Altova Authentic 2023 Desktop

Benutzer- und Referenzhandbuch

Altova Authentic 2023 Desktop
Benutzer- und Referenzhandbuch

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

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

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

Published: 2022

© 2016-2022 Altova GmbH

Inhaltsverzeichnis

1 Authentic Desktop und diese Dokumentation 10

1.1 Windows-Dateipfade 11
1.2 Informationen zu dieser Dokumentation 12

2 Benutzeroberfläche und Umgebung 13

2.1 Die grafische Benutzeroberfläche 14
2.1.1 Hauptfenster 15
2.1.2 Projektfenster 17
2.1.3 Info-Fenster 19
2.1.4 Eingabehilfen 19
2.1.5 Ausgabefenster "Meldungen" 19
2.1.6 Menüleiste, Symbolleisten und Statusleiste 20
2.2 Die Applikationsumgebung 22
2.2.1 Einstellungen und Anpassung 22
2.2.2 Tutorials, Projekte, Beispiele 22
2.2.3 Authentic Desktop Features, Hilfe und Altova-Produkte 23

3 Tutorial Authentic-Ansicht 24

3.1 Öffnen eines XML-Dokuments in der Authentic View-Ansicht 26
3.2 Die Benutzeroberfläche der Authentic-Ansicht 28
3.3 Node-Operationen 31
3.4 Eingabe von Daten in der Authentic-Ansicht 34
3.5 Eingabe von Attributwerten 37
3.6 Hinzufügen von Entities 38
3.7 Drucken des Dokuments 40

4 Benutzeroberfläche der Authentic-Ansicht 41

4.1 Übersicht über die Benutzeroberfläche ... 42
4.2 Authentic-Ansicht Symbolleistenschaltflächen ... 44
4.3 Authentic-Ansicht Hauptfenster ... 47
4.4 Authentic-Ansicht Eingabehilfen ... 50
4.5 Authentic-Ansicht Kontextmenüs ... 55

5 Bearbeitung in der Authentic-Ansicht 58
5.1 Automatisches Backup von Dateien ... 59
5.2 Grundlegendes zur Bearbeitung ... 61
5.3 Tabellen in der Authentic-Ansicht ... 66
5.3.1 SPS-Tabellen ... 66
5.3.2 CALS/HTML-Tabellen ... 67
5.3.3 Schaltflächen zum Bearbeiten von CALS/HTML-Tabellen ... 71
5.4 Bearbeiten einer DB ... 74
5.4.1 Navigieren in einer DB-Tabelle ... 74
5.4.2 DB-Abfragen ... 75
5.4.3 Ändern einer DB-Tabelle ... 79
5.5 Arbeiten mit Datumsangaben ... 81
5.5.1 Datumswahl ... 81
5.5.2 Texteingabe ... 82
5.6 Definieren von Entities ... 84
5.7 XML-Signaturen ... 87
5.8 Grafiken in der Authentic-Ansicht ... 89
5.9 Verwendung von Tasten in der Authentic-Ansicht ... 90

6 Authentic Skripterstellung 91

7 Browser-Ansicht 94

8 Globale Altova-Ressourcen 96
8.1 Definieren globaler Ressourcen ... 97
8.1.1 Dateien ... 99

8.1.2 Ordner ... 104
8.1.3 Datenbanken ... 106
8.2 Verwendung globaler Ressourcen ... 109
8.2.1 Zuweisen von Dateien und Ordnern ... 109
8.2.2 Wechseln der aktiven Konfiguration ... 112

9 Versionskontrolle 113

9.1 Einrichten der Versionskontrolle ... 115
9.2 Unterstützte Versionskontrollsysteme ... 116
9.3 Lokaler Arbeitsbereichordner ... 118
9.4 Applikationsprojekt ... 119
9.5 Zu Versionskontrolle hinzufügen ... 121
9.6 Arbeiten mit Versionskontrolle ... 124
9.6.1 Hinzufügen zur, Entfernen aus der Versionskontrolle ... 124
9.6.2 Auschecken, Einchecken ... 125
9.6.3 Dateien schreibgeschützt abrufen ... 128
9.6.4 Kopieren und Freigeben über die Versionskontrolle ... 129
9.6.5 Wechseln der Versionskontrolle ... 132
9.7 Versionskontrolle mit Git ... 134
9.7.1 Aktivieren der Git-Versionskontrolle mit dem Git SCC Plug-in ... 135
9.7.2 Hinzufügen eines Projekts zur Git-Versionskontrolle ... 135
9.7.3 Klonen eines Projekts anhand der Git-Versionskontrolle ... 137

10 Schema-Manager 139

10.1 Ausführen des Schema-Managers ... 142
10.2 Statuskategorien ... 145
10.3 Anwenden eines Patch oder Installation eines Schemas ... 147
10.4 Deinstallieren eines Schemas, Zurücksetzen ... 149
10.5 Befehlszeilenschnittstelle (CLI) ... 150
10.5.1 help ... 150
10.5.2 info ... 151
10.5.3 initialize ... 151
10.5.4 install ... 152

10.5.5 list ... 152
10.5.6 reset ... 153
10.5.7 uninstall ... 154
10.5.8 update ... 155
10.5.9 upgrade ... 155

11 Authentic Desktop in Visual Studio 157

11.1 Installieren des Authentic Plugin für Visual Studio ... 158
11.2 Unterschiede zur Standalone-Version ... 159

12 Authentic Desktop in Eclipse 160

12.1 Installation des Integrationspakets für Eclipse ... 161
12.2 Authentic Desktop-Perspektive in Eclipse ... 163
12.3 Andere Authentic Desktop-Einstiegspunkte in Eclipse ... 166

13 Menübefehle 167

13.1 Menü "Datei" ... 168
13.1.1 Neu ... 168
13.1.2 Öffnen ... 169
13.1.3 Neu laden ... 174
13.1.4 Kodierung ... 174
13.1.5 Schließen, Alle schließen, Alle inaktiven schließen ... 175
13.1.6 Speichern, Speichern unter, Alles speichern ... 175
13.1.7 Als Mail senden ... 181
13.1.8 Drucken ... 182
13.1.9 Druckvorschau, Druckereinrichtung ... 182
13.1.10 Zuletzt verwendete Dateien, Beenden ... 183
13.2 Menü "Bearbeiten" ... 184
13.2.1 Rückgängig, Wiederherstellen ... 184
13.2.2 Ausschneiden, Kopieren, Einfügen, Löschen ... 184
13.2.3 Alles markieren ... 185
13.2.4 Suchen, Weitersuchen ... 185
13.2.5 Ersetzen ... 186

13.3 Menü "Projekt" 187
13.3.1 Neues Projekt 190
13.3.2 Projekt öffnen 190
13.3.3 Projekt neu laden 190
13.3.4 Projekt schließen 190
13.3.5 Projekt speichern, Projekt speichern unter 191
13.3.6 Versionskontrolle 191
13.3.7 Dateien zu Projekt hinzufügen 206
13.3.8 Globale Ressource zu Projekt hinzufügen 207
13.3.9 URL zu Projekt hinzufügen 207
13.3.10 Aktive Datei zu Projekt hinzufügen 207
13.3.11 Aktive und verwandte Dateien zu Projekt hinzufügen 207
13.3.12 Projektordner zu Projekt hinzufügen 208
13.3.13 Externes Verzeichnis zum Projekt hinzufügen 208
13.3.14 Externes Webverzeichnis zum Projekt hinzufügen 211
13.3.15 Skript-Einstellungen 215
13.3.16 Eigenschaften 215
13.3.17 Zuletzt verwendete Projekte 218
13.4 Menü "XML" 219
13.4.1 Wohlgeformtheit prüfen 219
13.4.2 XML validieren 219
13.4.3 Bei Bearbeitung validieren 220
13.5 Menü "XSL/XQuery" 221
13.5.1 XSL-Transformation 221
13.5.2 XSL-FO-Transformation 222
13.5.3 XSL-Parameter / XQuery-Variablen 223
13.6 Menü "Authentic" 228
13.6.1 Neues Dokument 229
13.6.2 Datenbankdaten bearbeiten 229
13.6.3 StyleVision Stylesheet bearbeiten 230
13.6.4 Neue Zeile mit XML-Daten für Bearbeitung auswählen 230
13.6.5 XML-Signatur 231
13.6.6 XML Entities definieren 233
13.6.7 Markup-Code anzeigen 235
13.6.8 RichEdit 235

13.6.9 Zeile anhängen/einfügen/kopieren/löschen ... 236
13.6.10 Zeile nach unten/Zeile nach oben ... 236
13.6.11 HTML-, RTF-, PDF- Word 2007+-, Text-Dokument generieren ... 237
13.6.12 Vertrauenswürdige Pfade ... 237
13.7 Menü "Ansicht" ... 239
13.7.1 Authentic-Ansicht ... 239
13.7.2 Browser-Ansicht ... 239
13.8 Menü "Browser" ... 240
13.9 Menü "Extras" ... 241
13.9.1 Rechtschreibung ... 241
13.9.2 Rechtschreiboptionen ... 244
13.9.3 Skript-Editor ... 247
13.9.4 Makros ... 248
13.9.5 Benutzerdefinierte Tools ... 248
13.9.6 Globale Ressoucen ... 248
13.9.7 Aktive Konfiguration ... 249
13.9.8 Anpassen ... 250
13.9.9 Symbolleisten und Fenster wiederherstellen ... 267
13.9.10 Optionen ... 267
13.10 Menü "Fenster" ... 285
13.11 Menü "Hilfe" ... 287
13.11.1 Inhaltsverzeichnis, Index, Suche ... 287
13.11.2 Tastaturzuordnungen ... 288
13.11.3 Aktivierung, Bestellformular, Registrieren, Updates ... 288
13.11.4 Weitere Befehle ... 292
13.12 Befehlszeile ... 294

