid.
- 1407 An empty file name has been specified.
- 1421 Feature not available in this edition

GenerateSampleXML

Method: GenerateSam pbXML (obp b as GenerateSam pbXMLDb) as Document

Description

Generates a sample XML if the document is a schema or DTD. Use <u>Dialogs.GenerateSampleXMLDIg</u> to get an initialized set of options.

Available with TypeLibrary version 1.5

Errors

1400 The document object is no longer valid.

GenerateSchemaDocumentation

Method: GenerateSchem aDocum entation (obp b as Schem aDocum entationD b)

Description

Generate documentation for a schema definition file in HTML, MS-Word, or RTF format. The parameter objDlg is used to parameterize the generation process. Use <u>DiabgsSchemaDocumentationDlg</u> to get an initialized set of options. As a minimum, you will need to set the property <u>SchemaDocumentationDlgOutputFile</u> before starting the generation process. Note that this functionality is limited to Schema View only. See <u>DocumentCurrentViewMode</u> and

SPYViewModes.

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid parameters have been passed or an empty file name has been specified as output target.
- 1417 The document is not opened in schema view, maybe it is not an '.xsd' file.
- 1421 Feature is not available in this edition.
- 1422 Error during generation

GenerateWSDL20Documentation

Method: GenerateW SDL20Docum entation (obj b as W SDL20Docum entationD b)

Description

Generate documentation for a WSDL definition file in HTML, MS- Word, or RTF format. The parameter objDlg is used to parameterize the generation process. Use <u>DiabgsWSDL20DocumentationDb</u> to get an initialized set of options. As a minimum, you will need to set the property <u>WSDL20DocumentationDbCutputFile</u> before starting the generation process. Note that this functionality is limited to WSDL View only. See <u>DocumentCurrentViewMode</u> and <u>SPYViewModes</u>.

- 1400 The document object is no longer valid.
- 1407 Invalid parameters have been passed or an empty file name has been specified as output target.
- 1417 The document is not opened in schema view, maybe it is not an '.xsd' file.
- 1421 Feature is not available in this edition.
- 1422 Error during generation

GenerateWSDLDocumentation

Method: GenerateW SDLDocum entation (obp b as W SDLDocum entationD b)

Description

Generate documentation for a WSDL definition file in HTML, MS-Word, or RTF format. The parameter objDlg is used to parameterize the generation process. Use

DiabgsWSDLDocumentationDb to get an initialized set of options. As a minimum, you will need to set the property WSDLDocumentationDbOutputFile before starting the generation process. Note that this functionality is limited to WSDL View only. See <u>DocumentCumentViewMode</u> and SPYViewModes.

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid parameters have been passed or an empty file name has been specified as output target.
- 1417 The document is not opened in schema view, maybe it is not an '.xsd' file.
- 1421 Feature is not available in this edition.
- 1422 Error during generation

GenerateXBRLDocumentation

Method: GenerateXBRLDocum entation (*ob***Db** as XBRLDocum entationDb)

Description

Generate documentation for a WSDL definition file in HTML, MS-Word, or RTF format. The parameter objDlg is used to parameterize the generation process. Use <u>Dialogs XBRLDocum entationDlg</u> to get an initialized set of options. As a minimum, you will need to set the property <u>XBRLDocum entationDlgOutputFile</u> before starting the generation process. Note that this functionality is limited to XBRL View only. See <u>Docum entcurrentViewMode</u> and <u>SPYViewModes</u>.

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid parameters have been passed or an empty file name has been specified as output target.
- 1417 The document is not opened in schema view, maybe it is not an '.xsd' file.
- 1421 Feature is not available in this edition.
- 1422 Error during generation

GetDBStructureList

See also

Method: GetDBStructureList (pDatabase as DatabaseConnection) as ElementList

Description

GetDBStructureList creates a collection of elements from the Schema document for which tables in the specified database are created. The function returns a collection of ElementListTems where the properties ElementListTem Name contain the names of the tables.

See also CreateDBStructureFromXMLSchema.

Errors

- 1400 The object is no longer valid.
- 1427 Failed creating parser for the specified XML.
- 1428 Export of element list failed.
- 1429 Database selection missing.

Get Export Element List

See also

Method: GetExportElementList (*pFrom Child* as <u>XM LData</u>, *pExportSettings* as <u>ExportSettings</u>) as <u>ElementList</u>

Description

GetExportElem entList creates a collection of elements to export from the document, depending on the settings in pExportSettings and starting from the element pFromChild. The function returns a collection of ElementListTems where the properties <u>ElementListTem Name</u> contain the names of the tables that can be exported from the document. The property <u>ElementListTem FieldCount</u> contains the number of columns in the table. The property <u>ElementListTem RecordCount</u> contains the number of records in the table. The property <u>ElementListTem RecordCount</u> contains the number of records in the table. The property <u>ElementListTem elementKind</u> is unused. <u>UpdateXMLData()</u> might be indirectly needed as you have to pass the <u>XMLData</u> as parameter to this function.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.
- 1427 Failed creating parser for the specified XML.
- 1428 Export of element list failed.

GetPathName (obsolete)

```
Superseded by Document.FullName
// ----- javascript sam ple -----
// instead of:
// stPathNam e = Application ActiveDocum entGetPathNam e 0;
// use now:
stPathNam e = Application ActiveDocum entFulNam e;
```

See also

Method: GetPathName() as String

Description

The method GetPathName gets the path of the active document.

See also DocumentSetPathName (obsolete).

GridView

See also

Property: GridView as GridView

Description

This property provides access to the grid view functionality of the document.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid address for the return parameter was specified.
- 1417 Document needs to be open in enhanced grid view mode.

IsModified

See also

Property: IsM odified as Boolean

Description

True if the document is modified.

Errors

1400 The object is no longer valid.1407 Invalid address for the return parameter was specified.

IsValid

See also

Method: IsValid (strError as Variant) as Boolean

Return Value

True if the document is valid, false if not. To call IsValid(), the application GUI must be visible. (If you wish to validate without the GUI being visible, please use Altova RaptorXML Server.)

Description

IsValid validates the document against its associated schema or DTD. strError gives you the same error message as when you validate the file within the GUI.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.
- 1408 Unable to validate file.

IsValidEx

Method: BValidEx (*nXSDVersion* as <u>SPYValidateXSDVersion</u>, *nErrorLin* it as int, *nErrorForm* at as <u>SPYValidateErrorFormat</u>, out <u>strError</u> as Variant) as Boolean

Return Value

True if the document is valid, false if not.

Description

IsValidEx validates the document against its associated schema or DTD.

In parameters:

nXSDVersion which is an enumeration value of <u>SPYValidateXSDVersion</u> that selects the XSD version to validate against.

nErrorLimit which is an integer. Values must be 1 to 999.

nErrorFormat which is an enumeration value of <u>SPYValidateErrorFormat</u> that selects the XSD version to validate against.

Out parameter:

strError is the error message, and is the same as that received when validating the file within the GUI.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.

1408 Unable to validate file.

IsWellFormed

See also

Method: Isw ellForm ed (pData as XM LData, bW ith Chibren as Boolean, strError as Variant, nErrorPos as Variant, pBadXM LData as Variant) as Boolean

Return Value

True if the document is well formed.

Description

Iswellformed checks the document for well-formedness starting at the element pData.

If the document is not well formed, strError contains an error message, nErrorPos the position in the file and pBadXM LData holds a reference to the element which breaks the well-formedness. These out-parameters are defined as VARIANTs to support scripting languages like VBScript.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.

Example

See IsValid.

Name

See also

Property: Name as String (read-only)

Description

Use this property to retrieve the name - not including the path - of the document file. To change the file name for a document use the property <u>FullName</u>.

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

Parent

See also

Property: Parent as Documents (read-only)

Description

Access the parent of the document object.

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

Property: Parent as Application (read-only)

Path

See also

Property: Path as String (read-only)

Description

Use this property to retrieve the path - not including the file name - of the document file. To change the file name and path for a document use the property <u>FullName</u>.

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

Root Element

See also

Property: RootElement as XM LData (read-only)

Description

The property RootElement provides access to the root element of the XML structure of the document including the XML prolog data. To access the first element of a document's content navigate to the first child of kind *spyXMLDataElement* or use the <u>Document.DataRoot</u> property. If the <u>CurrentViewMode</u> is not *spyViewGrid* or *spyViewAuthentic* an <u>UpdateXMLData</u> may be necessary to get access to the latest XMLData.

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

Save

See also

Method: Save ()

Description

The method writes any modifications of the document to the associated file. See also DocumentFulName.

Errors

- 1400 The document object is no longer valid.
- 1407 An empty file name has been specified.
- 1403 Error when saving file, probably the file name is invalid.

SaveAs

See also

Method: SaveAs (strFieName as String)

Description

Save the document to the file specified. If saving was successful, the <u>FullName</u> property gets set to the specified file name.

Errors

- 1400 The document object is no longer valid.
- 1407 An empty file name has been specified.
- 1403 Error when saving file, probably the file name is invalid.

Saved

See also

Property: Saved as Boolean (read-only)

Description

This property can be used to check if the document has been saved after the last modifications. It returns the negation of <u>IsModified</u>.

Errors

- 1400 The document object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

SaveInString

See also

Method: SaveInString (pData as XM LData, bM arked as Boolean) as String

Parameters

pData

XM LData element to start. Set pData to <u>DocumentRootElement</u> if you want to copy the complete file.

bM arked

If bM arked is true, only the elements selected in the grid view are copied.

Return Value

Returns a string with the XML data.

Description

Save InString starts at the element pData and converts the XMLData objects to a string representation. <u>UpdateXMLData()</u> might be indirectly needed as you have to pass the <u>XMLData</u> as parameter to this function.

Errors

1400 The object is no longer valid.

1407 Invalid parameter or invalid address for the return parameter was specified.

SaveToURL

See also

Method: SaveToURL (strURL as String, strUser as String, strPassword as String)

Return Value

Description

SaveToURL () writes the document to the URL strURL. This method does not set the permanent file path of the document.

Errors

- 1400 The object is no longer valid.
- 1402 Invalid URL specified.
- 1403 Error while saving to URL.

SetActiveDocument

See also

Method: SetActiveDocument()

Description

The method sets the document as the active and brings it to the front.

Errors

1400 The object is no longer valid.

Set Encoding (obsolete)

```
Superseded by Document.Encoding
// ----- javascript sam ple -----
// instead of:
// Application A ctiveD ocum entSetEncoding ("UTF-16");
// use now :
Application A ctiveD ocum entEncoding = "UTF-16";
```

See also

Method: SetEncoding (*strEncoding* as String)

Description

SetEncoding sets the encoding of the document like the menu item "File/Encoding..." in Authentic Desktop. Possible values for strEncoding are, for example:

8859-1, 8859-2, ASCII, ISO-646, 850, 1252, 1255, SHIFT-JIS, MS-KANJI, BIG5, FIVE, UTF-7, UTF-7, UTF-8, UTF-16

SetExternallsValid

See also

Method: SetExtemalsValid (bValid as Boolean)

Parameters

bValid

Sets the result of an external validation process.

Description

The internal information set by this method is only queried on cancelling the default validation in any OnBeforeValidate handler.

Available with TypeLibrary version 1.5

Errors

1400 The object is no longer valid.

SetPathName (obsolete)

Superseded by Document.FullName

// ----- javascript sam ple ----// instead of:
// Application ActiveD ocum entSetPathNam e ("C :\ \m yXM LFiles \ \testxm 1");
// use now :
Application ActiveD ocum entFulN am e = "C :\ \m yXM LFiles \ \testxm 1";

See also

Method: SetPathName (*strPath* as String)

Description

The method SetPathName sets the path of the active document. SetPathName only copies the string and does not check if the path is valid. All succeeding save operations are done into this file.

StartChanges

See also

Method: StartChanges()

Description

After StartChanges is executed Authentic Desktop will not update its editor windows until <u>Docum entEndChanges</u> is called. This increases performance of complex tasks to the XML structure.

Errors

1400 The object is no longer valid.

Suggestions

Property: Suggestions as Array

Description

This property contains the last valid user suggestions for this document. The XMLSpy generated suggestions can be modified before they are shown to the user in the <u>OnBeforeShowSuggestions</u> event.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid parameter or invalid address for the return parameter was specified.

SwitchViewMode

See also

Method: SwitchViewMode (nMode as SPYViewModes) as Boolean

Return value

Returns true if view mode is switched.

Description

The method sets the current view mode of the document in Authentic Desktop. See also DocumentCurrentViewMode.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid address for the return parameter was specified.
- 1417 Invalid view mode specified.

TextView

See also

Property: TextView as <u>TextView</u>

Description

This property provides access to the text view functionality of the document.

Errors

- 1400 The object is no longer valid.
- 1407 Invalid address for the return parameter was specified.

Title

See also

Property: Title as String (read-only)

Description

T ± contains the file name of the document. To get the path and filename of the file use FullNam e.

Errors

1400 The document object is no longer valid.

1407 Invalid address for the return parameter was specified.

TransformXSL

See also

Method: Transform XSL()

Description

Transform XSL processes the XML document via the associated XSL file. See DocumentAssignXSL on how to place a reference to a XSL file into the document.

Errors

1400 The document object is no longer valid.

1411 Error during transformation process.

TransformXSLEx

See also

Method: Transform XSLEx (*nAction* as SPYDiabgAction)

Description

Transform XSLEx processes the XML document via the associated XSL file. The parameter specifies whether a dialog asking for the result document name should pop up or not. See DocumentAssignXSL on how to place a reference to a XSL file into the document.

Errors

- 1400 The document object is no longer valid.
- 1411 Error during transformation process.

TransformXSLFO

See also

Method: Transform XSLFO ()

Description

Transform XSLFO processes the XML document via the associated XSLFO file. See <u>AssignXSLFO</u> on how to place a reference to a XSLFO file into the document. You need to assign a FOP processor to Authentic Desktop before you can use this method.

Errors

- 1400 The document object is no longer valid.
- 1411 Error during transformation process.

TreatXBRLInconsistenciesAsErrors

Property: TreatXBRLhconsistenciesAsErrors as Boolean

Description

If this is set to true the Document.IsValid() method will return false for XBRL instances containing inconsistencies as defined by the XBRL Specification. The default value of this property is false.

Errors

1400 The document object is no longer valid.

1407 Invalid address for the return parameter was specified.

UpdateViews

See also

Method: UpdateViews()

Description

To redraw the Enhanced Grid View and the Tree View call UpdateViews. This can be important after you changed the XMLData structure of a document. This method does not redraw the text view of Authentic Desktop.

Errors

1400 The document object is no longer valid.

UpdateXMLData

See also

Method: UpdateXM LData () as Boolean

Description

The XMLData tree is updated from the current view. Please note that this can fail in case of the TextView if the current XML text is not well-formed. This is not necessary if <u>CurrentViewMode</u> is *spyViewGrid* or *spyViewAuthentic* because these views keep the XMLData updated.

Available with TypeLibrary version 1.5

Errors

1400 The document object is no longer valid.

3.2.11 Documents

See also

Properties

Count Item

Methods

NewAuthenticFie NewFie NewFieFromText OpenAuthenticFie OpenFie OpenURL OpenURLDiabg

Description

This object represents the set of documents currently open in Authentic Desktop. Use this object to open further documents or iterate through already opened documents.

Examples

```
'XM LSpy scripting environment - VB Script
'iterate through open docum ents
Dim objDocuments
Set objocum ents = Application Docum ents
For Each obj Doc In obj Documents
      to something useful with your document
      objDocSetActiveDocument()
Next
// ---
// XM LSpy scripting environment - JScript
// close all open docum ents
// ----
for (var iter = new Enum erator (Application Docum ents);
   !iteratEnd();
   iterm oveN ext())
{
      // MsgBox ("C losing file " + iter.item ()Name);
      iteritem ()C lose (true);
}
```

Count

See also

```
Property: Count as long
```

Description

Count of open documents.

1600	Invalid Documents object
1601	Invalid input parameter

ltem

See also

Method: Item (n as long) as Document

Description

Gets the document with the index n in this collection. Index is 1-based.

Errors

1600 Invalid Documents object

1601 Invalid input parameter

NewAut hent ic File

See also

Method: NewAuthenticFile (strSPSPath as String, strXMLPath as String) as Document

Parameters

strSPSPath The path to the SPS document.

strXM LPath The new XML document name.

Return Value

The method returns the new document.

Description

NewAuthenticFile creates a new XML file and opens it in Authentic View using SPS design strSPSPath.

NewFile

See also

Method: NewFile (strFile as String, strType as String) as Document

Parameters

strFile Full path of new file.

strType

Type of new file as string (i.e. " xml" , " xsd" , ...)

Return Value

Returns the new file.

Description

 ${\tt NewFie}$ creates a new file of type ${\tt strType}$ (i.e. "xml"). The newly created file is also the ActiveDocument.

NewFileFromText

See also

Method: NewFibFrom Text (strText as String, strType as String) as Document

Parameters

 $\ensuremath{\mathsf{strText}}$ The content of the new document in plain text.

strType

Type of the document to create (i.e. "xml").

Return Value

The method returns the new document.

Description

NewFileFromText creates a new document with strText as its content.

OpenAuthenticFile

See also

Method: OpenAuthenticFile (strSPSPath as String, strXMLPath as String) as Document

Parameters

strSPSPath The path to the SPS document.

strXM LPath

The path to the XML document (can be empty).

Return Value

The method returns the new document.

Description

OpenAuthenticFile opens an XML file or database in Authentic View using SPS design strSPSPath.

OpenFile

See also

Method: OpenFile (strPath as String, bDiabg as Boolean) as Document

Parameters

strPath Path and file name of file to open.

bDiabg Show dialogs for user input.

Return Value Returns the opened file on success.

Description

OpenFile opens the file strPath. If bDiabg is TRUE, a file-dialog will be displayed.

Example

Dim objoc As Document Set objoc = objpyDocumentsOpenFile(strFile,False)

OpenURL

See also

Method: OpenURL (*strURL* as String, *nURLType* as <u>SPYURLTypes</u>, *nLoading* as <u>SPYLoading</u>, *strUser* as String, *strPassword* as String) as Document

Parameters

strURL URL to open as document.

 ${\tt nURLType}$ Type of document to open. Set to -1 for auto detection.

nLoading Set nLoading to 0 (zero) if you want to load it from cache or proxy. Otherwise set nLoading to 1.

strUser Name of the user if required. Can be empty.

strPassword Password for authentification. Can be empty.

Return Value

The method returns the opened document.

Description

OpenURL opens the URL strURL.

OpenURLDialog

See also

Method: OpenURLDialog (*strURL* as String, *nURLType* as <u>SPYURLTypes</u>, *nLoading* as <u>SPYLoading</u>, *strUser* as String, *strPassword* as String) as Document

Parameters

strURL URL to open as document.

nURLType

Type of document to open. Set to -1 for auto detection.

nLoading

Set nLoading to 0 (zero) if you want to load it from cache or proxy. Otherwise set nLoading to 1.

strUser

Name of the user if required. Can be empty.

strPassword

Password for authentification. Can be empty.

Return Value

The method returns the opened document.

Description

OpenURLD abg displays the "open URL" dialog to the user and presets the input fields with the given parameters.

3.2.12 DTDSchemaGeneratorDlg

See also

Properties and Methods

Standard automation properties Application Parent

DTDS chem aForm at ValueList TypeDetection FrequentElements MergeAlEquaNamed ResolveEntities AttributeTypeDefinition G bbalAttributes OnlyStringEnum s MaxEnum Length OutputPath OutputPathDialogAction

Description

Use this object to configure the generation of a schema or DTD. The method <u>GenerateDTDOrSchemaEx</u> expects a DTDSchemaGeneratorDIg as parameter to configure the generation as well as the associated user interactions.

Application

Property: Application as Application (read-only)

Description

Access the Authentic Desktop application object.

Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

AttributeTypeDefinition

Property: AttributeTypeDefinition as SPYAttributeTypeDefinition

Description

Specifies how attribute definitions get merged.

Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

DTDSchemaFormat

Property: DTDSchemaFormat as SPYDTDSchem aForm at

Description

Sets the schema output format to DTD, or W3C.

Errors

3000 The object is no longer valid.

3001 Invalid address for the return parameter was specified.

Frequent Elements

Property: Frequent Elements as SPYFrequentElements

Description

Shall the types for all elements be defined as global? Use that value *spyGlobalComplexType* to define them on global scope. Otherwise, use the value *spyGlobalElements*.

Errors

The object is no longer valid.Invalid address for the return parameter was specified.

GlobalAttributes

Property: GlobalAttributes as Boolean

Description

Shall attributes with same name and type be resolved globally?

Errors

The object is no longer valid.Invalid address for the return parameter was specified.

MaxEnumLength

Property: MaxEnumLength as Integer

Description

Specifies the maximum number of characters allowed for enumeration names. If one value is longer than this, no enumeration will be generated.

Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

MergeAllEqualNamed

Property: MergeAllEqualNamed as Boolean

Description

Shall types of all elements with the same name be merged into one type?

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

OnlyStringEnums

Property: OnlyStringEnums as Boolean

Description

Specifies if enumerations will be created only for plain strings or all types of values.

Errors

3000 The object is no longer valid.

3001 Invalid address for the return parameter was specified.

Output Path

Property: Output Path as String

Description

Selects the file name for the generated schema/ DTD.

Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

Output PathDialogAction

Property: Output PathDialogAction as SPYDiabgAction

Description

Defines how the sub-dialog for selecting the schema/ DTD output path gets handled. Set this value to *spyDialogUserInput(2)* to show the dialog with the current value of the <u>OutputPath</u> property as default. Use *spyDialogOK(0)* to hide the dialog from the user.

Errors

3000 The object is no longer valid.3001 Invalid address for the return parameter was specified.

Parent

Property: Parent as Dialogs (read-only)

Description

Access the parent of the object.

Errors

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

ResolveEntities

Property: ResolveEntities as Boolean

Description

Shall all entities be resolved before generation starts? If yes, an info-set will be built.

Errors

3000 The object is no longer valid.

3001 Invalid address for the return parameter was specified.

TypeDetection

Property: TypeDetection as SPYTypeDetection

Description

Specifies granularity of simple type detection.

Errors

3000 The object is no longer valid.

3001 Invalid address for the return parameter was specified.

ValueList

Property: ValueList as Integer

Description

Generate not more than this amount of enumeration-facets per type. Set to - 1 for unlimited.

- 3000 The object is no longer valid.
- 3001 Invalid address for the return parameter was specified.

3.2.13 Element List

See also

Properties

Count Item

Methods

Rem oveElem ent

Description

Element lists are used for different purposes during export and import of data. Depending on this purpose, different properties of ElementListTem are used.

lt can hold

- a list of table names returned by a call to Application GetDatabaseTables,
- a list of field names retuned by a call to Application GetDatabase In portElem entList or Application GetTextIn portElem entList,
- a field name filter list used in Application In portFrom Database and Application In portFrom Text,
- a list of table names and counts for their rows and columns as returned by calls to GetExportElem entList Or
- a field name filter list used in Docum entExportToDatabase and Docum entExportToText.

Count

See also

Property: Count as long (read-only)

Description

Count of elements in this collection.

ltem

See also

Method: Item (n as long) as ElementListItem

Description

Gets the element with the index n from this collection. The first item has index 1.

RemoveElement

See also

Method: RemoveElement(Index as long)

Description

RemoveElement removes the element Index from the collection. The first Item has index 1.

3.2.14 ElementListItem

See also

Properties Name

ElementKind

FieldCount RecordCount

Description

An element in an ElementList. Usage of its properties depends on the purpose of the element list. For details see ElementList.

Element Kind

See also

Property: ElementKind as SPYXM LDataKind

Description

Specifies if a field should be imported as XML element (data value of spyXM LDataElement) or attribute (data value of spyXM LDataAttr).

FieldCount

See also

Property: Field ount as long (read-only)

Description

Count of fields (i.e. columns) in the table described by this element. This property is only valid after a call to Document GetExportElementList.

Name

See also

Property: Name as String (read-only)

Description

Name of the element. This is either the name of a table or a field, depending on the purpose of the element list.

RecordCount

See also

Property: RecordCount as long (read-only)

Description

Count of records (i.e. rows) in the table described by this element. This property is only valid after a call to DocumentGetExportElementList.

3.2.15 ExportSettings

See also

Properties

ElementList

EntitiesToText

ExportAlElements SubLevelLinit

From Attributes From SingleSubElements From TextValues

CreateKeys IndependentPrimaryKey

Namespace

ExportCompleteXML StartFromElement

Description

ExportSettings contains options used during export of XML data to a database or text file.

CreateKeys

See also

Property: CreateKeys as Boolean

Description

This property turns creation of keys (i.e. primary key and foreign key) on or off. Default is True.

Element List

See also

Property: ElementList as ElementList

Description

Default is empty list. This list of elements defines which fields will be exported. To get the list of available fields use DocumentGetExportElementList. It is possible to prevent exporting columns by removing elements from this list with ElementListRemoveElement before passing it to DocumentExportToDatabase or DocumentExportToText.

EntitiesToText

See also

Property: EntitiesToText as Boolean

Description

Defines if XML entities should be converted to text or left as they are during export. Default is True.

Export All Elements

See also

Property: ExportAllElements as Boolean

Description

If set to true, all elements in the document will be exported. If set to false, then ExportSettingsSubLevelLimit is used to restrict the number of sublevels to export. Default is true.

Export CompleteXML

See also

Property: ExportCompleteXML as Boolean

Description

Defines whether the complete XML is exported or only the element specified by <u>StartFromElement</u> and its children. Default is True.

FromAttributes

See also

Property: From Attributes as Boolean

Description

Set From Attributes to false if no export data should be created from attributes. Default is True.

FromSingleSubElements

See also

Property: From SingleSubElements as Boolean

Description

Set From SingleSubElements to false if no export data should be created from elements. Default is True.

FromTextValues

See also

Property: From TextValues as Boolean

Description

Set From TextValues to false if no export data should be created from text values. Default is True.

Independent PrimaryKey

See also

Property: IndependentPrimaryKey as Boolean

Description

Turns creation of independent primary key counter for every element on or off. If

ExportSettingsCreateKeys is False, this property will be ignored. Default is True.

Namespace

See also

Property: Namespace as SPYExportNamespace

Description

The default setting removes all namespace prefixes from the element names. In some database formats the colon is not a legal character. Default is spyNoNamespace.

Start From Element

See also

Property: StartFromElement as String

Description Specifies the start element for the export. This property is only considered when <u>ExportCompleteXML</u> is false.

SubLevelLimit

See also

Property: SubLevelim it as Integer

Description

Defines the number of sub levels to include for the export. Default is 0. This property is ignored if ExportSettings ExportAllElements is true.

3.2.16 FileSelectionDlg

See also

Properties and Methods

Standard automation properties Application Parent

Dialog properties FullName

Acceptance or cancellation of action that caused event DiabgAction

Description

The dialog object allows you to receive information about an event and pass back information to the event handler in the same way as with a user dialog. Use the FieselectionDigFulName to select or modify the file path and set the FieselectionDigDialogAction property to cancel or agree with the action that caused the event.

Application

See also

Property: Application as Application (read-only)

Description

Access the Authentic Desktop application object.

Errors

2400 The object is no longer valid.

2401 Invalid address for the return parameter was specified.

DialogAction

Property: DiabgAction as SPYDiabgAction

Description

If you want your script to perform the file selection operation without any user interaction necessary, simulate user interaction by either setting the property to *spyDialogOK(0)* or *spyDialogCancel(1)*.

To allow your script to fill in the default values but let the user see and react on the dialog, use the value *spyDialogUserInput(2)*. If you receive a FileSelectionDlg object in an event handler, *spyDialogUserInput(2)* is not supported and will be interpreted as *spyDialogOK(0)*.

Errors

- 2400 The object is no longer valid.
- 2401 Invalid value for dialog action or invalid address for the return parameter was specified.

FullName

Property: FullName as String

Description

Access the full path of the file the gets selected by the dialog. Most events that pass a

FileSelectionDlg object to you allow you modify this value and thus influence the action that caused the event (e.g. load or save to a different location).

Errors

- 2400 The object is no longer valid.
- 2401 Invalid address for the return parameter was specified.

Parent

See also

Property: Parent as Diabgs (read-only)

Description

Access the parent of the object.

- 2400 The object is no longer valid.
- 2401 Invalid address for the return parameter was specified.

3.2.17 FindInFilesDIg

See also

Properties and Methods

Standard automation properties Application Parent

Find RequbrExpression Replace DoReplace ReplaceOnDisk MatchWholeWord M atchC ase SearchLocation StartFolder IncludeSubfolders SearchhProjectFilesDoExternal FileExtension AdvancedXM LS earch XM LE lem entNam es XM LE lem entC ontents XM LAttributeNam es XM LAttributeContents XM LC om m ents XM LC Data XMLPI XM LRest ShowResult

Description

Use this object to configure the search (or replacement) for strings in files. The method <u>FindInFiles</u> expects a FindInFilesDIg as parameter.

AdvancedXMLSearch

Property: AdvancedXMLSearch as Boolean

Description

Specifies if the XML search properties (XMLElementNames, XMLElementContents, XMLAttributeNames, XMLAttributeContents, XMLComments, XMLCData, XMLPI and XMLRest) are considered. The default is false.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

Application

Property: Application as Application (read-only)

Description

Access the Authentic Desktop application object.

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

DoReplace

Property: DoReplace as Boolean

Description

Specifies if the matched string is replaced by the string defined in Replace. The default is false.

Errors

3500 The object is no longer valid.

3501 Invalid address for the return parameter was specified.

FileExtension

Property: FileExtension as String

Description

Specifies the file filter of the files that should be considered during the search. Multiple file filters must be delimited with a semicolon (eg: *.xml;*.dtd;a*.xsd). Use the wildcards * and ? to define the file filter.

Errors

3500 The object is no longer valid.3501 Invalid address for the return parameter was specified.

Find

Property: Find as String

Description

Specifies the string to search for.

Errors

3500 The object is no longer valid.3501 Invalid address for the return parameter was specified.

IncludeSubfolders

Property: IncludeSubfolders as Boolean

Description

Specifies if subfolders are searched too. The default is true.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

MatchCase

Property: MatchCase as Boolean

Description

Specifies if the search is case sensitive. The default is true.

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

MatchWholeWord

Property: MatchWholeWord as Boolean

Description

Specifies whether the whole word or just a part of it must match. The default is false.

Errors

3500 The object is no longer valid.

3501 Invalid address for the return parameter was specified.

Parent

Property: Parent as Diabgs (read-only)

Description

Access the parent of the object.

Errors

- 3500 The object is no longer valid.3501 Invalid address for the return parameter was specified.
- RegularExpression

Property: Regular Expression as Boolean

Description

Specifies if Find contains a regular expression. The default is false.

Errors

3500 The object is no longer valid.

3501 Invalid address for the return parameter was specified.

Replace

Property: Replace as String

Description

Specifies the replacement string. The matched string is only replaced if DoReplace is set true.

Errors

3500 The object is no longer valid.

3501 Invalid address for the return parameter was specified.

ReplaceOnDisk

Property: ReplaceOnDisk as Boolean

Description

Specifies if the replacement is done directly on disk. The modified file is not opened. The default is false.
- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

SearchInProjectFilesDoExternal

Property: SearchInProjectFilesDoExternal as Boolean

Description

Specifies if the external folders in the open project are searched, when a project search is performed. The default is false.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

SearchLocation

Property: SearchLocation as SPYFindInFilesSearchLocation

Description

Specifies the location of the search. The default is spyFindlnFiles_Documents.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

ShowResult

Property: ShowResult as Boolean

Description

Specifies if the result is displayed in the Find in Files output window. The default is false.

Errors

3500 The object is no longer valid.

3501 Invalid address for the return parameter was specified.

StartFolder

Property: StartFolder as String

Description

Specifies the folder where the disk search starts.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

XMLAttributeContents

Property: XMLAttributeContents as Boolean

Description

Specifies if attribute contents are searched when AdvancedXMLSearch is true. The default is true.

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

XMLAttributeNames

Property: XMLAttributeNames as Boolean

Description

Specifies if attribute names are searched when AdvancedXMLSearch is true. The default is true.

Errors

3500 The object is no longer valid.

3501 Invalid address for the return parameter was specified.

XMLCData

Property: XMLCData as Boolean

Description

Specifies if CData tags are searched when AdvancedXMLSearch is true. The default is true.

Errors

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

XMLComments

Property: XMLComments as Boolean

Description

Specifies if comments are searched when AdvancedXMLSearch is true. The default is true.

Errors

3500 The object is no longer valid.

3501 Invalid address for the return parameter was specified.

XMLElement Contents

Property: XMLElementContents as Boolean

Description

Specifies if element contents are searched when AdvancedXMLSearch is true. The default is true.

Errors

3500 The object is no longer valid.

3501 Invalid address for the return parameter was specified.

XMLElement Names

Property: XMLElement Names as Boolean

Description

Specifies if element names are searched when <u>AdvancedXMLSearch</u> is true. The default is true.

3500 The object is no longer valid.

3501 Invalid address for the return parameter was specified.

XMLPI

Property: XMLPI as Boolean

Description

Specifies if XML processing instructions are searched when <u>AdvancedXMLSearch</u> is true. The default is true.

Errors

3500 The object is no longer valid.

3501 Invalid address for the return parameter was specified.

XMLRest

Property: XMLRest as Boolean

Description

Specifies if the rest of the XML (which is not covered by the other XML search properties) is searched when <u>AdvancedXMLSearch</u> is true. The default is true.

- 3500 The object is no longer valid.
- 3501 Invalid address for the return parameter was specified.

3.2.18 FindInFilesResult

See also

Properties and Methods

Standard automation properties Application Parent

Count Item

<u>Path</u> Document

Description

This object represents a file that matched the search criteria. It contains a list of <u>FindInFilesResultMatch</u> objects that describe the matching position.

Application

Property: Application as Application (read-only)

Description

Access the Authentic Desktop application object.

Errors

3700 The object is no longer valid.3701 Invalid address for the return parameter was specified.

Count

Property: Count as long (read-only)

Description

Count of elements in this collection.

Document

Property: Path as Document (read-only)

Description

This property returns the **Document** object if the matched file is already open in XMLSpy.

Errors

3700 The object is no longer valid.

3701 Invalid address for the return parameter was specified.

ltem

Method: Item (n as long) as FindInFilesResultMatch

Description

Gets the element with the index n from this collection. The first item has index 1.

Parent

Property: Parent as FindInFilesResults (read-only)

Description

Access the parent of the object.

Errors

- 3700 The object is no longer valid.
- 3701 Invalid address for the return parameter was specified.

Path

Property: Path as String (read-only)

Description

Returns the path of the file that matched the search criteria.

- 3700 The object is no longer valid.
- 3701 Invalid address for the return parameter was specified.

3.2.19 FindInFilesResultMatch

See also

Properties and Methods

Standard automation properties Application Parent

Line Position Length LineText Replaced

Description

Contains the exact position in the file of the matched string.

Application

Property: Application as Application (read-only)

Description

Access the Authentic Desktop application object.

Errors

- 3800 The object is no longer valid.
- 3801 Invalid address for the return parameter was specified.

Length

Property: Length as Long (read-only)

Description

Returns the length of the matched string.

Errors

3800 The object is no longer valid.3801 Invalid address for the return parameter was specified.

Line

Property: Line as Long (read-only)

Description

Returns the line number of the match. The line numbering starts with 0.

3800	The object is no longer valid.
3801	Invalid address for the return parameter was specified.

LineText

Property: LineText as String (read-only)

Description

Returns the text of the line.

Errors

3800 The object is no longer valid.3801 Invalid address for the return parameter was specified.

Parent

Property: Parent as FindInFilesResult (read-only)

Description

Access the parent of the object.

Errors

3800	The object is no longer valid.
3801	Invalid address for the return parameter was specified.

Position

Property: Position as Long (read-only)

Description

Returns the start position of the match in the line. The position numbering starts with 0.

Errors

3800 The object is no longer valid.3801 Invalid address for the return parameter was specified.

Replaced

Property: Replaced as Boolean (read-only)

Description

True if the matched string was replaced.

- 3800 The object is no longer valid.
- 3801 Invalid address for the return parameter was specified.

3.2.20 FindInFilesResults

See also

Properties and Methods

Standard automation properties Application Parent

Count Item

Description

This is the result of the FindInFiles method. It is a list of FindInFilesResult objects.

Application

Property: Application as Application (read-only)

Description

Access the Authentic Desktop application object.

Errors

3600	The object is no longer valid.
3601	Invalid address for the return parameter was specified

Count

Property: Count as long (read-only)

Description

Count of elements in this collection.