14 Referenz für Programmierer 295

14.1 Skript-Editor ... 297
14.1.1 Erstellen eines Skripting-Projekts ... 298
14.1.2 Vordefinierte Befehle ... 312
14.1.3 Aktivieren von Skripts und Makros ... 322
14.2 IDE Plugins ... 325
14.2.1 Registrierung von IDE Plug-Ins ... 325

14.2.2 ActiveX Controls ... 326
14.2.3 Konfigurations-XML ... 326
14.2.4 ATL-Beispieldateien ... 330
14.2.5 IXMLSpyPlugIn ... 335
14.3 Applikations-API ... 341
14.3.1 Übersicht ... 342
14.3.2 Schnittstellen ... 372
14.3.3 Enumerationen ... 604
14.4 ActiveX Integration ... 619
14.4.1 Voraussetzungen ... 619
14.4.2 Hinzufügen der ActiveX Controls zur Toolbox ... 621
14.4.3 Integration auf Applikationsebene ... 622
14.4.4 Integration auf Dokumentebene ... 624
14.4.5 Beispiele zur ActiveX-Integration ... 627
14.4.6 Befehlsreferenz ... 641
14.4.7 Objektreferenz ... 648

15 Anhänge 670

15.1 Technische Daten ... 671
15.1.1 OS- und Arbeitsspeicheranforderungen ... 671
15.1.2 Altova-Prozessoren ... 671
15.1.3 Unicode-Unterstützung ... 672
15.1.4 Internet-Verwendung ... 672
15.2 Lizenzinformationen ... 674
15.2.1 Electronic Software Distribution ... 674
15.2.2 Software-Aktivierung und Lizenzüberwachung ... 675
15.2.3 Altova Endbenutzer-Lizenzvereinbarung zu Authentic ... 676

Index 677

1 Authentic Desktop und diese Dokumentation

Altova Authentic 2023 Desktop ist ein innovativer visueller Lösungsansatz zur Erstellung von XML-Dokumenten, bei dem der Endbenutzer sich nicht mit den technischen Aspekten von XML befassen muss. Authentic Desktop läuft auf Windows 7 SP1 mit Plattformupdate, Windows 8, Windows 10, Windows 11 und Windows Server 2008 R2 SP1 mit Plattformupdate oder höher. Die Authentic Desktop Enterprise Edition steht für 64-Bit und 32-Bit-Systeme zur Verfügung.

Letzte Aktualisierung: 17.10.2022

© 2016-2022 Altova GmbH

1.1 Windows-Dateipfade

Dateipfade unter Windows

Die in dieser Dokumentation angegebenen Dateipfade sind nicht für alle Betriebssysteme gleich. Sie sollten die folgenden Übereinstimmungen beachten:

- Verzeichnis (Eigene) Dateien: Das Verzeichnis (Eigene) Dateien befindet sich standardmäßig an den nachstehend angeführten Orten. Die Beispieldateien befinden sich in einem Unterverzeichnis dieses Verzeichnisses.

Windows 7/8/10/11	`C:\Benutzer\<Benutzername>\Dokumente`

- *Anwendungsverzeichnis:* Das Anwendungsverzeichnis ist jener Ordner, in dem sich Ihre Altova Anwendung befindet. Der Pfad zum Anwendungsverzeichnis ist standardmäßig folgender:

Windows 7/8/10/11	`C:\Programme\Altova\`
32-Bit-Version auf 64-Bit OS	`C:\Programme (x86)\Altova\`

Anmerkung: Authentic Desktop wird auch auf Windows Server 2008 R2 SP1 mit Plattformupdate oder höher unterstützt.

1.2 Informationen zu dieser Dokumentation

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

- eine Einführung 13, die eine Übersicht über die Benutzeroberfläche und die Authentic Desktop-Umgebung enthält.
- ein Tutorial 24, um Sie mit Authentic Desktop vertraut zu machen.
- eine Beschreibung der Benutzeroberfläche der Authentic-Ansicht 41, einer WYSIWYG-Ansicht des XML-Dokuments. Diese Ansicht gestattet Benutzern, XML-Dokumente zu erstellen und zu editieren, als würde es sich um einfache Textdokumente oder interaktive Formulare handeln. Der XML Markup-Code wird dem Benutzer nicht angezeigt, sodass er sich auf den Inhalt des Dokuments konzentrieren kann. Die Authentic View-Ansicht ist die Hauptansicht von Authentic Desktop.
- eine Beschreibung der Browser-Ansicht 94, in der das XML-Dokument on-the-fly transformiert und in einem Browser-Fenster angezeigt wird.
- eine Erläuterung der Altova-Funktion Globale Ressourcen 96, über die Sie Ressourcen schnell wechseln können.
- eine Erklärung der Verwendung von Authentic Desktop in Visual Studio 157 und Eclipse 160.
- eine Referenz zu den Menübefehlen 167 mit Beschreibungen aller Fenster und Menübefehle, die in Authentic Desktop zur Verfügung stehen.

 © 2016-2022 Altova GmbH

2 Benutzeroberfläche und Umgebung

Dieser Abschnitt enthält eine Beschreibung :

- der Benutzeroberfläche der Applikation 14 und
- der Applikationsumgebung 22.

Der Abschnitt "Benutzeroberfläche" 14 beginnt mit einer Übersicht über die Benutzeroberfläche. Anschließend werden die einzelnen Fenster der Benutzeroberfläche näher beschrieben. Außerdem erfahren Sie, wie Sie die Größe der Fenster anpassen und die Fenster verschieben können und mit den Fenstern und der Benutzeroberfläche arbeiten.

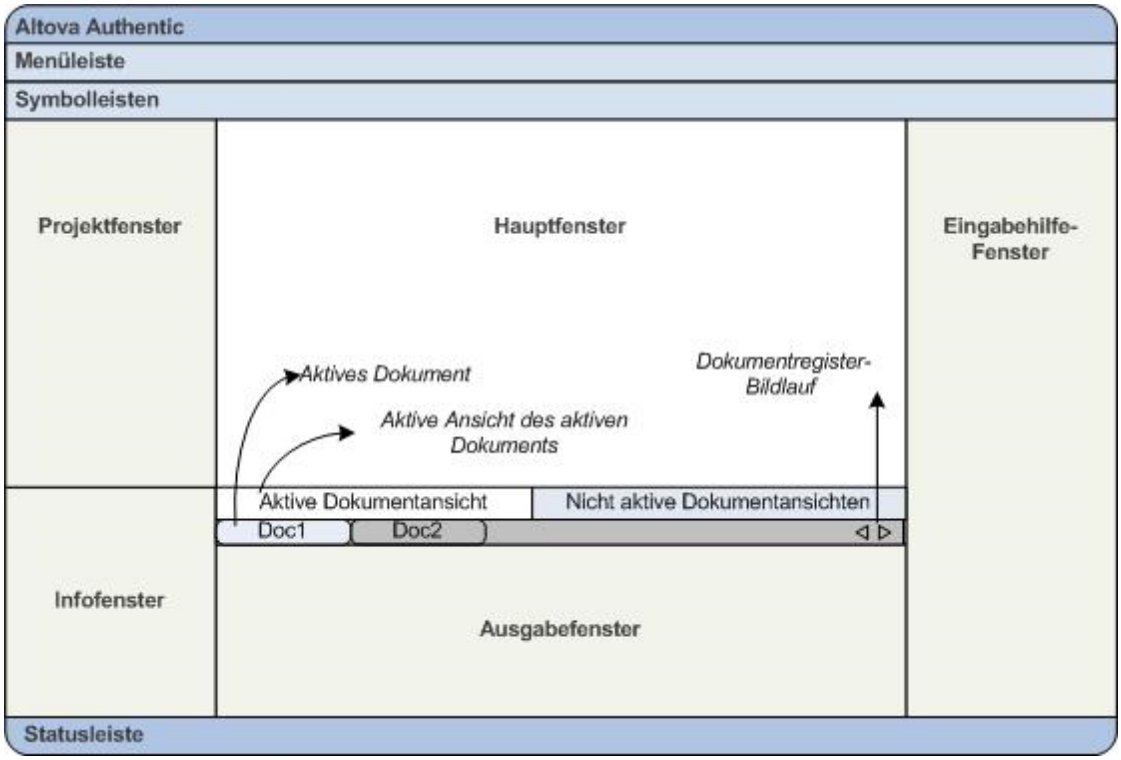
Im Abschnitt "Applikationsumgebung" 22 werden die verschiedenen Einstellungen erläutert, mit denen festgelegt wird, wie die Dateien angezeigt und bearbeitet werden können. Außerdem wird darin erklärt, wie und wo Sie Ihre Applikation anpassen können. In diesem Abschnitt erfahren Sie, wo wichtige Beispiel- und Tutorial-Dateien auf Ihrem Rechner installiert wurden. Weiter hinten in diesem Abschnitt finden Sie einen Link zur Altova Website, wo Sie eine Feature Matrix Ihrer Applikation finden, Ihr Benutzerhandbuch in verschiedenen Formaten abrufen können und Informationen über die verschiedenen Support-Optionen für Sie finden. Des Weiteren finden Sie dort Informationen über andere Produkte von Altova.

2.1 Die grafische Benutzeroberfläche

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

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

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



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

Ein- und Ausblenden der Seitenleisten

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

 © 2016-2022 Altova GmbH

Seitenleisten freischwebend anzeigen und andocken

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

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

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

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

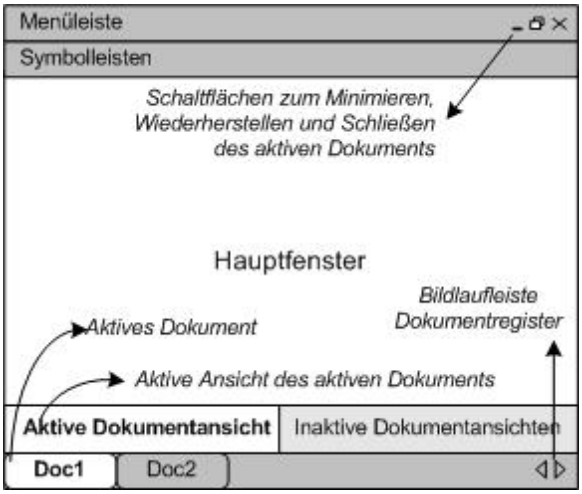
Automatisches Ausblenden der Seitenleisten

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

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

2.1.1 Hauptfenster

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



Dateien im Hauptfenster

- Sie können eine beliebige Anzahl an Dokumenten gleichzeitig geöffnet haben und bearbeiten.
- Jedes offene Dokument hat sein eigenes Fenster und ein Register mit seinem Namen am unteren Rand des Hauptfensters. Um ein Dokument zum aktiven zu machen, klicken Sie auf sein Register.
- Wenn mehrere Dateien geöffnet sind, werden manchmal auf Grund von Platzmangel in der Dokumentregisterleiste nicht alle Register angezeigt. Um Dokumentregister zu sehen, (i) verwenden Sie entweder die Scroll-Schaltflächen rechts unten in der Dokumentregisterleiste oder (ii) wählen Sie das gewünschte Dokument aus der Liste im unteren Bereich des Menüs Fenster[285] aus.
- Wenn das aktive Dokument maximiert ist, befinden sich die Schaltflächen zum **Minimieren, Wiederherstellen** und **Schließen** auf der rechten Seite der Menüleiste. Wenn ein Dokument überlappend, nebeneinander oder minimiert angezeigt wird, befinden sich die Schaltflächen zum **Maximieren, Wiederherstellen** und **Schließen** in der Titelleiste des Dokumentfensters.
- Wenn Sie eine Datei maximieren, werden auch alle anderen offenen Dateien maximiert.
- Um offene Dateien überlappend oder nebeneinander anzuzeigen, verwenden Sie die Befehle im Menü Fenster[285].
- Sie können offene Dateien auch durch Verwendung von **Strg+Tab** oder **Strg+F6** in der Reihenfolge aktivieren, in der sie geöffnet wurden.
- Durch Rechtsklick auf ein Dokumentregister wird ein Kontextmenü mit einer Auswahl an **Datei**-Befehlen wie z.B. **Drucken** und **Schließen** angezeigt.

Ansichten im Hauptfenster

Das aktive Dokument kann auf verschiedene Arten angezeigt und bearbeitet werden. Die verfügbaren Ansichten werden in einer Leiste oberhalb der Dokumentregister angezeigt (*siehe Abbildung oben*), wobei die aktive Ansicht markiert erscheint. Um eine Ansicht auszuwählen, klicken Sie auf die entsprechende Ansichtsschaltfläche oder rufen Sie im Menü Ansicht[239] den entsprechenden Befehl auf.

Verfügbar sind entweder Editier- oder Browser-Ansichten:

- Authentic-Ansicht[41]**:** Zum Bearbeiten von XML-Dokumenten in einer grafischen Ansicht auf Basis von StyleVision Power Stylesheets
- Browser-Ansicht[94]**:** Eine integrierte Browser-Ansicht, die sowohl CSS- als auch XSL-Stylesheets unterstützt.

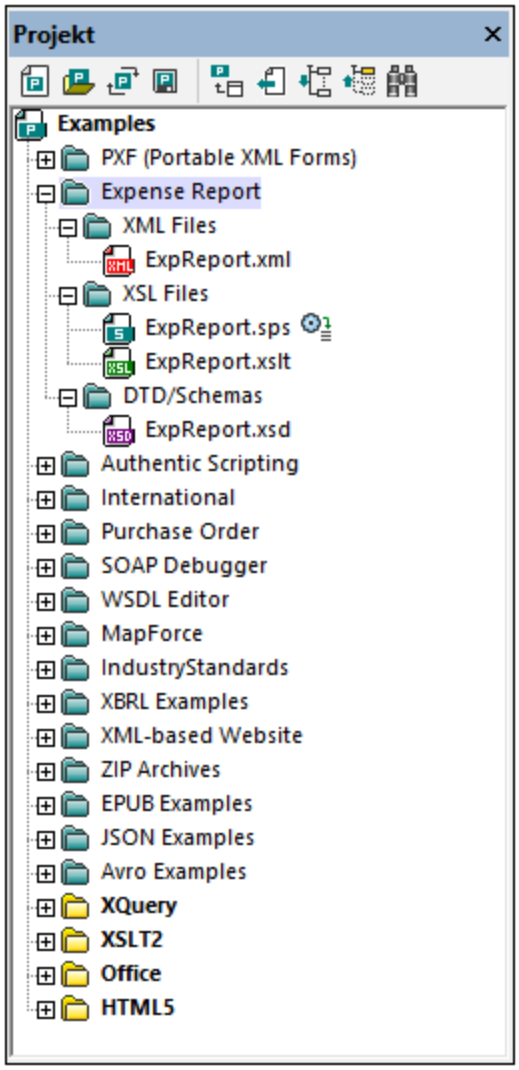
 © 2016-2022 Altova GmbH

Hinweis: Sie können die Standardansicht für bestimmte Dateierweiterungen im Dialogfeld **Extras | Optionen** 267 anpassen: Gehen Sie dazu auf dem Register "Dateiarten" zum Fenster "Standardansicht".

2.1.2 Projektfenster

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

Anmerkung: Im Projektfenster von Authentic Desktop wird standardmäßig das Projekt "Examples" geöffnet. Um das Standardprojekt "Examples" zu öffnen, gehen Sie *im Ordner (Eigene) Dokumente* 11 zum Ordner **`Examples`** der Applikation und doppelklicken Sie auf die Datei **`Examples.spp`**.



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

Projektoperationen

Die Befehle für Ordneroperationen stehen im Menü **Projekt** zur Verfügung. Einige Befehle stehen in den Kontextmenüs (Aufruf mittels Rechtsklick) des Projekts und seinen Ordnern zur Verfügung. Ein Teil der Befehle aus dem Menü **Projekt** steht aufgrund ihrer häufigen Verwendung auch in der Symbolleiste des Projektfensters zur Verfügung (*siehe Abbildung unten*).

Die Symbolleisten-Schaltflächen sind von links nach rechts: *Neues Projekt, Projekt öffnen, Projekt neu laden, Projekt speichern, Aktive Datei zu Projekt hinzufügen, Aktive Datei auswählen, Alle erweitern, Alle reduzieren, Suchen*. Die Namen dieser Befehle sind selbsterklärend und werden im Menü "Projekt" 187 erläutert.

In der Folge finden Sie eine Liste der wichtigsten Operationen im Zusammenhang mit dem Projektfenster.

- Im Projektfenster ist immer nur ein Projekt geöffnet. Wenn ein neues Projekt erstellt oder ein bestehendes geöffnet wird, wird das Projekt, das gerade im Projektfenster geöffnet ist, durch das neue ersetzt.
- Nachdem an einem Projekt Änderungen vorgenommen wurden, muss das Projekt (durch Auswahl des Befehls **Projekt | Projekt speichern)** gespeichert werden. Wenn ein Projekt nicht gespeicherte Änderungen enthält, so wird dies durch ein Sternchen neben dem Projektnamen angezeigt (*siehe Abbildung oben*).
- Das Projekt hat eine Baumstruktur bestehend aus Ordnern, Dateien und anderen Ressourcen. Diese können in jeder Ebene und bis zu einer beliebigen Tiefe hinzugefügt werden.
- Projektordner sind *semantische* Ordner, die eine logische Dateigruppierung darstellen und **nicht** der hierarchischen Organisation von Dateien auf Ihrer Festplatte entsprechen müssen.
- Ordner können physischen Verzeichnissen in Ihrem Dateisystem entsprechen und eine direkte Beziehung zu diesen haben. Solche Ordner werden *externe Ordner* genannt und im Projektfenster durch ein gelbes Ordnersymbol gekennzeichnet (im Gegensatz zu normalen Projektordnern, die grün angezeigt werden). Externe Projektordner müssen explizit mit dem Befehl **Aktualisieren** synchronisiert werden.
- Ein Ordner kann eine beliebige Mischung von Dateiarten enthalten. Alternativ dazu können Sie (im Dialogfeld "Eigenschaften" für diesen Ordner) für jeden Ordner Dateierweiterungen definieren, damit bestimmte Dateiarten jederzeit über einen Ordner aufgerufen werden können. Wenn eine Datei zum übergeordneten Ordner hinzugefügt wird, wird sie automatisch zu dem Unterordner hinzugefügt, für den diese Dateierweiterung definiert wurde.
- Wenn Sie die Maus über eine Bilddatei im Projektordner platzieren, wird eine Vorschau des Bilds (Formate `.png`, `.jpg`, `.gif`, `.bmp`, `.tiff` und `.ico`). angezeigt. Doppelklicken Sie auf das Bild, um es im Standard Bildansichts-/-bearbeitungsprogramm des Systems zu öffnen.
- Im Projektordner kann ein Ordner in einen anderen Ordner oder an eine andere Stelle im selben Ordner gezogen werden; eine Datei kann in einen anderen Ordner gezogen werden, nicht aber innerhalb desselben Ordners (in dem die Dateien alphabetisch geordnet sind) verschoben werden. Dateien und Ordner können auch aus dem Windows Datei-Explorer in die Projekt-Seitenleiste gezogen werden.
- Jeder Ordner hat bestimmte Eigenschaften, die im Dialogfeld "Eigenschaften" dieses Ordners definiert werden. Dazu gehören die Dateierweiterungen für diesen Ordner, das Schema, mit dem die XML-Dateien validiert werden, die XSLT-Datei, mit der die XML-Dateien transformiert werden sollen usw.