Item

Method: Item (n as long) as FindInFilesResult

Description

Gets the element with the index n from this collection. The first item has index 1.

Parent

Property: Parent as Application (read-only)

Description

Access the parent of the object.

Errors

3600 The object is no longer valid.3601 Invalid address for the return parameter was specified.

3.2.21 GenerateSampleXMLDIg

See also

Properties and Methods

Standard automation properties Application Parent

NonMandatoryAttributes NonMandatoryElements RepeatCount FillAttributesWithSampleData FillElementsWithSampleData ContentOfNillableElementsIsNonMandatory TryToUseNonAbstractTypes SchemaOrDTDAssignment LocalNameOfRootElement NamespaceURIOfRootElement OptionsDialogAction

Properties that are no longer supported

TakeFirstChoice - obsolete FillWithSampleData - obsolete Optimization - obsolete

Description

Used to set the parameters for the generation of sample XML instances based on a W3C schema or DTD.

Application

Property: Application as Application (read-only)

Description

Access the Authentic Desktop application object.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

ChoiceMode

Property: ChoiceMode as <u>SPYSampleXMLGenerationChoiceMode</u>

Description

Specifies which elements will be generated.

Errors

(C) 2018 Altova GmbH

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

ConsiderSampleValueHints

Property: ConsiderSampleValueHints as Boolean

Description

Selects whether to use <u>SampleValueHints</u> or not.

Errors

2200 The object is no longer valid.2201 Invalid address for the return parameter was specified.

Content Of Nillable Element sIs Non Mandatory

Property: ContentO fN illableElem entsIsNonM andatory as Boolean

Description

If true, the contents of elements that are nillable will not be treated as mandatory.

Errors

2200 The object is no longer valid.

2201 Invalid address for the return parameter was specified.

FillAttributesWithSampleData

Property: FillAttributesW ithSam pleData as Boolean

Description

If true, attributes will have sample content.

Errors

2200 The object is no longer valid.2201 Invalid address for the return parameter was specified.

FillElementsWithSampleData

Property: FillElementsW ithSampleData as Boolean

Description

If true, elements will have sample content.

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

FillWithSampleData - obsolete

Property: Fill ithSam pleData as Boolean

Description

Do no longer access this property. Use <u>FilkttributesW ithSam pleData</u> and Filklem entsW ithSam pleData, instead.

Errors

0001 The property is no longer accessible.

LocalNameOfRoot Element

Property: LocaNameOfRootElement as String

Description

Specifies the local name of the root element for the generated sample XML.

Errors

2200 The object is no longer valid.

2201 Invalid address for the return parameter was specified.

NamespaceURIOfRoot Element

Property: NamespaceURD fRootElement as String

Description

Specifies the namespace URI of the root element for the generated sample XML.

Errors

2200 The object is no longer valid.

2201 Invalid address for the return parameter was specified.

NonMandatoryAttributes

Property: NonMandatoryAttributes as Boolean

Description

If true attributes which are not mandatory are created in the sample XML instance file.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

NonMandatoryElements

Property: NonMandatoryElements as Boolean

Description

If true, elements which are not mandatory are created in the sample XML instance file.

Errors

2200 The object is no longer valid.

2201 Invalid address was specified for the return parameter.

Optimization - obsolete

Property: Optim ization as SPYSam pleXM LG enerationOptim ization

Description

Do not use this property any longer. Use ChoiceMode and NonMandatoryElements.

Errors

0001 The property is no longer accessible.

OptionsDialogAction

Property: OptionsDiabgAction as SPYDiabgAction

Description

To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid value has been used to set the property. Invalid address for the return parameter was specified.

Parent

Property: Parent as Diabgs (read-only)

Description

Access the parent of the object.

Errors

2200 The object is no longer valid.

2201 Invalid address for the return parameter was specified.

Repeat Count

Property: RepeatCount as long

Description

Number of elements to create for repeated types.

Errors

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

SampleValueHints

Property: SampleValueHints as <u>SPYSampleXMLGenerationSampleValueHints</u>

Description

Specifies how to select data for the generated sample file.

Errors

2200 The object is no longer valid.

2201 Invalid address for the return parameter was specified.

SchemaOrDTDAssignment

Property: Schem aO nDTDAssignm ent as SPYSam pbXM LG enerationSchem aO nDTDAssignm ent

Description

Specifies in which way a reference to the related schema or DTD - which is this document - will be generated into the sample XML.

Errors

2200 The object is no longer valid.

2201 Invalid address for the return parameter was specified.

TakeFirstChoice - obsolete

Property: TakeFirstChoice as Boolean

Description

Do no longer use this property.

Errors

0001 The property is no longer accessible.

TryToUseNonAbstractTypes

Property: TryToUseNonAbstractTypes as Boolean

Description

If true, tries to use a non-abstract type for xsi:type, if element has an abstract type.

- 2200 The object is no longer valid.
- 2201 Invalid address for the return parameter was specified.

3.2.22 GridView

See also

Methods

Deselect Select

SetFocus

Properties

CurrentFocus

1sV isible

Description GridView Class

Events

OnBeforeDrag

See also

Event: OnBeforeDrag() as Boolean

XMLSpy scripting environment - VBScript:

```
Function On BeforeDrag()
'On BeforeStartEditing = False 'to prohibit dragging
End Function
```

XMLSpy scripting environment - JScript:

```
function On BeforeDrag()
{
    // return false; /* to prohibit dragging */
}
```

XMLSpy IDE Plugin: KMLSpyPlughOnEvent(4, ...) // nEventId = 4

Description

This event gets fired on an attempt to drag an XMLData element on the grid view. Return *false* to prevent dragging the data element to a different position.

OnBeforeDrop

See also

Event: OnBeforeDrop (ob <u>XM LData</u> as XM LData) as Boolean

XMLSpy scripting environment - VBScript: Function On_BeforeDrop(objXMLData)

'On BeforeStartEditing = False 'to prohibit dropping End Function

XMLSpy scripting environment - JScript:

function On BeforeDrop (ob JM LData)

{
 // return false; /* to prohibit dropping */
}

XMLSpy IDE Plugin:

KMLSpyPlugInOnEvent(5, ...) // nEventId = 5

Description

This event gets fired on an attempt to drop a previously dragged XMLData element on the grid view. Return *false* to prevent the data element to be moved from its original position to the drop destination position.

OnBeforeStartEditing

See also

Event: OnBeforeStartEditing (obj M LData as XM LData, bEditingName as Bookan) as Bookan

XMLSpy scripting environment - VBScript:

```
Function On BeforeStartEditing (objKM LData, bEditingName)
On BeforeStartEditing = False 'to prohibit editing the field
End Function
```

XMLSpy scripting environment - JScript:

```
function On BeforeStartEditing(objKMLData, bEditingName)
{
     // return false; /* to prohibit editing the field */
}
```

XMLSpy IDE Plugin:

KM LSpyP lugin OnEvent (1, ...) // nEventId = 1

Description

This event gets fired before the editing mode for a grid cell gets entered. If the parameter *bEditingName* is true, the name part of the element will be edited, it its value is false, the value part will be edited.

OnEditingFinished

See also

Event: OnEditingFinished (objXM LData as XM LData, bEditingName as Boolean)

```
XMLSpy scripting environment - VBScript:
Function On EditingFinished (objKM LData, bEditingName)
End Function
```

XMLSpy scripting environment - JScript: function On EditingFinished (obt MLData, bEditingName)

XMLSpy IDE Plugin:

KMLSpyPlugInOnEvent(2, ...) // nEventId = 2

{

Description

This event gets fired when the editing mode of a grid cell gets left. The parameter *bEditingName* specifies if the name part of the element has been edited.

OnFocusChanged

See also

Event: OnFocusChanged (*objXM LData* **as** <u>XM LData</u>, *bSetFocus* as Boolean, *bEditingNam* e as Boolean)

XMLSpy scripting environment - VBScript:

Function On FocusChanged (ob XM LData, bSetFocus, bEditingName) End Function

XMLSpy scripting environment - JScript:

function On_FocusChanged(objXMLData, bSetFocus, bEditingName)
{
}

XMLSpy IDE Plugin:

XM LSpyP lugIn OnEvent (3, ...) // nEventId = 3

Description

This event gets fired whenever a grid cell receives or looses the cursor focus. If the parameter *bEditingName* is *true*, focus of the name part of the grid element has changed. Otherwise, focus of the value part has changed.

CurrentFocus

See also

Property: CurrentFocus as XM LData

Description

Holds the XML element with the current focus. This property is read-only.

Deselect

See also

Method: Deselect (pData as XM LData)

 $\begin{array}{l} \textbf{Description} \\ \textbf{Deselects the element } p \texttt{Data in the grid view.} \end{array}$

IsVisible

See also

Property: Isvisible as Boolean

Description

True if the grid view is the active view of the document. This property is read-only.

Select

See also

Method: Select (pData as XM LData)

 $\begin{array}{l} \textbf{Description} \\ \textbf{Selects the XML element } p \texttt{Data in the grid view.} \end{array}$

SetFocus

See also

Method: SetFocus (pFocusData as XM LData)

Description

Sets the focus to the element pFocusData in the grid view.

3.2.23 SchemaDocumentationDlg

See also

Properties and Methods

Standard automation properties Application Parent

Interaction and visibility properties OutputFile OutputFileDiabgAction OptionsDiabgAction ShowProgressBar ShowResult

Document generation options and methods

OutputForm at UseFixedDesign SPSFile EmbedDiagram s Diagram Form at MultipleOutputFiles EmbedCSShHTML CreateDiagram sFolder GenerateRelativeLinks

hclideAll hclideIndex hclideGbbaAttributes hclideGbbaElements hclideLocaAttributes hclideLocaElements hclideComplexTypes hclideComplexTypes hclideSimpETypes hclideRedEfines hclideRedEfines hclideRedEfines

AlDetails ShowDiagram ShowNamespace ShowType ShowChildren ShowUsedBy ShowProperties ShowSingEFacets ShowPatterns ShowEnumerations ShowAttributes ShowDentityConstraints ShowAnnotations ShowSourceCode

Description

This object combines all options for schema document generation as they are available through user interface dialog boxes in Authentic Desktop. The document generation options are initialized with the values used during the last generation of schema documentation. However, before using the object you have to set the <u>SetOutputFile</u> property to a valid file path. Use <u>OptionsDiabgAction</u>, <u>OutputFileDiabgAction</u> and <u>ShowProgressBar</u> to specify the level of user interaction desired. You can use <u>IncludeAlland Alloctails</u> to set whole option groups at once or the individual properties to operate on a finer granularity.

AllDetails

See also

Method: Alletails (ibDetailsOn as Boolean)

Description

Use this method to turn all details options on or off.

Errors

2900 The object is no longer valid.

Application

See also

Property: Application as Application (read-only)

Description

Access the Authentic Desktop application object.

Errors

2900 The object is no longer valid.2901 Invalid address for the return parameter was specified.

CreateDiagramsFolder

See also

Property: CreateDiagram sFolder as Boolean

Description

Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The default for the first run is false.

Errors

2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

DiagramFormat

See also

Property: Diagram Form at as SPYImageKind

Description

This property specifies the generated diagram image type. This property is not available for HTML

documentation. The property is initialized with the value used during the last call to Docum ent. GenerateSchem aDocum entation. The default for the first run is PNG.

Errors

2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

EmbedCSSInHTML

See also

Property: EmbedCSSINHTML as Boolean

Description

Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The default for the first run is true.

Errors

2900	The object is no longer valid.
2901	Invalid address for the return parameter was specified.

.. .

EmbedDiagrams

See also

Property: Em bedDiagram s as Boolean

Description

Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateSchemaDocumentation</u>. The default for the first run is true.

Errors

2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

GenerateRelativeLinks

See also

Property: GenerateRelativeLinks as Boolean

Description

Set this property to true, to create relative paths to local files. This property is not available for HTML documentation. The property is initialized with the value used during the last call to Docum ent.GenerateSchem aDocum entation. The default for the first run is false.

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

IncludeAll

See also

Method: hcludeAll(ibhclude as Boolean)

Description

Use this method to mark or unmark all include options.

Errors

2900 The object is no longer valid.

IncludeAttributeGroups

See also

Property: IncludeAttributeGroups as Boolean

Description

Set this property to true, to include attribute groups in the schema documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateSchemaDocumentation</u>. The default for the first run is true.

Errors

2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

IncludeComplexTypes

See also

Property: IncludeComplexTypes as Boolean

Description

Set this property to true, to include complex types in the schema documentation. The property is initialized with the value used during the last call to DocumentGenerateSchemaDocumentation. The default for the first run is true.

Errors

2900 The object is no longer valid.2901 Invalid address for the return parameter was specified.

IncludeGlobalAttributes

See also

Property: IncludeG bbalAttributes as Boolean

Description

Set this property to true, to include global attributes in the schema documentation. The property is initialized with the value used during the last call to DocumentGenerateSchemaDocumentation.

The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

IncludeGlobalElements

See also

Property: hcludeG bballements as Boolean

Description

Set this property to true, to include global elements in the schema documentation. The property is initialized with the value used during the last call to DocumentGenerateSchemaDocumentation. The default for the first run is true.

Errors

2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

IncludeGroups

See also

Property: IncludeG roups as Boolean

Description

Set this property to true, to include groups in the schema documentation. The property is initialized with the value used during the last call to $\underline{\texttt{DocumentGenerateSchemaDocumentation}}$. The default for the first run is true.

Errors

2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

IncludeIndex

See also

Property: Include Index as Boolean

Description

Set this property to true, to include an index in the schema documentation. The property is initialized with the value used during the last call to DocumentGenerateSchemaDocumentation. The default for the first run is true.

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

IncludeLocalAttributes

See also

Property: IncludeLocalAttributes as Boolean

Description

Set this property to true, to include local attributes in the schema documentation. The property is initialized with the value used during the last call to Document.com. The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

IncludeLocalElements

See also

Property: IncludeLocaElements as Boolean

Description

Set this property to true, to include local elements in the schema documentation. The property is initialized with the value used during the last call to Document.com. The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

IncludeRedefines

See also

Property: hcludeRedefines as Boolean

Description

Set this property to true, to include redefines in the schema documentation. The property is initialized with the value used during the last call to DocumentGenerateSchemaDocumentation. The default for the first run is true.

Errors

2900 The object is no longer valid.2901 Invalid address for the return parameter was specified.

IncludeReferencedSchemas

See also

Property: IncludeReferencedSchemas as Boolean

Description

Set this property to true, to include referenced schemas in the schema documentation. The

property is initialized with the value used during the last call to Docum entGenerateSchem aDocum entation. The default for the first run is true.

Errors

2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

IncludeSimpleTypes

See also

Property: IncludeS in pleTypes as Boolean

Description

Set this property to true, to include simple types in the schema documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateSchemaDocumentation</u>. The default for the first run is true.

Errors

2900	The object is no longer valid.
2901	Invalid address for the return parameter was specified.

MultipleOutput Files

See also

Property: MultipleOutputFiles as Boolean

Description

Set this property to true, to split the documentation files. The property is initialized with the value used during the last call to <u>DocumentGenerateSchemaDocumentation</u>. The default for the first run is false.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid value has been used to set the property. Invalid address for the return parameter was specified.

OptionsDialogAction

See also

Property: OptionsDiabgAction as SPYDiabgAction

Description

To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

- 2900 The object is no longer valid.
- 2901 Invalid value has been used to set the property. Invalid address for the return parameter was specified.
- Altova Authentic 2018 Desktop Enterprise Edition

Out put File

See also

Property: OutputFile as String

Description

Full path and name of the file that will contain the generated documentation. In case of HTML output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to Docum ent.GenerateSchem aDocum entation.

Errors

2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

Output FileDialogAction

See also

Property: OutputFileDiabgAction as SPYDiabgAction

Description

To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in <u>Output File</u> should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid value has been used to set the property. Invalid address for the return parameter was specified.

Out put Format

See also

Property: OutputForm at as SPYSchem aDocum entationForm at

Description

Defines the kind of documentation that will be generated: HTML (value=0), MS-Word (value=1), or RTF (value=2). The property gets initialized with the value used during the last call to Docum entGenerateSchem aDocum entation. The default for the first run is HTML.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid value has been used to set the property. Invalid address for the return parameter was specified.

Parent

See also

Property: Parent as Diabgs (read-only)

Description

Access the parent of the object.

Errors

2900 The object is no longer valid.2901 Invalid address for the return parameter was specified.

ShowAnnot at ions

See also

Property: ShowAnnotations as Boolean

Description

Set this property to true, to show the annotations to a type definition in the schema documentation. The property is initialized with the value used during the last call to Docum ent. GenerateSchem aDocum entation. The default for the first run is true.

Errors

2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

ShowAttributes

See also

Property: ShowAttributes as Boolean

Description

Set this property to true, to show the type definitions attributes in the schema documentation. The property is initialized with the value used during the last call to DocumentGenerateSchemaDocumentation. The default for the first run is true.

Errors

2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

ShowChildren

See also

Property: ShowChildren as Boolean

Description

Set this property to true, to show the children of a type definition as links in the schema documentation. The property is initialized with the value used during the last call to Docum ent. GenerateSchem aDocum entation. The default for the first run is true.

2900	The object is no longer valid.
2901	Invalid address for the return parameter was specified.

ShowDiagram

See also

Property: ShowDiagram as Boolean

Description

Set this property to true, to show type definitions as diagrams in the schema documentation. The property is initialized with the value used during the last call to Docum entGenerateSchem aDocum entation. The default for the first run is true.

Errors

2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

ShowEnumerations

See also

Property: ShowEnum erations as Boolean

Description

Set this property to true, to show the enumerations contained in a type definition in the schema documentation. The property is initialized with the value used during the last call to Docum ent. GenerateSchem aDocum entation. The default for the first run is true.

Errors

2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

ShowIdentityConstraints

See also

Property: Show BentityConstraints as Boolean

Description

Set this property to true, to show a type definitions identity constraints in the schema documentation. The property is initialized with the value used during the last call to Docum ent.GenerateSchem aDocum entation. The default for the first run is true.

Errors

2900 The object is no longer valid.2901 Invalid address for the return parameter was specified.

ShowNamespace

See also

Property: ShowNamespace as Boolean

Description

Set this property to true, to show the namespace of type definitions in the schema documentation. The property is initialized with the value used during the last call to Docum ent.GenerateSchem aDocum entation. The default for the first run is true.

Errors

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

ShowPatterns

See also

Property: ShowPatterns as Boolean

Description

Set this property to true, to show the patterns of a type definition in the schema documentation. The property is initialized with the value used during the last call to Docum entGenerateSchem aDocum entation. The default for the first run is true.

Errors

2900 The object is no longer valid.2901 Invalid address for the return parameter was specified.

ShowProgressBar

See also

Property: Show ProgressBar as Boolean

Description

Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

Errors

2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

ShowProperties

See also

Property: Show Properties as Boolean

Description

Set this property to true, to show the type definition properties in the schema documentation. The property is initialized with the value used during the last call to Docum entGenerateSchem aDocum entation. The default for the first run is true.

- 2900 The object is no longer valid.
- 2901 Invalid address for the return parameter was specified.

ShowResult

See also

Property: ShowResult as Boolean

Description

Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in Authentic Desktop. To show Word documentation, MS- Word will be started. The property gets initialized with the value used during the last call to DocumentGenerateSchemaDocumentation. The default for the first run is true.

Errors

2900 The object is no longer valid.2901 Invalid address for the return parameter was specified.

ShowSingleFacets

See also

Property: ShowSingleFacets as Boolean

Description

Set this property to true, to show the facets of a type definition in the schema documentation. The property is initialized with the value used during the last call to Docum ent. GenerateSchem aDocum entation. The default for the first run is true.

Errors

2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

ShowSourceCode

See also

Property: ShowSourceCode as Boolean

Description

Set this property to true, to show the XML source code for type definitions in the schema documentation. The property is initialized with the value used during the last call to Docum ent.GenerateSchem aDocum entation. The default for the first run is true.

Errors

2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

ShowType

See also

Property: ShowType as Boolean

Description

Set this property to true, to show the type of type definitions in the schema documentation. The property is initialized with the value used during the last call to

Docum ent GenerateSchem aDocum entation. The default for the first run is true.

Errors

2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

ShowUsedBy

See also

Property: ShowUsedBy as Boolean

Description

Set this property to true, to show the used-by relation for type definitions in the schema documentation. The property is initialized with the value used during the last call to Docum entGenerateSchem aDocum entation. The default for the first run is true.

Errors

2900 The object is no longer valid.2901 Invalid address for the return parameter was specified.

SPSFile

See also

Property: SPSFile as String

Description

Full path and name of the SPS file that will be used to generate the documentation.

Errors

2900 The object is no longer valid.

2901 Invalid address for the return parameter was specified.

UseFixedDesign

See also

Property: UseFixedDesign as Boolean

Description

Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

2900	The object is no longer valid.
2901	Invalid address for the return parameter was specified.

3.2.24 SpyProject

See also

Methods

CbseProject SaveProject SaveProjectAs

Properties

Rootlems ProjectFile

Description SpyProject Class

CloseProject

See also

Declaration: CbseProject(DiscardChanges as Boolean, bCbseFiles as Boolean, bDiabg as Boolean)

Parameters

bDiscardC hanges

Set bD iscardC hanges to FALSE if you want to save the changes of the open project files and the project.

bC bseFiles Set bC bseFiles to TRUE to close all open project files.

bDiabg Show dialogs for user input.

Description

CbseProject closes the current project.

ProjectFile

See also

Declaration: ProjectFile as String

Description Path and filename of the project.

RootItems

See also

Declaration: Rootlems as SpyProjectlems

Description

Root level of collection of project items.

SaveProject

See also

Declaration: SaveProject

Description

SaveProject saves the current project.

SaveProjectAs

See also

Declaration: SaveProjectAs (strPath as String, bDiabg as Boolean)

Parameters

strPath Full path with file name of new project file.

bDiabg If bDiabg is TRUE, a file-dialog will be displayed.

Description

SaveProjectAs stores the project data into a new location.

3.2.25 SpyProjectItem

See also

Methods

0 pen

Properties

Childlem s Parentlem FileExtensions Item Type Nam e Path ValidateW ith XMLForXSLT ransform ation XSLForXMLT ransform ation XSLT ransform ationFileExtension XSLT ransform ationFoller

Description

SpyProjectItem Class

ChildItems

See also

Declaration: Childlems as SpyProjectlems

Description

If the item is a folder, ChildItem s is the collection of the folder content.

FileExtensions

See also

Declaration: FileExtensions as String

Description

Used to set the file extensions if the project item is a folder.

ItemType

See also

Declaration: Item Type as SPYProjectItem Types

Description This property is read-only.

Name

See also

Declaration: Name as String

Description Name of the project item. This property is read-only.

Open

See also

Declaration: Open as Document

Return Value The project item opened as document.

Description Opens the project item.

ParentItem

See also

Declaration: Parentitem as SpyProjectitem

Description

Parent item of the current project item. Can be NULL (Nothing) if the project item is a top-level item.

Path

See also

Declaration: Path as String

Description

Path of project item. This property is read-only.

ValidateWith

See also

Declaration: ValidateW ith as String

Description

Used to set the schema/ DTD for validation.

XMLForXSLTransformation

See also

Declaration: XMLForXSLTransform ation as String

Description Used to set the XML for XSL transformation.

XSLForXMLTransformation

See also

Declaration: XSLForXMLT ransform ation as String

Description

Used to set the XSL for XML transformation.

XSLTransformationFileExtension

See also

Declaration: XSLT ransform ationFileExtension as String

Description Used to set the file extension for XSL transformation output files.

XSLTransformationFolder

See also

Declaration: XSLT ransform ation Folder as String

Description

Used to set the destination folder for XSL transformation output files.

3.2.26 SpyProjectItems

See also

Methods

AddFile AddFolder AddURL Removeltem

Properties

Count Item

Description SpyProjectItem s Class

AddFile

See also

Declaration: AddFile (strPath as String)

Parameters

 ${\tt strPath}$ Full path with file name of new project item

Description

The method adds a new file to the collection of project items.

AddFolder

See also

Declaration: AddFobler (strName as String)

Parameters

striName Name of the new folder.

Description

The method AddFolder adds a folder with the name strName to the collection of project items.

AddURL

See also

Declaration: AddURL (*strURL* as String, *nURLType* as <u>SPYURLTypes</u>, *strUser* as String, *strPassword* as String, *bSave* as Boolean)

Description

strURL URL to open as document.

nURLType
Type of document to open. Set to -1 for auto detection.

strUser

Name of the user if required. Can be empty.

strPassword Password for authentification. Can be empty.

bS ave **Save user and password information**.

Description

The method adds an URL item to the project collection.

Count

See also

Declaration: Count as long

Description

This property gets the count of project items in the collection. The property is read-only.

ltem

See also

Declaration: Item (n as long) as SpyProjectItem

Description

Retrieves the n-th element of the collection of project items. The first item has index 1.

Removeltem

See also

Declaration: Remove Item (plem as SpyProjectItem)

Description

Removeltem deletes the item pltem from the collection of project items.

3.2.27 TextImportExportSettings

See also

Properties for import only m portFile

Properties for export only

DestinationFolder FileExtension CommentIncluded RemoveDelimiter RemoveNewline

Properties for import and export

HeaderRow FieldDelimiter EncobsingCharacter Encoding EncodingByteOrder

Description

TextIn portExportSettings contains options common to text import and export functions.

CommentIncluded

See also

Property: Comment Included as Boolean

Description

This property tells whether additional comments are added to the generated text file. Default is true. This property is used only when exporting to text files.

DestinationFolder

See also

Property: DestinationFolder as String

Description

The property DestinationFolder sets the folder where the created files are saved during text export.

EnclosingCharacter

See also

Property: EnclosingCharacter as SPYTextEnclosing

Description

This property defines the character that encloses all field values for import and export. Default is spyNoEncbsing.

Encoding

See also

Property: Encoding as String

Description

The property Encoding sets the character encoding for the text files for importing and exporting.

EncodingByteOrder

See also

Property: EncodingByteO rder as SPYEncodingByteO rder

Description

The property EncodingByteO rdersets the byte order for Unicode characters. Default is spyNONE.

FieldDelimiter

See also

Property: FieldDelim iter as SPYTextDelim iters

Description

The property FieldDelim iter defines the delimiter between the fields during import and export. Default is spyTabulator.

FileExtension

See also

Property: FileExtension as String

Description

This property sets the file extension for files created on text export.

HeaderRow

See also

Property: HeaderRow as Boolean

Description

The property HeaderRow is used during import and export. Set HeaderRow true on import, if the first line of the text file contains the names of the columns. Set HeaderRow true on export, if the first line in the created text files should contain the name of the columns. Default value is true.

Import File

See also

Property: In portFile as String

Description

This property is used to set the text file for import. The string has to be a full qualified path.

RemoveDelimiter

See also

Property: RemoveDelimiter as Boolean

Description

The property RemoveDelimiter defines whether characters in the text that are equal to the delimiter character are removed. Default is false. This property is used only when exporting to text files.

RemoveNewline

See also

Property: RemoveNewline as Boolean

Description

The property RemoveNewline defines whether newline characters in the text are removed. Default is false. This property is used only when exporting to text files.

3.2.28 TextView

See also

Properties and Methods

Application Parent

LineFromPosition PositionFromLine LineLength SelText GetRangeText ReplaceText MoveCaret GoToLineChar SelectText SelectionStart SelectionEnd Text LineCount Length

Description

Events

OnBeforeShowSuggestions

See also

Event: OnBeforeShowSuggestions () as Boolean

Description

This event gets fired before a suggestion window is shown. The <u>Document</u> property <u>Suggestions</u> contains a string array that is recommended to the user. It is possible to modify the displayed recommendations during this event. Before doing so you have to assign an empty array to the <u>Suggestions</u> property. The best location for this is the <u>OnDocumentOpened</u> event. To prevent the suggestion window to show up return false and true to continue its display.

Examples

Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:

```
Function On_BeforeShowSuggestions ()
End Function
```

XMLSpy scripting environment - JScript:

```
function On_BeforeShowSuggestions ()
{
}
XMLSpy IDE Plugin:
XM LSpyPlugh OnEvent (33, ...) // nE
```

// nEventId = 33

OnChar

See also

Event: OnChar (nChar as Long, bExistSuggestion as Boolean) as Boolean

Description

This event gets fired on each key stroke. The parameter nChar is the key that was pressed and bExist Suggestions tells whether a Authentic Desktop generated suggestions window is displayed after this key. The Document property Suggestions contains a string array that is recommended to the user. It is possible to modify the displayed recommendations during this event. Before doing so you have to assign an empty array to the Suggestions property. The best location for this is the OnDocumentOpened event. To prevent the suggestion window to show up return false and true to continue its display.

It is also possible to create a new suggestions window when none is provided by Authentic Desktop. Set the <u>Document</u> property <u>Suggestions</u> to a string array with your recommendations and return true.

This event is fired before the <u>OnBeforeShowSuggestions</u> event. If you prevent to show the suggestion window by returning false then <u>OnBeforeShowSuggestions</u> is not fired.

Examples

Given below are examples of how this event can be scripted.

XMLSpy scripting environment - VBScript:

Function On_Char (nChar, bExistSuggestions) End Function

XMLSpy scripting environment - JScript:

function On_Char (hChar, bExistSuggestions)
{
}

XMLSpy IDE Plugin:

KMLSpyPlughOnEvent(35, ...) // nEventId = 35

Application

Property: Application as Application (read-only)

Description

Access the Authentic Desktop application object.

Errors

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

GetRangeText

Method: Get Range Text (nStart as Long, nEnd as Long) as String

Description

Returns the text in the specified range.

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

GoToLineChar

Method: GoToLineChar(nLine as Long, nChar as Long)

Description

Moves the caret to the specified line and character position.

Errors

3900 The object is no longer valid.3901 Invalid address for the return parameter was specified.

Length

Property: Length as Long

Description

Returns the character count of the document.

Errors

3900 The object is no longer valid.3901 Invalid address for the return parameter was specified.

LineCount

Property: LineCount as Long

Description

Returns the number of lines in the document.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

LineFromPosition

Method: LineFromPosition(nCharPos as Long) as Long

Description

Returns the line number of the character position.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

LineLength

Method: LineLength(nLine as Long) as Long

Description

Returns the length of the line.

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

MoveCaret

Method: MoveCaret(nDiff as Long)

Description

Moves the caret nDiff characters.

Errors

3900 The object is no longer valid.3901 Invalid address for the return parameter was specified.

Parent

Property: Parent as **Document** (read-only)

Description

Access the parent of the object.

Errors

3900 The object is no longer valid.3901 Invalid address for the return parameter was specified.

PositionFromLine

Method: PositionFromLine(nLine as Long) as Long

Description

Returns the start position of the line.

Errors

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

ReplaceText

Method: ReplaceText(nPosFrom as Long, nPosTill as Long, sText as String)

Description

Replaces the text in the specified range.

Errors

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

SelectionEnd

Property: SelectionEnd as Long

Description

Returns/sets the text selection end position.

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

SelectionStart

Property: SelectionStart as Long

Description

Returns/sets the text selection start position.

Errors

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

Select Text

Method: SelectText(nPosFrom as Long, nPosTill as Long)

Description

Selects the text in the specified range.

Errors

3900 The object is no longer valid.3901 Invalid address for the return parameter was specified.

SelText

Property: SelText as String

Description

Returns/sets the selected text.

Errors

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

Text

Property: Text as String

Description

Returns/sets the document text.

Errors

3900 The object is no longer valid.3901 Invalid address for the return parameter was specified.

(C) 2018 Altova GmbH

3.2.29 WSDLDocumentationDlg

See also

Properties and Methods

Standard automation properties <u>Application</u> <u>Parent</u>

Interaction and visibility properties

GlobalElementsAndTypesOnly OptionsDialogAction OutputFile OutputFileDialogAction SeparateSchemaDocument ShowProgressBar ShowResult

Document generation options and methods

OutputFormat UseFixedDesign SPSFile EmbedDiagrams DiagramFormat MultipleOutputFiles EmbedCSSInHTML CreateDiagramsFolder

IncludeAll IncludeBinding IncludeImportedWSDLFiles IncludeMessages IncludeOverview IncludePortType IncludeService IncludeTypes

AllDetails ShowBindingDiagram ShowExtensibility ShowMessageParts ShowPort ShowPortTypeDiagram ShowServiceDiagram ShowServiceDiagram ShowSourceCode ShowTypesDiagram ShowUsedBy

Description

This object combines all options for WSDL document generation as they are available through user interface dialog boxes in Authentic Desktop. The document generation options are initialized with the values used during the last generation of WSDL documentation. However, before using the object you have to set the <u>OutputFile</u> property to a valid file path. Use <u>OptionsDialogAction</u>, <u>OutputFileDialogAction</u> and <u>ShowProgressBar</u> to specify the level of user interaction desired. You can use <u>IncludeAll</u> and <u>AllDetails</u> to set whole option groups at once or the individual properties to operate on a finer granularity.

AllDetails

See also

Method: Alletails (ibDetailsOn as Boolean)

Description

Use this method to turn all details options on or off.

Errors

4300 The object is no longer valid.

Application

See also

Property: Application as Application (read-only)

Description

Access the Authentic Desktop application object.

Errors

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

CreateDiagramsFolder

See also

Property: C reateD ingram sFolder as Boolean

Description

Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The property is initialized with the value used during the last call to DocumentGenerateW SDLDocumentation. The default for the first run is false.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

DiagramFormat

See also

Property: Diagram Form at as SPYImageKind

Description

This property specifies the generated diagram image type. This property is not available for HTML documentation. The property is initialized with the value used during the last call to DocumentGenerateW SDLDocumentation. The default for the first run is PNG.

Errors

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

EmbedCSSInHTML

See also

Property: EmbedCSSIhHTML as Boolean

Description

Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The property is initialized with the value used during the last call to DocumentGenerateWSDLDocumentation. The default for the first run is true.

Errors

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

EmbedDiagrams

See also

Property: EmbedDiagram s as Boolean

Description

Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to DocumentGenerateWSDLDocumentation. The default for the first run is true.

Errors

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

GlobalElementsAndTypesOnly

See also

Property: G bballementsAndTypesOnly as Boolean

Description

Returns/sets a value indicating whether a full Schema documentation is done or only Global Elements and Types are documented.

Errors

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

IncludeAll

See also

Method: hcludeAll(ibhclude as Boolean)

Description

Use this method to mark or unmark all include options.

Errors

4300 The object is no longer valid.

IncludeBinding

See also

Property: hcludeBinding as Boolean

Description

Set this property to true, to include bindings in the WSDL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateWSDLDocumentation</u>. The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

IncludeImportedWSDLFiles

See also

Property: hclude Im portedW SDLFiles as Boolean

Description

Set this property to true, to include imported WSDL finles in the WSDL documentation. The property is initialized with the value used during the last call to DocumentGenerateWSDLDocumentation. The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

IncludeMessages

See also

Property: IncludeMessages as Boolean

Description

Set this property to true, to include messages in the WSDL documentation. The property is

initialized with the value used during the last call to <u>Docum entGenerateW SDLDocum entation</u>. The default for the first run is true.

Errors

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

IncludeOverview

See also

Property: hcludeOverview as Boolean

Description

Set this property to true, to include an overview in the WSDL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateWSDLDocumentation</u>. The default for the first run is true.

Errors

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

IncludePortType

See also

Property: IncludePortType as Boolean

Description

Set this property to true, to include port types in the WSDL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateWSDLDocumentation</u>. The default for the first run is true.

Errors

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

IncludeService

See also

Property: IncludeService as Boolean

Description

Set this property to true, to include services in the WSDL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateWSDLDocumentation</u>. The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

IncludeTypes

See also

Property: IncludeTypes as Boolean

Description

Set this property to true, to include types in the WSDL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateWSDLDocumentation</u>. The default for the first run is true.

Errors

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

MultipleOut put Files

See also

Property: MultipleOutputFiles as Boolean

Description

Set this property to true, to split the documentation files. The property is initialized with the value used during the last call to <u>DocumentGenerateWSDLDocumentation</u>. The default for the first run is false.

Errors

3900 The object is no longer valid.
3901 Invalid value has been used to set the property. Invalid address for the return parameter was specified.

OptionsDialogAction

See also

Property: OptionsDiabgAction as SPYDiabgAction

Description

To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors

3900 The object is no longer valid.

3901 Invalid value has been used to set the property. Invalid address for the return parameter was specified.

Out put File

See also

Property: OutputFile as String

Description

Full path and name of the file that will contain the generated documentation. In case of HTML

output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to Docum ent.GenerateW SDLDocum entation.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

Output FileDialogAction

See also

Property: OutputFibDiabgAction as SPYDiabgAction

Description

To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in <u>OutputFile</u> should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid value has been used to set the property. Invalid address for the return parameter was specified.

Output Format

See also

Property: OutputForm at as SPYSchem aDocum entationForm at

Description

Defines the kind of documentation that will be generated: HTML (value=0), MS-Word (value=1), or RTF (value=2). The property gets initialized with the value used during the last call to DocumentGenerateWSDLDocumentation. The default for the first run is HTML.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid value has been used to set the property. Invalid address for the return parameter was specified.

Parent

See also

Property: Parent as Diabogs (read-only)

Description

Access the parent of the object.

Errors

3900	The object is no longer valid.
3901	Invalid address for the return parameter was specified.

SeparateSchemaDocument

See also

Property: SeparateSchem aDocum ent **as** Boolean

Description

Returns/sets a value indicating whether the Schema documentation should be placed in a separate document.

Errors

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

ShowBindingDiagram

See also

Property: ShowBindingDiagram as Boolean

Description

Set this property to true, to show binding diagrams in the WSDL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateWSDLDocumentation</u>. The default for the first run is true.

Errors

3900 The object is no longer valid.3901 Invalid address for the return parameter was specified.

ShowExt ensibility

See also

Property: ShowExtensibility as Boolean

Description

Set this property to true, to show service and binding extensibilities in the WSDL documentation. The property is initialized with the value used during the last call to DocumentGenerateWSDLDocumentation. The default for the first run is true.

Errors

3900 The object is no longer valid.3901 Invalid address for the return parameter was specified.

Show Message Parts

See also

Property: ShowMessageParts as Boolean

Description

Set this property to true, to show message parts of messges in the WSDL documentation. The property is initialized with the value used during the last call to DocumentGenerateWSDLDocumentation. The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

ShowPort

See also

Property: ShowPort as Boolean

Description

Set this property to true, to show service ports in the WSDL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateWSDLDocumentation</u>. The default for the first run is true.

Errors

3900 The object is no longer valid.3901 Invalid address for the return parameter was specified.

ShowPortTypeDiagram

See also

Property: ShowPortTypeDiagram as Boolean

Description

Set this property to true, to show port type diagrams in the WSDL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateWSDLDocumentation</u>. The default for the first run is true.

Errors

3900 The object is no longer valid.3901 Invalid address for the return parameter was specified.

ShowPortTypeOperations 87

See also

Property: ShowPortTypeOperations as Boolean

Description

Set this property to true, to show port type operations in the WSDL documentation. The property is initialized with the value used during the last call to DocumentGenerateWSDLDocumentation. The default for the first run is true.

Errors

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

ShowProgressBar

See also

Property: Show ProgressBar as Boolean

Description

Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

Errors

3900 The object is no longer valid.3901 Invalid address for the return parameter was specified.

ShowResult

See also

Property: ShowResult as Boolean

Description

Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in Authentic Desktop. To show Word documentation, MS- Word will be started. The property gets initialized with the value used during the last call to DocumentGenerateWSDLDocumentation. The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

ShowServiceDiagram

See also

Property: ShowServiceDiagram as Boolean

Description

Set this property to true, to show service diagrams in the WSDL documentation. The property is initialized with the value used during the last call to Documentation. The property is initialized with the value used during the last call to Documentation. The property is initialized with the value used during the last call to Documentation. The property is initialized with the value used during the last call to Documentation. The default for the first run is true.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

ShowSourceCode

See also

Property: ShowSourceCode as Boolean

Description

Set this property to true, to show source code for the includes in the WSDL documentation. The property is initialized with the value used during the last call to

DocumentGenerateWSDLDocumentation. The default for the first run is true.

Errors

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

ShowTypesDiagram

See also

Property: Show Types Diagram as Boolean

Description

Set this property to true, to show type diagrams in the WSDL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateWSDLDocumentation</u>. The default for the first run is true.

Errors

3900 The object is no longer valid.3901 Invalid address for the return parameter was specified.

ShowUsedBy

See also

Property: ShowUsedBy as Boolean

Description

Set this property to true, to show the used-by relation for types, bindings and messages definitions in the WSDL documentation. The property is initialized with the value used during the last call to DocumentGenerateWSDLDocumentation. The default for the first run is true.

Errors

3900 The object is no longer valid.3901 Invalid address for the return parameter was specified.

UseFixedDesign

See also

Property: UseFixedDesign as Boolean

Description

Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

Errors

3900 The object is no longer valid.

3901 Invalid address for the return parameter was specified.

SPSFile

See also

Property: SPSFile as String

Description

Full path and name of the SPS file that will be used to generate the documentation.

Errors

3900 The object is no longer valid.3901 Invalid address for the return parameter was specified.

3.2.30 WSDL20DocumentationDlg

See also

Properties and Methods

Standard automation properties <u>Application</u> <u>Parent</u>

Interaction and visibility properties

GlobalElementsAndTypesOnly OptionsDialogAction OutputFile OutputFileDialogAction SeparateSchemaDocument ShowProgressBar ShowResult

Document generation options and methods

OutputFormat UseFixedDesign SPSFile EmbedDiagrams DiagramFormat MultipleOutputFiles EmbedCSSInHTML CreateDiagramsFolder

IncludeAll IncludeBinding IncludeImportedWSDLFiles IncludeInterface IncludeOverview IncludeService IncludeTypes

AllDetails ShowBindingDiagram ShowExtensibility ShowEndpoint ShowFault ShowInterfaceDiagram ShowOperation ShowServiceDiagram ShowSourceCode ShowTypesDiagram ShowUsedBy

Description

This object combines all options for WSDL document generation as they are available through user interface dialog boxes in Authentic Desktop. The document generation options are initialized with the values used during the last generation of WSDL documentation. However, before using the object you have to set the <u>OutputFile</u> property to a valid file path. Use <u>OptionsDialogAction</u>, <u>OutputFileDialogAction</u> and <u>ShowProgressBar</u> to specify the level of user interaction desired. You can use <u>IncludeAll</u> and <u>AllDetails</u> to set whole option groups at once or the individual properties to operate on a finer granularity.

AllDetails

See also

Method: AlDetails (ibDetailsOn as Boolean)

Description

Use this method to turn all details options on or off.

Errors

4300 The object is no longer valid.

Application

See also

Property: Application as Application (read-only)

Description

Access the Authentic Desktop application object.

Errors

4300 The object is no longer valid.4301 Invalid address for the return parameter was specified.

CreateDiagramsFolder

See also

Property: CreateDiagram sFolder as Boolean

Description

Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The property is initialized with the value used during the last call to Docum entGenerateW SD20LDocum entation. The default for the first run is false.

Errors

4300 The object is no longer valid.

4301 Invalid address for the return parameter was specified.

DiagramFormat

See also

Property: Diagram Form at as SPYImageKind

Description

This property specifies the generated diagram image type. This property is not available for HTML

documentation. The property is initialized with the value used during the last call to DocumentGenerateWSDL20Documentation. The default for the first run is PNG.

Errors

4300 The object is no longer valid.

4301 Invalid address for the return parameter was specified.

EmbedCSSInHTML

See also

Property: EmbedCSSINHTML as Boolean

Description

Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The property is initialized with the value used during the last call to DocumentGenerateWSDL20Documentation. The default for the first run is true.

Errors

4300 The object is no longer valid.4301 Invalid address for the return parameter was specified.

EmbedDiagrams

See also

Property: Em bedDiagram s as Boolean

Description

Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to DocumentGenerateWSDL20Documentation. The default for the first run is true.

Errors

4300 The object is no longer valid.4301 Invalid address for the return parameter was specified.

GlobalElement sAndTypesOnly

See also

Property: G bbaElem entsAndTypesOnly as Boolean

Description

Returns/sets a value indicating whether a full Schema documentation is done or only Global Elements and Types are documented.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

IncludeAll

See also

Method: hcludeAll(ibhclude as Boolean)

Description

Use this method to mark or unmark all include options.

Errors

4300 The object is no longer valid.

IncludeBinding

See also

Property: IncludeBinding as Boolean

Description

Set this property to true, to include bindings in the WSDL documentation. The property is initialized with the value used during the last call to Document.com https://www.document.com https://www.document.com"/>https://www.document.com https://www.document.com"/>https://www.document.com https://www.document.com"/>https://www.document.com https://www.document.com"/>https://wwww.document.com <a href="ht

Errors

4300 The object is no longer valid.

4301 Invalid address for the return parameter was specified.

IncludeImportedWSDLFiles

See also

Property: Include In portedW SDLFiles as Boolean

Description

Set this property to true, to include imported WSDL finles in the WSDL documentation. The property is initialized with the value used during the last call to DocumentGenerateWSDL20Documentation. The default for the first run is true.

Errors

4300 The object is no longer valid.4301 Invalid address for the return parameter was specified.

IncludeInterface

See also

Property: Include Interface as Boolean

Description

Set this property to true, to include interfaces in the WSDL documentation. The property is initialized with the value used during the last call to DocumentGenerateWSDL20Documentation.

The default for the first run is true.

Errors

4300 The object is no longer valid.

4301 Invalid address for the return parameter was specified.

IncludeOverview

See also

Property: IncludeOverview as Boolean

Description

Set this property to true, to include an overview in the WSDL documentation. The property is initialized with the value used during the last call to Documentation. The default for the first run is true.

Errors

4300 The object is no longer valid.4301 Invalid address for the return parameter was specified.

IncludeService

See also

Property: IncludeService as Boolean

Description

Set this property to true, to include services in the WSDL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateWSDL20Documentation</u>. The default for the first run is true.

Errors

4300 The object is no longer valid.

4301 Invalid address for the return parameter was specified.

IncludeTypes

See also

Property: hcludeTypes as Boolean

Description

Set this property to true, to include types in the WSDL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateWSDL20Documentation</u>. The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

MultipleOutput Files

See also

Property: MultipleOutputFiles as Boolean

Description

Set this property to true, to split the documentation files. The property is initialized with the value used during the last call to <u>DocumentGenerateWSDL20Documentation</u>. The default for the first run is false.

Errors

4300 The object is no longer valid.

4301 Invalid value has been used to set the property. Invalid address for the return parameter was specified.

OptionsDialogAction

See also

Property: OptionsDiabgAction as SPYDiabgAction

Description

To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid value has been used to set the property. Invalid address for the return parameter was specified.

Out put File

See also

Property: OutputFile as String

Description

Full path and name of the file that will contain the generated documentation. In case of HTML output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to DocumentGenerateWSDL20Documentation.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

Output FileDialogAction

See also

Property: OutputFileDialogAction as SPYDialogAction

Description

To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in <u>OutputFile</u> should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

- 4300 The object is no longer valid.
- 4301 Invalid value has been used to set the property.
 - Invalid address for the return parameter was specified.

Out put Format

See also

Property: OutputForm at as SPYSchem aDocum entationForm at

Description

Defines the kind of documentation that will be generated: HTML (value=0), MS-Word (value=1), or RTF (value=2). The property gets initialized with the value used during the last call to DocumentGenerateWSDL20Documentation. The default for the first run is HTML.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid value has been used to set the property. Invalid address for the return parameter was specified.

Parent

See also

Property: Parent as Diabgs (read-only)

Description

Access the parent of the object.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

SeparateSchemaDocument

See also

Property: SeparateSchem aDocum ent as Boolean

Description

Returns/sets a value indicating whether the Schema documentation should be placed in a separate document.

Errors

- 3900 The object is no longer valid.
- 3901 Invalid address for the return parameter was specified.

ShowBindingDiagram

See also

Property: ShowBindingDiagram as Boolean

Description

Set this property to true, to show binding diagrams in the WSDL documentation. The property is initialized with the value used during the last call to Documentation.com Documentation. The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

ShowEndpoint

See also

Property: ShowEndpoint as Boolean

Description

Set this property to true, to show service endpoints in the WSDL documentation. The property is initialized with the value used during the last call to DocumentGenerateWSDL20Documentation. The default for the first run is true.

Errors

4300 The object is no longer valid.

4301 Invalid address for the return parameter was specified.

ShowExt ensibility

See also

Property: ShowExtensibility as Boolean

Description

Set this property to true, to show service and binding extensibilities in the WSDL documentation. The property is initialized with the value used during the last call to Docum entGenerateW SDL20Docum entation. The default for the first run is true.

Errors

4300 The object is no longer valid.4301 Invalid address for the return parameter was specified.

ShowFault

See also

Property: ShowFault as Boolean

Description

Set this property to true, to show faults in the WSDL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateWSDL20Documentation</u>. The default for the first run is true.

Errors

4300 The object is no longer valid.

4301 Invalid address for the return parameter was specified.

ShowInterfaceDiagram

See also

Property: Show Interface Diagram as Boolean

Description

Set this property to true, to show interface diagrams in the WSDL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateWSDL20Documentation</u>. The default for the first run is true.

Errors

4300 The object is no longer valid.4301 Invalid address for the return parameter was specified.

ShowOperation

See also

Property: ShowOperation as Boolean

Description

Set this property to true, to show interface and binding operations in the WSDL documentation. The property is initialized with the value used during the last call to DocumentGenerateWSDL20Documentation. The default for the first run is true.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

ShowProgressBar

See also

Property: Show ProgressBar as Boolean

Description

Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

ShowResult

See also

Property: ShowResult as Boolean

Description

Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in Authentic Desktop. To show Word documentation, MS- Word will be started. The property gets initialized with the value used during the last call to DocumentGenerateWSDL20Documentation. The default for the first run is true.

Errors

4300 The object is no longer valid.

4301 Invalid address for the return parameter was specified.

ShowServiceDiagram

See also

Property: ShowServiceDiagram as Boolean

Description

Set this property to true, to show service diagrams in the WSDL documentation. The property is initialized with the value used during the last call to $\underline{DocumentGenerateWSDL20Documentation}$. The default for the first run is true.

Errors

4300 The object is no longer valid.

4301 Invalid address for the return parameter was specified.

ShowSourceCode

See also

Property: ShowSourceCode as Boolean

Description

Set this property to true, to show source code for the includes in the WSDL documentation. The property is initialized with the value used during the last call to DocumentGenerateWSDL20Documentation. The default for the first run is true.

Errors

4300 The object is no longer valid.4301 Invalid address for the return parameter was specified.

ShowTypesDiagram

See also

Property: ShowTypesDiagram as Boolean

Description

Set this property to true, to show type diagrams in the WSDL documentation. The property is initialized with the value used during the last call to Document.com https://www.document.com https://www.document.com"/>https://www.document.com https://www.document.com"/>https://www.document.com https://www.document.com"/>https://www.document.com https://www.document.com"/>https://wwww.document.com <a href="

Errors

4300 The object is no longer valid.

4301 Invalid address for the return parameter was specified.

ShowUsedBy

See also

Property: ShowUsedBy as Boolean

Description

Set this property to true, to show the used-by relation for types, bindings and messages definitions in the WSDL documentation. The property is initialized with the value used during the last call to DocumentGenerateWSDL20Documentation. The default for the first run is true.

Errors

4300 The object is no longer valid.4301 Invalid address for the return parameter was specified.

SPSFile

See also

Property: SPSFile as String

Description

Full path and name of the SPS file that will be used to generate the documentation.

Errors

4300 The object is no longer valid.

4301 Invalid address for the return parameter was specified.

UseFixedDesign

See also

Property: UseFixedDesign as Boolean

Description

Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

Errors

- 4300 The object is no longer valid.
- 4301 Invalid address for the return parameter was specified.

3.2.31 XBRLDocumentationDlg

See also

Properties and Methods

Standard automation properties Application Parent

Interaction and visibility properties

OptionsDiabgAction OutputFile OutputFileDiabgAction ShowProgressBar ShowResult

Document generation options and methods

OutputForm at UseFixedDesign SPSFile EmbedDiagram s Diagram Form at EmbedCSSIhHTML CreateDiagram sFolder

hchdeAll hchdeOverview hchdeOverview hchdeG bbaElements hchdeG bbaElements hchdeDefnitionLinkroles hchdePresentationLinkroles hchdeCabulationLinkroles

AlDetails ShowDiagram ShowSubstitutiongroup ShowItem type ShowBalance ShowPeriod ShowAbstract ShowNillable ShowLabels ShowReferences ShowLinkbaseReferences

ShortQualifiedName ShowImportedElements

Description

This object combines all options for XBRL document generation as they are available through user interface dialog boxes in Authentic Desktop. The document generation options are initialized with the values used during the last generation of XBRL documentation. However, before using the object you have to set the <u>OutputFile</u> property to a valid file path. Use <u>OptionsDialogAction</u>, <u>OutputFileDialogAction</u> and <u>ShowProgressBar</u> to specify the level of user interaction desired. You can use <u>IncludeAlland Alloctais</u> to set whole option groups at once or the individual properties to operate on a finer granularity.

AllDetails

See also

Method: AllDetails (ibDetailsOn as Boolean)

Description

Use this method to turn all details options on or off.

Errors

4400 The object is no longer valid.

Application

See also

Property: Application as Application (read-only)

Description

Access the Authentic Desktop application object.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

CreateDiagramsFolder

See also

Property: CreateDiagram sFolder as Boolean

Description

Set this property to true, to create a directory for the created images. Otherwise the diagrams will be created next to the documentation. This property is only available when the diagrams are not embedded. The property is initialized with the value used during the last call to Docum ent.GenerateXBRLDocum entation. The default for the first run is false.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

DiagramFormat

See also

Property: Diagram Form at as <u>SPYImageKind</u>

Description

This property specifies the generated diagram image type. This property is not available for HTML documentation. The property is initialized with the value used during the last call to DocumentGenerateXBRLDocumentation. The default for the first run is PNG.

4400 The object is no longer valid.

4401 Invalid address for the return parameter was specified.

EmbedCSSInHTML

See also

Property: EmbedCSSINHTML as Boolean

Description

Set this property to true, to embed the CSS data in the generated HTML document. Otherwise a separate file will be created and linked. This property is only available for HTML documentation. The property is initialized with the value used during the last call to DocumentGenerateXBRLDocumentation. The default for the first run is true.

Errors

4400 The object is no longer valid.4401 Invalid address for the return parameter was specified.

EmbedDiagrams

See also

Property: Em bedDiagram s as Boolean

Description

Set this property to true, to embed the diagrams in the generated document. This property is not available for HTML documentation. The property is initialized with the value used during the last call to DocumentGenerateXBRLDocumentation. The default for the first run is true.

Errors

4400 The object is no longer valid.

4401 Invalid address for the return parameter was specified.

IncludeAll

See also

Method: hcludeAll(ibhclude as Boolean)

Description

Use this method to mark or unmark all include options.

Errors

4400 The object is no longer valid.

IncludeCalculationLinkroles

See also

Property: IncludeCalculationLinkroles as Boolean

Description

Set this property to true, to include calculation linkroles in the XBRL documentation. The property is initialized with the value used during the last call to Documentation.com The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

IncludeDefinitionLinkroles

See also

Property: IncludeDefinitionLinkroles as Boolean

Description

Set this property to true, to include definition linkroles in the XBRL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateXBRLDocumentation</u>. The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

IncludeGlobalElements

See also

Property: IncludeG bballem ents as Boolean

Description

Set this property to true, to include global elements in the XBRL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateXBRLDocumentation</u>. The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

IncludeNamespacePrefixes

See also

Property: IncludeNamespacePrefixes as Boolean

Description

Set this property to true, to include namespace prefixes in the XBRL documentation. The property is initialized with the value used during the last call to DocumentGenerateXBRLDocumentation.
The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

IncludeOverview

See also

Property: IncludeOverview as Boolean

Description

Set this property to true, to include an overview in the XBRL documentation. The property is initialized with the value used during the last call to DocumentGenerateXBRLDocumentation. The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

IncludePresentationLinkroles

See also

Property: IncludePresentationLinkroles as Boolean

Description

Set this property to true, to include presentation linkroles in the XBRL documentation. The property is initialized with the value used during the last call to DocumentGenerateXBRLDocumentation. The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

OptionsDialogAction

See also

Property: OptionsDiabgAction as SPYDiabgAction

Description

To allow your script to fill in the default values and let the user see and react on the dialog, set this property to the value *spyDialogUserInput(2)*. If you want your script to define all the options in the schema documentation dialog without any user interaction necessary, use *spyDialogOK(0)*. Default is *spyDialogOK*.

- 4400 The object is no longer valid.
- 4401 Invalid value has been used to set the property. Invalid address for the return parameter was specified.

Out put File

See also

Property: OutputFile as String

Description

Full path and name of the file that will contain the generated documentation. In case of HTML output, additional '.png' files will be generated based on this filename. The default value for this property is an empty string and needs to be replaced before using this object in a call to Docum ent.GenerateXBRLDocum entation.

Errors

4400 The object is no longer valid.

4401 Invalid address for the return parameter was specified.

Output FileDialogAction

See also

Property: OutputFileDiabgAction as SPYDiabgAction

Description

To allow the user to select the output file with a file selection dialog, set this property to *spyDialogUserInput(2)*. If the value stored in <u>Output File</u> should be taken and no user interaction should occur, use *spyDialogOK(0)*. Default is *spyDialogOK*.

Errors

4400 The object is no longer valid.

4401 Invalid value has been used to set the property. Invalid address for the return parameter was specified.

Out put Format

See also

Property: OutputForm at as SPYSchem aDocum entationForm at

Description

Defines the kind of documentation that will be generated: HTML (value=0), MS-Word (value=1), or RTF (value=2). The property gets initialized with the value used during the last call to DocumentGenerateXBRLDocumentation. The default for the first run is HTML.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid value has been used to set the property. Invalid address for the return parameter was specified.

Parent

See also

Property: Parent as Diabgs (read-only)

Description

Access the parent of the object.

Errors

4400 The object is no longer valid.

4401 Invalid address for the return parameter was specified.

Short Qualified Name

See also

Property: ShortQualifiedName as Boolean

Description

Set this property to true, to use short qualified names in the XBRL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateXBRLDocumentation</u>. The default for the first run is true.

Errors

4400 The object is no longer valid.

4401 Invalid address for the return parameter was specified.

ShowAbstract

See also

Property: ShowAbstract as Boolean

Description

Set this property to true, to show abstracts in the XBRL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateXBRLDocumentation</u>. The default for the first run is true.

Errors

4400 The object is no longer valid.

4401 Invalid address for the return parameter was specified.

ShowBalance

See also

Property: ShowBalance as Boolean

Description

Set this property to true, to show balances in the XBRL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateXBRLDocumentation</u>. The default for the first run is true.

4400	The object is no longer valid.
4401	Invalid address for the return parameter was specified.

ShowDiagram

See also

Property: ShowDiagram as Boolean

Description

Set this property to true, to show diagrams in the XBRL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateXBRLDocumentation</u>. The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

ShowImportedElements

See also

Property: Show Imported Elements as Boolean

Description

Set this property to true, to show imported elements in the XBRL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateXBRLDocumentation</u>. The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

ShowItemtype

See also

Property: Show Item type as Boolean

Description

Set this property to true, to show item types in the XBRL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateXBRLDocumentation</u>. The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

ShowLabels

See also

Property: ShowLabels as Boolean

Description

Set this property to true, to show labels in the XBRL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateXBRLDocumentation</u>. The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

ShowLinkbaseReferences

See also

Property: ShowLinkbaseReferences as Boolean

Description

Set this property to true, to show linkbase references in the XBRL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateXBRLDocumentation</u>. The default for the first run is true.

Errors

4400 The object is no longer valid.

4401 Invalid address for the return parameter was specified.

ShowNillable

See also

Property: ShowNillable as Boolean

Description

Set this property to true, to show nillable properties in the XBRL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateXBRLDocumentation</u>. The default for the first run is true.

Errors

4400 The object is no longer valid.4401 Invalid address for the return parameter was specified.

ShowPeriod

See also

Property: ShowPeriod as Boolean

Description

Set this property to true, to show periods in the XBRL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateXBRLDocumentation</u>. The default for the first run is true.

Errors

4400 The object is no longer valid.

4401 Invalid address for the return parameter was specified.

ShowProgressBar

See also

Property: ShowProgressBar as Boolean

Description

Set this property to true, to make the window showing the document generation progress visible. Use false, to hide it. Default is false.

Errors

4400 The object is no longer valid.

4401 Invalid address for the return parameter was specified.

ShowReferences

See also

Property: ShowReferences as Boolean

Description

Set this property to true, to show references in the XBRL documentation. The property is initialized with the value used during the last call to <u>DocumentGenerateXBRLDocumentation</u>. The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

ShowResult

See also

Property: ShowResult as Boolean

Description

Set this property to true, to automatically open the resulting document when generation was successful. HTML documentation will be opened in Authentic Desktop. To show Word documentation, MS- Word will be started. The property gets initialized with the value used during the last call to DocumentGenerateXBRLDocumentation. The default for the first run is true.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

ShowSubst it ut iongroup

See also

Property: ShowSubstitutiongroup as Boolean

Description

Set this property to true, to show substitution groups in the XBRL documentation. The property is initialized with the value used during the last call to Documentation.com The property is initialized with the value used during the last call to Documentation.com The property is initialized with the value used during the last call to Documentation.com The default for the first run is true.

Errors

4400 The object is no longer valid.

4401 Invalid address for the return parameter was specified.

SPSFile

See also

Property: SPSFile as String

Description

Full path and name of the SPS file that will be used to generate the documentation.

Errors

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

UseFixedDesign

See also

Property: UseFixedDesign as Boolean

Description

Specifies whether the documentation should be created with a fixed design or with a design specified by a SPS file (which requires StyleVision).

- 4400 The object is no longer valid.
- 4401 Invalid address for the return parameter was specified.

3.2.32 XMLData

See also

Properties

Kind Name TextValue

HasChildren MayHaveChildren Parent

Methods

GetFirstChib GetNextChib GetCurrentChib

hsertChild hsertChildAfter hsertChildBefore AppendChild

EraseAlChildren EraseChild EraseCurrentChild

IsSameNode

CountChildren CountChildrenKind

GetChibl GetChibAttribute GetChibElement GetChibKind GetNamespacePrefixForURI

HasChildrenKind SetTextValueXM LEncoded

Description

The XMLData interface provides direct XML-level access to a document. You can read and directly modify the XML representation of the document. However, please, note the following restrictions:

- The XMLData representation is only valid when the document is shown in grid view or authentic view.
- When in authentic view, additional XMLData elements are automatically inserted as parents of each visible document element. Typically this is an XMLData of kind <code>spyXMlDataElement</code> with the <u>Name</u> property set to 'Text'.
- When you use the XMLData interface while in a different view mode you will not receive errors, but changes are not reflected to the view and might get lost during the next view switch.

Note also:

- Setting a new text value for an XML element is possible if the element does not have nontext children. A text value can be set even if the element has attributes.
- When setting a new text value for an XML element which has more than one text child, the latter will be deleted and replaced by one new text child.
- When reading the text value of an XML element which has more than one text child, only

the value of the first text child will be returned.

AppendChild

See also

Declaration: AppendC hild (pNewData as XM LData)

Description

Append hid appends pNewData as last child to the XM LData object.

Errors

- 1500 The XMLData object is no longer valid.
- 1505 Invalid XMLData kind was specified.
- 1506 Invalid address for the return parameter was specified.
- 1507 Element cannot have Children
- 1512 Cyclic insertion new data element is already part of document
- 1514 Invalid XMLData kind was specified for this position.
- 1900 Document must not be modified

Example

Dim objCurrentParent As XM LData Dim objNewChild As XM LData

Set objNewChild = objSpyActiveDocum entCreateChild(spyXMLDataElement) Set objCumentParent = objSpyActiveDocum entRootElement

objCurrentParentAppendChild objNewChild

Set ob New Child = Nothing

Count Children

See also

Declaration: CountChildren as long

Description

CountChildren gets the number of children.

Available with TypeLibrary version 1.5

Errors

1500 The XMLData object is no longer valid.

Count ChildrenKind

See also

Declaration: CountChildrenKind (*nKind* as SPYXMLDataKind) as long

Description

CountChildrenKind gets the number of children of the specific kind.

Available with TypeLibrary version 1.5

Errors

1500 The XMLData object is no longer valid.

EraseAllChildren

See also

Declaration: EraseAlChildren

Description

EraseAlChildren deletes all associated children of the XMLData object.

Errors

1500 The XMLData object is no longer valid.

1900 Document must not be modified

Example

The sample erases all elements of the active document.

Dim objCumentParent As XM LD ata

Set objCumentParent = objSpyActiveDocumentRootElement objCumentParentEraseAllChildren

EraseChild

Method: EraseChild (Child as XMLData)

Description Deletes the given child node.

Errors

- 1500 Invalid object.
- 1506 Invalid input xml
- 1510 Invalid parameter.

EraseCurrent Child

See also

Declaration: EraseCurrentChild

Description

EraseCurrentChild deletes the current XMLData child object. Before you call EraseCurrentChild you must initialize an internal iterator with XMLDataGetFirstChild After deleting the current child, EraseCurrentChild increments the internal iterator of the XMLData element. No error is returned when the last child gets erased and the iterator is moved past the end of the child list. The next call to EraseCurrentChild however, will return error 1503.

- 1500 The XMLData object is no longer valid.
- 1503 No iterator is initialized for this XMLData object, or the iterator points past the last child.
- 1900 Document must not be modified

Examples

```
// XM LSpy scripting environment - JScript
// erase all children of XM LD ata
11
// lets get an XM LD ata element, we assume that the
// cursor selects the parent of a list in grid view
var objList = Application ActiveDocum entGridView CurrentFocus;
// the following line would be shorter, of course
//objListEraseAlChibren ();
// but we want to demonstrate the usage of EraseCurrentChild
if ((objList ⊨ null) && (objListHasChildren))
 try
 {
  objEle = objListGetFirstChild(-1);
  while (obÈle ⊨ null)
    objListEraseCurrentChild();
    // no need to call GetNextChild
 catch (em)
  // 1503 - we reached end of child list
   {if ((errnum ber & 0xffff) = 1503) throw (err); }
```

Get Child

See also

Declaration: GetChild (position as long) as XM LData

Return Value

Returns an XML element as XM LData object.

Description

GetChild() returns a reference to the child at the given index (zero-based).

Available with TypeLibrary version 1.5

Errors

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

GetChildAttribute

Method: GetChildAttribute (strName as string) child as XMLData object (NULL on error)

Description

Retrieves the attribute having the given name.

Errors

1500 Invalid object.

1510 Invalid parameter.

Get ChildElement

Method: GetChidElement (strName as string, nIndex as long) child as XMLData object (NULL on error)

Description

Retrieves the Nth child element with the given name.

Errors

1500 Invalid object.

1510 Invalid parameter.

Get ChildKind

See also

Declaration: GetChildKind (position as long, nKind as SPYXMLDataKind) as XMLData

Return Value

Returns an XML element as XM LData object.

Description

GetChitKind() returns a reference to a child of this kind at the given index (zero-based). The position parameter is relative to the number of children of the specified kind and not to all children of the object.

Available with TypeLibrary version 1.5

Errors

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

Get Current Child

See also

Declaration: GetCurrentChild as XM LData

Return Value

Returns an XML element as XMLData object.

Description

GetCurrentChild gets the current child. Before you call GetCurrentChild you must initialize an internal iterator with XMLDataGetFirstChild.

- 1500 The XMLData object is no longer valid.
- 1503 No iterator is initialized for this XMLData object.

1510 Invalid address for the return parameter was specified.

Get First Child

See also

Declaration: GetFirstChild (nKind as SPYXM LDataKind) as XM LData

Return Value

Returns an XML element as XM LData object.

Description

GetFirstChild initializes a new iterator and returns the first child. Set nKind = -1 to get an iterator for all kinds of children.

REMARK: The iterator is stored inside the XMLData object and gets destroyed when the XMLData object gets destroyed. Be sure to keep a reference to this object as long as you want to use GetCurrentChild, GetNextChild or EraseCurrentChild.

Errors

- 1500 The XMLData object is no longer valid.
- 1501 Invalid XMLData kind was specified.
- 1504 Element has no children of specified kind.
- 1510 Invalid address for the return parameter was specified.

Example

See the example at XM LData GetNextC hild.

GetNamespacePrefixForURI

Method: GetNamespacePrefixForURI(strURI as string) strNS as string

Description

Returns the namespace prefix of the supplied URI.

Errors

- 1500 Invalid object.
- 1510 Invalid parameter.

Get Next Child

See also

Declaration: GetNextChild as XM LData

Return Value

Returns an XML element as XMLData object.

Description

GetNextChild steps to the next child of this element. Before you call GetNextChild you must initialize an internal iterator with XMLDataGetFirstChild.

Check for the last child of the element as shown in the sample below.

Errors

- 1500 The XMLData object is no longer valid.
- 1503 No iterator is initialized for this XMLData object.
- 1510 Invalid address for the return parameter was specified.

Examples

```
get elements of all kinds
Set objCumentChild = objParentGetFirstChild(-1)
```

Do

```
to something useful with the child
```

```
step to next child
Set objCumentChild = objParentGetNextChild
Loop Until (ErrNumber - vbObjectError = 1503)
```

```
// -----
// XMLSpy scripting environment - JScript
// iterate through children of XMLData
// -----
```

```
try
{
```

```
var ob}XM LD ata = ... // initialize som ehow
var objC hild = ob}XM LD ata GetFirstC hild (-1);
```

while (true)

```
{
    // do som ething usefull w ith objChild
    objChild = objXM LD ata GetNextChild();
    }
}
catch (ent)
{
    if ((ernum ber & 0xffff) == 1504)
    ;// element has no children
    else if ((ernum ber & 0xffff) == 1503)
    ;// last child reached
    else
    throw (ent);
}
```

GetTextValueXMLDecoded

Method: GetTextValueXMLDecoded ()as string

Description

Gets the decoded text value of the XML.

Errors

1500 lr	nvalid c	object.
---------	----------	---------

1510 Invalid parameter.

HasChildren

See also

Declaration: HasChildren as Boolean

Description

The property is true if the object is the parent of other XMLData objects. This property is readonly.

Errors

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

HasChildrenKind

See also

Declaration: HasChildrenKind (*nKind* as SPYXMLDataKind) as Boolean

Description

The method returns true if the object is the parent of other XM LData objects of the specific kind.

Available with TypeLibrary version 1.5

Errors

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

Insert Child

See also

Declaration: InsertChild (pNewData as XM LData)

Description

InsertChild inserts the new child before the current child (see also <u>XM LData GetFirstChild</u>, <u>XM LData GetNextChild</u> to set the current child).

- 1500 The XMLData object is no longer valid.
- 1503 No iterator is initialized for this XMLData object.
- 1505 Invalid XMLData kind was specified.
- 1506 Invalid address for the return parameter was specified.
- 1507 Element cannot have Children
- 1512 Cyclic insertion new data element is already part of document
- 1514 Invalid XMLData kind was specified for this position.
- 1900 Document must not be modified

Insert ChildAfter

Method: hsertChibBefore (Node as XMLData, NewData as XMLData)

Description

Inserts a new XML node (supplied with the second parameter) after the specified node (first parameter).

Errors

- 1500 Invalid object.
- 1506 Invalid input xml
- 1507 No children allowed
- 1510 Invalid parameter.
- 1512 Child is already added
- 1514 Invalid kind at position

Insert ChildBefore

Method: InsertChildBefore (Node as XMLData, NewData as XMLData)

Description

Inserts a new XML node (supplied with the second parameter) before the specified node (first parameter).

Errors

- 1500 Invalid object.
- 1506 Invalid input xml
- 1507 No children allowed
- 1510 Invalid parameter.
- 1512 Child is already added
- 1514 Invalid kind at position

IsSameNode

See also

Declaration: IsSam eNode (pNodeToCompare as XMLData) as Boolean

Description

Returns true if pNodeToCompare references the same node as the object itself.

Errors

- 1500 The XMLData object is no longer valid.
- 1506 Invalid address for the return parameter was specified.

Kind

See also

Declaration: Kind as SPYXM LDataKind

Description

Kind of this XM LData object. This property is read-only.

Errors

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

MayHaveChildren

See also

Declaration: MayHaveChildren as Boolean

Description

Indicates whether it is allowed to add children to this XM LData object. This property is read-only.

Errors

1500	The XMLData object is no longer valid.	
4 = 4 0		

1510 Invalid address for the return parameter was specified.

Name

See also

Declaration: Name as String

Description

Used to modify and to get the name of the XM LData object.