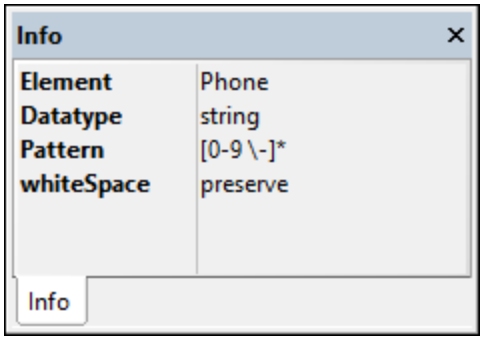
 © 2016-2022 Altova GmbH

- Die Batch-Verarbeitung von Dateien in einem Ordner erfolgt durch Rechtsklick auf den Ordner und Auswahl des entsprechenden Befehls aus dem Kontextmenü (z.B. **XML validieren** oder **Wohlgeformtheit prüfen**).

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

2.1.3 Info-Fenster

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



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

2.1.4 Eingabehilfen

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

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

2.1.5 Ausgabefenster "Meldungen"

Im Fenster "Meldungen" werden Meldungen über in Authentic Desktop ausgeführte Aktionen sowie Fehlermeldungen und andere Ausgabemeldungen angezeigt. Wenn z.B. ein XML-Dokument validiert wird und gültig ist, wird im Meldungsfenster eine entsprechende Meldung angezeigt. Andernfalls wird eine Meldung, die den Fehler beschreibt, angezeigt. Beachten Sie, dass es Links (schwarzen Text) zu Nodes und Node-Inhalt im XML-Dokument gibt sowie Links zum entsprechenden Abschnitt in der Spezifikation im Internet, in dem die betreffende Regel beschrieben wird (blauer Text).

Validieren von Ordnern und Dateien im Projektfenster

Der Befehl **Validieren** (im Menü XML) wird normalerweise auf das aktive Dokument angewendet. Sie können den Befehl jedoch auch auf eine Datei oder eine Gruppe von Dateien im aktiven Projekt anwenden. Wählen Sie die gewünschte Datei bzw. den gewünschten Ordner im Projekt-Fenster (durch Anklicken) aus und klicken Sie auf **XML | Validieren** 219 oder **F8**. Ungültige Dateien in einem Projekt werden geöffnet und im Hauptfenster aktiviert und die Fehlermeldung "*Die Datei ist nicht gültig*" wird angezeigt.

Anmerkung: Sie können die Wohlgeformtheitsprüfung auch im Projektfenster durchführen (**Wohlgeformtheit prüfen** 219 oder **F7**).

2.1.6 Menüleiste, Symbolleisten und Statusleiste

Menüleiste

Die Menüleiste (*siehe Abbildung* 14) enthält die verschiedenen Menüs der Applikation. Es gelten die folgenden Konventionen:

- Wenn Befehle in einem Menü in einer Ansicht oder an einer bestimmten Stelle im Dokument **nicht** zur Verfügung stehen, sind sie ausgegraut und deaktiviert.
- Bei Auswahl einiger Menübefehle erscheint ein Untermenü mit einer Liste zusätzlicher Optionen. Solche Menübefehle weisen rechts neben dem Befehlsnamen einen nach rechts weisenden Pfeil auf.
- Bei einigen Menübefehlen wird ein Dialogfeld aufgerufen, in dem Sie weitere Informationen eingeben müssen, damit der Befehl ausgeführt werden kann. Diese Befehle sind durch drei Auslassungspunkte (...) nach dem Namen des Befehls gekennzeichnet.
- Um einen Menübefehl aufzurufen, klicken Sie auf den Menünamen und anschließend auf den Befehl. Enthält ein Menüeintrag ein Untermenü, wird dieses geöffnet, wenn Sie die Maus über den Menüeintrag positionieren. Klicken Sie auf den gewünschten Untermenüeintrag.
- Sie können ein Menü auch durch Drücken der entsprechenden Tastenkombination öffnen. Die Tastenkombination für die einzelnen Menüs lautet **Alt+TASTE**, wobei TASTE für den unterstrichenen Buchstaben im Menünamen steht. Die Tastenkombination für das Menü **Datei** ist z.B. **Alt+a**.
- Die Auswahl eines Menübefehls (also eines Befehls in einem Menü) erfolgt durch das nacheinander-Auswählen (i) des Menüs durch die entsprechende Tastenkombination (siehe voriger Punkt) und (ii) anschließendes Drücken der Tastenkombination für den jeweiligen Befehl (**Alt+TASTE**, wobei TASTE dem unterstrichenen Buchstaben im Befehlsnamen entspricht). Um z.B. eine neue Datei zu erstellen, (**Datei | Neu**), drücken Sie **Alt+a** und dann **Alt+N**.
- Einige Menübefehle können **direkt** durch Drücken eines speziellen **Tastenkürzels** oder einer Tastenkombination (**Strg+TASTE**) aufgerufen werden. Bei Befehlen, denen ein Tastenkürzel zugewiesen wurde, wird rechts vom Befehl das Tastenkürzel oder die Tastenkombination angezeigt. So können Sie z.B. eine neue Datei mit der Tastenkombination **Strg+N** erstellen und eine XML-Datei durch Drücken von **F8** validieren. Im Dialogfeld "Anpassen" (**Extras | Anpassen**) können Sie auf dem Register "Tastatur" Ihre eigenen Tastenkürzel erstellen 255.

Symbolleisten

Die Symbolleisten (*siehe Abbildung* 14) enthalten Schaltflächen zum Aufrufen von Menübefehlen. Wenn Sie den Mauszeiger über die Schaltfläche halten, wird der Name des Befehls angezeigt. Um den Befehl auszuführen, klicken Sie auf die Schaltfläche.

 © 2016-2022 Altova GmbH

Die Symbolleisten-Schaltflächen sind in Gruppen angeordnet. Im Dialogfeld **Extras | Anpassen | Symbolleisten** 252 können Sie einstellen, welche Symbolleistengruppen angezeigt werden sollen. Diese Einstellungen gelten für die aktuelle Ansicht. Um Einstellungen für eine andere Ansicht vorzunehmen, wechseln Sie zu dieser Ansicht und nehmen Sie anschließend unter **Extras | Anpassen | Symbolleisten** 252 Ihre Einstellungen vor. Sie können Symbolleistengruppen auf der GUI auch an eine andere Stelle ziehen, indem Sie darauf klicken und mit dem Ziehpunkt (oder an der Titelleiste) an die gewünschte Stelle verschieben. Wenn Sie auf den Ziehpunkt doppelklicken, wird die Symbolleiste abgedockt und frei schwebend angezeigt; wenn Sie auf die Titelleiste doppelklicken, wird die Symbolleiste an der Stelle angedockt, an der sie sich vorher befunden hat.

Statusleiste

Die Statusleiste befindet sich am unteren Rand des Applikationsfensters (*siehe Abbildung* 14). In ihr werden die folgenden Informationen angezeigt: (i) Statusinformationen über das Laden von Dateien und (ii) Informationen über Menübefehle und Befehlsschaltflächen in den Symbolleisten, wenn Sie den Cursor darüber platzieren. Wenn Sie die 64-Bit-Version von%APPNAME%> verwenden, wird in der Statusleiste das Suffix (x64) hinter dem Applikationsnamen angezeigt. Für die 32-Bit-Version gibt es kein Suffix.

2.2 Die Applikationsumgebung

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

Dieser Abschnitt ist folgendermaßen gegliedert:

- *Einstellungen und Anpassung* 22*:* Hier wird beschrieben wie und wo wichtige Einstellungen und anpassbare Optionen definiert werden können.
- *Tutorials, Projekte, Beispiele:* 22 Hier finden Sie die Pfade der verschiedenen im Applikationspaket enthaltenen Dateien, die nicht Teil des Programms selbst sind.
- *Produktfeatures, Dokumentation und Altova-Produkte* 23*:* Hier finden Sie Links zur Altova Website, wo Sie Informationen über Produktfeatures, ein Hilfe in zusätzlichen Formaten und andere Altova-Produkte finden.

2.2.1 Einstellungen und Anpassung

Dieser Abschnitt enthält eine kurze Übersicht darüber, wie Sie Authentic Desktop personalisieren können.

Einstellungen
Auf den verschiedenen Registern im Dialogfeld "Optionen" können verschiedene wichtige Einstellungen von Authentic Desktop konfiguriert werden. Gehen Sie die verschiedenen Optionen selbst durch, um zu sehen, welche Optionen zur Verfügung stehen.

Anpassung
Sie können verschiedene Aspekte von Authentic Desktop anpassen, wie unter anderem auch das Aussehen der Benutzeroberfläche. Die entsprechenden Optionen dazu finden Sie im Dialogfeld "Anpassen" (Aufruf über den Menübefehl **Extras | Anpassen** 250). Die verschiedenen Konfigurationsoptionen sind im Abschnitt **Menübefehle** 167 beschrieben.

2.2.2 Tutorials, Projekte, Beispiele

Das Authentic Desktop Installationspaket enthält Tutorials, Projekte und Beispieldateien.