Errors

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

Parent

See also

Declaration: Parent as XM LData

Return value

Parent as XM LData object. Nothing (or NULL) if there is no parent element.

Description

Parent of this element. This property is read-only.

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

SetTextValueXMLEncoded

Method: SetTextValueXM LEncoded (strValas String)

Description

Sets the encoded text value of the XML.

Errors

1500 Invalid object.

1513 Modification not allowed.

TextValue

See also

Declaration: TextValue as String

Description

Used to modify and to get the text value of this XM LData object.

- 1500 The XMLData object is no longer valid.
- 1510 Invalid address for the return parameter was specified.

3.3 Enumerations

This is a list of all enumerations used by the Authentic Desktop API. If your scripting environment does not support enumerations use the number-values instead.

3.3.1 ENUMApplicationStatus

Description

Enumeration to specify the current Application status.

eApplicationRunning eApplicationAfterLicenseCheck	= 0 = 1
eApplicationBeforeLicenseCheck	= 2
eApplicationConcurrentLicenseCheckFailed	= 3
eApplicationProcessingCommandLine	= 4

3.3.2 SPYAttributeTypeDefinition

Description

Attribute type definition that can be selected for generation of Sample XML. This type is used with the method <u>GenerateDTDOrSchema</u> and <u>GenerateDTDOrSchemaEx</u>.

spyMergedGlobal	= 0
spyDistinctGlobal	= 1
spyLocal	= 2

3.3.3 SPYAuthenticActions

Description

Actions that can be performed on <u>AuthenticRange</u> objects.

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

3.3.4 SPYAut hent ic Document Position

Description

Relative and absolute positions used for navigating with <u>AuthenticRange</u> objects.

Possible values:

spyAuthenticDocumentBegin = 0

spyAuthenticDocumentEnd = 1

spyAuthenticRangeBegin = 2

spyAuthenticRangeEnd = 3

3.3.5 SPYAuthenticElementActions

Description

Actions that can be used with the obsolete object GetAllowed Elements (superseded by AuthenticRange CanPerform Action With).

k	ActionInsertAt	= 0
k	_ActionApply	= 1
k	ActionClearSurr	= 2
k	ActionAppend	= 3
k	ActionInsertBefore	= 4
k	ActionRemove	= 5

3.3.6 SPYAut hent ic Element Kind

Description

Enumeration of the different kinds of elements used for navigation and selection within the <u>AuthenticRange</u> and <u>AuthenticView</u> objects.

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

3.3.7 SPYAuthenticMarkupVisibility

Description

Enumeration values to customize the visibility of markup with MarkupVisibility.

Possible values:

spyAuthenticMarkupHidden = 0

spyAuthenticMarkupSmall = 1

- spyAuthenticMarkupLarge = 2
- spyAuthenticMarkupMixed = 3

3.3.8 SPYAuthenticToolbarButtonState

Description

Authentic toolbar button states are given by the following enumeration:

authenticToolbarButtonDefault	= 0
authenticToobarButtonEnabled	= 1
authenticToobarButtonDisabled	= 2

3.3.9 SPYDatabaseKind

Description

Values to select different kinds of databases for import. See <u>DatabaseConnection DatabaseKind</u> for its use.

spyDB_Access	= 0
spyDB_SQLServer	= 1
spyDB_Oracle	= 2
spyDB_Sybase	= 3
spyDB_MySQL	= 4
spyDB_DB2	= 5
spyDB_Other	= 6
spyDB_Unspecified	= 7
spyDB_PostgreSQL	= 8
spyDB_iSeries	= 9

3.3.10 SPYDialogAction

Description

Values to simulate different interactions on dialogs. See <u>Dialogs</u> for all dialogs available.

Possible values:

spyDialogOK	= 0
spyDialogCancel	= 1
spyDialogUserInput	= 2

// simulate click on OK button

// simulate click on Cancel button

= 2 // show diabg and allow user interaction

3.3.11 SPYDOMType

Description

Enumeration values to parameterize generation of C++ code from schema definitions.

Possible values:

spyDOMType_msxml4	= 0	Obsolete
spyDOMType_xerces	= 1	
spyDOMType_xerces3	= 2	
spyDOMType_msxml6	= 3	

spyDOMType_xerces indicates Xerces 2.x usage; spyDOMType_xerces3 indicates Xerces 3.x
usage.

3.3.12 SPYDTDSchemaFormat

Description

Enumeration to identify the different schema formats.

spyDTD	= 0
spyW3C	= 1

3.3.13 SPYEncodingByteOrder

Description

Enumeration values to specify encoding byte ordering for text import and export.

spyNONE	= 0
spyLITTLE_ENDIAN	= 1
spyBIG_ENDIAN	= 2

3.3.14 SPYExport Namespace

Description

Enumeration type to configure handling of namespace identifiers during export.

Possible values:

spyNoNamespace = 0 spyReplaceColonWithUnderscore = 1

3.3.15 SPYFindInFilesSearchLocation

Description The different locations where a search can be performed. This type is used with the <u>FindInFilesDlg</u> dialog.

spyFindInFiles Documents	= 0
spyFindInFiles Project	= 1
spyFindInFiles Folder	= 2

3.3.16 SPYFrequent Elements

Description

Enumeration value to parameterize schema generation.

Possible values:

spyGlobalElements = 0 spyGlobalComplexType = 1

3.3.17 SPYImageKind

Description

Enumeration values to parameterize image type of the generated documentation. These values are used in SchemaDocumentationDialogDiagram Form at.

spyImageType_PNG	= 0
spyImageType_EMF	= 1
3.3.18 SPYImportColumnsType

Description

Enumeration to specify different Import columns types.

spyImportColumns Element	= 0
spyImportColumns Attribute	= 1

3.3.19 SPYKeyEvent

Description

Enumeration type to identify the different key events. These events correspond with the equally named windows messages.

spyKeyDown	= 0
spyKeyUp	= 1
spyKeyPressed	= 2

3.3.20 SPYKeyStatus

Description

Enumeration type to identify the key status.

spyLeftShiftKeyMask	= 1
spyRightShiftKeyMask	= 2
spyLeftCtrlKeyMask	= 4
spyRightCtrlKeyMask	= 8
spyLeftAltKeyMask	= 16
spyRightAltKeyMask	= 32

3.3.21 SPYLibType

Description

Enumeration values to parameterize generation of C++ code from schema definitions.

spyLibType_static	= 0
spyLibType_dll	= 1

3.3.22 SPYLoading

Description

Enumeration values to define loading behaviour of URL files.

Possible values:

spyUseCacheProxy = 0 spyReload = 1

3.3.23 SPYMouseEvent

Description

Enumeration type that defines the mouse status during a mouse event. Use the enumeration values as bitmasks rather then directly comparing with them.

Examples

```
'to check for ctrl-leftbutton-down in VB
```

End If

'to check for double-click with any button in VBScript If (((i_eM ouseEventAnd spyDoubleClickMask) <> 0) Then

```
'react on double-click
```

End If

Possible values:

spyNoButtonMask	= 0
spyMouseMoveMask	= 1
spyLeftButtonMask	= 2
spyMiddleButtonMask	= 4
spyRightButtonMask	= 8
spyButtonUpMask	= 16
spyButtonDownMask	= 32
spyDoubleClickMask	= 64
spyShiftKeyDownMask	= 128
spyCtrlKeyDownMask	= 256
spyLeftButtonDownMask	= 34
spyMiddleButtonDownMask	= 36
spyRightButtonDownMask	= 40
spyLeftButtonUpMask	= 18
spyMiddleButtonUpMask	= 20
spyRightButtonUpMask	= 24
spyLeftDoubleClickMask	= 66
spyMiddleDoubleClickMask	= 68
spyRightDoubleClickMask	= 72

// spyLeftButtonMask | spyButtonDownMask // spyMiddleButtonMask | spyButtonDownMask // spyRightButtonMask | spyButtonDownMask // spyLeftButtonMask | spyButtonUpMask // spyMiddleButtonMask | spyButtonUpMask // spyRightButtonMask | spyButtonUpMask // spyRightButtonMask | spyButtonUpMask // spyRightButtonMask | spyDoubleClickMask // spyRightButtonMask | spyDoubleClickMask

3.3.24 SPYNumberDateTimeFormat

Description

Enumeration value to configure database connections.

Possible values:

spySystemLocale = 0 spySchemaCompatible = 1

3.3.25 SPYProgrammingLanguage

Description

Enumeration values to select the programming language for code generation from schema definitions.

Only available/enabled in the Enterprise edition. An error is returned, if accessed by any other version.

spyUndefinedLanguage	= -1
spyJava	= 0
spyCpp	= 1
spyCSharp	= 2

3.3.26 SPYProjectItemTypes

Description

Enumeration values to identify the different elements in project item lists. See <u>SpyProjectItem.ItemType</u>.

spyUnknownItem	= 0
spyFileltem	= 1
spyFolderItem	= 2
spyURLItem	= 3

3.3.27 SPYProjectType

Description

Enumeration values to parameterize generation of C# from schema definitions.

spyVisualStudioProject	= 0	Obsolete
spyVisualStudio2003Project	= 1	Obsolete
spyBorlandProject	= 2	Obsolete
spyMonoMakefile	= 3	Obsolete
spyVisualStudio2005Project	= 4	For C++ code also
spyVisualStudio2008Project	= 5	For C++ code also
spyVisualStudio2010Project	= 6	For C++ code also

3.3.28 SpySampleXMLGenerationChoiceMode

Description

This enumeration is used in GenerateSampleXMLDlg.ChoiceMode:

spySampleXMLGen_FirstBranch = 0 spySampleXMLGen_AllBranches = 1 spySampleXMLGen_ShortestBranch = 2

3.3.29 SPYSampleXMLGenerationOptimization (Obsolete)

This enumeration is OBSOLETE since v2014.

Description

Specify the elements that will be generated in the Sample XML. This enumeration is used in <u>GenerateSampleXMLDIg</u>.

spySampleXMLGen_Optimized	= 0
spySampleXMLGen_NonMandatoryElements	= 1
spySampleXMLGen_Everything	= 2

3.3.30 SpySampleXMLGenerationSampleValueHints

Description

This enumeration is used in GenerateSampleXMLDlg.SampleValueHints

spySampleXMLGen_FirstFit = 0 spySampleXMLGen_RandomFit = 1 spySampleXMLGen_CycleThrough = 2

3.3.31 SPYSampleXMLGenerationSchemaOrDTDAssignment

Description

Specifies what kind of reference to the schema/DTD should be added to the generated Sample XML.

This enumeration is used in <u>GenerateSampleXMLDlg</u>.

spySampleXMLGen_AssignRelatively	= 0
spySampleXMLGen_AssignAbsolutely	= 1
spySampleXMLGen_DoNotAssign	= 2

3.3.32 SPYSchemaDefKind

Description

Enumeration type to select schema diagram types.

spyKindElement	= 0
spyKindComplexType	= 1
spyKindSimpleType	= 2
spyKindGroup	= 3
spyKindModel	= 4
spyKindAny	= 5
spyKindAttr	= 6
spyKindAttrGroup	= 7
spyKindAttrAny	= 8
spyKindldentityUnique	= 9
spyKindldentityKey	= 10
spyKindldentityKeyRef	= 11
spyKindldentitySelector	= 12
spyKindldentityField	= 13
spyKindNotation	= 14
spyKindInclude	= 15
spyKindImport	= 16
spyKindRedefine	= 17
spyKindFacet	= 18
spyKindSchema	= 19
spyKindCount	= 20

3.3.33 SPYSchemaDocumentationFormat

Description

Enumeration values to parameterize generation of schema documentation. These values are used in SchemaDocumentationDiabgDutputFormat.

spySchemaDoc_HTML	= 0
spySchemaDoc_MSWord	= 1
spySchemaDoc_RTF	= 2
spySchemaDoc PDF	= 3

3.3.34 SPYSchemaExtensionType

Description

Enumeration to specify different Schema Extension types.

spySchemaExtension None	= 0
spySchemaExtension SQL XML	= 1
spySchemaExtension_MS_SQL_Server	= 2
spySchemaExtension_Oracle	= 3

3.3.35 SPYSchemaFormat

Description

Enumeration to specify different Schema Format types.

spySchemaFormat	Hierarchical	=	0
spySchemaFormat	Flat	=	1

3.3.36 SPYText Delimiters

Description

Enumeration values to specify text delimiters for text export.

spyTabulator	= 0
spySemicolon	= 1
spyComma	= 2
spySpace	= 3

3.3.37 SPYText Enclosing

Description

Enumeration value to specify text enclosing characters for text import and export.

spyNoEnclosing	= 0
spySingleQuote	= 1
spyDoubleQuote	= 2

3.3.38 SPYTypeDetection

Description

Enumeration to select how type detection works during <u>GenerateDTDOrSchema</u> and <u>GenerateDTDOrSchemaEx</u>.

Possible values:

spyBestPossible = 0 spyNumbersOnly = 1 spyNoDetection = 2

3.3.39 SPYURLTypes

Description

Enumeration to specify different URL types.

Possible values:

spyURLTypeAuto = -1 spyURLTypeXML = 0 spyURLTypeDTD = 1

3.3.40 SPYValidateXSDVersion

Description

Enumeration values that select what XSD version to use. The XSD version that is selected depends on both (i) the presence/absence—and, if present, the value—of the /xs:schema/ @vc:minVersion attribute of the XSD document, and (ii) the value of this enumeration.

spyValidateXSDVersion_1_0 selects XSD 1.0 if vc:minVersion is absent, or is present with
any value.

spyValidateXSDVersion_1_1 selects XSD 1.1 if vc:minVersion is absent, or is present with
any value.

spyValidateXSDVersion_AutoDetect selects XSD 1.1 if vc:minVersion=1.1. If the vc:minVersion attribute is absent, or is present with a value other than 1.1, then XSD 1.0 is selected.

spyValidateXSDVersion_AutoDetect	=	0
spyValidateXSDVersion_1_1	=	1
spyValidateXSDVersion_1_0	=	2

3.3.41 SPYValidateErrorFormat

Description

Enumeration values that select the format of the error message.

spyValidateErrorFormat_Text	=	0
spyValidateErrorFormat_ShortXML	=	1
spyValidateErrorFormat_LongXML	=	2

3.3.42 SPYViewModes

Description

Enumeration values that define the different view modes for XML documents. The mode *spyViewAuthentic(4)* identifies the mode that was intermediately called DocEdit mode and is now called Authentic mode. The mode *spyViewJsonSchema* identifies a mode which is mapped to the Schema Design View on the GUI but is distinguished internally.

spyViewGrid	= 0	
spyViewText	= 1	
spyViewBrowser	= 2	
spyViewSchema	= 3	
spyViewContent	= 4	// obsolete
spyViewAuthentic	= 4	
spyViewWSDL	= 5	
spyViewZIP	= 6	
spyViewEditionInfo	= 7	
spyViewXBRL	= 8	
spyViewJsonSchema	= 9	

3.3.43 SPYVirtualKeyMask

Description

Enumeration type for the most frequently used key masks that identify the status of the virtual keys. Use these values as bitmasks rather then directly comparing with them. When necessary, you can create further masks by using the 'logical or' operator.

Examples

```
'VBScript sam ple:check if ctrl-key is pressed
If ((inV intualKeyStatus And spyC trlKeyM ask) <> 0)) Then
        'ctrl-key is pressed
End If
'VBScript sam ple:check if ONLY ctrl-key is pressed
If (inV intualKeyStatus == spyC trlKeyM ask) Then
        'exactly ctrl-key is pressed
End If
// JScript sam ple:check if any of the right virtual keys is pressed
if ((inV intualKeyStatus & (spyR ightShiftKeyM ask | spyR ightC trlKeyM ask | spyR ightA ltKeyM ask)) \= 0)
{
        ; 'right virtual key is pressed
}
```

spyNoVirtualKeyMask	= 0
spyLeftShiftKeyMask	= 1
spyRightShiftKeyMask	= 2
spyLeftCtrlKeyMask	= 4
spyRightCtrlKeyMask	= 8
spyLeftAltKeyMask	= 16
spyRightAltKeyMask	= 32
spyShiftKeyMask	= 3
spyCtrlKeyMask	= 12
spyAltKeyMask	= 48

- // spyLeftShiftKeyMask | spyRightShiftKeyMask
- // spyLeftCtrlKeyMask | spyRightCtrlKeyMask
- 48 // spyLeftAltKeyMask | spyRightAltKeyMask

3.3.44 SPYXMLDataKind

Description

The different types of XMLData elements available for XML documents.

_ 1
= 1
= 2
= 3
= 4
= 5
= 6
= 7
= 8
= 9
= 10
= 11
= 12
= 13
= 14
= 15
= 16

4 ActiveX Integration

このセクションで説明されている Authentic Desktop ユーザーインターフェイス とActiveX コントロール 機能を使用するアプリケーションに統合することができます。 ActiveX 技術によりC++、C#、 VB.NET、HTML などの統合のために使用される広範囲の言語が有効化されます(HTML内に統 合されている ActiveX コンポーネントは、Microsoft Internet Explorer バージョンとActiveX をサポート すると共に作動する必要があります)。全てのコンポーネントはフル OLE コントロールです。 Java への 統合はラッパークラスにより与えられます。

ActiveX コントロールを、カスタムコードに統合するには、Authentic Desktop Integration Package をインストールしてください(次を参照してください: <u>http://www.altova.com/</u> <u>ide_integration.html)</u>。すべての Authentic Desktop を最初に、そして、Authentic Desktop Integration Package そして他の必要条件を言語 とプラットフォームに応じてインストールしてくだ さい(次を参照してください: <u>必要条件</u>)。

統合の2つの異なるレベル間を柔軟に選択することができます:アプリケーションレベルとドキュメントレベル

アプリケーションレベルの統合は、(メニュー、ツールバー、ペインを含む) Authentic Desktop のフルイン ターフェイスを ActiveX コントロールをカスタムアプリケーションとして埋め込むことを意味します。例えば、 多くのシナリオでは、カスタムアプリケーションを、Authentic Desktop グラフィカル ユーザーインターフェ イスを埋め込む1つのフォームから構成することができます。

ドキュメントレベルの統合よりもこのアプローチよりも簡単に実装することができますが、カスタム必要条件 に従いを柔軟的に Authentic Desktop グラフィカル ユーザーインターフェイスを構成するためには適切で はない可能性があります。ドキュメントレベルの統合は、アプリケーションに Authentic Desktop を1つ1つ 埋め込むことを意味します。メイン Authentic Desktop コントロールのみではなく、メインドキュメントエ ディターウィンドウの実装も含まれます。また、オプションで、追加ウィンドウの実装も含まれます。このア プローチは GUI の構成に柔軟性を与えますが、選択された言語により ActiveX コントロールとの高度なイ ンタラクションが必要とされます。

セクションアプリケーションレベルの統合とドキュメントレベルの統合 は、それぞれのレベルの主要なス テップに関して説明しています。ActiveX 統合のサンプル セクションは、C#、HTML、および、Javaの サンプルを提供します。これらのサンプルを確認することにより、正しい選択が直くに行うことができま す。オブジェクトレファレンス セクションは、プロパティとメソッドに加え統合のために使用できるすべて の COM オプジェクトに付いて説明しています。

Authentic Desktop を Visual Studio プラグインとして使用するための情報に関しては、次を参照してください: <u>Visual Studio</u> 内の Authentic Desktop。

4.1 必要条件

Authentic Desktop ActiveX コントロールをカスタムアプリケーションに統合するには、以下が使用中のコンピューターにインストールされている必要があります:

- Authentic Desktop
- Authentic Desktop Integration Package は、次からダウンロードできます: <u>http://</u> www.altova.com/ide_integration.html

64 ビット ActiveX コントロールを統合するには、Authentic Desktop の 64 ビット バージョンと Authentic Desktop Integration Package をインストールする必要 があります。Visual Studio を使用して、Microsoft .NET プラットフォームの下で開発 されたアプリケーションに関しては、Authentic Desktop の 32 ビット と64 ビット バージョンとAuthentic Desktop Integration Package について下で 説明 されているとおりインストールします。

Microsoft .NET (C#, VB.NET) with Visual Studio

Authentic Desktop ActiveX コントロールを、Microsoft .NET の下で開発された 32 ビット アプリケーションに統合するには、以下を使用中のコンピューターにインストールしてください:

- Microsoft .NET Framework 4.0 または以降
- Visual Studio 2008/2010/2012/2013/2015/2017
- Authentic Desktop 32 ビットとAuthentic Desktop Integration Package 32 ビット
- ActiveX コントロールがVisual Studio ツールボックスに追加されている必要があります(次を参照してください: ActiveX コントロールをツールボックスに追加する).

64 ビット ActiveX コントロールを統合する場合、次の必要条件が上記に加え満たされる必要があ ります:

- Authentic Desktop 32 ビットとAuthentic Desktop Integration Package 32 ビットをインスト ールしてください (Visual Studio は 32 ビットで作動するため、32 ビット ActiveX コントロール を Visual Studio デザイナーに与える必要があります。)
- Authentic Desktop 64 ビットとAuthentic Desktop Integration Package 64 ビット をインスト ールしてください(実際の64 ビット ActiveX コントロールをランタイムにカスタムアプリケーション に与えます。)
- Visual Studio内で、64 ビットビルド更新を作成し、この更新を使用してアプリケーションを ビルドします。サンプルに関しては、次を参照してください:サンプルC# ソリューションの実 行。

Java

Eclipse 開発環境を使用して、Authentic Desktop ActiveX コントロールを Java アプリケーションに 統合するには、次を使用中のコンピューターにインストールして ください:

- Java Runtime Environment (JRE) または Java Development Kit (JDK) 7 または以降
- Eclipse
- Authentic Desktop and Authentic Desktop Integration Package

メモ: Windows XP とWindows Server 2003 で統合パッケージが正確に作動するためには、再頒 布可能パッケージを手動でインストールしてください: <u>https://www.microsoft.com/en-US/</u> download/details.aspx?id=48145.

メモ: Authentic Desktop ActiveX コントロールの64 ビット バージョンを作動するには、Eclipse の 64 ビット バージョン、および、Authentic Desktop とAuthentic Desktop Integration Package の 64 ビット バージョン ofを使用 してください。

クライアントコンピューター上 での Authentic Desktop 統合 とデプロイ

.NET アプリケーションを作成し他のクライアントに配布する場合は、クライアントコンピューター上に次をインストールしてください:

- Authentic Desktop
- Authentic Desktop Integration Package
- カスタムの統合コード、または、アプリケーション。

4.2 ActiveX コントロールをツールボックスに追加する

Visual Studio を使用して開発されたアプリケーション内で、Authentic Desktop ActiveX コントロールを使用するには、コントロールが最初に Visual Studio ツールボックスに追加 される必要 があります:

- 1. Visual Studio の「ツール」メニューから、「ツールボックスアイテムの選択」をクリックします。
- 「COM コンポーネント」タブ、から、Authentic DesktopControl, Authentic DesktopControl ドキュメント、および、Authentic DesktopControl プレースホルダーの横のチェックボックスを選 択します。

上記のコントロールが使用することができない場合、下のステップに従います:

- 「COM コンポーネント」タブから「参照」をクリックします。AuthenticControl.ocx を Authentic Desktop インストールからフォルダーを選択します。Authentic Desktop Integration Package をインストールすることを忘れないでください。それ以外の場合、この ファイルを使用することはできません、次を参照してください:<u>必要条件</u>.
- 2. を更に高いパーミッションと共に再起動するようにプロンプトされると 「異なる資格情報 下で再起動する」をクリックします。



上記のステップが完了すると Authentic Desktop ActiveX コントロールは、Visual Studio ツールボックス内で使用できるようになります。

Toolbo	• = ×	
Search	Toolbox 🔎 -	
⊳ All W	/indows Forms	
▷ Com	mon Controls	
▷ Cont	ainers	
🔺 Men	us & Toolbars	
h.	Pointer	
ħ⊒-	ContextMenuStrip	
	MenuStrip	
	StatusStrip	
	ToolStrip	
	ToolStripContainer	
6	AuthenticDesktopControl	
6	AuthenticDesktopControl PlaceHolder	
6	AuthenticDesktopyControl Document	
⊳ Data		
▷ Com	ponents	
Printing		
Dialogs		
VPF Interoperability		

- General
- メモ: アプリケーションレベルの統合では、AuthenticDesktopControl ActiveX コントロール のみが 使用 されます (次を参照してください:アプリケーションレベルの統合)。 AuthenticDesktopControl ドキュメント とAuthenticDesktopControl プレースホルダー コントロールは、ドキュメントレベルの統合のために使用 されます (次を参照してください:ド キュメントレベルの統合)。

4.3 アプリケーションレベルの統合

アプリケーションレベルの統合により使用中のアプリケーションのウィンドウに Authentic Desktop のイン ターフェイスを埋め込むことができます。 この種類の統合により、すべてのメニュー、ツールバー、ステ ータスバー、ドキュメントウィンドウ、および、ヘルパーウィンドウを含む Authentic Desktop のユーザー インターフェイスを使用することができます。アプリケーションのユーザーインターフェイスのカスタム化は Authentic Desktop が提供する機能に限られています。 これには、ヘルパーウィンドウの整理 とサイ ズ調整、及びメニューとソールバーのカスタム化が含まれています。

アプリケーションレベルで統合する場合、統合する必要のある唯一のActiveX コントロールは AuthenticDesktopControl です。AuthenticDesktopControlDocument または AuthenticDesktopControlPlaceHolder ActiveX コントロール をインスタンス化、または、アクセス使用としないでください。

初期化、または、Authentic Desktop の振る舞いを自動化するには、<u>AuthenticDesktopControl</u>. のために説明されているプロパティ、メソッド、とイベントを使用してください。Authentic Desktop 機能 への複雑なアクセスのために<u>AuthenticDesktopControl.application</u>を使用することを考慮して ください。

Authentic Desktop アプリケーションをHTML ページ、に埋め込むサンプルに関しては、次を参照して ください: アプリケーションレベルの HTML 統合。

Visual Studio を使用する C# または VB.NET での Authentic Desktop ActiveX コントロールをアプリケーションレベルで統合する基本的なアプリケーションを作成するステップは、以下のとおりです:

- すべての必要条件が満たされていることを確認してください(次を参照してください:必要条件).
- 2. 新規の Visual Studio Windows Forms プロジェクトを新規の空のフォーム共に作成してください。
- 3. まだ行っていない場合、ActiveX コントロールをツールボックスに追加します (次を参照してく ださい: ActiveX コントロールをツールボックスに追加する)。
- 4. ルーツボックスから、AuthenticDesktopControl をドラッグして、新規のフォームにドロップしてください。
- 5. フォームで AuthenticDesktopControl を選択し、プロパティウィンドウ内で IntegrationLevel プロパティを ICActiveXIntegrationOnApplicationLevel に設定しま す。

Pr	Properties 🔹 🗖 🗙			
axAuthenticDesktopControl AxAuthenticControlLib.AxAuthenticDeskto -				
0	11 12 F 12 F			
	BorderStyle	0 - None	*	
	CausesValidation	True		
	ContextMenuStrip	(none)		
	Dock	None		
	EnableUserPrompts	True		
-	GenerateMember	True		
	IntegrationLevel	${\sf ICActiveXIntegrationOnApplicationLevel} \lor$		
Ŧ	Location	316, 172	1	
	Locked	False		
Ŧ	Margin	3, 3, 3, 3		
Ŧ	MaximumSize	0, 0		
Ŧ	MinimumSize	0, 0		
	Modifiers	Private		
Ŧ	Padding	0, 0, 0, 0		
ŧ	Size	165, 109		
	TabIndex	6		
	TabStop	True		
	Tag			
	UseWaitCursor	False		
	Visible	False	Ŧ	

- 6. ビルド(x86、x64)するプラットフォームに一致するビルドプラットフォームの更新を作成します。 ビルドの更新の作成の方法は以下のとおりです:
 - a. Visual Studio 内のソリューションを右クリックし、「更新 マネージャー」を選択します。
 - b. 「アクティブなソリューションプラットフォーム」から、「新規作成…」を選択し、x86 またはx64 更新を選択します (このサンプルでは、x86)。

New Solution Platform
Type or select the new platform:
x86 🗸
Copy <u>s</u> ettings from:
Any CPU 👻
Create new project platforms
OK Cancel

Visual Studio内でソリューションをビルドして実行する準備が整うと、ターゲットプラットフォーム (x86、x64)に一致する更新をしようしてビルドすることを確認してください。

4.4 ドキュメントレベルの統合

アプリケーションレベルの統合に比べ、ドキュメントレベルの統合複雑ですが、ActiveX コントロール。 を使用することによりアプリケーションに Authentic Desktop 機能を柔軟的に埋め込むことができま す。このアプローチにより、コードは Authentic Desktop ユーザーインターフェイスの次のパーツに部分 的にアクセスすることができます:

- ドキュメントの編集 ウィンドウ
- プロジェクトウィンドウ
- 情報ウインドウ
- メッセージウィンドウ
- 入力ヘルパー ウィンドウ(要素、属性、エンティティ)
- 出力ウィンドウ

アプリケーションレベルの統合で説明されているとおり、ActiveX アプリケーションレベルの統合のためには、AuthenticDesktopControlのみが必要とされる唯一のコントロールです。しかしながら、ActiveX ドキュメントレベルの統合の場合は、機能 Authentic Desktop が次の ActiveX コントロール により与えられます:

- 1. AuthenticDesktopControl
- 2. <u>AuthenticDesktopControl</u> ドキュメント
- 3. AuthenticDesktopControl プレースホルダ

これらのコントロールは Authentic Desktop のアプリケーション インストール フォルダーで使用 することのできる AuthenticControl.ocx ファイルにより与えられます。Visual Studio を使用 して ActiveX 統合を開発するとこれらのコントロールを Visual Studio ツールボックスに追加する必要があります(次を参照してください: ActiveX コントロールをツールボックスに追加する)。

ドキュメントレベルでの ActiveX コントロールのアプリケーションへの統合の基本のステップは、以下のとおりです:

- 最初に、アプリケーション内でAuthenticDesktopControlをインスタンス化します。
 「AuthenticDesktopControl ドキュメント」と「AuthenticDesktopControl プレースホルダ」コントロールへのサポートが有効化されるために、このコントロールの統合は必須です。
 <u>IntegrationLevel</u>プロパティをI「CActiveXIntegrationOnDocumentLevel」(または
 「1」)に設定することは重要です。ユーザーに対してコントロールを非表示にするには、
 Visible」プロパティを「False」に設定します。
- メモ: ドキュメントレベルで統合する場合、未知の結果が生成される可能性があるため、 AuthenticDesktopControlの「開く」メソッドを使用しないでください。 AuthenticDesktopControlドキュメントとAuthenticDesktopControlプレースホルダの 対応するメソッドを代わりに使用して開いてください。
 - 2. AuthenticDesktopControl ドキュメント のインスタンスを少なくとも1つアプリケーション内に 作成します。このコントロールは、アプリケーションに Authentic Desktop のドキュメントの編 集ウィンドウを与え、必要であれば、複数回インスタンス化することができます。

既存のファイルを開くために、メソッド 「開く」を使用します。ドキュメントに関連した機能に アクセスするには、メソッドとプロパティ「ドキュメント」を介してアクセスすることのできる「パ ス」と「保存」を使用します。

メモ: コントロールは、読み取り専用モードをサポートしません。プロパティ「読み取り専用」の値は無視されます。
3. 任意で、アプリケーションに、アプリケーションで使用することのできる(ドキュメントウィンドウ 以外の)各追加ウィンドウのために AuthenticDesktopControl プレースホルダ コントロール を追加することができます。

AuthenticDesktopControl プレースホルダ のインスタンスによりアプリケーションウィンドウに 選択してAuthentic Desktopの追加ウィンドウを埋め込むことができます。ウィンドウの種類 (例えば、プロジェクトウィンドウ)はプロパティ「PlaceholderWindowID」により定義するこ とができます。有効なウィンドウの識別子のためには、次を参照してください: AuthenticDesktopControlPlaceholderWindow.

メモ: 各 ウィンドウ識 別 子 のために「AuthenticDesktopControl プレースホルダ」のみを使用 してください。

Authentic Desktop プロジェクトウィンドウを選択するプレースホルダーコントロールのために、 追加メソッドを使用することができます。「OpenProject」を使用して、Authentic Desktop プロジェクトをロードすることができます。Authentic Desktop 自動化インターフェイスからプロ パティプロジェクトとメソッドを使用して、他のプロジェクトに関連したオペレーションを操作しま す。

例えば、Visual Studio を使用した C# または VB.NET内では、Authentic Desktop ActiveX コントロ ールをドキュメントレベルで統合する基本的なアプリケーションの作成ステップは、以下にリストされる ものに類似しています。必要に応じて適用は複雑になる可能性があります。しかしながら、ActiveX ドキュメントレベルの統合のための必要条件として、下の命令を理解することは重要です。

- 1. 新規の空のフォームと共に新規の Visual Studio Windows Forms プロジェクトを作成します。
- 2. まだ行っていない場合、ActiveX コントロールを ツールボックスに追加します (次を参照してください: ActiveX コントロールをツールボックスに追加する)。
- 3. ツールボックスから <u>AuthenticDesktopControl</u> をドラッグして新規のフォームにドラッグします。
- AuthenticDesktopControl の「IntegrationLevel」プロパティを 「CActiveXIntegrationOnDocumentLevel」に、と「Visible」プロパティを「False」に 設定します。コードから、または、「プロパティ」ウィンドウから行うことができます。
- 5. <u>AuthenticDesktopControl</u> <u>ドキュメント</u> をツールボックスからフォームにドラッグします。 こ のコントロールは、使用中のアプリケーションに Authentic Desktop のメインウィンドウを与え、 ドキュメントのためにサイズを調整することができます。
- 代わりに、フォームに1つまたは複数の <u>AuthenticDesktopControl</u> プレースホルダ コントロ ールを追加することができます (アプリケーションが必要とする各追加ウィンドウの型など。例 えば、「プロジェクト」ウィンドウなど)。このような追加プレースホルダー コントロールをメイン ドキュメントコントロールの下、または、左右に通常配置します。例:



- ろ「AuthenticDesktopControl プレースホルダ」コントロールの 「PlaceholderWindowID」プロパティを有効なウィンドウ識別子に設定してください。有 効な値のリストに関しては、次を参照してください: AuthenticDesktopControlPlaceholderWindow。
- 8. アプリケーションにコマンドを以下のように追加します (少なくとも、ドキュメントを開き、保存し、閉じる必要があります)。

Authentic Desktop コマンドをクエリする

ドキュメントレベルで統合する場合は、Authentic Desktop メニュー または ツールバーをアプリケーションで使用することはできません。代わりに、必要とするコマンドを取得し、状態を確認し、プログラム的に実行することができます:

- 使用することのできるすべてのコマンドを取得するには、AuthenticDesktopControlの <u>CommandsList</u>プロパティを使用します。
- メニュー構造により整理されているコマンドを取得するには、メイン Menu プロパティを使用します。
- 表示 される順番 に整理 されているコマンドを取得するには、<u>Toolbars</u> プロパティを使用 します。
- Authentic Desktop にコマンドを送信するには、<u>Exec</u>メソッドを使用します。
- コマンドが現在有効化されているか、または、無効化されているかクエリするために、 <u>QueryStatus</u>メソッドを使用します。

これによりAuthentic Desktop コマンドをアプリケーションのメニューとソールバーに柔軟的に統合することができます。

Authentic Desktop のインストールは、Authentic Desktop 内で使用されているコマンドラベルを与え

ます。GIF 書式内のアイコンのために Authentic Desktop インストールのフォルダー 「<ApplicationFolder>\Examples\ActiveX\Images」を参照してください。fコマンド名に対応するファイル名は、コマンド参照 セクション内にリストされています。

一般的な注意点

Authentic Desktop の振る舞いを自動化するには、AuthenticDesktopControl、 AuthenticDesktopControl <u>ドキュメント</u>とAuthenticDesktopControl <u>プレースホルダ</u>内で説明されているプロパティメソッド、とイベントを使用してください。

Authentic Desktop 機能への高度なアクセスに関しては、次のプロパティを考慮してください:

- <u>AuthenticDesktopControl.Application</u>
- <u>AuthenticDesktopControlDocument.Document</u>
- <u>AuthenticDesktopControlPlaceHolder.Project</u>

これらのプロパティは Authentic Desktop 自動化インターフェイス (Authentic Desktop API)へのアクセスを提供します

メモ: ドキュメントを開くには、適切なドキュメントコントロールの <u>AuthenticDesktopControlDocument.Open</u> または <u>AuthenticDesktopControlDocument.New</u> を使用してください。 プロジェクトを開くには、 Authentic Desktop プロジェクトウィンドウを埋め込むプレースホルダ コントロール上の <u>AuthenticDesktopControlPlaceHolder.OpenProject</u> を常に使用してください。

異なるプログラミング環境内で必要なコントロールをインスタント化しアクセスする方法を説明するサンプルに関しては、次を参照してください: <u>ActiveX 統合のサンプル</u>。

4.5 ActiveX 統合のサンプル

このセクションは、異なるコンテナー環境とプログラム言語を使用するAuthentic Desktop ドキュメント レベルの統合のサンプル、を含みます (HTML セクションには、追加でアプリケーションレベルの統合の サンプルが含まれます)。全てのサンプルのソースコードは、Authentic Desktop インストール内のフォ ルダー <ApplicationFolder>\Examples\Activex で使用することができます。

4.5.1 C#

C# and Visual Studio のための基本の ActiveX 統合 サンプルソリューションは、フォルダー <ApplicationFolder>\Examples\ActiveX\C#内にあります。ソースコードをコンパイルする前に、 サンプルを実行して、全ての必要条件が満たされていることを確認してください(次を参照してください) い: サンプル C# ソリューションの実行)。

サンプル C# ソリューションの実行

フォルダー ApplicationFolder>LexamplesActiveXC#内で使用することのできるサンプル Visual Studio ソリューションは、Authentic Desktop ActiveX コントロールの使用方法について説明しています。 このソリューションをビルドして実行する前に、次のステップに注意してください:

ステップ 1: 必要条件をチェックする

Visual Studio 2010 または以降は、サンプルソリューションを開く必要があります。必要条件の完全なリストは、次を参照してください:<u>必要条件</u>。

ステップ 2:書 き込 みのパーミッションを保有するディレクトリにサンプルをコピーする

Visual Studio を管理者として作動することを回避するには、デフォルトの場所から実行する代わりに、書き込みのパーミッションを持つディレクトリへのソースコードをコピーします。

ステップ 3: すべての必要 とされるコントロールプロパティをチェックし設定する

サンプルアプリケーションは、AuthenticDesktopControlDocument の1つのインスタンスと AuthenticDesktopControlPlaceHolder コントロールの 複数のインスタンス を含んでいます。下のテー プルに表示 されているとおりにこれらのコントロールの次のプロパティが設定 されていることを確認してく ださい:

コントロール名	プロパティ	プロパティ値
axAuthenticDesktopControl	IntegrationLevel	ICActiveXIntegrationOnDocu mentLevel
axAuthenticDesktopControlHelperW ndEntities	PlaceholderWindo wlD	2
axAuthenticDesktopControlHelperW ndAttributes	PlaceholderWindo wlD	1
axAuthenticDesktopControlHelperW ndElements	PlaceholderWindo wlD	0
axAuthenticDesktopControlHelperW ndInfo	PlaceholderWindo wlD	18
axAuthenticDesktopControlHelperW ndProject	PlaceholderWindo wlD	4

ActiveX コントロールのプロパティをビュー、または、設定する方法は、以下のとおりです:

- 1. デザイナー ウィンドウ内 で MDIMain.cs フォームを開きます。
- ★E: 64 ビット Windows では、デザイナーウィンドウで開く前に Visual Studio ソリューションを 「x86」にビルドを更新する必要がある可能性があります。64 ビットアプリケーションとしてサンプルをビルドする場合は、次を参照してください:必要条件.