Pfade von Tutorials, Projekten und Beispieldateien

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

```
C:\Benutzer\<Benutzername>\Dokumente\Altova\Authentic2023\AuthenticExamples\
```

Der Ordner `My Documents\Altova\Authentic2023` wird für jeden Benutzer, der auf dem PC einen Account hat, im entsprechenden Ordner dieses Benutzers unter `<username>` installiert. Bei diesem Installationssystem

 © 2016-2022 Altova GmbH

stehen jedem Benutzer seine eigenen `AuthenticExamples`-Ordner in einem eigenen Arbeitsbereich zur Verfügung.

Pfade der Tutorial-, Projekt- und Beispieldateien

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

2.2.3 Authentic Desktop Features, Hilfe und Altova-Produkte

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

Liste der Authentic Desktop-Features

Die Altova-Website bietet eine Liste der Authentic Desktop Features.

Authentic Desktop Hilfe

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

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

Support-Optionen

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

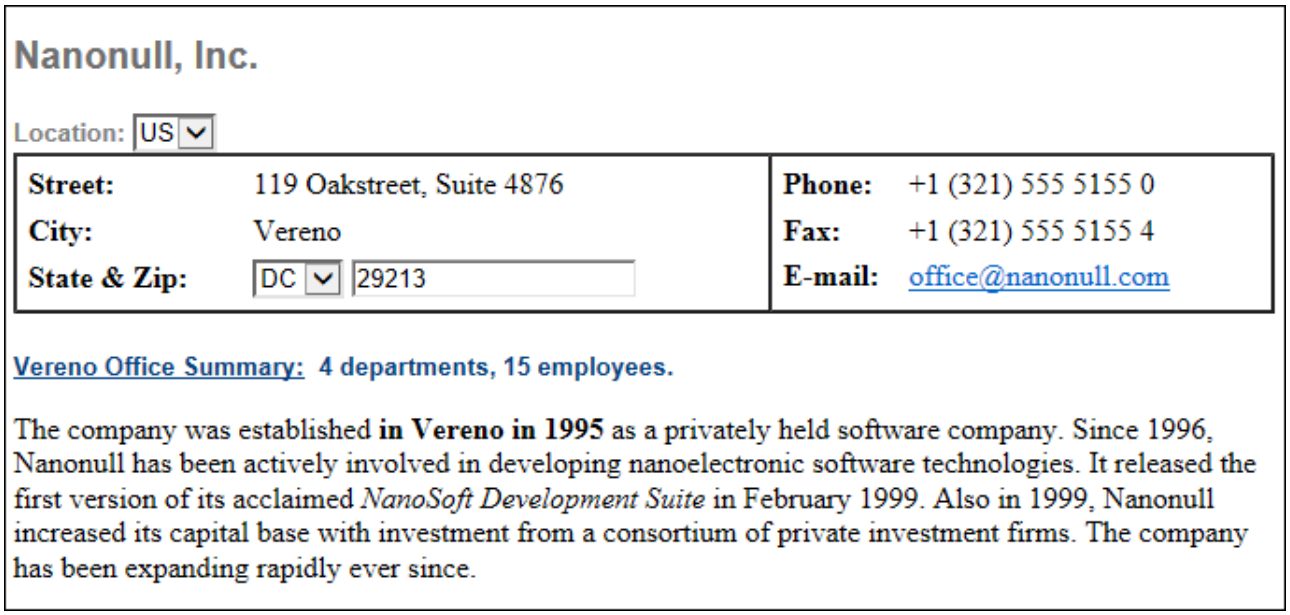
- Links zu unseren FAQ-Seiten
- Diskussionsforen zu Altova-Produkten und allgemeinen XML-Themen
- Online Support-Formulare, über die Sie Anfragen an den Support senden können, falls Sie über ein Support- und Wartungspaket verfügen. Unser Support-Team wird sich um Ihre Anfrage kümmern.

Altova-Produkte

Eine Liste aller Altova-Produkte finden Sie auf der Altova Website.

3 Tutorial Authentic-Ansicht

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



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

Dieses Tutorial enthält eine schrittweise Anleitung zu Folgendem:

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

Denken Sie daran, dass dieses Tutorial als Einführung gedacht ist und daher nicht auf komplizierte Aspekte eingeht. Zusätzliches Referenzmaterial und nähere Informationen zu einzelnen Funktionen finden Sie in den Abschnitten zur Oberfläche der Authentic View-Ansicht 41.

Voraussetzungen für das Tutorial

Sie finden alle für das Tutorial benötigten **Dateien** im Ordner `AuthenticExamples` des Altova-Applikationsordners. Es sind die Dateien:

 © 2016-2022 Altova GmbH

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

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

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

3.1 Öffnen eines XML-Dokuments in der Authentic View-Ansicht

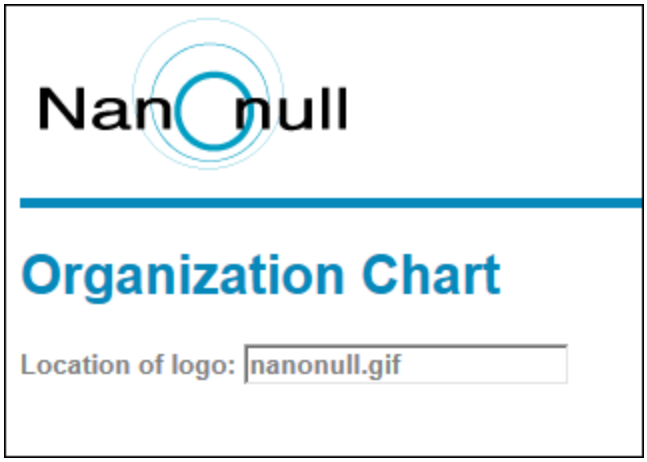
Sie können in der Authentic View-Ansicht ein vorhandenes XML-Dokument bearbeiten oder ein neues XML-Dokument erstellen. In diesem Tutorial werden Sie ein vorhandenes XML-Dokument in der Authentic View-Ansicht öffnen (wie in diesem Abschnitt beschrieben) und erfahren, wie Sie es bearbeiten können (Beschreibung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

Öffnen eines vorhandenen XML-Dokuments

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

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

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



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

Erstellen eines neuen XML-Dokuments auf Basis eines SPS

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

Über Datei | Neu

1. Klicken Sie auf **Datei | Neu**.
2. Wählen Sie im Dialogfeld "Neues Dokument erstellen" das gewünschte SPS aus.

 © 2016-2022 Altova GmbH

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

Über Authentic | Neues Dokument

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

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

3.2 Die Benutzeroberfläche der Authentic-Ansicht

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

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

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

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

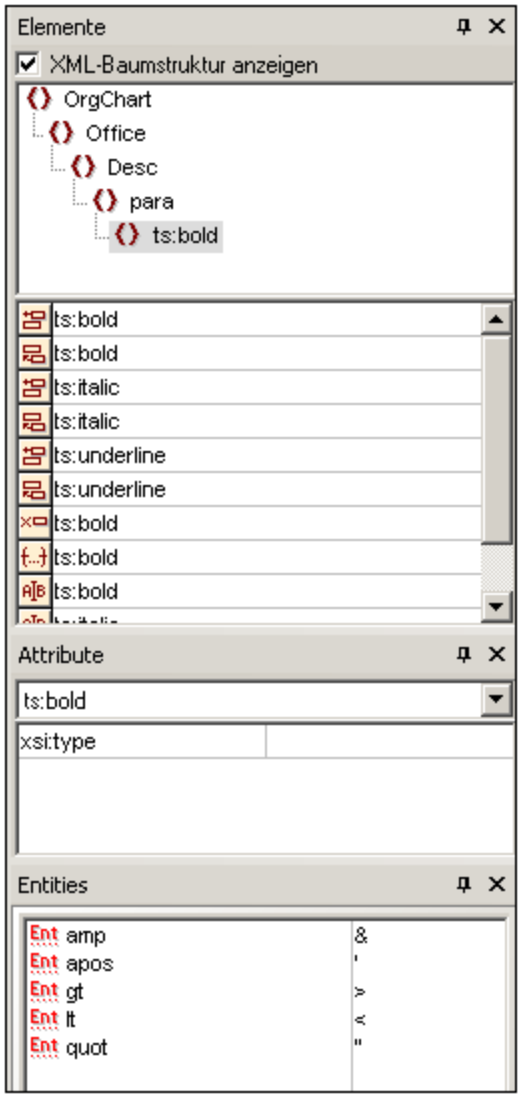
Standardmäßig werden die Node-Tags in der Authentic View-Ansicht nicht angezeigt. Sie können sie durch Auswahl des Menübefehls **Authentic | Große Markup-Symbole einblenden** (oder der Symbolleisten-Schaltfläche) einblenden. Große Markup-Tags enthalten den Namen des jeweiligen Node. Alternativ dazu können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

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

Eingabehilfen

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

 © 2016-2022 Altova GmbH



Element-Eingabehilfe

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

Attribute-Eingabehilfe

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

werden die Attribute dieses Elements in der Eingabehilfe angezeigt, wo sie anschließend bearbeitet werden können.

Entities-Eingabehilfe
Die Entities-Eingabehilfe ist nicht kontextsensitiv. Hier werden alle im Dokument deklarierten Entities angezeigt. Wenn Sie auf eine Entity doppelklicken, wird sie an der Cursorposition eingefügt. Eine Beschreibung, wie man Entities für ein Dokument hinzufügt, finden Sie im Abschnitt Benutzeroberfläche der Authentic-Ansicht 41.

Kontextmenü

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

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

Anmerkung: Nähere Informationen über die Benutzeroberfläche finden Sie unter Benutzeroberfläche der Authentic-Ansicht 41.

 © 2016-2022 Altova GmbH

3.3 Node-Operationen

Es gibt in der Authentic View-Ansicht von XML-Dokumenten zwei grundlegende Node-Typen: **Element-Nodes** und **Attribut-Nodes**. Diese Nodes sind alle mit Tags versehen, die Sie einblenden können 28. Es gibt auch andere Nodes im Dokument, wie z.B. Text-Nodes (die mit keinen Markup-Tags versehen sind) und CDATA-Abschnitt-Nodes (welche mit Markup versehen sind, um sie vom umgebenden Text abzugrenzen).

Die in diesem Abschnitt beschriebenen Node-Operationen beziehen sich nur auf Element-Nodes und Attribut-Nodes. Beim Ausprobieren der hier beschriebenen Operationen sollten Sie am besten die großen Markup-Symbole einblenden 28

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

Durchführen von Node-Operationen

Node-Operationen können entweder durch Auswahl eines Befehls im Kontextmenü 30 oder durch Klicken auf den Eintrag für die Node-Operation in der Element-Eingabehilfe 28 durchgeführt werden. In einige Fällen kann ein Element oder Attribut hinzugefügt werden, indem Sie in der Authentic View-Ansicht des Dokuments auf den Link Node hinzufügen 32 klicken. In speziellen Fällen, in denen Elemente als Absätze oder Listeneinträge definiert wurden, können Sie durch Drücken der Eingabetaste 32 ein neues gleichrangiges Element derselben Art erstellen, wenn sich der Cursor in einem derartigen Element befindet. In diesem Abschnitt erfahren Sie außerdem, wie Nodes über die Funktionen Element anwenden 32, Node entfernen 33 und Element löschen 33 erstellt und gelöscht werden können.

Einfügen von Elementen

Elemente können an den folgenden Stellen eingefügt werden:

- an der Cursorposition innerhalb eines Element-Node. Die an dieser Stelle verfügbaren Elemente werden im Untermenü des Kontextmenüs **Einfügen** aufgelistet. In der Element-Eingabehilfe werden Elemente, die an dieser Stelle eingefügt werden können, durch ein Symbol markiert. Platzieren Sie im Dokument `NanonullOrg.xml` den Cursor in das Element `para` und erstellen Sie über das Kontextmenü und die Element-Eingabehilfe die Elemente `bold` und `italic`.
- vor oder nach dem ausgewählten Element oder einem seiner Vorfahren, wenn dies laut Schema gestattet ist. Wählen Sie das erforderliche Element aus dem bzw. den angezeigten Untermenü(s) aus. In der Element-Eingabehilfe werden Elemente, die vor oder nach dem ausgewählten Element eingefügt werden können, durch die Symbole bzw. gekennzeichnet. Beachten Sie, dass Sie in der Element-Eingabehilfe Elemente nur vor/nach dem ausgewählten Element einfügen können, nicht aber vor/nach dem Vorfahrenelement. Probieren Sie diesen Befehl aus, indem Sie den Cursor zuerst in das Element `para` und anschließend in die Tabelle mit der Liste der Angestellten platzieren.

Link "Node hinzufügen"

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

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

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

Erstellen neuer Elemente mit der Eingabetaste

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

Anwenden von Elementen

In Elementen mit gemischtem Inhalt (denjenigen, die sowohl Text als auch Child-Elemente enthalten) können Sie Textinhalt markieren und darauf eines der zulässigen Child-Elemente anwenden. Der ausgewählte Text wird zum Inhalt des angewendeten Elements. Um Elemente anzuwenden, wählen Sie im Kontextmenü den Befehl "Anwenden" und wählen Sie anschließend eines der vorgeschlagenen Elemente aus. (Wenn auf den ausgewählten Text keine Elemente angewendet werden können, wird der Befehl "Anwenden" im Kontextmenü nicht angezeigt.) In der Element-Eingabehilfe werden Elemente, die für eine Auswahl angewendet werden können, durch das Symbol gekennzeichnet. Wählen Sie im Dokument `NanonullOrg.xml` Text im Element gemischten Inhalts `para` aus und probieren Sie aus, wie sich die Elemente `bold` und `italic` bei Anwendung auswirken.

 © 2016-2022 Altova GmbH

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

Entfernen von Nodes

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

Löschen von Elementen

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

Tabellen und Tabellenstruktur

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

- *SPS-Tabellen (statische und dynamische)*. Die allgemeine Struktur von SPS-Tabellen wird durch den Stylesheet Designer festgelegt. Innerhalb dieser grundlegenden Struktur sind die einzigen zulässigen Änderungen an der Struktur inhaltsbezogene. So können Sie z. B. neue Zeilen zu einer dynamischen SPS-Tabelle hinzufügen.
- *XML-Tabellen*, in welchen Sie entscheiden, dass der Inhalt eines bestimmten Node (z.B. eines Node für personenspezifische Daten) als Tabelle dargestellt werden soll. Wenn der Stylesheet Designer die Erstellung dieses Node als XML-Tabelle ermöglicht hat, können Sie die Struktur der Tabelle festlegen sowie ihren Inhalt bearbeiten. XML-Tabellen werden im Abschnitt Verwendung von Tabellen in der Authentic View-Ansicht 66 näher beschrieben.

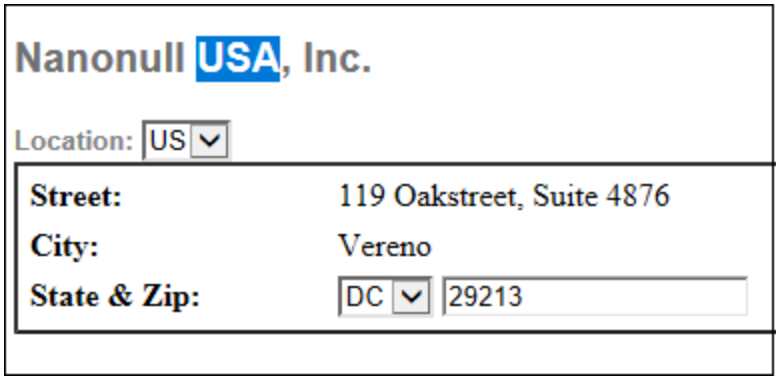
3.4 Eingabe von Daten in der Authentic-Ansicht

Daten werden im XML-Dokument direkt im Hauptfenster der Authentic View-Ansicht eingegeben. Zusätzlich dazu können Daten für Attribute (also der Attributwert) über die Attribute-Eingabehilfe eingegeben werden 37 .Daten werden entweder (i) direkt als Text oder (ii) durch Auswahl einer Option in einem Dateneingabeelement, eingegeben, das anschließend auf einen vordefinierten Texteintrag gemappt wird.

Hinzufügen von Textinhalt

Sie können Elementinhalt und Attributwerte im Hauptfenster der Authentic View-Ansicht direkt als Text eingeben. Platzieren Sie dazu den Cursor an die gewünschte Stelle und geben Sie den Text ein. Sie können Text auch aus der Zwischenablage in das Dokument kopieren. Inhalt kann auch mit den normalen Inhaltsbearbeitungsmechanismen wie **Feststelltaste** und **Entfernen** bearbeitet werden. Zum Beispiel können Sie den zu bearbeitenden Text markieren und den Ersetzungstext mit gedrückter **Feststelltaste** eingeben.

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



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

Einfügen von Sonderzeichen und Entities

Bei der Eingabe von Daten werden folgende Inhaltsarten speziell behandelt:

- *Sonderzeichen, die als XML-Markup-Symbole verwendet werden* (kaufmännisches Und, Apostroph, Größer als, Kleiner als und Anführungszeichen). Diese Zeichen stehen als vordefinierte Entities 38 zur Verfügung und können durch Doppelklick auf die entsprechende Entity in der Entities-Eingabehilfe im Dokument eingefügt werden. Wenn diese Zeichen häufig vorkommen (z.B. in Programmcodelisten), können Sie innerhalb von CDATA-Abschnitten eingegeben werden. Klicken Sie zum Einfügen eines CDATA-Abschnitts mit der rechten Maustaste an die gewünschte Stelle und wählen Sie im Kontextmenü den Befehl **CDATA-Abschnitt einfügen**. Der XML-Prozessor ignoriert alle Markup-Zeichen innerhalb von CDATA-Abschnitten. Dies bedeutet auch, dass Sonderzeichen innerhalb von CDATA-Abschnitten als das Zeichen selbst und nicht die Entity-Referenz eingegeben werden müssen.
- *Sonderzeichen, die nicht über die Tastatur eingegeben werden können,* sollten durch Kopieren des Zeichens aus der Zeichentabelle Ihres Systems an die gewünschte Stelle im Dokument eingegeben werden.

 © 2016-2022 Altova GmbH

- *Ein häufig verwendeter Textstring* kann als Entity definiert werden 84, die in der Entities-Eingabehilfe angezeigt wird. Die Entity wird an der gewünschten Stelle eingefügt 38. Platzieren Sie dazu den Cursor an die gewünschte Stelle und doppelklicken Sie in der Entities-Eingabehilfe auf die Entity. Diese Funktion eignet sich dazu, um Dokumente auf aktuellem Stand zu halten, da der Wert des Textstrings nur einmal gespeichert ist. Wenn der Wert geändert werden muss, müssen Sie nur die Entity-Definition ändern.