Debug	•	хб4	•	•	Sta	art 👻
Object Br	0000	хб4				
Object bit	Uvvs	x86				
Browse:	M	Configuration M	lan	ager.		

2. Visual Studio のドキュメント概要 ウィンドウを開きます(「表示」メニューから、「他のウィンドウ|ドキュメントの概要」をクリックします)。

Document Outline - MainFrame 🔹 🗖 🗙
🔺 🔚 MainFrame Form
🗗 axAuthenticDesktopControlHelperWndEntities AxAuthenticDesktopControlPlaceHolder
axAuthenticDesktopControlHelperWndAttributes AxAuthenticDesktopControlPlaceHolder
🗗 axAuthenticDesktopControlHelperWndElements AxAuthenticDesktopControlPlaceHolder
🗗 axAuthenticDesktopControlHelperWndInfo AxAuthenticDesktopControlPlaceHolder
🖬 axAuthenticDesktopControlHelperWndProject AxAuthenticDesktopControlPlaceHolder
axAuthenticDesktopControlDocument AxAuthenticDesktopControlDocument
axAuthenticDesktopControl AxAuthenticDesktopControl
🔺 🔄 mainMenu MenuStrip
🔺 🔄 fileMenultem ToolStripMenultem
🖹 newToolStripMenuItem ToolStripMenuItem
🔄 OpenDocumentMenuItem ToolStripMenuItem
🖹 saveToolStripMenuItem ToolStripMenuItem
🖹 exitToolStripMenuItem ToolStripMenuItem
🔺 🔄 projectToolStripMenuItem ToolStripMenuItem
🔄 openToolStripMenuItem ToolStripMenuItem
🖹 saveProjectToolStripMenuItem ToolStripMenuItem
🖹 closeToolStripMenuItem ToolStripMenuItem

3. ドキュメント概要 ウィンドウ内の ActiveX コントロールをクリックし、プロパティウィンドウ内の 必要 とされるプロパティを編集 します。例:

Pr	operties	- □	×
a	AuthenticDesktopControl AxA	$uthentic {\sf ControlLib}. A x {\sf Authentic} {\sf Desktop} {\sf Control}$	•
	💱 🖓 F 🖉		
Ŧ	(ApplicationSettings)		*
Ŧ	(DataBindings)		
	(Name)	axAuthenticDesktopControl	
	AccessibleDescription		
	AccessibleName		
	AccessibleRole	Default	
	AllowDrop	False	
	Anchor	Top, Left	
	Appearance	0	
	BorderStyle	0 - None	
	CausesValidation	True	
	ContextMenuStrip	(none)	
	Dock	None	
	EnableUserPrompts	True	
	GenerateMember	True	
	IntegrationLevel	ICActiveXIntegrationOnDocumentLevel 🗸	
Ŧ	Location	316, 172	•

IntegrationLevel

ステップ 4: ビルドプラットフォームを設定する

- ビルド(x86、x64)するプラットフォームに一致するビルドプラットフォームの更新を作成します。 ビルドの更新の作成の方法は以下のとおりです:
 - a. Visual Studio内のソリューションを右クリックし、「更新マネージャー」を選択します。
 - b. 「アクティブなソリューションプラットフォーム」から、「新規作成…」を選択し、x86 または x64 更新を選択します (このサンプルでは、x86)。

New Solution Platform			
Type or select the new platform:			
x86 🗸			
Copy <u>s</u> ettings from:			
Any CPU 🔹			
☑ Create new project platforms			
OK Cancel			

Visual Studio 内でソリューションをビルドして実行する準備が整うと、ターゲットプラットフォーム (x86、x64) に一致するように更新をしようしてビルドすることを忘れないでください。それ以外の場合、ランタイムエラーが起こる可能性があります。

4.5.2 HTML

このセクションのコードリストは、アプリケーションレベルとドキュメントレベルでの AuthenticDesktopControl の統合方法を示しています。全てのサンプルのためのソースコードは Authentic Desktop インストールのフォルダー <ApplicationFolder>\Examples\ActiveX\HTML内 にあります。サンプルは、32 ビットアプリケーションとして作動する場合、Internet Explorer内でのみ 作動することができます。

メモ: 64 ビットモード上で作動する場合、nternet Explorer 10 または以降は、ActiveX コントロー ルをロードしません。

アプリケーションレベルの HTML 統合

このサンプルは、アプリケーションレベルでの HTML ページへの Authentic Desktop コントロールの簡単な統合について説明しています。統合に関しては、次のセクションで説明されています:

- HTML コード内で AuthenticDesktopControl を開始します。
- ボタンがドキュメントをロードするように実装し、コード生成タスクを自動化します。
- アプリケーションイベントのためのアクションを定義する。

このサンプルのためのコードは、Authentic Desktop インストール内の次の場所で使用することができます:

<ApplicationFolder>\Examples\ActiveX\HTML\AuthenticActiveX ApplicationLevelhtm.

コントロールインスタンスを作成する

HTML Object タグは、AuthenticDesktopControl のインスタンスを作成 するために使用 されます。 Classid は、AuthenticDesktopControl です。幅と高さがウィンドウのサイズを指定します。アプリケー ションレベルがデフォルトのため、追加のパラメーターは必要ありません。

デフォルトのドキュメントを開くボタンを追加する

- 部のタスクの自動化のシンプルなサンプルとして、ページにボタンを追加します:

<input type="button" value="Open Marketing Expenses" onclick="BtnOpenMEFile()">

クリックされると 定義済みドキュメントが AuthenticDesktopControl 内で開かれます。異なるインストール上でサンプルを作動するため、AuthenticDesktopControl に対して相対的なファイルを検索するためにメソッドを使用します。

カスタムイベントへの接続

サンプルは、プリンシパルを表示するために AuthenticDesktopControl カスタムイベントのための 2つの イベントのコールバックを実装します:

```
</SCRIPT>
```

ドキュメントレベルの HTML 統合

このサンプルは、ドキュメントレベルで HTML ページに Authentic Desktop コントロールを統合する方法 について説明しています。次のトピックについて説明されています:

- HTML コード内でAuthenticDesktopControl ActiveX コントロール オブジェクトをインスタンス化します。
- Authentic Desktop ファイルの編集を許可する AuthenticDesktopControlDocument ActiveX コントロールをインスタンス化します。
- AuthenticDesktopControl プロジェクトウィンドウのための AuthenticDesktopControlPlaceHolder をインスタンス化します。
- Authentic Desktop ヘルパーウィンドウをホストするために AuthenticDesktopControlPlaceHolder をインスタンス化します。
- 頻繁に使用される Authentic Desktop コマンドのためのシンプルなカスタマーツールバーを作成します。
- Authentic Desktop の COM 自動化 インターフェイスを使用するボタンを追加します。
- コマンドボタンを更新するためのイベントハンドラーを使用します。

このサンプルは、Authentic Desktop インストールの<ApplicationFolder>\Examples\ActiveX \HTML\ フォルダー内のファイル Authentic DesktopActiveX_ApplicationLevel.htm 内で使用 することがあります。

AuthenticDesktopControl をインスタンス化する

AuthenticDesktopControl HTML OBJECT タグは、AuthenticDesktopControl のインスタンスを作成 するために使用されます。 クラス ID は AuthenticDesktopControl です。ユーザーインターフェイスのた めのコントロール間 ジェージャーとして使用される幅と高さは、0 に設定されています。 統合 レベルは OBJECT タグ内 でパラメーターとして指定されています。

```
<OBJECT id=""
width="0"
height="0"
VIEWASTEXT>
<PARAM NAME="IntegrationLevel" VALUE="1">
</OBJECT>
```

エディターウィンドウの作成

HTML OBJECT タグは、編集 ウィンドウを埋め込むために使用されます。

```
<OBJECT id="objDoc1"
width="600"
height="500"
VIEWASTEXT>
</OBJECT>
```

プロジェクトウィンドウの作成

HTML OBJECT タグは、AuthenticDesktopControlPlaceHolder ウィンドウを作成するために使用されます。最初の追加カスタムパラメーターは、Authentic Desktop プロジェクトウィンドウを表示するためのプレースホルダを定義します。2番目のパラメーターは、Authentic Desktop インストールで提供されたサンプルプロジェクトの1つをロードします(以下の場所で見つけることができます: <yourusername>/MyDocuments フォルダー)。

```
<OBJECT id="objProjectWindow"
width="200"
height="200"
VIEWASTEXT>
</OBJECT>
```

ヘルパーウィンドウのためにプレースホルダを作成する

HTML OBJECT タグは、異なる Authentic Desktop ヘルパーウィンドウをホストすることのできる AuthenticDesktopControlPlaceHolder ActiveX コントロールをインスタンス化 するために使用 されま す。最初から、ヘルパーウィンドウは表示 されません。サンプルファイルを参照してください。

```
<OBJECT id="objEHWindow"
width="200"
height="200"
VIEWASTEXT>
<PARAM name="PlaceholderWindowID" value="">
</OBJECT>
```

3つのボタンにより表示される実際のウィンドウを切り替えることができます。 JavaScript 実行ボタンを クリックすることにより、プロパティ PlaceHolderWindow D を定義されている対応する値に設定します。

```
<SCRIPT ID="Javahandlers" LANGUAGE="javascript">
//
//
// specify which of the windows shall be shown in the placeholder control.
function BtnHelperWindow(i_ePlaceholderWindowID)
{
        objEHWindow.PlaceholderWindowID = i_ePlaceholderWindowID;
}
</SCRIPT>
```

4.5.3 Java

Authentic Desktop ActiveX コンポーネントに Java コードからアクセスすることができます。Java 統合は 下にリストされるライブラリで提供されています。これらのライブラリは、Authentic Desktop と Authentic Desktop Integration Package の両方をインストールした後、Authentic Desktop インスト ール内のフォルダー <ApplicationFolder>\Examples\JavaAPI で使用することができます(次を参 照してください:必要条件)。

- AltovaAutomation.dll: Altova 自動サーバーのための JNI ラッパー (Authentic Desktop の 32 ビット インストールの場合)
- AltovaAutomation_x64.dll: Altova 自動サーバーのための JNI ラッパー (Authentic Desktop の 64 ビット インストールの場合)
- AltovaAutomation.jar: Altova 自動サーバーにアクセスします
- AuthenticActiveX.jar: Authentic ActiveX インターフェイスをラップする Java クラス
- AuthenticActiveX_JavaDoc.zip: Javadoc インターフェイスのためのヘルプドキュメントを含む Javadoc ファイル
- メモ: Java ActiveX 統合を使用するには、 dll と jar ファイルが Java クラス 検索パス内に含まれている必要があります。

サンプル Java プロジェクト

サンプル Java プロジェクトは製品のインストールにより提供されます。 Java プロジェクト をテストし、使用用途に応じて変更することができます。詳細に関しては、次を参照してください: サンプル Java プロジェクト。

JavaRules に ActiveX コントロール名 をマッピングする際 のルール

ActiveX コントロールのドキュメントに関しては、次を参照してください: オブジェクトレファレンス。オブ ジェクトの名前変換は Java の場合、他の言語と比較して、若干異なります。具体的には、 ActiveX コントロールとJava ラッパー間のマッピングのためのルールは、以下のとおりです:

クラスとクラス名

Authentic Desktop ActiveX インターフェイスの全 てのコンポーネントのために、コンポーネントの名前を持つ Java クラスが存在します。

メソッド名

Javadoc インターフェイス上のメソッド名は、COM インターフェイス上で使用 される名前と同じ ですが、Java 名前変換に準拠する小文字で開始します。COM プロパティにアクセスする には、プロパティ名をプレフィックスする get とset から始まる Java メソッド使用 することができ ます。 ぷプロパティが書き込みをサポートしない場合、setter メソッドを使用 することはできま せん。サンプル: AuthenticDesktopControlの IntegrationLevel プロパティ、Java メソッド getIntegrationLevel とsetIntegrationLevel を使用 することができます。

列挙

ActiveX インターフェイス内で定義される全ての列挙のために、Java 列挙は同じ名前と値を 使用して定義されます。

• イベント とイベントハンドラー

イベントをサポートする自動化インターフェイス内の全てのインターフェイスのために、同じ名前とイベントを使用した Javadoc インターフェイスを使用することができます。単一のイベントのオーバーロードを簡素化するために、全てのイベントのためのデフォルトの実装を使用する Java クラスが提供されます。 この Java クラスの名前は、イベントインターフェイスの名

前と DefaultHandler'です。例:

AuthenticDesktopControl:アプリケーションにアクセスするための Java クラス AuthenticDesktopControlEvents:AuthenticDesktopControl のためのイベント インターフェ イス

AuthenticDesktopControlEventsDefaultHandler:AuthenticDesktopControlEvents のためのデフォルトのハンドラー

マッピングルールの例 外

上記のルールの例外の一部は、以下のとおりです。これらは下にリストされています:

インターフェイス	Java 名
AuthenticDesktopControlDocument, メ ソッド New	newDocument
Document, メソッド SetEncoding	setFileEncoding
AuthenticView, メソッド Goto	gotoElement
AuthenticRange, メソッド Goto	gotoElement
AuthenticRange, メソッド Clone	cloneRange

このセクション

このセクションは、Java コードからアクセスすることのできる基本の Authentic Desktop ActiveX 機能の一部を紹介しています。次のサブセクションに整理されています:

- <u>サンプル Java</u> <u>プロジェクト</u>
- <u>ActiveX コントロールの作成</u>
- <u>コントロール内のデータのロード</u>
- 基本のイベントハンドリング
- <u>۲ ح ۲</u>
- <u>UI アップデートイベントハンドリング</u>
- <u>XML ツリーの作成</u>

サンプル Java プロジェクト

Authentic Desktop インストールパッケージには、アプリケーション フォルダー: <ApplicationFolder> \Examples\ActiveX\Java\ の ActiveX サンプルフォルダーにあるサンプル Java プロジェクトが含まれ ています。

Java サンプルは、Java. を使用して作成された共有されるデスクトップアプリケーション内に AuthenticDesktopControl を統合する方法を示しています。

バッチファイル BuildAndRun.bat を使用してコマンドラインからテストすることができ、また、Eclipse内からサンプルプロジェクトをコンパイルして実行することもできます。これらのプロシージャの使用方法を参照してください。

ファイルリスト Java サンプルフォルダーには、サンプルプロジェクトを作動するために必要な全てのファイルが含まれて

います。これらのファイルは下にリストされています:

.classpath	Eclipse プロジェクトヘルパーファイル
.project	Eclipse プロジェクトファイル
AltovaAutomation.dll	Java-COM ブリッジ: DLL パート (32-ビットインスト ール用)
AltovaAutomation_x64.dll	Java-COM ブリッジ: DLL パート (64-ビットインスト ール用)
AltovaAutomation.jar	Java-COM ブリッジ: Java ライブラリパート
AuthenticActiveX.jar	Authentic Desktop ActiveX コントロールの Java クラス
AuthenticActiveX_JavaDoc.zip	Java API のためのヘルプドキュメントを含む Javadoc ファイル
AuthenticDesktopContainer.java	Java サンプルソースコード
AuthenticDesktopContainerEventHandler .java	Java サンプルソースコード
BuildAndRun.bat	コンパイルするバッチファイル、および、コマンドライ ンプロンプトからサンプルコードを実行します。 Java 仮想マシンがパラメータとして存在する箇所 にフォルダーが期待されます。
XMLTreeDialog.java	Java サンプルソースコード

サンプルの内 容

サンプルは、AWT フレームウィンドウ内 に AuthenticDesktop ドキュメント エディターウィンドウ,プロ ジェクトウィンドウ、情報ウィンドウとAuthentic 入力 ヘルパーを表示します。 Authentic のために定 義されているファイルメニューを読み取り、同じ構造を持つ AWT メニューを作成します。 このメニュー またはプロジェクトを使用して、ドキュメントエディター内でファイルを開き作業することができます。

サンプルを必要に応じて変更することができます.

コードリスト内で次の特定の機能が説明されています:

- <u>ActiveX コントロールの作成</u>:自動サーバーとして考えられている Authentic Desktop を返します。または、作動中の場合、Authentic Desktop をアクティブ化します。
- <u>コントロール内のデータのロード</u>: Authentic Desktop にインストールされているサンプルドキュ メントの1つを検索し開きます。
- <u>基本のイベントハンドリング</u>: すべての開かれているドキュメントのビューをテキストビューに変えます。 コードも開かれているドキュメント内での反復の方法を示しています。
- <u>メニュー</u>: アクティブなドキュメントを検証し、メッセージボックス内に結果を表示します。 コードは出力パラメータの使用方法について説明しています。
- <u>UI アップデートイベントハンドリング</u>: Authentic Desktop イベントの処理方法について説明 しています。

• XML ツリーの作成: XML ツリーの作成 とモーダルアクティブ化 について説明しています。

サンプルフォルダーへのパスの更新

与えられているサンプルを作動する前に、AuthenticDesktopContainer.java ファイルを編集します。 具体的には、Authentic Desktop サンプルファイルが使用中のオペレーティングシステム上で保存 されている実際のフォルダーを参照する次のパスを確認します:

// Locate samples installed with the product.
final String strExamplesFolder = System.getenv("USERPROFILE") + "\\My
Documents\\Altova\\Authentic2018\\AuthenticExamples\\";

コマンドラインからサンプルを実行する

コマンドラインからサンプルを作動します:

- すべての必要条件が満たされていることを確認してください(次を参照してください:必要条件).
- 2. コマンドプロンプトウィンドウを開き、サンプル Java プロジェクト フォルダーに対して現在のディレクトリを変更し、次を入力します:

buildAndRun.bat "<Path-to-the-Java-bin-folder>"

3. **「Enter」**を押します。

AuthenticDesktopContainer.java内のJava ソースがコンパイルされ、実行されます。

Eclipse 内 でサンプルをコンパイルし実 行 する

サンプル Java プロジェクトを Eclipse にインポートする:

- すべての必要条件が満たされていることを確認してください(次を参照してください:<u>必要条件</u>).
- 2. ファイル メニュー から、「インポート」をクリックします。
- 3. 「ワークスペースで既存のプロジェクトを開く」を選択し、次の場所にある Eclipse プロジェクトファイルを参照します:<ApplicationFolder>\Examples\ActiveX\Java\。このフォルダーでは書き込みのアクセスがないため、インポートダイアログボックス上の「ワークプレースにプロジェクトをコピーする」チェックボックスを選択することが奨励されます。

サンプルアプリケーションを作動するには、Package Explorer内のプロジェクトを右クリックし、コマンド「実行 | Java アプリケーション」を選択します。

コード内のコメントを使用するとJava API クラスのためのヘルプを使用することができ、とEclipse内で Javadoc ビューを有効化するには、メニューコマンド「Window | ビューの表示 | JavaDoc」を選択 してください。

ActiveX コントロールの作成

下のコードリストは、ActiveX コントロールの作成方法を示しています。コンストラクターは、Java ラッパーオブジェクトを作成します。カンバスにより派生したオブジェクトをパネル、または、フレームに追加 すると、ラップされた ActiveX の生成がトリガーされます。

```
01 /**
    * Authentic Desktopmanager control-always needed
02
   */
03
04 public static AuthenticDesktopControl
                                             authenticDesktopControl= null:
05
06
    * Authentic Desktop document editing control
07
08 *,
09 public static AuthenticDesktopControDocument authenticDesktopDocument = null;
10
11
12
    * Toolwindows - Authentic Desktop place-holder controls
13
14 private static AuthenticDesktopControPaceHober authenticDesktopInfoTooWindow = null;
15 private static AuthenticDesktopControPheceHolder authenticDesktopEHElementTooWindow = null;
16 private static AuthenticDesktopControPheceHobler authenticDesktopProjectTooWindow = null;
17
18
     // Create the Authentic Desktop ActiveX control, The parameter determines that we want
   // to place document controls and place-hobler controls individually.
19 // It gives us fullcontrolover the menu, as well.
20
    authenticDesktopControl= new AuthenticDesktopControl(
    CActiveXIntegrationLevelCActiveXIntegrationOnDocumentLevelgetValue();
21
22
    authenticDesktopDocument = new AuthenticDesktopControDocument();
23
     authenticDesktopDocumentsetPreferredSize(new Dimension (640,480));
24
    fram e add (authenticDesktopDocument, BorderLayoutCENTER );
25
26 // Create a project window and open the sample project in it
27
     authenticDesktopProjectTooWindow = new AuthenticDesktopControPaceHolder(
    XMLSpyControPacehoberWindow XMLSpyControProjectWindowTooWndgetValue();
28
    authenticDesktopProjectTooWindow setPreferredSize (new Dimension (200, 200));
```

コントロール内のデータのロード

下のコードリストは、ActiveX コントロール内 にデータをロードを表示する方法を表示しています。

```
1 // Locate samples installed with the product.
```

2

- finalString strExamplesFobler = System getenv("USERPROFILE")+
- "\\MyDocuments\\Alova\\Authentic2018\\AuthenticExamples\\";
- authenticDesktopProjectTooWindow = new

 $\label{eq:linear} Authentic Desktop Contro P heet o bler (XM LS py Contro P heet o bler W indow XM LS py Contro P roject W indow Too W ndget Value ();$

基本のイベントハンドリング

下のコードリストは、基本のイベントの処理方法について表示しています。AuthenticDesktopControl のopen メソッド、または、メニューまたはプロジェクトツリーを使用してファイルを開こうとすると onOpenedOrFocused イベントは、アタッチ済みのイベントハンドラーに送信されます。このイベントの ための基本の処理により、Authentic DesktopDocumentControlのopen メソッドを呼び出すことによ リファイルを開きます。

```
01 // Open the PXF file when button is pressed
     btn0penPxfaddActionListener(new ActionListener) {
03
       public void actionPerformedActionEvente) {
04
        try {
05
          authenticDesktopControlopen(strExamplesFobler + "OrgChartpxf");
        }catch AutomationException e1) {
06
07
         el printStackTrace ();
08
        }
09
10
      });
      public void onOpenedOrFocused(String istrfileName, boolean ibOpenW thThisControl, boolean
11
ibFileA lreadyO pened ) throws AutomationException
12
   {
```

// Handle the New/Open events coming from the Project tree or from the menus 13 14 if (libFileA lreadyO pened) 15 16 // This is basically an SDI interface, so open the file in the already existing document control 17 trv { 18 AuthenticDesktopContainerauthenticDesktopDocumentopen(istrFieName); 19 AuthenticDesktopContainerauthenticDesktopDocumentrequestFocusInWindow (); 20 {catch Exception e) { e printStackTrace (); 21 22

23 } 24 }

メニュー

下のコードリストは、メニューアイテムの作成方法を表示しています。 各 Authentic DesktopCommand オブジェクトは、コマンドの ID へのセットされている ActionCommand を持つ、対応 する MenuItem オブジェクト、を取得します。全てのメニューアイテムにより生成されているアクション は、(閉じられたメカニズムのを再度解釈するなどの)特定の処理を行う同じ関数により処理されま す。または、exec メソッドを呼び出すことによりAuthenticDesktopControl オブジェクトに実行を 委任することができます。メニューの作成中にフィルされる menuMap オブジェクトは、後で使用されま す (次のセクションを参照してください: UI アップデートイベントハンドリング).

```
01 // Load the file menu when the button is pressed
02
      btnMenuaddActionListener(new ActionListener) {
       public void actionPerform ed (ActionEvente) {
03
04
        try {
05
          // Create the menubar that will be attached to the frame
06
         MenuBarmb = new MenuBar();
07
          // Load the main menu's first item - the File menu
         XMLSpyComm and xm BpyMenu = xm BpyControlgetMainMenu (getSubComm and s)getItem (0);
08
          // Create Java menu tems from the Commands objects
09
10
         Menu fileMenu = new Menu ();
11
         handlerObjectfillenu (fileMenu, xm BpyMenu getSubCommands ();
12
         fileMenu setLabel(xm B pyMenu getLabel) replace ("&", ""));
13
         mbadd(fileMenu);
14
          framesetMenuBar(mb);
15
          fram e validate ();
16
        }catch (AutomationException e1) {
17
         el printStackTrace ();
18
         // Disable the button when the action has been performed
19
20
         (AbstractButton) e getSource () setEnabled (false );
21
       }
22
      });
   /** * Populates a menu with the commands and submenus contained in an XMLSpyCommands object */
23
24
      public void fillenu Menu newMenu, XMLSpyCommands xm BpyMenu) throws AutomationException
25
26
27
     // Foreach command/submenu in the xm BpyMenu
     for (int i= 0 ; i< xm BpyMenugetCount();++i)</pre>
28
29
      XMLSpyComm and xm BpyComm and = xm BpyMenu getItem (i);
30
      if (xm BpyComm and getIsSeparator≬)
31
       newMenuaddSeparator();
32
      else
33
34
       XMLSpyCommands subCommands = xm BpyCommandgetSubCommands);
35
        // Is it a command (eaf),or a submenu?
36
       if (subCommands.isNull() || subCommands.getCount() == 0)
37
38
         // Command -> add it to the menu, set its ActionCommand to its D and store it in the menuMap
39
        MenuItem mi= new MenuItem (xm BpyCommandgetLabel) replace ("&",""));
40
        misetActionCommand("" + xm BpyCommandgetD();
41
        miaddActionListener(this);
42
        newMenuadd(mi);
43
        menuMapput(xm BpyCommandgetD (,mi);
44
45
       else
46
47
         // Submenu -> create submenu and repeat recursively
```

```
48
        Menu newSubMenu = new Menu ();
49
        filM enu (newSubMenu, subComm ands );
50
        newSubMenusetLabel(xmBpyCommandgetLabel()replace("&",""));
51
        newMenuadd(newSubMenu);
52
53
      }
54
     }
55
56
   }
57
    /**
    * Action handler for the menu items
58
    * Called when the user selects a menu item ; the item is action command corresponds to the command table for
59
XMLSpy
60
61 public void actionPerformed(ActionEvente)
62
63
     try
64
65
      inticmd = IntegerparseInt(egetActionCommand());
66
      // Handle explicitly the Close commands
67
      switch (iCmd)
68
69
       case 57602:
                       //Cbse
70
71
                       // Cbse All
       case 34050:
        AuthenticDesktopContainer.intXm BpyDocument();
72
73
74
75
76
        break;
       de fault:
        AuthenticDesktopContainerxm BpyControlexec (Cmd);
        break;
      }
77
78
     }
    catch (Exception ex)
79
80
      exprintStackTrace ≬;
81
     }
82
83 }
```

UI アップデー トイベン トハンドリング

下のコードリストは、UI-Update イベントハンドラーの作成方法を表示しています。

```
/**
01
              * Call-back from the XMLSpyControl.
02
             * Called to enable/disable commands
03
              *,
04
05 @Override
06 public void on Update Cm dU I() throws Automation Exception
07
              ł
08
                 // A command should be enabled if the result of queryS tatus contains the Supported (1) and Enabled (2) flags
09
               for (javautilMapEntry<Integer,MenuItem>pair:menuMapentrySet())
                 pairgetValue (setEnabled (AuthenticDesktopContainerauthenticDesktopControlqueryStatus (pairgetKey())>2);
10
11 }
           .
/**
12
             * Call-back from the XMLSpyControl.
13
14
             * Usually called while enabling/disabling commands due to UI updates
            */
15
16 @Override
17 public boolean on IsActiveEditor (String istrFilePath ) throws AutomationException
18
19
                try {
20
                   return
\label{eq:authentic} Points and a state of the state of
21
               } catch (Exception e) {
22
                    return false;
23
                 }
24
          }
```

XML ツリーの作成

下のリストは、XML データオブジェクトをツリー内にノードとしてロードします。

```
01 // access required XMLSpy Java-COM classes
02 in port com altova autom ation XMLSpyXMLData;
03
04 // access AWT and Swing components
05 im port java awt.*;
06 in port javax swing *;
07 im port javax swing tree *;
08
09 /**
10 * A simple example of a tree control bading the structure from an XMLData object.
11 * The class receives an XMLData object, bads its nodes in a JTree, and prepares
12 * formodalactivation.
13 *
14 * Feelfree to modify and extend this sample.
15 *
16 * @authorAltova GmbH
17 */
18 class XMLTreeDabg extends JDabg
19 {
     /**
20 /
    * The tree control
21
22
23 private JTree myTree;
24
25
26
    * Root node of the tree control
27
28 private DefaultMutableTreeNode top;
29
30
    * Constructor that prepares the modaldabg containing the filled tree control
31
32
    * Oparam xml The data to be displayed in the tree
33
    * Oparam parent Parent frame
34
35
   public XMLTreeDiabg(XMLData xml, Frame parent)
36
    {
37
     // Construct the modaldabg
38
    super(parent, "XML tree", true );
39
     // Arrange controls in the diabg
40
    top = new DefaultMutableTreeNode ("root");
41
    myTree = new JTree (top);
42
    setContentPane (new JScrolPane (myTree));
43
     // Build up the tree
44
    fill ree (top, xm l);
45
    myTreeexpandRow(0);
46
   }
47
   /**
48
   * Loads the nodes of an XML element under a given tree node
49
50
    * @ param node Target tree node
    * @param elem Source XML element
51
     *
52
   private void fill ree (Default Mutable TreeNode node, XMLData elem)
53
54
    {
55
     try
56
     {
57
      // There are severalways to treate through child elements: ether using the getFirstChild/getNextChild,
58
      // or by incrementing an index up to countChildren and calling getChild [as shown be bw].
      // If you only want to get children of one kind, you should use countChildrenKind/getChildKind,
59
      // or provide a kind to the getFirstChild before terating with the getNextChild.
60
61
      intnSize = elem countChildren();
62
      for (int i= 0 ; i< nSize ;++i)</pre>
63
64
        // Create a new tree node for each child element, and continue recursively
65
       XMLData newElem = elem getChild();
66
       DefaultMutableTreeNode newNode = new DefaultMutableTreeNode (newElem getName());
67
       node add(newNode);
68
       fillree (newNode, newElem );
69
      }
```

4.6 コマンドレファレンス

このセクションは、Authentic Desktop内で使用することのできる全てのメニューコマンドの名前と識別子をリストしています。各サブセクションはAuthentic Desktop. コマンドテーブルの対応するトップレベルメニューからのコマンドが以下のとおりに整理されています:

- 「メニューアイテム」列は、コマンドのメニューテキストを Authentic Desktop で表示 される順番に表示します。コマンドの機能を簡単に識別することができます。
- 「コマンド名」列は、Authentic Desktop インストール ディレクトリの ActiveX\Images フォル ダーから同じ名前を持つアイコンを取得するために使用される文字列を指定します。
- 「D」列は、このコマンドを実行またはクエリするメソッドの引数として与えられる列の数値識別子を表示します。

コマンドを実行するには、<u>AuthenticDesktopControl.Exec</u>または <u>AuthenticDesktopControlDocument.Exec</u>メソッドを利用します。コマンドのステータスをクエリする には、<u>AuthenticDesktopControl.QueryStatus</u>または <u>AuthenticDesktopControlDocument.QueryStatus</u>メソッドを使用します。

インストールされた Authentic Desktop のエディションにより、コマンドの一部 がサポートされない場合があります。

4.6.1 ワァイル」メニュー

「ファイル」メニューには以下のコマンドが含まれます:

メニューアイテム	コマンド名	ID
新規作成	ID_FILE_NEW	57600
開く	ID_FILE_OPEN	57601
再ロード	IDC_FILE_RELOAD	34065
エンコード	IDC_ENCODING	34061
閉 じる	ID_FILE_CLOSE	57602
全て閉じる	IDC_CLOSE_ALL	34050
アクティブなファイル以外閉 じる	IDC_CLOSE_OTHERS	34271
保存	ID_FILE_SAVE	57603
名前を付けて保存	ID_FILE_SAVE_AS	57604
全て保存する	ID_FILE_SAVE_ALL	34208
電子メールで送信する	ID_FILE_SEND_MAIL	57612
印刷	ID_FILE_PRINT	57607
印刷 プレビュー	IDC_PRINT_PREVIEW	34104
印刷設定	ID_FILE_PRINT_SETUP	57606
最近使ったファイル	ID_FILE_MRU_FILE1	57616
終了	ID_APP_EXIT	57665

4.6.2 編集」メニュー

「編集」メニューには以下のコマンドが含まれます:

メニューアイテム	コマンド名	ID
元に戻す	ID_EDIT_UNDO	57643
や垣し	ID_EDIT_REDO	57644
切り取り	ID_EDIT_CUT	57635
<u>コピー</u>	ID_EDIT_COPY	57634
貼り付け	ID_EDIT_PASTE	57637
削除	ID_EDIT_CLEAR	57632
全て選択	ID_EDIT_SELECT_ALL	57642
検索	ID_EDIT_FIND	57636
次を検索	ID_EDIT_REPEAT	57640
置換	ID_EDIT_REPLACE	57641

4.6.3 プロジェクト」メニュー

「プロジェクト」メニューには以下のコマンドが含まれます:

メニューアイテム	コマンド名	ID
新規のプロジェクト	IDC_ICPROJECTGUI_NEW	37200
プロジェクトを開く	IDC_ICPROJECTGUI_OPEN	37201
プロジェクトを再 ロードする	IDC_ICPROJECTGUI_RELOAD	37202
プロジェクトを閉 じる	IDC_ICPROJECTGUI_CLOSE	37203
プロジェクトを保存する	IDC_ICPROJECTGUI_SAVE	37204
プロジェクトに名前を付けて保存する	IDC_ICPROJECTGUI_SAVE_AS	37207
ソース管理を有効化する	ID_SCC_ENABLE	38602
プロジェクトにファイルを追加する	IDC_ICPROJECTGUI_ADD_FILES_TO_P ROJECT	37205
グローバルリソースをプロジェクトに追加す る	IDC_ICPROJECTGUI_ADD_GLOBAL_RE SOURCE_TO_PROJECT	37239
URL をプロジェクトに追加する	IDC_ICPROJECTGUI_ADD_URL_TO_PR OJECT	37206
アクティブなファイルをプロジェクトに追加す る	IDC_ICPROJECTGUI_ADD_ACTIVE_FIL E_TO_PROJECT	37208
アクティブならびに関係 するファイルをプロ ジェクトに追加する	IDC_ICPROJECTGUI_ADD_ACTIVE_AN D_RELATED_FILES_TO_PROJECT	37209
プロジェクトフォルダーをプロジェクトに追加 する	IDC_ICPROJECTGUI_ADD_FOLDER_TO _PROJECT	37210
外部フォルダーをプロジェクトに追加する	IDC_ICPROJECTGUI_ADD_EXT_FOLDE R_TO_PROJECT	37211
外部 Web フォルダーをプロジェクトに追加 する	IDC_ICPROJECTGUI_ADD_EXT_URL_F OLDER_TO_PROJECT	37212
スクリプト設定	IDC_PROJECT_SCRIPT_SETTINGS	34136
プロパティ	IDC_ICPROJECTGUI_PROJECT_PROPE RTIES	37223
最近使ったプロジェクト	IDC_ICPROJECTGUI_RECENT	37224

4.6.4 「XML」メニュー

MML」メニューには以下のコマンドが含まれます:

メニューアイテム	コマンド名	ID
整形式のチェック	IDC_CHECK_WELL_FORM	34049
XML の検証	IDC_VALIDATE	32954

4.6.5 「XSL/ XQuery」メニュー

XSL/XQuery」メニューには以下のコマンドが含まれます:

メニューアイテム	コマンド名	ID
XSL 変換	IDC_TRANSFORM_XSL	33006
XSL-FO 変換	IDC_TRANSFORM_XSLFO	33007
XSL パラメーター /XQuery 変数	IDC_TRANSFORM_XSL_PARAMS	33008

4.6.6 「Authentic」メニュー

「Authentic」メニューには以下のコマンドが含まれます:

メニューアイテム	コマンド名	ID
新規ドキュメント	IDC_AUTHENTIC_NEW_FILE	34036
データベースデータの編集	IDC_AUTHENTIC_EDIT_DB	34035
StyleVision スタイルシートの編集	IDC_EDIT_SPS	34060
XML データの新たな行を選択し編集	IDC_CHANGE_WORKING_DB_XML_CE	32861
XML 署名	IDC_AUTHENTICGUI_XMLSIGNATURE	32862
XML エンティティを定義	IDC_DEFINE_ENTITIES	32805
マークアップを隠 す	IDC_MARKUP_HIDE	32855
小さなマークアップの表示	IDC_MARKUP_SMALL	32858
大きなマークアップの表示	IDC_MARKUP_LARGE	32856
混合マークアップの表示	IDC_MARKUP_MIXED	32857
太字	IDC_AUTHENTICGUI_RICHEDIT_TOGGL EBOLD	32813
イタリック	IDC_AUTHENTICGUI_RICHEDIT_TOGGL EITALIC	32814
下線	IDC_AUTHENTICGUI_RICHEDIT_TOGGL EUNDERLINE	32815
取じ消し線	IDC_AUTHENTICGUI_RICHEDIT_TOGGL ESTRIKETHROUGH	32816
前景色	IDC_AUTHENTICGUI_RICHEDIT_COLOR _FOREGROUND	32824
背景色	IDC_AUTHENTICGUI_RICHEDIT_COLOR _BACKGROUND	32830
左寄せ	IDC_AUTHENTICGUI_RICHEDIT_ALIGN_ LEFT	32818
	IDC_AUTHENTICGUI_RICHEDIT_ALIGN_ CENTER	32819
 右寄せ	IDC_AUTHENTICGUI_RICHEDIT_ALIGN_ RIGHT	32820
行の追加	IDC_ROW_APPEND	32806
行の挿入	IDC_ROW_INSERT	32809

メニューアイテム	コマンド名	ID
行の複製	IDC_ROW_DUPLICATE	32808
行を上に移動	IDC_ROW_MOVE_UP	32811
行を下に移動	IDC_ROW_MOVE_DOWN	32810
行の削除	IDC_ROW_DELETE	32807
HTML ドキュメントの生成	IDC_PXF_GENERATE_HTML	34283
RTF ドキュメントの生成	IDC_PXF_GENERATE_RTF	34284
PDF ドキュメントの生成	IDC_PXF_GENERATE_PDF	34285
Word 2007+ ドキュメントの生成	IDC_PXF_GENERATE_DOCX	34286
信頼できる場所	IDC_TRUSTED_LOCATIONS	34288

「表示」メニューには以下のコマンドが含まれます:

メニューアイテム	コマンド名	ID
Authentic ビュー	IDC_VIEW_CONTENT	34177
ブラウザービュー	IDC_VIEW_BROWSER	34176
テキストビュー設定	IDC_TEXTVIEW_SETTINGS	34119

4.6.8 プラウザー」メニュー

「ブラウザー」メニューには以下のコマンドが含まれます:

メニューアイテム	コマンド名	ID
戻る	IDC_STEP_BACK	32958
進む	IDC_STEP_FORWARD	32957
中止	IDC_BROWSER_STOP	34047
最新の状態に更新	IDC_BROWSER_REFRESH	34046
最大	IDC_BROWSER_FONT_LARGEST	34041
大	IDC_BROWSER_FONT_LARGE	34040
中	IDC_BROWSER_FONT_MEDIUM	34042
小	IDC_BROWSER_FONT_SMALL	34043
最小	IDC_BROWSER_FONT_SMALLEST	34044

4.6.9 『ツール」メニュー

「ツール」メニューには以下のコマンドが含まれています:

メニューアイテム	コマンド名	ID
スペル	IDC_SPELL_CHECK	34154
スペル オプション	IDC_SPELL_OPTIONS	34155
スクリプトエディター	ID_SCRIPTFORMEDITOR_EDIT_PROJE CT	39666
無し	ID_SCRIPTFORMEDITOR_EXECUTE_M ACRO_MENU_UPPDATE	39600
	IDC_TOOLS_ENTRY	34292
グローバルリソース	IDC_GLOBALRESOURCES	37401
	IDC_GLOBALRESOURCES_SUBMENU ENTRY1	37408
カスタマイズ	IDC_APP_TOOLS_CUSTOMIZE	32959
オプション	IDC_SETTINGS	34133
	ID_SCRIPTING_MACROITEMS	34249

4.6.10 ヘルプ」メニュー

「ウィンドウ」メニューには以下のコマンドが含まれます:

メニューアイテム	コマンド名	ID
重ねて表示	ID_WINDOW_CASCADE	57650
上下に並べて表示	ID_WINDOW_TILE_HORZ	57651
左右に並べて表示	ID_WINDOW_TILE_VERT	57652
プロジェクトウィンドウ	IDC_PROJECT_WINDOW	34128
情報ウィンドウ	IDC_INFO_WINDOW	34085
入力ヘルパー	IDC_ENTRY_HELPERS	34062
出力ウィンドウ	IDC_OUTPUT_DIALOGBARS	34004
プロジェクトと入力ヘルパー	IDC_PROJECT_ENTRYHELPERS	34006
全てオン オフ	IDC_ALL_BARS	34031

4.6.11 ヘルプ」メニュー

「ヘルプ」メニューには以下のコマンドが含まれます:

メニューアイテム	コマンド名	ID
目次	IDC_HELP_CONTENTS	34076
インデックス	IDC_HELP_INDEX	34077
検索	IDC_HELP_SEARCH	34079
キーボードマップ	IDC_HELP_KEYMAPDLG	34078
ライセンスの登録	IDC_ACTIVATION	34005
注文フ ォー ム	IDC_OPEN_ORDER_PAGE	34094
登録	IDC_REGISTRATION	34131
最新情報のチェック	IDC_CHECK_FOR_UPDATES	34275
サポートセンター	IDC_OPEN_SUPPORT_PAGE	34096
Web の FAQ	IDC_SHOW_FAQ	34153
コンポーネントや無料 ツールのダウンロード	IDC_OPEN_COMPONENTS_PAGE	34093
インターネット上の Authentic	IDC_OPEN_XML_SPY_HOME	34098
Authentic トレーニング	ID_HELP_XMLSPYTRAINING	34210
Authentic について	ID_APP_ABOUT	57664

4.7 オブジェクトレファレンス

オブジェクト:

Authentic DesktopCommand Authentic DesktopCommands AuthenticDesktopControl AuthenticDesktopControlDocument AuthenticDesktopControlPlaceHolder

標準の Authentic Desktop 機能へのアクセスを与えるには、Authentic Desktop 自動化インター フェイスのオブジェクトにもアクセスすることができます。詳細に関しては、次を参照してください: AuthenticDesktopControlApplication, AuthenticDesktopControlDocument.Document AuthenticDesktopControlPlaceHolder.Project。

4.7.1 Authentic DesktopCommand

Jultra D Label Name Sseparator Toolfp StatusText Accelerator SubCommands

詳細:

コマンドオブジェクトは次のオプションであることができます:実行可能なコマンド,コマンドコンテナー (例えば、メニュー、サプメニュー、または、ツールバー)、または、メニューセパレーター。現在の Command オブジェクト内に保管 される情報の種類を決定するには、ID、IsSeparator、と SubCommands プロパティをクエリしてください。

コマンドオブジェクト	使用可能な場合
実行可能なコマンド	 ID はゼロよりも大きい場合 IsSeparator が false の場合 SubCommands が空白の場合
コマンド コンテナー	 ID はゼロの場合 IsSeparator が true の場合 SubCommands が Command オブジェクト を含む場合
セパレーター	 ID はゼロの場合 IsSeparator が true の場合

Accelerator

プロパティ: Acceleratorをstringとして

詳細:

コマンドのために定義 された アクセラレータキーを返します。コマンドにアクセラレーターキーが割り当て られていない場合、このプロパティは、空の文字列を返します。 アクセラレータキーの文字列の表示 は次のフォーマットを取ります:

[ALT+] [CTRL+] [SHIFT+] key

Windows プラットフォーム SDK 関数 GetKeyNameText を使用して key が変換された箇所。

ID

プロバティ: D を long として

詳細:

このプロパティは、コマンドの一意の識別子を取得します。コマンドのIDは(<u>Exec</u>を使用して)コマンドを実行するために、または、(<u>QueryStatus</u>を使用して)ステータスをクエリするために必須です。コマンドが他のコマンド(例えば、トップレベルのメニュー)のためのコンテナー、または、セパレーターの場合、IDは0です。

IsSeparator

プロパティ: IsSeparatorを boolean として

詳細:

コマンドオブジェクトがメニューセパレーターの場合、の場合、プロパティは、true を返します。それ以外の場合は、false を返します。次も参照してください:コマンド.

Label

プロパティ: Labelを string として

詳細:

このプロパティは、Authentic Desktop のグラフィカルなユーザーインターフェイス内に表示されるとおりのコマンドのテキストを取得します。コマンドがセパレーターの場合、「ラベル」は、空の文字列です。 このプロパティは、関連した GUI テキストを持たない、一部のツールバーコマンドのために空の文字列を返します。

Name

プロパティ: Name を string として

詳細:

このプロパティは一意のコマンド名を取得します。使用できる箇所で、この値はコマンドのアイコンファ イルを取得するために使用できます。使用することのできるアイコンファイルは、使用中のAuthentic Desktop インストールのフォルダー **<ApplicationFolder>\Examples\ActiveX\Images**内にありま す。

StatusText

プロパティ: Labelを string として

詳細:

コマンドが選択されると、ステータステキストは、Authentic Desktopのステータスバー内で表示される テキストです。他のコマンドのセパレーターまたは、コンテナオではないコマンドオブジェクトに対しての み適用することができます。それ以外の場合、プロパティは空の文字列です。

SubCommands

プロバティ: SubCommands をCommands として

詳細:

SubCommands プロパティは、現在のコマンドのサブコマンドである Command オブジェクトのコレクションを 取得します。 プロパティは、他のコマンド(メニュー、サブメニュー、または、ツールバー)のためのコンテナ ーであるコマンドにのみ適用されます。 このようなコンテナーコマンドは、0 に設定された ID を持ち、お よび、false に設定された ISSeparator プロパティを持ちます。

ToolTip

プロパティ: TooTp を string として

詳細:
このプロパティは、各コマンドのためのヒントを表示するテキストを取得します。コマンドがツールのヒントのテキストを持たない場合、プロパティは、空の文字列を返します。

4.7.2 Authentic DesktopCommands

プロパティ: Count Tem

詳細:

AuthenticDesktopControl のコマンドレベルとID へのアクセスを取得するための Command オブジェクトのコレクションです。これらのコマンドは、Exec メソッドを使用して実行することができます。また、ステータスをQueryStatusを使用してクエリすることができます。

Count

プロパティ: Count を long として

詳細:

このコレクションのレベル上の Comm and オブジェクトの数。

Item

プロパティ: Item (n を long として) を Comm and として

詳細:

このコレクション内のインデックス n を持つコマンドを取得します。インデックスは 1 ベースです。

4.7.3 AuthenticDesktopControl

プロパティ:

htegrationLevel Appearance Application BorderStyle CommandsList CommandsStructure 俳推奨) EnableUserPrompts MainMenu Toolbars

メソッド:

Open Exec QueryStatus

イベント:

OnUpdateCmdUI OnOpenedOrFocused OnCbseEditingWindow OnFileChangedAlert OnContextChanged OnDocumentOpened OnValidationWindowUpdated

このオブジェクトは完全な ActiveX コントロールで、Authentic Desktop ライブラリがアプリケーションレベル モードで使用 されている場合、表示 されます。

プロパティ

以下のプロパティが定義されています:

htegrationLevel EnableUserPrompts Appearance BorderStyle

コマンドに関連したプロパティ

CommandsList MainMenu Toobars CommandsStructure 俳推奨)

AuthenticDesktopAPI へのアクセス: Application

Appearance

プロパティ: Appearance を short として

ディスパッチ ID: -520

詳細:

0 に対 して等 しくない値 は、コントロールの周 リでクライアントエッジを表示 します。 デフォルトの値 は 0 です。

Application

プロパティ: Application を Application として

ディスパッチ ID: 1

詳細:

Application プロパティは、完全な Authentic Desktop 自動サーバー APIの Application オブジェクトへのアクセスを与えます。プロパティは読み取り専用です

BorderStyle

プロパティ: BorderStyle を short として

ディスパッチ ID: -504

詳細:

1の値は、薄いボーダーを持つコントロールを表示します。 デフォルトの値は 0 です。

CommandsList

プロバティ: Comm and List を Comm and として (読み取り専用)

ディスパッチ ID: 1004

詳細:

このプロパティは、AuthenticDesktopControl を持つ定義された全てのコマンドのフラットリストを返します。メニュー構造に従い、コマンドを整理するには、MainMenu を使用します。ツールバーコマンドを取得するには、Toolbars を使用します。

```
public void GetAllAuthenticCommands()
{
    // Get all commands from the Authentic ActiveX control assigned to the
current form
    AuthenticControlLib.XMLSpyCommands commands =
    this.axAuthenticDesktopControl1.CommandList;
    // Loop through all commands
    for (int i = 0; i < commands.Count; i++)
    {
        // Get each command by index and output it to the console
        AuthenticControlLib.XMLSpyCommand cmd =
        axAuthenticDesktopControl1.CommandList[i];
        Console.WriteLine("{0} {1} {2}", cmd.ID, cmd.Name,
        cmd.Label.Replace("&", ""));
     }
}</pre>
```

C# サンプル

EnableUserPrompts

プロバティ: EnableUserPrompts を boolean として

ディスパッチ ID: 1006

詳細:

このプロパティを false に設定すると コントロール内 のユーザープロンプトを無効化 します。デフォルトの値は true です。

IntegrationLevel

JUNF1: htegrationLevelをCActiveX htegrationLevelとして

ディスパッチ ID: 1000

詳細:

The IntegrationLevel プロパティ determines コントロールのオペレーションモード。次も参照してください: アプリケーションレベルの統合 とドキュメントレベルの統合。

メモ: AuthenticDesktopControlオブジェクトの作成後すくに、このプロパティを設定することは重要です。

MainMenu

プロパティ: MainMenu をCommand として(読み取り専用)

ディスパッチ ID: 1003

詳細:

このプロパティは、 Authentic Desktop Control メインメニュー内 で使用 することのできる構造の情報 と コマンドを command オブジェクトとして提供 します。オブジェクトは、 Authentic Desktop の使用 するこ とのできる全 てのサブメニューを含んでいます (例「ファイル」、「編集」、「表示」など)。サブメニュー オブジェクトにアクセスするには、MainMenu プロパティの SubCommands プロパティを使用 します。各サ ブメニューは Command オブジェクトでもあります。各サブメニューのために、対応 する子 コマンドセパレー ターを取得 するために、 SubCommands プロパティを使用 して更 に反復 することができます (このテクニッ クが使用 される可能性があります。例えば、アプリケーションメニューをプログラム的に作成 するなど)。 メニューコマンドの一部は、 SubCommands プロパティを持つ他のメニューコマンドのためのコンテナー (親」)としての役割を果たします。全てのメニューコマンドの構造をプログラム的に取得 するには、再 帰的 な関数を作成 する必要があります。

```
public void GetAuthenticMenus()
{
    // Get the main menu from the Authentic ActiveX control assigned to the
current form
    AuthenticControlLib.XMLSpyCommand mainMenu =
this.axAuthenticDesktopControl1.MainMenu;
    // Loop through entries of the main menu (e.g. File, Edit, etc.)
    for (int i = 0; i < mainMenu.SubCommands.Count; i++)</pre>
```

```
{
    AuthenticControlLib.XMLSpyCommand menu = mainMenu.SubCommands[i];
    Console.WriteLine("{0} menu has {1} children items (including
separators)", menu.Label.Replace("&", ""), menu.SubCommands.Count);
    }
}
```

C# サンプル

Toolbars

プロパティ: Toobars を Commands として (読み取り専用)

ディスパッチ ID: 1005

詳細:

このプロパティは、AuthenticDesktopControl ツールバーの構造の情報を command オブジェクトとして 与えます。Command オブジェクトには、Authentic Desktopの使用することのできる全てのツールバー が含まれています。ツールバーにアクセスするには、Toolbars プロパティの SubCommands プロパティ を使用します。各ツールバーは、Command オブジェクトでもあります。各ツールバーのために、コマン ドを使用するために SubCommands プロパティを更に反復します (例えば、アプリケーションのツールバ ープログラム的に作成するためなどにこのテクニックは使用される場合があります)。

```
public void GetAuthenticToolbars()
{
   // Get the application toolbars from the Authentic ActiveX control
assigned to the current form
   AuthenticControlLib.XMLSpyCommands toolbars =
this.axAuthenticDesktopControl1.Toolbars;
   // Iterate through all toolbars
   for (int i = 0; i < toolbars.Count; i++)</pre>
     AuthenticControlLib.XMLSpyCommand toolbar = toolbars[i];
     Console.WriteLine();
     Console.WriteLine("The toolbar \"{0}\" has the following commands:",
toolbar.Label);
      // Iterate through all commands of this toolbar
      for (int j = 0; j < toolbar.SubCommands.Count; j++)</pre>
      {
         AuthenticControlLib.XMLSpyCommand cmd = toolbar.SubCommands[j];
         // Output only command objects that are not separators
         if ( ! cmd.IsSeparator)
            Console.WriteLine("{0}, {1}, {2}", cmd.ID, cmd.Name,
cmd.Label.Replace("&", ""));
         }
      }
    }
}
```



メソッド

以下のメソッドが定義されます:

0 pen Exec QueryStatus

Exec

Method: Exec (nCm dD を long として) を boolean として

ディスパッチ ID: 6

詳細:

このメソッドは、ID nCmdID を持つ Authentic Desktop コマンドを呼び出し、コマンドが実行可能な場合、メソッドは、true を返します。使用可能な全てのコマンドのリストを取得するには、 CommandsList を使用してください。コマンドの状態を取得するには、QueryStatus を使用してください。

Open

Method: Open (strFilePath を string として) を boolean として

ディスパッチ ID: 5

詳細:

メソッドの結果は、引数 strFilePath内でパスされる拡張子により異なります。ファイル拡張子が sps の場合、新規ドキュメントが開かれます。ファイル拡張子が svp の場合、対応する プロジェクトが開 かれます。異なるファイル拡張子がメソッドにパスされるとコントロールはファイルを新規のコンポーネ ントとしてアクティブなドキュメントにロードします。

ドキュメントレベルの統合モード内でコントロール使用する際、このメソッドを使用して、ドキュメントまたはプロジェクトをロードしないでください。代わりに、AuthenticDesktopControDocumentOpenと AuthenticDesktopControDeceHoberOpenProjectを使用してください。

QueryStatus

Method: QueryStatus (nCm dD を long として) を long として

ディスパッチ ID: 7

詳細:

QueryStatus は、nCm dD により指定 されたコマンドのチェックされた チェックが解除 されたステータスを 返します。このステータスはビットマスクとして返されます。

	ビット	値	名前	意味
行可能	0	1 2	サポートされている 有効化されている	コマンドがサポートされている場合設定します。 コマンドが有効化されている場合設定します (実
	2	4	チェックされている	コマンドがチェックされている場合設定します。

これは、QueryStatus が 0 コマンドを返すと ID は、有効な Authentic Desktop コマンドとして認識 されないことを意味します。QueryStatus が 1 または 5 の値を返すとコマンドは無効化されます。

Events

AuthenticDesktopControl ActiveX コントロールは次の接続ポイントイベントを提供します:

OnUpdateCmdUI OnOpenedOrFocused OnCbseEditingWindow OnFileChangedAlert OnContextChanged

OnDocum entOpened OnValidationWindowUpdated

OnCloseEditingWindow

イベント: OnC bseEditingW indow (i_strFilePath を String として) を boolean として

ディスパッチ ID:1002

詳細:

Authentic Desktop が既に開かれているドキュメントを閉じる場合、このイベントはトリガーされます。 このイベントへの応答として、クライアントは *i_strFilePath* に関連するエディターウィンドウを閉じます。 このイベントが *true* を返すと、クライアントがドキュメントを閉じたことを指します。特定の処理が必要 とされておらず、AuthenticDesktopControl がエディターを閉じようと試み、関連したドキュメントコント ロールを破棄する場合、クライアントは、*false* を返すことができます。

OnContextChanged

イベント: OnContextChanged (i_strContextName をString として、ibActive をbool として) を bool として

ディスパッチ ID: 1004

詳細:

このイベントは、is not used in Authentic Desktop

OnDocumentOpened

イベント: OnDocum entOpened (ob Docum entを Document として)

ディスパッチ ID: 1

詳細:

このイベントはドキュメントが開かれる都度トリガーされます。引数 obpocumentは、Authentic Desktop 自動化インターフェイス、からの Documentオブジェクトであり、ドキュメントの詳細をクエリす るために使用することができ、または、追加オペレーションを行います。ドキュメントレベルで統合され る場合、イベント Authentic Desktop Contro Document On Document Opened を代わりに使用すること が奨励されます。

OnFileChangedAlert

イベント: OnFileChangedAlert(i strFilePathをStringとして)をboolとして

ディスパッチ ID: 1001

詳細:

AuthenticDesktopControl を使用してファイルがロードされると is このイベントはトリガーされます。他のアプリケーションによりハードディスク上で変更されます。イベントを処理すると クライアントは、 true を返します。または、Authentic Desktop が通常の方法で処理すると false を返します。例えば、ユーザーに再ロードを促すなど。

OnLicenseProblem

イベント: OnLicenseProblem (i strLicenseProblemTextをStringとして)

ディスパッチ ID: 1005

詳細:

AuthenticDesktopControl がこのコントロールのための有効なライセンスを使用できないことを検知す ると、このイベントはトリガーされます。制限されているユーザーファイセンスの場合、コントロールが初 期化されると発生する可能性があります。インテグレーターはこのイベントをしようして、このコントロー ルの機能へのアクセスを無効化することができます。このイベントから戻ると、コントロールは機能への アクセスをブロックします (例、コントロール内の空のウィンドウを表示し、リクエストに対してエラーを返 すなど)。

OnOpenedOrFocused

イベント: OnOpenedOrFocused (i_strFilePath をString として、 i bOpenWithThisControlをboolとして)

ディスパッチ ID: 1000

詳細:

アプリケーションレベルで統合する場合、このイベントはクライアントにドキュメントが開かれたことまたは Authentic Desktop によリアクティブに設定されたことを通知します。

ドキュメントレベルで統合する場合、このイベントは、クライアントにドキュメントウィンドウをファイル i_strFilePath で開くように命令します。ファイルが既に開かれている場合、対応するドキュメン トウィンドウをアクティブなウィンドウにしてください。

i_bOpenWithThisControl が true の場合、内部アクセスが必要とされるため、ドキュメントは、 AuthenticDesktopControl により開かれる必要があります。それ以外の場合、ファイルは、異なるエディ ターで開くことができます。

OnToolWindowUpdated

イベント: On Too W indow Updated (pTool Wnd を long として)

ディスパッチ ID: 1006

詳細:

ツールウィンドウが更新されるとこのイベントはトリガーされます。

On Update CmdUI

イベント:OnUpdateCmdUI()

ディスパッチ ID: 1003

詳細:

頻繁に呼び出され、AuthenticDesktopControlQueryStatus を使用してAuthentic Desktop コマンドの状況をチェックする良い機会をインテグレーターに与えます。このコールバック内で long オペレーションを実行しないでください。

OnValidationWindowUpdated

イベント: OnValidationW indowUpdated ()

ディスパッチ ID: 3

詳細:

新しい情報と共に検証出力ウィンドウが更新されるとこのイベントはトリガーされます。

4.7.4 AuthenticDesktopControlDocument

プロパティ: Appearance BorderStyle Document ISM odified Path ReadOnly

メソッド:

Exec New Open QueryStatus Rebad Save SaveAs

イベント:

OnDocum entOpened OnDocum entC bsed OnModifiedFlagChanged OnContextChanged OnFileChangedAlert OnActivate

AuthenticDesktopControl がドキュメントレベルモード内で統合されている場合、各ドキュメントは、型 AuthenticDesktopControDocumentのオブジェクト内で表示されています。 AuthenticDesktopControDocumentには、1度に1つのドキュメントのみが含まれていますが、場合によっては異なるファイルを表示するために使用される場合があります。

このオブジェクトは完全な ActiveX コントロールです。

プロパティ

以下のプロパティが定義されています:

ReadOnly ISM odified Path Appearance BorderS tyle

AuthenticDesktopAPI へのアクセス: Document

Appearance

プロパティ: Appearance を short として

ディスパッチ ID: -520

詳細:

```
0 に対して等しくない値は、ドキュメントコントロールの周りにクライアントエッジを表示します。 デフォルトの値は0です。
```

BorderStyle

プロパティ: BorderStyle を short として

ディスパッチ ID: -504

詳細:

1の値は、薄いボーダー持つコントロールを表示します。デフォルトの値は0です。

Document

プロパティ: Document をドキュメントとして

ディスパッチ ID: 1

詳細:

Document プロパティは、Authentic Desktop 自動サーバー APIの Document オブジェクト へのアクセスを与えます。 このインターフェイスは、コントロール内 にロードされるドキュメントと使用 することのできる追加機能を与えます。 プロパティは読み取 「専用です

IsModified

プロパティ: EM odified を boolean として (読み取り専用)

ディスパッチ ID: 1006

詳細:

ドキュメントコンテンツが最後に開かれてから変更された場合、IsModified は true を返します。 再ロードし、オペレーションを保存します。それ以外の場合は、false を返します。

Path

プロパティ: Path を string として

ディスパッチ ID: 1005

詳細:

コントロール内にロードされたドキュメントのフルパス名に設定、または、取得します。

ReadOnly

プロパティ: ReadOnly を boolean として

ディスパッチ ID: 1007

詳細:

このプロパティを使用すると、ドキュメントの読み取り専用モードをオンとオフに切り替えることができます。ReadOnlyがtrueの場合、変更はできないことに注意してくざさい。

メソッド

以下のメソッドが定義されています: ドキュメントハンドリング: New Open Reload Save SaveAs

コマンドハンドリング:: Exec QueryStatus

Exec

メソッド: Exec (nCm dD を long として) を boolean として

ディスパッチ ID: 8

詳細:

このメソッドは、ID nCmdID を持つ Authentic Desktop コマンドを呼び出し、コマンドが実行可能な場合、メソッドは、true を返します。現在 アクティブなドキュメントがアプリケーションで使用 できる場合のみ、このメソッドを呼び出すことができます。

メニュー構造に従いコマンドを整理するには、AuthenticDesktopControlの<u>MainMenu</u>プロパティを使用します。ツールバーコマンドを取得するには、AuthenticDesktopControlの<u>Toolbars</u>プロパティを使用します。

New

メソッド: New () を boolean として

ディスパッチ ID: 1000

詳細: このメソッドはコントロール内で新規ドキュメントを初期化します。

Open

メソッド: Open (strFileName を string として) を boolean として

ディスパッチ ID: 1001

詳細:

Open は、ファイル strFileName を新規のドキュメントとして、コントロールにロードします。

QueryStatus

メソッド: QueryStatus (nCm dD を long として) を long として ディスパッチ ID: 9

詳細:

QueryStatus は、nCm dD により指定されたコマンドのチェックされた チェックが解除されたステータスを返します。このステータスはビットマスクとして返されます。

	ビット	値	名前	意味	
	0 1	1 2	サポートされて 有効化されてい	 いる , 1る	 コマンドがサポートされている場合設定します。 コマンドが有効化されている場合設定します (実
行可能					
	2	4	チェックされてい	13	コマンドがチェックされている場合設定します。

これは、QueryStatus が 0 コマンドを返すと ID は、有効な Authentic Desktop コマンドとして認識 されないことを意味します。QueryStatus が 1 または 5 の値を返すとコマンドは無効化されます。ア プリケーション内で現在 アクティブなドキュメントが存在する場合、クライアントは、ドキュメントコントロ ールの QueryStatus メソッドを呼び出します。

Reload

メソッド: Rebad () を boolean として

ディスパッチ ID: 1002

詳細:

Rebad ファイルシステムからドキュメントコンテンツを更新する。

Save

メソッド: Save () を boolean として

ディスパッチ ID: 1003

詳細:

Save は、現在のドキュメントをロケーション Path に保存する。

SaveAs

メソッド: SaveAs (strFileName を string として) を boolean として

ディスパッチ ID: 1004

詳細:

SaveAs は Path を strFileName に設定し、ドキュメントをこの場所に保存します。

Events

AuthenticDesktopControlDocument ActiveX コントロールは次の接続ポイントイベントを提供します:

OnDocumentOpened OnDocumentCbsed OnModifiedFlagChanged OnContextChanged OnFileChangedAlert OnActivate OnSetEditorTitle

OnActivate

イベント: OnActivate ()

ディスパッチ ID: 1005

詳細:

ドキュメントコントロールが有効化 されると、フォーカスのある、ユーザー入力 として使用 することのできるこのイベントがトリガーされます。

OnContextChanged

 $4 \times 2 h$: On ContextChanged (i_strContextName & String $b \in b$ and $b \in b \in b$ and $b \in b \in b \in b$

ディスパッチ ID: 1004

詳細:なし

OnDocumentClosed

イベント: OnDocum entC bsed (obpocum entを Document として)

ディスパッチ ID: 1001

詳細:

このイベントはドキュメントがこのコントロールにロードされると閉じられます。引数 obpocum ent は、 Authentic Desktop 自動化インターフェイスからの Docum ent オブジェクトです、で慎重に使用してく ださい。

OnDocumentOpened

イベント: OnDocum entOpened (obDocum entを Document として)

ディスパッチ ID: 1000

詳細:

このイベントは、このコントロール内でドキュメントが開かれるとトリガーされます。引数 objocum ent は、Authentic Desktop 自動化インターフェイスからの Docum ent オブジェクトです。ドキュメントの詳 細をクエリするために使用することができ、または、追加オペレーションを行います。

OnDocumentSaveAs

イベント: OnContextDocum entSaveAs (i strFileName を String として)

ディスパッチ ID: 1007

詳細:

このドキュメントが新しい名前で内部で保存される場合このイベントがトリガーされます。

OnFileChangedAlert

イベント: OnFileChangedAlert ()をboolとして

ディスパッチ ID: 1003

詳細:

このドキュメントにロードされると このイベントはトリガーされます。コントロールはハードディスク上で他のアプリケーションに変更されます。イベントを処理すると クライアントは、true を返します。または、Authentic Desktop が通常の方法で処理すると false を返します。例えば、ユーザーに再ロードを促すなど。

OnModifiedFlagChanged

イベント: OnModifiedFlagChanged (ibIsModified を boolean として)

ディスパッチ ID:1002

詳細:

このイベントはドキュメントの状態を、変更された、および、変更されていない状態に切り替えるとトリガーされます。ドキュメントコンテンツが元のコンテンツと異なる場合、パラメーター *i_blsModifed* is *true* それ以外の場合は、false を返します。

OnSetEditorTitle

イベント: OnSetEditorTitle ()

ディスパッチ ID: 1006

詳細:

このイベントは、含まれているドキュメントが内部で名前を与えられると挙げられます。

4.7.5 AuthenticDesktopControlPlaceHolder

すべてのプロジェクトプレースホルダーウィンドウに使用することのできるプロパティ: PlacehoblerWindow D

プロジェクトプレースホルダーウィンドウのためのプロパティ: Project

プロジェクトプレースホルダーウィンドウのためのメソッド:

OpenProject CbseProject

AuthenticDesktopControlPaceHolder コントロールは、概要、ライブラリまたは プロジェクトウィンドウ などの追加の Authentic Desktop ウィンドウを表示するために使用されます。他の ActiveX コントロ ールと同様に使用することができ、クライアントアプリケーション内に配置することができます。

プロパティ

以下のプロパティが定義されています:

PlaceholderWindowD

AuthenticDesktopAPI へのアクセス: Project

Label

プロパティ: Labelを String (読み取り) 使用)として

ディスパッチ ID: 1001

詳細:

このプロパティはプレースホルダのタイトルへのアクセスを与えます。 プロパティは読み取り専用です

PlaceholderWindowID

プロパティ: PlaceholderWindowID を <u>AuthenticDesktopControlPlaceholderWindow</u> として

ディスパッチ ID: 1

詳細:

このプロパティを使用すると コントロールのクライアントエリア内 で表示 される Authentic Desktop ウィ ンドウが通知されます。PlacehoblerWindowD は、AuthenticDesktopControlPlaceholderWindow 列挙の有効な値に設定することができます。コントロールはすくに状態を変更し、すくに新規の Authentic Desktop ウィンドウを表示します。

Project

プロパティ: Project Project として (読み取)専用)

ディスパッチ ID: 2

詳細:

Project プロパティは、Authentic Desktop 自動サーバー APIの Project オブジェクト へのアクセスを 与えます。このインターフェイスは、コントロールへロードされるプロジェクトに使用 される追加機能を 与えます。プレースホルダウィンドウが Authentic DesktopXProjectWindow (=3)の値を持つ PlacehoblerWindow D の場合のみ、プロパティは有効なプロジェクトインターフェイスを返します。プロ パティは読み取り専用です

メソッド

以下のメソッドが定義されています:

OpenProject CbseProject

OpenProject

メソッド: OpenProject (strFileName を string として) を boolean として

ディスパッチ ID: 3

詳細:

OpenProject は strFileName を新規のプロジェクトとしてコントロールにロードします。 プレースホルダ ウィンドウが XM LSpyXProjectWindow (=3)と異なる <u>PlacehoblerWindow</u> を持つ場合、メソッドは失敗します。

CloseProject

メソッド: C bseProject()

ディスパッチ ID: 4

詳細:

C bseProject はプロジェクトにロードされたコントロールを閉じます。プレースホルダウィンドウが XM LSpyXProjectWindow (=3)と異なる PlaceholderWindow D を持つ場合、メソッドは失敗します。