Anmerkung: Wenn Markup-Tags in der Authentic View -Ansicht ausgeblendet werden, werden leere Elemente leicht übersehen. Um sicherzustellen, dass Sie kein leeres Element übersehen haben aktivieren Sie ein der Optionen große Markup-Symbole (oder kleine Markup-Symbole) anzeigen. 28

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

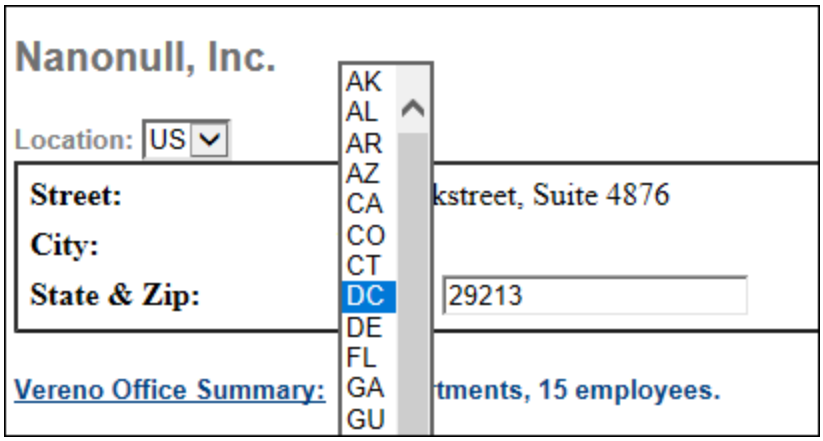
Hinzufügen von Inhalt über ein Dateneingabe-Steuerelement

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

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

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

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



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

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

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

© 2016-2022 Altova GmbH

3.5 Eingabe von Attributwerten

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

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

Attributwerte im Hauptfenster

Attributwerte können als normaler Text oder als Text in einem Eingabefeld eingegeben werden oder als vom Benutzer aus einer Auswahl gewählter Eintrag, der auf einen XML-Wert gemappt wird. Attributwerte werden auf dieselbe Art eingegeben, wie Element-Inhalt: siehe Eingabe von Daten in der Authentic View-Ansicht 34. In solchen Fällen erfolgt die Unterscheidung zwischen Element-Inhalt und Attributwerten durch das StyleVision Power Stylesheet und die Daten werden entsprechend behandelt.

Attributwerte in der Attribut-Eingabehilfe

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

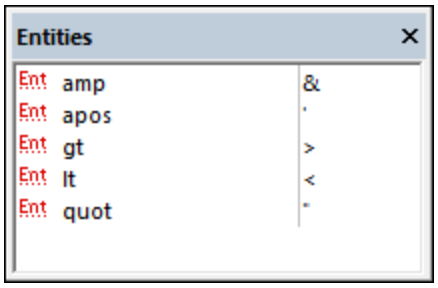
1. Wählen Sie das Element `CompanyLogo` aus, indem Sie auf den Tag CompanyLogo klicken. Die Attribute des Elements `CompanyLogo` werden in der Attribut-Eingabehilfe angezeigt.
2. Ändern Sie in der Attribut-Eingabehilfe den Wert des Attributs `href` von `nanonull.gif` in `Altova_right_300.gif` (ein Bild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

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

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

3.6 Hinzufügen von Entities

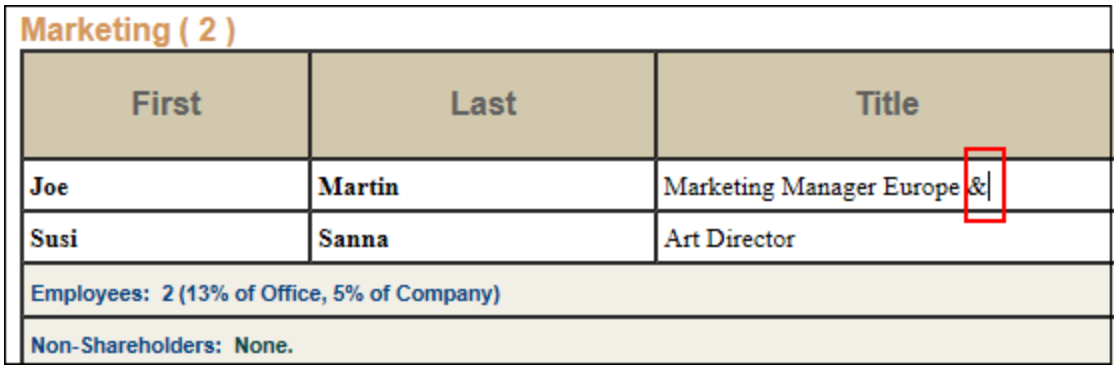
Bei einer Entity in der Authentic View-Ansicht handelt es sich normalerweise (nicht notwendigerweise) um XML-Daten wie z.B. ein einzelnes Zeichen, einen Textstring oder auch ein Fragment eines XML-Dokuments. Auch eine Binärdatei wie z.B. eine Bilddatei kann eine Entity sein. Alle verfügbaren Entities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



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

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

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



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

 © 2016-2022 Altova GmbH

Definieren Ihrer eigenen Entities

Als Dokument-Bearbeiter können Sie auch Ihre eigenen Entities definieren. Eine Beschreibung dazu finden Sie im Abschnitt Definieren von Entities in der Authentic View-Ansicht 84.

3.7 Drucken des Dokuments

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

So drucken Sie die Datei `NanonullOrg.xml`:

1. Wechseln Sie in den Modus "Markup-Symbole ausblenden", wenn Sie gerade einen anderen Modus verwenden, da sonst auch die Markup-Symbole gedruckt werden.
2. Wählen Sie **Datei | Druckvorschau**, um eine Vorschau aller Seiten anzuzeigen. In der Abbildung unten sehen Sie einen Ausschnitt einer um 50 % verkleinerten Druckvorschauseite. Beachten Sie, dass die Formatierung auf der Seite dieselbe ist, wie in der Authentic View-Ansicht.

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

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

 © 2016-2022 Altova GmbH

4 Benutzeroberfläche der Authentic-Ansicht

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

Dieser Abschnitt enthält:

- eine Übersicht über die Benutzeroberfläche
- eine Beschreibung der Symbolleistensymbole der Authentic View-Ansicht
- eine Beschreibung der Ansichtsmodi im FensterAuthentic View-Ansicht-
- eine Beschreibung der Eingabehilfen und deren Verwendung
- eine Beschreibung der verschiedenen in der Authentic View-Ansicht verfügbaren Kontextmenüs

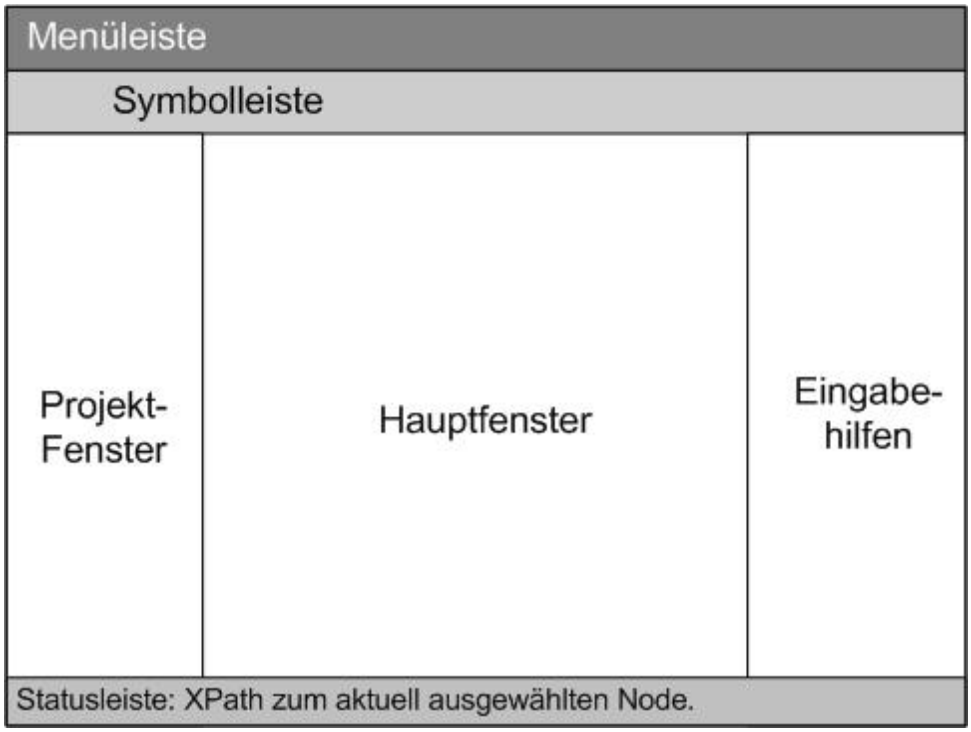
Weitere Informationen zur Authentic View-Ansicht finden Sie:

- im Tutorial zur Authentic View-Ansicht, in dem beschrieben wird, wie Sie mit der Authentic View-Ansicht arbeiten. Sie finden das Tutorial in der Dokumentation zu Altova <%SPY-GEN%> und zur Altova Authentic Desktop (siehe Abschnitt Tutorials) sowie online.
- Eine ausführliche Beschreibung der Menübefehle der Authentic View-Ansicht finden Sie im Abschnitt "Benutzerreferenz" Ihrer Produktdokumentation.

Altova Website: XML-Content-Bearbeitung, Erstellung von XML-Dokumenten

4.1 Übersicht über die Benutzeroberfläche

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



Menüleiste

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

Symbolleiste

Eine Beschreibung der Schaltflächen der Symbolleiste finden Sie im Abschnitt Symbolleistenschaltflächen der Authentic View-Ansicht 44.

Projekt-Fenster

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

Info-Fenster

Dieses Fenster enthält Informationen über den gerade in der Authentic-Ansicht ausgewählten Node.