イベント

AuthenticDesktopControlPlaceholder ActiveX コントロールは次の接続ポイントイベントを提供します

0 nM odifiedF lagC hanged

OnModifiedFlagChanged

イベント: OnModifiedFlagChanged (iblsModified を boolean として)

ディスパッチ ID: 1

詳細:

このイベントは Authentic DesktopXProjectWindow (=3)の PlacehoblerWindow D を持 つプレースホル ダコントロールのためのみにトリガーされます。プロジェクトのコンテンツが変更された、または、変更さ れていない状態の間で変更があった場合、イベントは実行されます。プロジェクトコンテンツが元のコンテンツと異なる場合、パラメーター i_blsModifed は true を返し、それ以外の場合は、false を返します。

OnSetLabel

イベント: OnSetLabel(istrNewLabelをstringとして)

ディスパッチ ID: 1000

詳細:

プレースホルダウィンドウのタイトルが変更されると挙げられます。

4.7.6 列挙

以下の列挙が定義されています:

CActiveX htegrationLevel AuthenticDesktopC ontrolP aceho blerW indow

ICActiveXIntegrationLevel

AuthenticDesktopControlの IntegrationLevel プロパティのための可能な値。

```
ICActiveXIntegrationOnApplicationLevel = 0
ICActiveXIntegrationOnDocumentLevel = 1
```

$\label{eq:linear} Authentic Deskt op Control Placeholder Window$

この列挙にはサポートされる追加 Authentic Desktop ウィンドウのリストが含まれています。

AuthenticDesktopControlNoToolWnd				
AuthenticDesktopControlEntryHelperTopToolWnd	=	0		
AuthenticDesktopControlEntryHelperMiddleToolWnd	=	1		
AuthenticDesktopControlEntryHelperBottomToolWnd	=	2		
AuthenticDesktopControlValidatorOutputToolWnd				
AuthenticDesktopControlProjectWindowToolWnd	=	4		
AuthenticDesktopControlInfoToolWnd				

Chapter 4

付録

付録

以下の付録には、Authentic Desktopに関する技術的な情報や、ライセンスに関する重要な情報が収められています。各付録には以下のようにサブセクションが収められています:

<u>技術情報</u>

- OS ならびにメモリの必要条件
- Altova XML パーサー
- Altova XSLT とXQuery エンジン
- Unicode のサポート
- インターネットへの接続

ライセンス情報

- 電子的なソフトウェアの配布
- 知的財産権 と 著作権
- エンドユーザーライセンス使用許諾契約書

技術データ 1

このセクションは、ソフトウェアの技術面に関する役に立つ背景情報を含んでいます。以下のように 整理されています:

- <u>OS とメモリ要件</u> ٠
- •
- Altova XML バリデーター Altova XSLT とXQuery エンジン •
- <u>Unicode</u> <u>のサポート</u> •
- インターネットの使用 ٠

1.1 OS とメモリ要件

オペレーティングシステム

Altova ソフトウェア アプリケーションは、以下のプラットフォームでご使用いただけます:

- プラットフォーム更新済みの Windows 7 SP1、Windows 8、Windows 10
- プラットフォーム更新済みの Windows Server 2008 R2 SP1 または以降

メモリ

ソフトウェアが C++ で書 かれているため、Java Runtime Environment をダウンロードする必要 はなく Java ベースのアプリケーションに比べ、通常少ないメモリを必要 とします。しかしながら、各ドキュメント は完全に解析するため、また、ビューと編集の速度を向上するためにメモリにダウンロードされます。 メモリの要件は、ドキュメントのサイズを増やします。

メモリ要件は、制限のない「元に戻す」履歴により影響を受けます。大きなドキュメントの大きなセクションの切り取り、貼り付け操作を繰り返し行うと使用できるメモリがすくに消費されます。

1.2 Altova XML バリデーター

XML ドキュメントを開くと、アプリケーションは、内蔵の XML バリデーターを使用して、指定されている場合、スキーマに対して整形式をチェックし、ツリーとインフォセットを作成します。 XML バリデーターは、ドキュメントを編集する際 にインテリジェントな編集 ヘルプを提供し、発生する検証 エラーを表示するために使用 されます。

内蔵の XML バリデーターは、W3C の XML スキーマ 1.0 と1.1 仕様の最終勧告を実装しています。 . New developments recommended by the W3C XML スキーマ作業 グループにより 勧告 される新しい 項目は、XML バリデーターに継続的に組み込まれるため、Altova 製品は最高水準の開発環境を 届けることができます。

1.3 Altova XSLT と XQuery エンジン

Altova 製品 は、Altova XSLT 1.0、2.0、および 3.0 エンジン とAltova XQuery 1.0 と3.1 エンジンを 使用しています。各エンジンのためのドキュメントと実装に固有の振る舞いに関しては、製品で使 用されるエンジンの各ドキュメントの付属書 (エンジン情報)で確認することができます。

メモ: Altova MapForce は、XSLT 1.0、2.0 および XQuery 1.0 エンジンを使用したコードを生成します。

1.4 Unicode のサポート

Altova XML 製品は、Unicode を完全にサポートします。XML ドキュメントを編集するには、ドキュメント内 で使用 されている Unicode 文字 をサポートするフォントが必要です。

フォントの多くは、Unicode 範囲全体の特定のサブセットを含む場合があり、このため、通常は対応 する表記システムをターゲットとします。テキストの一部が、文字化けして表示された場合、理由とし ては、選択されたフォントが必要とする字形を含まない場合があげられます。ですから、特に、異なる 言語、または、異なる言語システムのXMLドキュメントを編集する場合、範囲全体をカバーする フォントを使用することが役にたちます。典型的な Unicode フォントは、Windows PC の Arial Unicode MS で確認することができます。

アプリケーションフォルダーの /Examples フォルダー内 で、異なる言語システムで表記 された次の文章を含む UnicodeUTF-8 html というXHTML ファイルを確認 してください:

- When the world wants to talk, it speaks Unicode
- Wenn die Welt miteinander spricht, spricht sie Unicode
- 世界的に話すなら、Unicodeです。)

XHTML ファイルを開くと、Unicode の可能性を確認することができ、使用中の PC の使用することのできるフォントによりサポートされている表記システムが表示されます。

1.5 インターネットの使用

Altova アプリケーションは、次の状況でインターネット 接続を開始します:

- 登録ダイアログ(「ヘルプ|ソフトウェアのライセンス認証」)内の 評価キーコードをリクエスト」をクリックした場合、登録ダイアログボックス内の3つのフィールドが通常の http (ポート80)接続を使用し、サーバーに転送され、無料の評価キーが顧客に通常の SMTP 電子メールを使用して送り返されます。
- Altova 製品の一部では、インターネットからファイルを開くことができます(「ファイル|開く) URL に切り替える」)。この場合、ドキュメントは、次のプロトコルメソッドと接続の1つを使用して取得されます: HTTP (通常、ポート80)、FTP (通常、ポート20/21)、HTTPS (通常、ポート443)。HTTP サーバーをポート8080で作動することもできます(URL ダイアログ内で、サーバー名とコロンの後にポートを指定します)。
- XML スキーマ、または、DTD を参照する XML ドキュメント、と URL によい指定 されている ドキュメントを開くと 参照 されている スキーマドキュメントは、HTTP 接続(ポート 80) また は URL によい指定 されている他のプロトコール(上のポイント2参照)によい抽出 されます。 XML ファイルが検証 されている場合、スキーマドキュメントも抽出 されます(オプションダイアロ グのファイルタブ内の(「ツール | オプション」))。アプリケーションに命令 している場合、ドキュ メントが開かれるど検証 が自動的に行われる場合もあります。
- WSDL とSOAP を使用する Altova アプリケーションでは、Web サービスを使用する接続は、WSDL ドキュメントにより定義 されています。
- XMLSpy内で、「電子メールで送信」コマンドを使用する場合、(「ファイル | 電子メールで送信」)現在選択されている範囲、または、ファイルは、ユーザーのマシンにインストールされている MAPI コンプライアント電子メールプログラムにより送信されます。
- ソフトウェアの荒いセンス認証とLiveUpdateの一部として、Altova ソフトウェア使用許諾書 内で更に詳しい説明を確認することができます。

2 ライセンス情報

ライセンス情報

- ソフトウェアの配布 に関する情報
- このソフトウェアに関連する知的所有権に関する情報
- このソフトウェア製品の使用に関する<u>Authentict</u>のためのエンドユーザー使用許諾契約書

本製品を使用する前に、上記の情報をよくお読みください。ソフトウェアのインストール時に上記の すべての条件に同意したとみなされ、お客様は上記の条件に拘束されることを同意したとみなされ ます。

2.1 電子的なソフトウェアの配布

この製品は電子的なソフトウェアの配布により利用することが可能で、この配布方法により、以下のユニークなメリットがあります:

- 購入を決定する前に、無料でソフトウェアを試用することができます。
- ソフトウェアの購入を決定した際には、<u>Altova Web</u> <u>サイト</u>にて注文を行います。すくにライセンス 登録された製品の使用を開始することができます。
- オンラインにて注文を行うと常に最新のソフトウェアをご利用いただけます。
- 製品パッケージには包括的なヘルプシステムが画面上に表示されます。最新バージョンのユーザーマニュアルは www.altova.com 上にあり、(i) HTML フォーマットによる閲覧、ならびに(ii) PDF フォーマットのダウンロードと印刷に対応しております。

30日間の評価期間

この製品をダウンロードした後は、最大で30日の間無料で製品の評価を行うことができます。20日間を超えた頃から、製品がライセンス登録されていないことがソフトウェアにより表示されます。このメッセージはアプリケーションが起動されるたびに表示され、30日間を超えてプログラムを使用するには、キーコードとして送られる Altova ソフトウェア使用許諾契約書,を購入し、ソフトウェアアクティベーションダイアログにて入力し、製品をアンロックする必要があります。ライセンスの購入はAltova Webサイトのオンラインショップにてお求めいただけます。

組織内 でソフトウェアの評価を行う

評価版のソフトウェアを組織内のネットワークにて配布したい場合、またはインターネットに接続されていないコンピューターにてソフトウェアを使用する場合、どのような状態でも改変さていていないことを条件に、セットアッププログラムだけの配布を行うことが可能です。ソフトウェアインストーラーへアクセスした人は、例外なく30日間の評価ライセンスキーコードをリクエストして、試用期間が経過した後は、製品を使い続けるためにライセンスの購入を行う必要があります。

詳細については、 このセクション最後にある <u>Altova</u> ソフトウェアライセンス使用許諾契約書 を参照く ださい。

2.2 知的所有権

Altova ソフトウェアならびに Altova により認められたコピーは、Altova ならびにその供給者が知的所有権を有しています。ソフトウェアの構造や構成、コードは Altova ならびにその供給者にとって価値のある企業秘密かつ機密情報となります。このソフトウェアは米国の著作権法により保護されてお 以国際条項ならびにソフトウェアが使用されている国の法律が適用されます。Altova はこのソフト ウェアに関する特許、企業秘密、商標に関する所有権ならびにこれらに限定されないその他の知 的財産権を有しており、Altova の所有権には、このソフトウェアならびにそれに付随する印刷物に抱 合された画像、写真、アニメーション、映像、音声、音楽、テキスト、そして「アプレット」が含まれま す。著作権侵害に関する届け出は、Altova Web Site に記されている Altova 著作権エージェント へお送りください。

Altova ソフトウェアには国際条項により保護されているサードパーティソフトウェアも含まれており、 http://www.altova.com/legal_3rdparty.html にて詳細が記されている著作権法なども含め、しかしこれに限定されないかたちで知的所有権により保護されます。

その他全ての名前または商標は、対応する所有者が有するものです。

2.3 Altova Authentic エンドユーザー使用許諾契約書

The license agreement below applies to the following Altova products:

Authentic® Enterprise Edition

THIS IS A LEGAL DOCUMENT -- RETAIN FOR YOUR RECORDS

ALTOVA® END-USER LICENSE AGREEMENT

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

Important - Read Carefully. Notice to User:

This End User License Agreement ("Agreement") is a legal document between you and Altova GmbH ("Altova"). It is important that you read this document before using the Altova-provided software ("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 Software, you agree to be bound by the terms of this Agreement 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 Software, and you must destroy any downloaded copies of the Software in your possession or control. You may print a copy of this Agreement as part of the installation process at the time of acceptance. Alternatively, a copy of this Agreement may be found at http://www.altova.com/license agreements.html and a copy of the Privacy Policy may be found at http://www.altova.com/privacy.

1. SOFTWARE LICENSE

(a) License Grant for Authentic Desktop Enterprise Edition ("ADEE"). Upon your acceptance of this Agreement, Altova grants you a non-exclusive, non-transferable (except as provided below), limited license, without the right to grant sublicenses, to install and use a copy of ADEE software on one compatible personal computer or workstation up to the Permitted Number of computers. You <u>may not</u> distribute or reproduce the ADEE software other than as expressly permitted. Subject to the limitations set forth in Section 1(a)(i) you may install and use a copy of this Software on more than one of your compatible personal computers or workstations if you have purchased a Named-User license. Subject to the limitations set forth in Sections 1(a)(ii) and (iii), users may use the Software concurrently on a network. The Permitted Number of computers and or users and the type of license, e.g. Installed, Named User, and Concurrent-User, shall be determined and specified at such time as you elect to purchase the Software. Installed user licenses are intended to be fixed and not concurrent. In other words, you cannot uninstall for one user in order to reinstall that license to a different user and then uninstall and reinstall back to the original user. Users should be static. Notwithstanding the foregoing, permanent switchovers are acceptable (i.e., an employee has left the company, machine is retired). During the evaluation period, hereinafter defined, only a single user may install and use the ADEE software on one (1) personal computer or workstation. You may install one (1) copy of the ADEE software on a computer file server within your internal network solely for the purpose of downloading and installing this ADEE software onto other computers within your internal network up to the Permitted Number of computers in a commercial environment only. No other network use is permitted, including without limitation using the ADEE software either directly or through commands, data or instructions from or to a computer not part of your internal network, for Internet or Web-hosting services or by any user not licensed to use this copy of ADEE software through a valid license from Altova, except as set forth in Sections 1(a)(ii) and (iii) below Altova makes no warranties or representations about the performance of Software in a terminal server environment and the foregoing are expressly excluded from the limited warranty in Section 4 hereof and technical support is not available with respect to issues arising from use in such an environment.

(i) Named-Use. If you have licensed the "Named-User" version of the ADEE software, you may install the Software on up to five (5) compatible personal computers or workstations of which you are the primary user thereby allowing you to switch from one computer to the other as necessary provided that only one (1) instance of the ADEE software will be used by you as the Named-User at any given time. If you have purchased multiple Named-User licenses, each individual Named-User will receive a separate license key code.

(ii) Concurrent Use in Same Local Area Network (LAN). If you have licensed a "Concurrent-User" version of the ADEE software, you may install the ADEE software on any compatible computers in a commercial environment only, up to ten (10) times the Permitted Number of users, provided that only the Permitted Number of users actually use the Software at the same time and further provided that the computers on which the ADEE software is installed are on the same physical computer network. The Permitted Number of concurrent users shall be delineated at such time as you elect to purchase the ADEE licenses. Each separate physical network or office location requires its own set of separate Concurrent-User Licenses for those wishing to use the Concurrent-User versions of the Software in more than one location or on more than one network, all subject to the above Permitted Number limitations and based on the number of users using or needing access to the software. If a computer is not on the same physical network, then a locally installed user license or a license dedicated to concurrent use in a virtual environment is required.

(iii) Concurrent Use in Virtual Environment. If you have licensed a "Concurrent-User" versions of ADEE software, you may install a copy of the ADEE Software on a terminal server (Microsoft Terminal Server, Citrix Metaframe, etc.), application virtualization server (Microsoft App- V, Citrix XenApp, VMWare ThinApp, etc.) or virtual machine environment within your internal network for the sole and exclusive purpose of permitting individual users within your organization to access and use the Software through a terminal server, application virtualization session, or virtual machine environment from another computer provided that the total number of users that access or use the ADEE Software concurrently at any given point in time on such network, virtual machine or terminal server does not exceed the Permitted Number; and provided that the total number of users authorized to use the ADEE Software through the terminal server, application virtualization session, or virtual machine environment does not exceed ten (10) times the Permitted Number of users. Accordingly, the limitations set forth in Section 1(a)(ii) regarding the number of installations and the requirement that the usage be on the same physical network shall not apply to terminal server, application virtualization session, or virtual machine environments. In a virtual environment, you must deploy a means of preventing users from exceeding the Permitted Number of concurrent users. Altova makes no warranties or representations about the performance of Software in a terminal server, application virtualization session, or virtual machine environment and the foregoing are expressly excluded from the limited warranty in Section 5 hereof and technical support is not available with respect to issues arising from use in such environments.

(iv) Key Codes for ADEE. Prior to your purchase and as part of the registration for the thirty (30) - day evaluation period, as applicable, you will receive an evaluation key code. You will receive a purchase key code when you elect to purchase the ADEE software licenses from either Altova GmbH or an authorized reseller. The purchase key code will enable you to activate the Software beyond the initial evaluation period. You may not re-license, reproduce or distribute any key code except with the express written permission of Altova.

(b) License Grant for Authentic Browser-Plugin Enterprise Edition ("ABEE "). Upon your acceptance of this Agreement, Altova grants you a non-exclusive, non-transferable limited license, without the right to grant sublicenses, to install and use ABEE software on a per server basis for a twelve (12) month term, commencing on the date of your license purchase and expiring on the date that is twelve (12) months thereafter (the "ABEE License Term"). Altova also grants you a non-exclusive, non-transferable, limited worldwide license, without the right to grant sublicenses, to use software to develop web pages, web applications, or applications that include ABEE software, to reproduce the ABEE software on your website or server and to distribute the ABEE software from your website or server over a computer network, but only in its executable object code form, and only to end users for the limited purpose of enabling them to view, share, and/ or edit XML files during the ABEE software after the expiration of its license term, you must purchase a new ABEE software license. If you have purchased an ABEE software license then under the terms of the Agreement, support and maintenance (or SMP as further detailed below) for the software is included as part of the license purchase and you will be entitled to receive the benefits

set forth below during the ABEE License Term which is coterminous with the Support Period. <u>Unlike other</u> Altova software products, you <u>cannot</u> renew SMP for the ABEE software and at the expiration of the ABEE License Term and Support Period, you must purchase a new ABEE software license if you wish to continue to use, reproduce or distribute the ABEE software.

(i) Key Codes for ABEE. Prior to your purchase you may request a thirty (30) day evaluation key code, which will be sent to you. You will receive a purchase key code when you elect to purchase the ABEE software licenses from either Altova GmbH or an authorized reseller. The purchase key code will enable you to use the ABEE software during the ABEE License Term. You may not re-license, reproduce or distribute any key code except with the express written permission of Altova.

(ii) ABEE Software Specific Restrictions. In addition to the restrictions and obligations provided in other sections of this Agreement that are applicable to the ABEE software, your limited license to distribute the ABEE software set forth above, is further subject to all of the following restrictions; (i) ABEE software may only be licensed but may not be sold, and (ii) you must use, reproduce or distribute the ABEE software provided by Altova AS IS and may not impair, alter or remove the Altova End User License Agreement, (which will appear in the installation process and which an end user must accept in order to be able to install or operate the Software or any other files).

(c) Backup and Archival Copies. You may make one (1) backup and one (1) archival copy of the 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 Software. You may not transfer the rights to a backup or archival copy unless you transfer all rights in the Software as provided in this Agreement.

(d) Upgrades and Updates. If the Software that you have licensed is an upgrade or an update, then the latest update or upgrade that you download and install replaces all or part of the Software previously licensed. The update or upgrade and the associated license keys, as applicable, does not constitute the granting of a second license to the software in that you may not use the upgrade or updated copy in addition to the copy of the software that it is replacing and whose license has terminated.

(e) **Title.** Title to the Software is not transferred to you. Ownership of all copies of the Software and of copies made by you is vested in Altova, subject to the rights of use or distribution, as applicable, granted to you in this Agreement. All rights not specifically granted in this Agreement are reserved by Altova.

(f) **Reverse Engineering.** Except and to the limited extent as may be otherwise specifically provided by applicable law in the European Union, 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 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 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.

(g) Other Restrictions. You may not loan, rent, lease, sublicense, distribute or otherwise transfer all or any portion of the Software to third parties except to the limited extent set forth in Section 3 or as otherwise expressly provided. You may not copy the Software except as expressly set forth above, and any copies that you are permitted to make pursuant to this Agreement must contain the same copyright, patent and other intellectual property markings that appear on or in the Software. You may not modify, adapt or translate the Software. You may not, directly or indirectly, encumber or suffer to exist any lien or security interest on the Software; knowingly take any action that would cause the Software to be placed in the public domain; or use the Software in any computer environment not specified in this Agreement. You may not permit any use of or access to the Software by any third party in connection with a commercial service offering, such as for a cloud-based or web-based SaaS offering.

You will comply with applicable law and Altova 's instructions regarding the use of the Software. You agree to notify your employees and agents who may have access to the Software of the restrictions

contained in this Agreement and to ensure their compliance with these restrictions.

(h) NO GUARANTEE. THE SOFTWARE IS NEITHER GUARANTEED NOR WARRANTED TO BE ERROR-FREE NOR SHALL ANY LIABILITY BE ASSUMED BY ALTOVA IN THIS RESPECT. NOTWITHSTANDING ANY SUPPORT FOR ANY TECHNICAL STANDARD, THE SOFTWARE IS NOT INTENDED FOR USE IN OR IN CONNECTION WITH, WITHOUT LIMITATION, THE OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION, COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL EQUIPMENT, MEDICAL DEVICES OR LIFE SUPPORT SYSTEMS, MEDICAL OR HEALTH CARE APPLICATIONS, OR OTHER APPLICATIONS WHERE THE FAILURE OF THE SOFTWARE OR ERRORS IN DATA PROCESSING COULD LEAD TO DEATH, PERSONAL INJURY OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE. YOU AGREE THAT YOU ARE SOLELY RESPONSIBLE FOR THE ACCURACY AND ADEQUACY OF THE SOFTWARE AND ANY DATA GENERATED OR PROCESSED BY THE SOFTWARE FOR YOUR INTENDED USE AND YOU WILL DEFEND, INDEMNIFY AND HOLD ALTOVA, ITS OFFICERS AND EMPLOYEES HARMLESS FROM ANY THIRD PARTY CLAIMS, DEMANDS, OR SUITS THAT ARE BASED UPON THE ACCURACY AND ADEQUACY OF THE SOFTWARE IN YOUR USE OR ANY DATA GENERATED BY THE SOFTWARE IN YOUR USE.

2. INTELLECTUAL PROPERTY RIGHTS

You acknowledge that the 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 Software are the valuable trade secrets and confidential information of Altova and its suppliers. The 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 the Software and all accompanying printed materials. You will take no actions which adversely affect Altova 's intellectual property rights in the 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. Altova®, XMLSpy®, Authentic®, StyleVision®, MapForce®, UModel®, Dat abase Spy®, DiffDog[®], SchemaAgent®, SemanticWorks®, MissionKit®, Markup Your Mind®, Nanonull™, RaptorXML™, RaptorXML Server™, RaptorXML +XBRL Server™, Powered By RaptorXML™, FlowForce Server™, StyleVision Server™, and MapForce Server™ are trademarks of Altova GmbH. (pending or registered in numerous countries). Unicode and the Unicode Logo are trademarks of Unicode, Inc. Windows, Windows XP, Windows Vista, Windows 7, and Windows 8 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 Agreement does not grant you any intellectual property rights in the Software. Notifications of claimed copyright infringement should be sent to Altova 's copyright agent as further provided on the Altova Web Site.

3. LIMITED TRANSFER RIGHTS

Notwithstanding the foregoing, you may transfer all your rights to use the Software to another person or legal entity provided that: (a) you also transfer this Agreement, the Software and all other software or hardware bundled or pre-installed with the Software, including all copies, updates and prior versions, and all copies of font software converted into other formats, to such person or entity; (b) you retain no copies, including backups and copies stored on a computer; (c) the receiving party secures a personalized key code from Altova; and (d) the receiving party accepts the terms and conditions of this Agreement and any other terms and conditions upon which you legally purchased a license to the Software. Notwithstanding the foregoing, you may not transfer education, pre-release, or not-for-resale copies of the Software.

4. PRE- RELEASE AND EVALUATION PRODUCT ADDITIONAL TERMS

If the product you have received with this license is pre-commercial release or beta Software ('Pre-release Software "), then this Section 4 applies. In addition, this section applies to all evaluation and/ or demonstration copies of Altova software ('Evaluation Software ") and continues in effect until you purchase a license. To the extent that any provision in this section is in conflict with any other term or condition in this Agreement, this section shall supersede such other term(s) and condition(s) with respect to the Pre-release and/ or Evaluation Software, but only to the extent necessary to resolve the conflict. You acknowledge that the Pre-release Software is a pre-release

version, does not represent final product from Altova, and may contain bugs, errors and other problems that could cause system or other failures and data loss. CONSEQUENTLY, THE PRE-RELEASE AND/ OR EVALUATIÓN SOFTWARE IS PROVIDED TO YOU "AS- IS " WITH NO WARRANTIES FOR USE OR PERFORMANCE, AND ALTOVA DISCLAIMS ANY WARRANTY OR LIABILITY OBLIGATIONS TO YOU OF ANY KIND, WHETHER EXPRESS OR IMPLIED. WHERE LEGALLY LIABILITY CANNOT BE EXCLUDED FOR PRE-RELEASE AND/ OR EVALUATION SOFTWARE, BUT IT MAY BE LIMITED, ALTOVA 'S LIABILITY AND THAT OF ITS SUPPLIERS SHALL BE LIMITED TO THE SUM OF FIFTY DOLLARS (USD \$50) IN TOTAL. If the Evaluation Software has a time-out feature, then the software will cease operation after the conclusion of the designated evaluation period. Upon such expiration date, your license will expire unless otherwise extended. Your license to use any output created with the Evaluation Software that contains generated program code (including Unrestricted Source Code) such as Java, C++, C, VB.NET or XSLT and associated project files and build scripts as well as generated XML, XML Schemas, documentation, UML diagrams, and database structures terminates automatically upon the expiration of the designated evaluation period but the license to use such output is revived upon your purchase of a license for the Software that you evaluated and used to create such output. Access to any files created with the Evaluation Software is entirely at your risk. You acknowledge that Altova has not promised or guaranteed to you that Pre-release Software will be announced or made available to anyone in the future, that Altova has no express or implied obligation to you to announce or introduce the Pre-release Software, and that Altova may not introduce a product similar to or compatible with the Pre-release Software. Accordingly, you acknowledge that any research or development that you perform regarding the Pre-release Software or any product associated with the Pre-release Software is done entirely at your own risk. During the term of this Agreement, if requested by Altova, you will provide feedback to Altova regarding testing and use of the Pre-release Software, including error or bug reports. If you have been provided the Pre-release Software pursuant to a separate written agreement, your use of the Software is governed by such agreement. You may not sublicense, lease, loan, rent, distribute or otherwise transfer the Pre-release Software. Upon receipt of a later unreleased version of the Pre-release Software or release by Altova of a publicly released commercial version of the Software, whether as a stand-alone product or as part of a larger product, you agree to return or destroy all earlier Pre-release Software received from Altova and to abide by the terms of the license agreement for any such later versions of the Pre-release Software.

5. LIMITED WARRANTY AND LIMITATION OF LIABILITY

(a) Limited Warranty and Customer Remedies. Altova warrants to the person or entity that first purchases a license for use of the Software pursuant to the terms of this Agreement that (i) the Software will perform substantially in accordance with any accompanying Documentation for a period of ninety (90) days from the date of receipt, and (ii) any support services provided by Altova shall be substantially as described in Section 6 of this Agreement. Some states and jurisdictions do not allow limitations on duration of an implied warranty, so the above limitation may not apply to you. To the extent allowed by applicable law, implied warranties on the Software, if any, are limited to ninety (90) days. Altova's and its suppliers' entire liability and your exclusive remedy shall be, at Altova's option, either (i) return of the price paid, if any, or (ii) repair or replacement of the Software that does not meet Altova 's Limited Warranty and which is returned to Altova with a copy of your receipt. This Limited Warranty is void if failure of the Software has resulted from accident, abuse, misapplication, abnormal use, Trojan horse, virus, or any other malicious external code. Any replacement Software will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer. This limited warranty does not apply to Evaluation and/ or Pre-release Software.

(b) No Other Warranties and Disclaimer. THE FOREGOING LIMITED WARRANTY AND REMEDIES STATE THE SOLE AND EXCLUSIVE REMEDIES FOR ALTOVA OR ITS SUPPLIER'S BREACH OF WARRANTY. ALTOVA AND ITS SUPPLIERS DO NOT AND CANNOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THE SOFTWARE. EXCEPT FOR THE FOREGOING LIMITED WARRANTY, AND FOR ANY WARRANTY, CONDITION, REPRESENTATION OR TERM TO THE EXTENT WHICH THE SAME CANNOT OR MAY NOT BE EXCLUDED OR LIMITED BY LAW APPLICABLE TO YOU IN YOUR JURISDICTION, ALTOVA AND ITS SUPPLIERS MAKE NO WARRANTIES, CONDITIONS, REPRESENTATIONS OR TERMS, EXPRESS OR IMPLIED, WHETHER BY STATUTE, COMMON LAW, CUSTOM, USAGE OR OTHERWISE AS TO ANY OTHER MATTERS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, ALTOVA AND ITS SUPPLIERS DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, SATISFACTORY QUALITY, INFORMATIONAL CONTENT OR ACCURACY, QUIET ENJOYMENT, TITLE AND NON-INFRINGEMENT, WTH REGARD TO THE SOFTWARE, AND THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT SERVICES. THIS LIMITED WARRANTY GIVES
YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHERS, WHICH VARY FROM STATE/ JURISDICTION TO STATE/ JURISDICTION.

(c) Limitation of Liability. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW EVEN IF A REMEDY FAILS ITS ESSENTIAL PURPOSE, 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 SOFTWARE OR THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT SERVICES, EVEN IF ALTOVA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN ANY CASE, ALTOVA 'S ENTIRE LIABILITY UNDER ANY PROVISION OF THIS AGREEMENT SHALL BE LIMITED TO THE AMOUNT ACTUALLY PAID BY YOU FOR THE SOFTWARE PRODUCT. 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 shall be limited to the greatest extent permitted by law and the limitations or exclusions of warranties and liability contained herein do not prejudice applicable statutory consumer rights of person acquiring goods otherwise than in the course of business. The disclaimer and limited liability above are fundamental to this Agreement between Altova and you.

Infringement Claims. Altova will indemnify and hold you harmless and will defend or (d) settle any claim, suit or proceeding brought against you by a third party that is based upon a claim that the content contained in the Software infringes a copyright or violates an intellectual or proprietary right protected by United States or European Union law ("Claim"), but only to the extent the Claim arises directly out of the use of the Software and subject to the limitations set forth in Section 5 of this Agreement except as otherwise expressly provided. You must notify Altova in writing of any Claim within ten (10) business days after you first receive notice of the Claim, and you shall provide to Altova at no cost such assistance and cooperation as Altova may reasonably request from time to time in connection with the defense of the Claim. Altova shall have sole control over any Claim (including, without limitation, the selection of counsel and the right to settle on your behalf on any terms Altova deems desirable in the sole exercise of its discretion). You may, at your sole cost, retain separate counsel and participate in the defense or settlement negotiations. Altova shall pay actual damages, costs, and attorney fees awarded against you (or payable by you pursuant to a settlement agreement) in connection with a Claim to the extent such direct damages and costs are not reimbursed to you by insurance or a third party, to an aggregate maximum equal to the purchase price of the Software. If the Software or its use becomes the subject of a Claim or its use is enjoined, or if in the opinion of Altova 's legal counsel the Software is likely to become the subject of a Claim, Altova shall attempt to resolve the Claim by using commercially reasonable efforts to modify the Software or obtain a license to continue using the Software. If in the opinion of Altova 's legal counsel the Claim, the injunction or potential Claim cannot be resolved through reasonable modification or licensing, Altova, at its own election, may terminate this Agreement without penalty, and will refund to you on a pro rata basis any fees paid in advance by you to Altova. THE FOREGOING CONSTITUTES ALTOVA 'S SOLE AND EXCLUSIVE LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT. This indemnity does not apply to situations where the alleged infringement, whether patent or otherwise, is the result of a combination of the Altova software and additional elements supplied by you.

6. SUPPORT AND MAINTENANCE

Altova offers multiple optional "Support & Maintenance Package(s)" ("SMP") for the version of Software product edition that you have licensed, which you may elect to purchase in addition to your Software license. 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 and maintenance as well as your upgrade eligibility depend on your decision to purchase SMP and the level of SMP that you have purchased:

(a) If you have not purchased SMP, you will receive the Software "AS IS" and will not receive any maintenance releases or updates. However, Altova, at its option and in its sole discretion on a case by case basis, may decide to offer maintenance releases to you as a courtesy, but these maintenance releases will not include any new features in excess of the feature set at the time of your purchase of the Software. In addition, Altova will provide free technical support to you for thirty (30) days after the date of your purchase (the "Support Period" for the purposes of this paragraph 6(a), and Altova, in its sole discretion on a case by case basis, may also provide free courtesy technical support during your thirty (30) day evaluation period. Technical support is provided via a Web-based support form only, and there is no guaranteed response time. (b) If you have purchased SMP, then solely for the duration of its delineated Support Period, you are eligible to receive the version of the 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. The specific upgrade edition that you are eligible to receive based on your Support Period is further detailed in the SMP that you have purchased. 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 Software problem or error to Altova. If Altova determines that a reported reproducible material error in the Software exists and significantly impairs the usability and utility of the Software, Altova agrees to use reasonable commercial efforts 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 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 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 Software. Altova is not obligated to fix errors that are immaterial. Immaterial errors are those that do not significantly impact use of the Software as determined by Altova in its sole discretion. 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 Software and Altova will not provide you with generic consultation, assistance, or advice under any circumstances.

Updating Software may require the updating of software not covered by this Agreement before installation. Updates of the operating system and application software not specifically covered by this Agreement are your responsibility and will not be provided by Altova under this Agreement. Altova 's obligations under this Section 6 are contingent upon your proper use of the Software and your compliance with the terms and conditions of this Agreement at all times. Altova shall be under no obligation to provide the above technical support if, in Altova 's opinion, the 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 Software without Altova 's written approval; (iii) causes external to the Software, such as natural disasters, the failure or fluctuation of electrical power, or computer equipment failure; (iv) your failure to maintain the Software at Altova 's specified release level; or (v) use of the Software with other software without Altova as 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 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. SOFTWARE ACTIVATION, UPDATES AND LICENSE METERING

(a) License Metering. The Software includes a built- in license metering module that is designed to assist you with monitoring license compliance in small local area networks (LAN). The metering module attempts to communicate with other machines on your local area network (LAN). You permit Altova to use your internal network for license monitoring for this purpose. This license metering module may be used to assist with your license compliance but should not be the sole method. Should your firewall settings block said communications, you must deploy an accurate means of monitoring usage by the end user and preventing users from using the Software more than the Permitted Number.

(b) License Compliance Monitoring. You are required to utilize a process or tool to ensure that the Permitted Number is not exceeded. Without prejudice or waiver of any potential violations of the Agreement, Altova may provide you with additional compliance tools should you be unable to accurately account for license usage within your organization. If provided with such a tool

by Altova, you (a) are required to use it in order to comply with the terms of this Agreement and (b) permit Altova to use your internal network for license monitoring and metering and to generate compliance reports that are communicated to Altova from time to time.

The Software may use your internal network and Software Activation. (C) Internet connection for the purpose of transmitting license-related data at the time of installation, registration, use, or update to an Altova Master 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 Master License Server. You agree that Altova may use these measures and you agree to follow any applicable requirements. You further agree that use of license key codes that are not or were not generated by Altova and lawfully obtained from Altova, or an authorized reseller as part of an effort to activate or use the Software violates Altova 's intellectual property rights as well as the terms of this Agreement. You agree that efforts to circumvent or disable Altova 's copyright protection mechanisms, the license management mechanism, or the Altova Master License Server violate Altova 's intellectual property rights as well as the terms of this Agreement. Altova expressly reserves the rights to seek all available legal and equitable remedies to prevent such actions and to recover lost profits, damages and costs.

(d) LiveUpdate. 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.

(e) Use of Data. 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 Agreement. By your acceptance of the terms of this Agreement and/ or use of the Software, you authorize the collection, use and disclosure of information collected by Altova for the purposes provided for in this Agreement and/ or the Privacy Policy. Altova has the right in its sole discretion to amend this provision of the 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.

(f) Audit Rights. You agree that Altova may audit your use of the Software for compliance with the terms of this Agreement at any time, upon reasonable notice. In the event that such audit reveals any use of the Software by you other than in full compliance with the terms of this Agreement, you shall reimburse Altova for all reasonable expenses related to such audit in addition to any other liabilities you may incur as a result of such non-compliance.

(g) Notice to European Users. Please note that the information as described in paragraph 7(d) above may be transferred outside of the European Economic Area, for purposes of processing, analysis, and review, by Altova, Inc., a company located in Beverly, Massachusetts, U.S.A., or its subsidiaries or Altova 's subsidiaries or divisions, or authorized partners, located worldwide. You are advised that the United States uses a sectoral model of privacy protection that relies on a mix of legislation, governmental regulation, and self-regulation. You are further advised that the Council of the European Union has found that this model does not provide "adequate" privacy protections as contemplated by Article 25 of the European Union's Data Directive. (Directive 95/ 46/ EC, 1995 O.J. (L 281) 31). Article 26 of the European Union's Data Directive allows for transfer of personal data from the European Union to a third country if the individual has unambiguously given his consent to the transfer of personal information, regardless of the third country's level of protection. By agreeing to this Agreement, you consent to the transfer of all such information to the United States and the processing of that information as described in this Agreement and the Privacy Policy.

8. TERM AND TERMINATION

This Agreement may be terminated (a) by your giving Altova written notice of termination; (b) by Altova, at its option, giving you written notice of termination if you commit a breach of this Agreement and fail to cure such breach within ten (10) days after notice from Altova; or (c) at the request of an authorized Altova reseller in the event that you fail to make your license payment or other monies due and payable. This Agreement automatically terminates upon the expiration of the ABEE License Term. In addition the Agreement governing your use of a previous version of the Software that you have upgraded or updated is terminated upon your acceptance of the terms and

conditions of the Agreement accompanying such upgrade or update. Upon any termination of the Agreement, you must cease all use of the Software that this Agreement governs, 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 Software remain in your possession or control. The terms and conditions set forth in Sections 1(c)- (h), 2, 5, 7, 9, 11- 14 survive termination as applicable.

9. RESTRICTED RIGHTS NOTICE AND EXPORT RESTRICTIONS

The 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 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 Agreement. Manufacturer is Altova GmbH, Rudolfsplatz 13a/9, A-1010 Vienna, Austria/EU. You may not use or otherwise export or re-export the Software or Documentation except as authorized by United States law and the laws of the jurisdiction in which the Software was obtained. In particular, but without limitation, the Software or Documentation may not be exported or re-exported (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 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.

10. U.S. GOVERNMENT ENTITIES

Notwithstanding the foregoing, if you are an agency, instrumentality or department of the federal government of the United States, then this Agreement shall be governed in accordance with the laws of the United States of America, and in the absence of applicable federal law, the laws of the Commonwealth of Massachusetts will apply. Further, and notwithstanding anything to the contrary in this Agreement (including but not limited to Section 5 (Indemnification)), all claims, demands, complaints and disputes will be subject to the Contract Disputes Act (41 U.S.C. § § 7101 et seq.), the Tucker Act (28 U.S.C. § 1346(a) and § 1491), or the Federal Tort Claims Act (28 U.S.C. § § 1346(b), 2401-2402, 2671-2672, 2674-2680), FAR 1.601(a) and 43.102 (Contract Modifications); FAR 12.302(b), as applicable, or other applicable governing authority. For the avoidance of doubt, if you are an agency, instrumentality, or department of the federal, state or local government of the U.S. or a U.S. public and accredited educational institution, then your indemnification obligations are only applicable to the extent they would not cause you to violate any applicable law (e.g., the Anti-Deficiency Act), and you have any legally required authorization or authorizing statute.

11. THIRD PARTY SOFTWARE

The 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/legal3rdparty.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.

12. JURISDICTION, CHOICE OF LAW, AND VENUE

If you are located in the European Union and are using the Software in the European Union and not in the United States, then this Agreement 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 Software resides in the Handelsgericht, Wen (Commercial Court, Vienna) and you further agree and expressly consent to the exercise of personal jurisdiction in the Handelsgericht, Wen (Commercial Court, Vienna) in connection with any such dispute or claim.

If you are located in the United States or are using the Software in the United States then this Agreement will be governed by and construed in accordance with the laws 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 Software resides in the federal or state courts of the Commonwealth of Massachusetts and you further agree and expressly consent to the exercise of personal jurisdiction in the federal or state courts of the Commonwealth 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 Software in the United States, then this Agreement 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 Software resides in the Handelsgericht, Wen (Commercial Court, Vienna) and you further agree and expressly consent to the exercise of personal jurisdiction in the Handelsgericht Wen (Commercial Court, Vienna) in connection with any such dispute or claim. This Agreement 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.

13. TRANSLATIONS

Where Altova has provided you with a foreign translation of the English language version, you agree that the translation is provided for your convenience only and that the English language version will control. If there is any contradiction between the English language version and a translation, then the English language version shall take precedence.

14. GENERAL PROVISIONS

This Agreement 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 Agreement 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 Agreement 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 Agreement. This Agreement 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 Ágreement 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 Agreement. If, for any reason, any provision of this Agreement is held invalid or otherwise unenforceable, such invalidity or unenforceability shall not affect the remainder of this Agreement, and this Agreement 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: 2016/10//07

Index

.NET,

Authentic Desktop スタンドアロンとの違い,138 Authentic Desktop の統合,136

A

ActiveX, アプリケーションレベルの統合 689 ドキュメントレベルの統合,692 統合の必要条件 685 ActiveX controls, support, 373 ActiveX コントロール, Visual Studio ツールボックスへの追加 687 Altova Scripting Projects, 328 Altova XML パーサー, について,753 Altova エンジン, Altova 製品内で,754 Altova グローバルリソース, グローバルリソース,92 グローバルリソースを参照する,92 Altova サポート,21 Altova ソフトウェアの注文,315 Altova ソフトウェアの登録、315 Altova 製品,21 API, documentation, 391 overview, 393 Application. ActiveDocument, 422 AddMacroMenuItem, 422 AddXSLT XQParameter, 422 Application, 423 ClearMacroMenu, 423 CurrentProject, 424 Dialogs, 424 Documents, 424 GetDatabaseImportElementList, 425

GetDatabaseSettings, 425 GetDatabaseTables, 426 GetExportSettings, 426 GetTextImportElementList, 427 GetTextImportExportSettings, 427 GetXSLT XQParameterCount, 428 GetXSLT XQParameterName, 428 GetXSLT XQParameterXPath, 428 ImportFromDatabase, 428 ImportFromSchema, 429 ImportFromText, 430 ImportFromWord, 430 NewProject, 431 OnBeforeOpenDocument, 420 OnBeforeOpenProject, 420 OnDocumentOpened, 421 OnProjectOpened, 421 OpenProject, 432 Parent, 432 Quit, 432 ReloadSettings, 433 RemoveXSLT XQParameter, 433 RunMacro, 433 ScriptingEnvironment, 434 ShowApplication, 434 ShowForm, 435 URLDelete, 435 URLMakeDirectory, 436 WarningNumber, 436 WarningText, 436 **Application Events, 348** ASPRJ files, 328 ATL. plug-in sample files, 377 Authentic Desktop. ユーザーマニュアル ,3 統合,684 Authentic Desktop 統合, のサンプル ,701 Authentic Desktopコマンド, AuthenticDesktopControl, 727 AuthenticDesktopControl 内,730 Authentic Integration Package, 137, 140 Authentic Plugin for VS .NET, インストール,137 Authentic View, 63, 239 CDATA セクション ,32 DB データの編集,237 GUI の概要 ,39

Authentic View, 63, 239 PXF ファイルから出力ドキュメントを生成,247 SPS テーブル ,61 XML テーブル ,63 XML テーブルアイコン,68 XML テーブルの使用 ,63 XML ドコメン の印刷 ,37 XMI ドキュメントを開く,24 XML/テキストとして貼り付け、51 エンティティ,32 エンティティの挿入 36 コンテキストメニュー ,26 ツールバーアイコン 41 データエントリデバイス 32 データの挿入,32 テーブル (SPS とXML), 60 テーブル内,29 テキストの書式設定 41 テキストの範囲 <u>,51</u> ドキュメントの表示44 ノードの削除,29 ノードの挿入,29 ノードの追加 29 マークアップタグの表示 26 マークアップの表示 41.44 メインウィンドウ 44 主要な機能の使用方法 54 新たな XML ファイルを開く,236 切り替え,250 属性値の挿入 35 特殊文字 32 入力ヘルパー ,26,47 要素のクリア 29 要素の適用,29 Authentic View Events, 348 Authentic View テンプレート,24 Authentic スクリプト, セキュリティー設定,248 信頼された場所,248 Authentic スクリプトの信頼された場所,248 Authentic メニュー ,235 マークアップの表示 41 動的なテーブルの編集 41 AuthenticDataTransfer, dropEffect, 440 getData, 440 ownDrag, 441 type, 441 AuthenticDesktopControl, 731

C# を使用した統合 ,697 HTML を使用下統合,702 オブジェクトレファレンス,726 ドキュメント,684 ドキュメントレベルでの統合のサンプルサンプル 696 のサンプル 統合 at application level, 701 AuthenticDesktopControlDocument, 739 AuthenticDesktopControlPlaceHolder, 745 AuthenticRange, AppendRow, 446 Application, 447 CanPerformAction, 447 CanPerformActionWith, 447 Close, 448 Collaps ToBegin, 448 Collaps ToEnd, 448 Copy, 448 Cut. 449 Delete, 449 DeleteRow, 449 DuplicateRow, 450 ExpandTo, 451 FirstTextPosition, 451 FirstXMLData, 452 FirstXMLDataOffset, 452 GetElementAttributeNames, 453 GetElementAttributeValue, 454 GetElementHierarchy, 454 GetEntityNames, 455 Goto, 455 GotoNext, 456 GotoNextCursorPosition, 456 GotoPrevious, 457 GotoPrevious Curs or Position, 457 Has Element Attribute, 458 InsertEntity, 458 InsertRow, 458 Is CopyEnabled, 459 Is CutEnabled, 459 Is Delete Enabled, 459 Is Empty, 460 Is Equal, 460 Is FirstRow, 460 Is InDynamic Table, 460 Is LastRow, 461 Is Paste Enabled, 461 Is TextStateApplied, 461 LastTextPosition, 462 LastXMLData, 462

AuthenticRange, LastXMLDataOffset, 463 MoveBegin, 464 MoveEnd, 464 MoveRowDown, 465 MoveRowUp, 465 Parent, 465 Paste, 466 PerformAction, 466 Select, 467 SelectNext, 467 SelectPrevious, 468 SetElementAttributeValue, 468 SetFromRange, 469 Text, 470 AuthenticView, 485 Application, 478 As XMLString, 478 DocumentBegin, 480 DocumentEnd, 480 Event, 481 Goto, 482 Is RedoEnabled, 482 Is UndoEnabled, 483 MarkupVisibility, 483 OnBeforeCopy, 471 OnBeforeCut, 472 OnBeforeDelete, 472 OnBeforeDrop, 472 OnBeforePaste, 473 OnDragOver, 474 OnKeyboardEvent, 474 OnMouseEvent, 475 OnSelectionChanged, 476 Parent, 483 Print, 483 Redo, 484 Selection, 484 Undo, 485 WholeDocument, 486 XMLDataRoot, 486 Auto-Macro setting, 352

С

C#, Authentic Desktop の統合 697

CDATA セクション,

Authentic View 内に挿入 .55 Class ID,

Authentic Desktop 統合,701 CodeGeneratorDIg, Application, 487 CPPSettings DOMType, 488 CPPSettings LibraryType, 489 CPPSettings UseMFC, 489 CSharpSettings ProjectType, 489 OutputPath, 489 OutputPathDialogAction, 490 OutputResultDialogAction, 490 Parent, 490 ProgrammingLanguage, 491 PropertySheetDialogAction, 491 TemplateFileName, 491 COM-API, documentation, 391

Configure, XMLSPY UI, 374 CR&LF, 289

D

DatabaseConnection, ADOConnection, 492 As Attributes, 493 CreateMissingTables, 493 CreateNew, 493 DatabaseKind, 494 ExcludeKeys, 494 File, 494 IncludeEmptyElements, 495 NumberDateTimeFormat, 496 **ODBCConnection**, 496 SQLSelect, 497 TextFieldLen, 497 DB, 71, 73 Authentic View 内のクエリ,71 Authentic View 内の編集,71,78 Authentic View内のテーブルのナビゲート,72 DB クエリ内のパラメーター,73 クエリの作成,73 内の Authentic Viewの表示のフィルター,73 Debugging macros, 357 Delete,

Delete, Application.URLDelete, 435 Dialogs, Application, 498 CodeGeneratorDlg, 498 DTDSchemaGeneratorDlg, 500 FileSelectionDlg, 499 GenerateSampleXMLDlg, 499 Parent, 499 SchemaDocumentationDlg, 499 directories, creating with Application.URLMakeDirectory, 436 Document, 513 Application, 506 AssignDTD, 506 AssignSchema, 506 AssignXSL, 506 AssignXSLFO, 507 Authentic View, 507 Close, 508 ConvertDTDOrSchema, 508 CreateChild, 510 CreateSchemaDiagram, 511 CurrentViewMode, 511 DataRoot, 511 DocEditView, 511 Encoding, 512 EndChanges, 512 ExecuteXQuery, 513 ExportToDatabase, 513 ExportToText, 514 FullName, 515 GenerateDTDOrSchema, 515, 516 GenerateProgramCode, 516 GenerateSampleXML, 517 GenerateSchemaDocumentation, 517 GetExportElementList, 519 GetPathName, 519 GridView, 520 Is Modified, 520 Is Valid, 520 Is WellFormed, 521 Name, 521 OnBeforeCloseDocument, 504 OnBeforeSaveDocument, 503 OnBeforeValidate, 504 OnCloseDocument, 505 OnViewActivation, 505 Path, 522

RootElement, 522 Save, 523 SaveAs, 523 Saved, 523 SaveInString, 523 SaveToURL, 524 SetActiveDocument, 524 SetEncoding, 524 SetExternalIs Valid, 526 SetPathName, 526 StartChanges, 526 SwitchViewMode, 527 Title, 527 Trans formXSL, 528 Trans formXSLFO, 528 UpdateViews, 529 UpdateXMLData, 529 XQuery, 513 Document Events, 348 Documents. Count, 530 Item, 531 NewAuthenticFile, 531 NewFile, 531 NewFileFromText, 532 OpenAuthenticFile, 532 OpenFile, 532 OpenURL, 533 OpenURLDialog, 533 DTD. 289. 291 DTDSchemaGeneratorDIg, Application, 535 AttributeTypeDefinition, 535 DTDSchemaFormat, 535 FrequentElements, 536 GlobalAttributes, 536 MaxEnumLength, 536 MergeAllEqualNamed, 536 OnlyStringEnums, 537 OutputPath, 537 OutputPathDialogAction, 537 Parent, 537 ResolveEntities, 537 TypeDetection, 538 ValueList, 538

Ε

Eclipse のための Authentic プラグイン, インストール,140 Eclipse プラットフォーム, Authentic Integration Package, 140 Authentic パースペクティブ,148 ∠ Authentic Desktop, 139 Eclipse 内の Authentic パースペクティブ,148 ElementList, Count, 539 Item, 539 RemoveElement, 539 ElementListItem, ElementKind, 540 FieldCount, 540 Name, 540 RecordCount, 540 Enter +- ,88 Enter キー参照,88 使用の効果,88 Enumerations, SPYAttributeTypeDefinition, 641 SPYAuthentic Actions, 642 SPYAuthenticDocumentPosition, 643 SpyAuthenticElementActions, 644 SPYAuthenticElementKind, 645 SPYAuthentic Markup Visibility, 646 SPYDatabaseKind, 648 SPYDialogAction, 649 SPYDOMType, 650 SPYDTDSchemaFormat, 651 SPYEncodingByteOrder, 652 SPYExportNamespace, 653 SPYFrequentElements, 655 SPYKeyEvent, 658 SPYLibType, 660 SPYLoading, 661 SPYMouseEvent, 662 SPYNumberDateTimeFormat, 663 SPYProgrammingLanguage, 664 SPYProjectItemTypes, 665 SPYProjectType, 666 SPYSampleXMLGenerationOptimization, 668 SPYSampleXMLGenerationSchemaOrDTDAssignment, 670 SPYSchemaDefKind, 671

SPYSchemaDocumentationFormat, 672 SPYTextDelimiters, 675 SPYTextEnclosing, 676 SPYTypeDetection, 677 SPYURLTapes, 678 SPYViewModes, 681 SPYVirtualKeyMask, 682 SPYXMLDataKind, 683 Event, 420, 421, 471, 472, 473, 474, 475, 476, 503, 504, 505, 562, 563, 564 Event handlers, in Scripting Project, 348 overview, 334 Events, 348, 399 and event handlers, 336 ExportSettings, CreateKeys, 541 ElementList, 541 Entities ToText, 541 ExportAllElements, 542 FromAttributes, 542 FromSingleSubElements, 542 FromTextValues, 542 IndependentPrimaryKey, 542 Namespace, 543 SubLevelLimit, 543 F FileSelectionDlg, Application, 544 DialogAction, 544

FullName, 544 Parent, 545

Form Object Palette, 330 Form Object properties, 343 Forms,

and built- in commands, 360 and event handling, 346 and Form Objects, 343 creating new, 341 in Scripting Projects, 340 invocation of, 336 naming, 341 overview, 334 properties of, 341 setting tab sequence of objects, 343 775

Eclipse のための Authentic プラグイン,140

G

Generate Sample XML, 641, 668, 670 GenerateSampleXMLDIg, Application, 557 FillWithSampleData, 559 NonMandatoryAttributes, 559 NonMandatoryElements, 559 Parent, 560 RepeatCount, 560 TakeFirstChoice, 561 Global declarations, 338 overview, 334 Global scripting project, of Authentic Desktop, 328 Grid View Events, 348 GridView, CurrentFocus, 564 Deselect, 564 Is Visible, 564 OnBeforeDrag, 562 OnBeforeDrop, 562 OnBeforeStartEditing, 563 OnEditingFinished, 563 OnFocus Changed, 564 Select, 565 SetFocus, 565

GUI 詳細,8

Η

HTML, Authentic Desktop の統合,702 HTML サンプル, AuthenticDesktopControl 統合,701 of AuthenticDesktopControl 統合,701 HTML 出力, PXF ファイルからAuthentic View 内に生成,247

J

Java, 704 JRE,

Language, scripting language - changing, 336 loading, 533

Μ

Macros,

creating with Scripting Editor, 352 debugging, 357 editing with Scripting Editor, 352 execution of, 336 functions for, in Global Declarations, 338 how to use in Scripting Project, 351 overview, 334 running, 354 setting as Auto-Macro in Scripting Editor, 352 Microsoft® SharePoint® Server, 212 MIME, 291 MSXML, 295

0

OASIS, XML カタログ ,225 OS, Altova 製品のための ,752 Overview, of XMLSpy API, 393

Ρ

Parent, 522 PDF 出力, PXF ファイルからAuthentic View 内に生成 247 Plug- in,

ATL sample files, 377 registration, 372 User interface configuration, 374 Plug-in, XMLSPY, 371 Programming points, in Scripting Project, 358 Properties and Events pane, 330 PUBLIC, 識別子 - カタログ ,225 PXF ファイル, Authentic View から出力ドキュメントを生成 ,247

R

Register, plug-in, 372 Return キー, Enter キーを参照 .88 RTF 出力, PXF ファイルからAuthentic View 内に生成 .247

S

save, 524 schema, 429 SchemaDocumentationDlg, AllDetails, 567 Application, 567 Include All, 569 Include Attribute Groups, 569 IncludeComplexTypes, 569 IncludeGlobalElements, 570 IncludeGroups, 570 IncludeIndex, 570 IncludeLocalElements, 571 IncludeRedefines, 571 IncludeSimpleTypes, 572 Options DialogAction, 572 OutputFile, 573 OutputFileDialogAction, 573 OutputFormat, 573 Parent, 573 ShowAnnotations, 574 ShowAttributes, 574 ShowChildren, 574 ShowConstraints, 575 ShowDiagram, 575

ShowEnumerations, 575 ShowNamespace, 575 ShowPatterns, 576 ShowProgressBar, 576 ShowProperties, 576 ShowResult, 577 ShowSingleFacets, 577 ShowSourceCode, 577 ShowType, 577 ShowUsedBy, 578 Scripting Editor, GUI description, 330 Main Window, 330 Scripting Environment, 325 usage overview, 327 Scripting language, 336 Scripting Project, and Events, 348 application event handlers, 348 Event Handlers, 334 Forms, 334 Forms in, 340 Global Declarations, 334 Global Declarations in, 338 Macros, 334 Macros in, 351 programming points, 358 steps for creating, 336 Scripting Project Tree pane, 330 Scripting Projects, for Authentic Desktop, 328 for Authentic Desktop Project, 328 SharePoint® Server, 212 Source control, installing a source-control plug-in, 110 SPP ファイルの位置,180 SPS. 新たな XML ファイルに関連付ける,153 SPSテーブル 動的なテーブルの編集 41 SPS テーブル 内の Authentic View, 使用方法 ,61 SpyProject, CloseProject, 579 ProjectFile, 579 RootItems, 579 SaveProject, 580 SaveProjectAs, 580

ChildItems, 581 FileExtensions, 581 ItemType, 581 Name, 581 Open, 582 ParentItem, 582 Path, 582 ValidateWth, 582 XMLForXSLTrans formation, 582 XSLForXMLTrans formation, 582 XSLTrans formationFileExtension, 583

SpyProjectItems,

AddFile, 584 AddFolder, 584 AddURL, 584 Count, 585 Item, 585 RemoveItem, 585

Т

terminate, 432 Text View Events, 348 TextImportExportSettings, DestinationFolder, 586 EnclosingCharacter, 586 Encoding, 586 EncodingByteOrder, 587 FieldDelimiter, 587 FieleExtension, 587 HeaderRow, 587 ImportFile, 587

U

UCS-2, 293 Unicode のサポート, Altova 製品内で,755 URL, 435, 436, 524, 533 メールで送信,169 User interface, configure using plug-in, 374 UTF-16, 293

V

Visual Studio, Authentic Desktop ActiveX コントロールをツールボックスへ追加 ,687 Visual Studio .Net, と Authentic Desktop, 136 と Authentic Desktop の違い,138 VS .NET, と Authentic Integration Package, 137

W

Web サーバー,319
Windows, Altova 製品のためのサポート,752
フロート、ドッキング、タブ & 自動非表示 &
Word 2007+ 出力, PXF ファイルからAuthentic View 内に生成 247

X

XML. Oasis カタログ,225 スペルチェッカー ,260 XML DB. 新たにデータ行をAuthentic View ヘロード,239 新規 XML デーダ行のロード,72 XML DB 内のデータを編集,239 xML テーブル 内の Authentic View, 編集のためのアイコン,68 XML テーブル内の Authentic View, 使用方法 ,63 XML ドキュメント, Authentic View 内で開く,24 XML パーサー, について,753 XML メニュー ,222 XML 署名,85,240 XMLData, AppendChild, 629 EraseAllChildren, 630

XMLData, EraseCurrentChild, 630 GetChild, 631 GetChildKind, 632 GetCurrentChild, 632 GetFirstChild, 633 GetNextChild, 633 Has Children, 635 Has ChildrenKind, 635 InsertChild, 635 Is SameNode, 636 Kind, 636 MayHaveChildren, 637 Name, 637 Parent, 637 TextValue, 638 XMLSPY, 151 plug-in registration, 372 ヘルプ ,21 機能,21 XMLSpy API, documentation, 391 overview, 393 XMLSPY plug-in, 371 XMLSpv コマンドテーブル,713 XMLSPY を使用して編集,289 XMLSpyLib, 391, 394 Application, 419 Authentic Data Trans fer, 440 AuthenticRange, 445 Authentic View, 471 CodeGeneratorDlg, 487 DatabaseConnection, 492 Dialogs, 498 Document, 502 Documents, 530 DTDSchemaGeneratorDlg, 535 ElementList, 539 ElementListItem, 540 ExportSettings, 541 FileSelectionDlg, 544 GenerateSampleXMLDlg, 557 GridView, 562 ProjectItem, 581 SchemaDocumentationDlg, 566 SpyProject, 579 SpyProjectItems, 584 TextImportExportSettings, 586 XMLData, 628

XML- 準拠,291 XQuery, XQuery ドキュメントを変数を受け渡し,231 XQuery プロセッサー, Altova 製品内で,754 XSL/ XQuery メニュー,227 XSLT, プロセッサ,295 XSLT パラメーター, インターフェースを介してスタイルシートへ受け渡し,231 XSLT プロセッサー, Altova 製品内で,754 XSLT 変換,228 FO への変換,229 PDF への変換,229

Ζ

アイコン、 ツールバー /メニューへ追加 ,273 大き〈表示 ,285 アクティブな構成、 グローバルリソース ,272 イメージ書式設定、 内の Authentic View, 87 インターネット,319 インターネットの使用、 Altova 製品内で,756 ウィンドウ、 プロジェクト,305,310 開く311 左右に並べて表示,304 重ねて表示 302 上下に並べて表示,303 情報,306,310 入力ヘルパー,307,310 ウィンドウ メニュー,301 エイリアス グローバルリソース参照 92 エクスプローラー,291 エンコーディング デフォルト,293 ファイル ,161 エンティティ、 Authentic View に挿入 36 Authentic View 内に挿入 ,55 Authentic View 内の定義 55

(C) 2018 Altova GmbH

エンティティ、 内の Authentic View の定義 ,83 エンティティ入力ヘルパー、 Authentic View, 47 エンドユーザー使用許諾契約書,757 お気に入り、313 カスタマイズ,19 コンテキストメニュー ,280 コンテキストメニューのカスタマイズ ,285 ツールバー /メニューコマンド,273 マクロ 282 メニュー ,280 カスタムカタログ ,225 カスタム辞書,260 カタログ, Oasis XML, 225 キーコード、 Altova ソフトウェアのための 315 キーボードショートカット,277 キーマップ,314 クエリ. Authentic View内の DB 表示,73 グローバル 設定 289 グローバルリソース、 定義 .93 グローバル リソース XML ファイル ,93 グローバルリソース アクティブな構成 272 ツールバーにて有効にする ,275 データベースの型の定義,103 ファイルの型の定義 96 ファイル型とフォルダー型の使用,106 フォルダーの型の定義,101 構成の変更,109 使用,105,109 定義 271 グローバルリソースの 構成 .93 グローバルリソースの構成,109 コピーコマンド,176 コマンド,273 キーマップにて表示 ,314 コンテキストメニュー,280 ツールバー /メニューへ追加 ,273 メニューから削除,280 メニューのリセット,280 コマンドライン、320 コンテキストメニュー,

カスタマイズ 285 コマンド,280 サーチ, 検索を参照,179 サポートオプション,21 サポートセンター,319 ショートカット、 ツールチップにて表示,285 割り当て 削除 277 スキーマ、 設定 ,289 スクリプト,297 スクリプトエディター 開始 268 スクリプト言語,297 スタートグループ 追加 (コンテキストメニュー),285 ステータスパー ,16 スプラッシュ画面,294 スペルチェッカー、 カスタム辞書,260 スペルチェックのオプション,264 セルの最大幅,294 ソースコントロール チェックアウトを元に戻す,193 プロバイダーの変更 ,200 ソース管理,297 サポートされているプロバイダー ,188 ステータスを最新の状態に更新,200 ソース管理に追加,194 チェックアウト,191 ファイルの削除,195 フォルダーの取得,190 プロジェクトを開く,188 プロパティ,199 共有 ,195 差分の表示,198 最新バージョンの取得,190 有効化、無効化,189 履歴の表示 ,197 ソース管理マネジャー,200 ソフトウェアのライセンスの認証,315 タブ文字,289 チェック、 スペルチェッカー ,260 ツール. 外部アプリケーションを参照,276 ツールチップ、 ショートカットを表示 ,285

ツールチップ 表示,285 ツールバー ,16 コマンドを追加 .273 ツールバーならびにメニューコマンドのリセット,275 マクロの追加 .282 新規作成 ,275 大きなアイコンで表示 285 有効化/無効化,275 ツールメニュー,259 データペース、 Authentic Viewにて編集,237 DB を参照 ,71 テーブル Authentic View内,29 自動的に生成 291 動的な_(SPS) テーブルの編集 41 テーブル 内の Authentic View, SPS (静的と動的) テーブルの使用 .61 XML テーブルの編集のためのアイコン,68 テーブル内の Authentic View. XML テーブルの使用 ,63 テキスト、 Authentic View 内の書式設定 55 Authentic View 内の編集,55 ドキュメント内を検索,178 検索と置き換え,179 テキストの範囲、 内の Authentic View, 51 テクニカルサポート,319 デフォルト、 エンコーディング,293 メニュー ,280 デフォルトエディター,291 デフォルトビュー, メインウィンドウにおける設定 291 テンプレート. Authentic View における XML ドキュメントのテンプレート, 236 テンプレート XML ファイル, Authentic View 内で開く,24 テンプレートファイル、 新規ドキュメントに対して,153 ドキュメント、 スペルチェッカー ,260 ドキュメントレベル、 XMLSpy の統合のサンプル 696 ドッキングウィンドウ,8 ドック可能なウィンドウ,305.306

トピック 目次にて参照 313 パーサー、 Altova 製品に内蔵の,753 パラメーター、 DB クエリ内,73 インターフェースを介してスタイルシートへ受け渡し.231 バリデーター、 Altova 製品内で,753 ビッグエンディアン,293 ピュー・ ブラウザー ビュー ,251 ビューの変更、 Authentic View, 41 ビューメニュー,249 ファイル エンコーディング,161 ソース管理に追加,194 タブ ,289 デフォルトエンコーディング,293 メールで送信,169 印刷オプション,171 開く,155 開くオプション 289 最後に使用,173 新たに作成 ,153 閉じる,162 保存 ,163 ファイル メニュー ,152 ファイルタイプ ,291 ファイルの保存、 エンコーディング,161 ファイル拡張子、 カスタマイズ ,225 フォントサイズ、 ブラウザービューにて,257 ブックマーク,313 ブラウザー,294 ビュー ,251 ブラウザー ビュー ,252 ブラウザー メニュー ,252 ブラウザービュー、 フォントサイズ .257 ページの読み込みを中止,255 最新の状態に更新,256 進む ,254 別のウィンドウ,258 戻る 253 プラットフォーム,

プラットフォーム、 Altova 製品のための,752 プログラマーのレファレンス,322 プログラム設定,289 プロジェクト,206 URL の追加 ,204 アクティブなファイルを追加,205,206 グローバルリソースの追加,203 ソース管理に追加,194 ファイルの追加,202 フォルダーの追加 207 プロパティ 218 開く,184 外部ウェブフォルダーの追加 212 外部フォルダーの追加 ,208 概要,180 関連するファイルを追加,206 再ロード,185 最近使用されたプロジェクト,221 新規作成 ,183 閉じる,186 保存,187 プロジェクトウィンドウ,12,305,310 表示 俳表示の切り替え 309 プロジェクトメニュー,180 ヘルプ キーマップ ,314 目次 313 ヘルプ メニュー ,312 ヘルプシステム,313 ホットキー,277 マークアップ 内の Authentic View, 41, 44 マークアップ (Authentic View にて), 隠す,244 小さな/大きな/混合マークアップを表示 ,244 マークアップの非表示、41 マークアップを隠す,44 マークアップを隠す (Authentic View にて),244 マクロ・ アプリケーションマクロの起動 269 メニュー /ツールバーへ追加,282 編集ボタン 285 メインウィンドウ,10 メインウィンドウにおけるドキュメント,10 メインカタログ ,225 メニュー ,273 Authentic, 235 XML, 222

XSL/ XQuery, 227 ウィンドウ 301 カスタマイズ ,280 コマンドの追加 /削除 ,273 コマンドを削除 ,280 ツール,259 デフォルト/XMLSpy, 280 ビュー ,249 プロジェクト,180 ヘルプ ,312 マクロの追加,282 編集,174 メニュー ブラウザー ,252 メニューバー,16 メモリ要件,752 やり直しコマンド,175 ユーザーインターフェイス詳細,8 ユーザーマニュアル .3 ユーザーマニュアル Authentic Desktop, 6 ユーザーリファレンス,151 ライセンス、 Altova ソフトウェアのための 315 情報,757 リセット、 ショー トカット,277 ツールバーならびにメニューコマンド,275 メニューコマンド,280 リトルエンディアン,293 リンク・ Authentic View 内の ,55 移動ウィンドウ,8 印刷。 Authentic View から,37 印刷オプション,171 印刷設定,172 隠す,305,306,307,310 解析, XSLT. 295 改行,289 開く、 ファイル ,155 開くオプション、 ファイル ,289 外部 XSL プロセッサ,295 外部アプリケーション、 ファイルから開く,276 外部で解析されるエンティティ,291 割り当て、 ショートカットをコマンドへ割り当て 277

技術データ,751 空要素,291 検索, ドキュメント内のテキスト,178 と置き換え(ドキュメント内のテキストにて),179 検証,19.225 元に戻すコマンド,175 構文による色分け,291,294 行, 削除 (Authentic View にて),245 上 下へ移動 246 挿入 (Authentic View にて),245 追加 (Authentic View にて),245 複製 (Authentic View にて),245 混合マークアップ (Authentic View にて),244 再ロード,289 ファイルの変更,160 最後に使用されたファイル、 リスト,173 最適な幅,294 削除, コンテキストメニューのコマンドを削除 ,280 ショートカット,277 ツールバー ,275 ツールバーからアイコンを削除,273 ツールバーからコマンドを削除 273 行 (Authentic View にて),245 削除コマンド,176 視覚的な書式設定 内の Authentic View, 87 試用期間 Altova ソフトウェア製品の試用,758,759 自動検証,291 自動検証の無効化,291 自動非表示ウィンドウ,8 辞書, スペルチェッカー ,260 既存のものを修正 260 新たに追加 ,260 手今日,289 重ねて表示。 ウィンドウ 302 出力ウィンドウ、 表示 / 非表示の切り替え ,308 出力フォーマット,289 小さなマークアップ,44 小さなマークアップ (Authentic View), 244 小さなマークアップの表示,41 章,313

上 /下へ移動. 行 (Authentic View にて),246 情報 ウィンドウ,306.310 情報ウィンドウ,14 新規ファイル、 作成,153 整形式のチェック,223 正規表現, 検索文字列にて,178 静的な (SPS) テーブル 内の Authentic View. 使用方法 ,61 切り取りコマンド,176 切り替え,305,306,307,310 設定,19.289 スクリプト,297 選択されたノードに対しての XPath. 39 全て選択コマンド,177 挿入,245 属性のプレビュー,294 属性值, Authentic View 内に挿入,35 属性入力ヘルパー、 Authentic View, 47 大きなマークアップ,44 大きなマークアップ (Authentic View にて),244 大きなマークアップの表示、41 置換え、 テキスト,178 ドキュメント内のテキスト,179 著作権情報,757 貼り付け、 XML, 51 XML として ,55 テキスト,51 テキストとして、55 貼付けコマンド,176 電子メール、 ファイルを送信、169 登録, 設定,289 統合, アプリケーション内の Authentic Desktop, 684 動的な (SPS) テーブル 内の Authentic View. 使用方法 61 動的なテーブル、 編集 41 内の Authentic View テーブル, 使用方法 60 日付,

日付, 手動での変更,82 日付の選択、 内の Authentic View の使用 ,81 入力ヘルパー,15,307,310 表示/非表示の切り替え,309 背景情報,751 配置, 左右に並べて,304 上下に並べて,303 配布, Altova ソフトウェア製品の配布,758,759 Altova のソフトウェア製品,757 非 XML ファイル ,291 表示,305,306,307,310 表示方法,294 評価キー、 Altova ソフトウェアのための 315 評価期間, Altova のソフトウェア製品,757 複合型マークアップ,44 複合型マークアップの表示,41 複数ユーザー,289 文字セット, エンコーディング,293 文法,291 変換, XSLT 変換を参照,229 変更を検知,289 編集, マクロボタン,285 編集メニュー,174 法的な情報,757 目次,313 要素入力ヘルパー、 Authentic View, 47 隣り合わせ,294 列挙, AuthenticDesktopControl, 748