 © 2016-2022 Altova GmbH

Hauptfenster

Das Hauptfenster ist der Bereich, in dem das XML-Dokument angezeigt und bearbeitet wird. Eine Beschreibung dazu finden Sie im Abschnitt Authentic View-Ansicht Hauptfenster 47.

Eingabehilfen

Dieser Bereich enthält drei Eingabehilfen-Fenster: Elemente, Attribute und Entities. Welche Einträge in diesen Fenstern (Element- und Attribute-Eingabehilfen) zu sehen sind, ist vom Kontext abhängig, d.h. davon, an welcher Stelle im Dokument sich der Cursor gerade befindet. Um ein Element oder eine Entity in das Dokument einzugeben, doppelklicken Sie in der Eingabehilfe darauf. Der Wert eines Attributs wird in der Attribute-Eingabehilfe im Feld "Wert" des entsprechenden Attributs eingegeben. Nähere Informationen dazu finden Sie im Abschnitt Authentic View-Ansicht Eingabehilfen 50.

Statusleiste

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

Kontextmenüs

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

4.2 Authentic-Ansicht Symbolleistenschaltflächen

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

Ein- und Ausblenden von XML Markup

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

	Markup-Symbole ausblenden. Alle XML-Tags mit Ausnahme der reduzierten Tags werden ausgeblendet. Wenn Sie im Modus "Markup-Symbole ausblenden" auf einen Tag, der reduziert wurde, doppelklicken (wodurch er normalerweise erweitert wird), wird der Inhalt des Nodes angezeigt und die Tags ausgeblendet.
	Kleine Markup-Symbole einblenden. Die XML-Element/Attribut-Tags werden ohne Namen angezeigt.
	Große Markup-Symbole einblenden. Die XML-Elemente/Attribute werden mit ihren Namen angezeigt.
	Gemischte Markup-Symbole einblenden. Im StyleVision Power Stylesheet können Sie für jedes XML-Element oder Attribut einzeln festlegen, wie es im angezeigt werden soll (mit großen oder kleinen Markup-Symbolen oder gar nicht). Der Modus heißt "Gemischte Markup-Symbole", weil darin für einige Elemente mit Markup-Symbolen angezeigt werden, andere dagegen ohne. Im gemischten Markup-Modus der Authentic View-Ansicht ist die Anzeige der Markup-Symbole also benutzerdefiniert. Beachten Sie allerdings, dass diese Einstellungen von der Person vorgenommen werden, die das StyleVision Power Stylesheet erstellt hat. Sie kann nicht vom Benutzer der Authentic View-Ansicht vorgenommen werden.

Bearbeiten von dynamischen Tabellenstrukturen

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

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

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

 © 2016-2022 Altova GmbH

	Zeile anhängen
	Zeile in Tabelle einfügen
	Aktuelle Tabellenzeile kopieren (d.h. der Inhalt einer Zelle wird dupliziert)
	Aktuellen Zeile um eine Zeile nach oben verschieben
	Aktuellen Zeile um eine Zeile nach unten verschieben
	Aktuelle Zeile löschen

Anmerkung: Diese Befehle sollten nur bei **dynamischen SPS-Tabellen** und nicht innerhalb von statischen SPS-Tabellen verwendet werden. Die verschiedenen Arten der in der Authentic View-Ansicht verwendeten Tabellen werden weiter hinten im Abschnitt Tabellen in der Authentic-Ansicht 66 näher beschrieben.

Erstellen und Bearbeiten von XML-Tabellen

Sie können Ihre eigenen Tabellen einfügen, wenn Sie Daten in Tabellenform darstellen möchten. Solche Tabellen werden als XML-Tabellen eingefügt. Sie können die Struktur einer XML-Tabelle bearbeiten und die Tabelle formatieren. Die Symbole dazu stehen in der Symbolleiste zur Verfügung (siehe Abbildung unten). Im Abschnitt XML-Tabellenbearbeitungsbefehle 71 finden Sie nähere Informationen zu diesen Befehlen.

Die Befehle zu diesen Symbolen **stehen nicht als Menübefehle** zur Verfügung. Beachten Sie außerdem, dass die entsprechende Funktion erst im StyleVision Power Stylesheet aktiviert und konfiguriert werden muss, damit Sie XML-Tabellen verwenden können. Eine ausführliche Beschreibung der in der Authentic View-Ansicht verwendeten Tabellen bzw. wie diese erstellt und bearbeitet werden, finden Sie weiter hinten in der Dokumentation im Abschnitt Tabellen in der Authentic-Ansicht 66.

Symbole zur Textformatierung

Text wird in der Authentic View-Ansicht formatiert, indem man darauf ein XML-Element oder -Attribut anwendet, das die gewünschte Formatierung aufweist. Wurde eine solche Formatierung definiert, kann der Autor des StyleVision Power Stylesheet in der Symbolleiste der Authentic View-Ansicht Schaltflächen zur Verfügung stellen, mit denen die Formatierung angewendet wird. Um eine Textformatierung mit Hilfe eines Textformatierungssymbols anzuwenden, markieren Sie den gewünschten Text und klicken Sie auf das entsprechende Symbol.

Symbole für die Navigation in DB-Zeilen

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

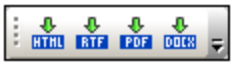
	Mit diesem Symbol wird das Dialogfeld "Datenbankabfrage bearbeiten" aufgerufen. Hier können Sie eine Abfrage eingeben. In der Authentic View-Ansicht werden die abgefragten Datensätze angezeigt.

XML-Datenbankbearbeitung

Mit dem Befehl **Neue Zeile mit XML-Daten für Bearbeitung auswählen** können Sie in einer XML-Datenbank wie z.B. IBM DB2 eine neue Zeile in der entsprechenden Tabelle auswählen. Diese Zeile wird in der Authentic View-Ansicht angezeigt, kann dort bearbeitet werden und wird dann wieder in der Datenbank gespeichert.

Portable XML Form (PXF)-Symbolleisten-Schaltflächen

In der Authentic View-Ansicht von <%SPY-GEN%> und Authentic Desktop stehen die folgenden PXF-Schaltflächen zur Verfügung:



Durch Klicken auf die jeweilige Schaltfläche wird ein HTML-, RTF-, PDF- und/oder DocX-Dokument generiert.

Diese Schaltflächen sind aktiv, wenn eine PXF-Datei in der Authentic View-Ansicht geöffnet ist. Die einzelnen Schaltflächen sind aktiv, wenn die PXF-Datei so konfiguriert wurde, dass sie das XSLT-Stylesheet für das jeweilige Ausgabeformat enthält. Wenn die PXF-Datei z.B. so konfiguriert wurde, dass sie XSLT Stylesheets für HTML und RTF enthält, so sind nur die Symbolleisten-Schaltflächen für die HTML- und RTF-Ausabe aktiv, während diejenigen für die PDF- und DocX (Word 2007+)-Ausgabe deaktiviert sind.

 © 2016-2022 Altova GmbH

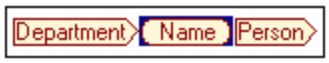
4.3 Authentic-Ansicht Hauptfenster

In der Authentic View-Ansicht gibt es vier Ansichtsmodi: Große Markup-Symbole; Kleine Markup-Symbole; Gemischte Markup-Symbole; und Alle Markup-Symbole ausblenden. Damit können Sie die Markup-Informationen für Dokumente mit unterschiedlicher Genauigkeit anzeigen. Um zwischen diesen Ansichtsmodi zu wechseln, verwenden Sie die Befehle im Menü "Authentic" oder die Symbole der Symbolleiste (siehe vorhergehender Abschnitt, Authentic View-Ansicht Symbolleistenschaltflächen 44).

Große Markup-Symbole

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

Das Element `Name` in der Abbildung oben ist **erweitert**, d.h. sein Start- und End-Tag sowie der Inhalt der Elements werden angezeigt. Durch Doppelklicken auf den Start- oder End-Tag eines Elements/Attributs können Sie dieses **reduzieren.** Um ein reduziert angezeigtes Element/Attribut wieder zu erweitern, doppelklicken Sie auf den reduzierten Tag.



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

Kleine Markup-Symbole

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

Nanonull, Inc.

Location: US

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

Vereno Office Summary: 4 departments, 15 employees.

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

Beachten Sie, dass Start-Tags ein Symbol enthalten, während End-Tags leer sind. Außerdem haben Element-Tags eine eckige Klammer im Symbol, während Attribut-Tags als Symbol ein Ist-Gleich-Zeichen haben (*siehe Abbildung unten*).

2006-04-01: *Boston*, *USA*

Um ein Element/Attribut zu reduzieren bzw. zu erweitern, doppelklicken Sie auf den entsprechenden Tag. Im unten gezeigten Beispiel sehen Sie ein reduziertes Element (blau markiert). Beachten Sie die Form des Tags des reduzierten Elements und die des Start-Tags des erweiterten Elements links davon.

Office Summary: 4 departments, 15 employees.

Gemischte Markup-Symbole

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

 © 2016-2022 Altova GmbH

Alle Markup-Symbole ausblenden

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

Anzeige von Inhalt

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

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

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

Bei den anderen Dateneingabe-Hilfen wird durch Ihre Auswahl der entsprechende XML-Wert erzeugt, der im StyleVision Power Stylesheet definiert wurde. In einer Auswahlliste könnte also beispielsweise "approved" (das in der Dropdown-Liste der Auswahlliste verfügbar wäre) ausgewählt werden, und dieser Eintrag könnte auf einen XML-Wert "1" oder auf "approved" oder einen beliebigen anderen Eintrag gemappt werden, während "not approved" dem Wert "0" oder "not approved" oder einem beliebigen anderen Wert zugeordnet würde.

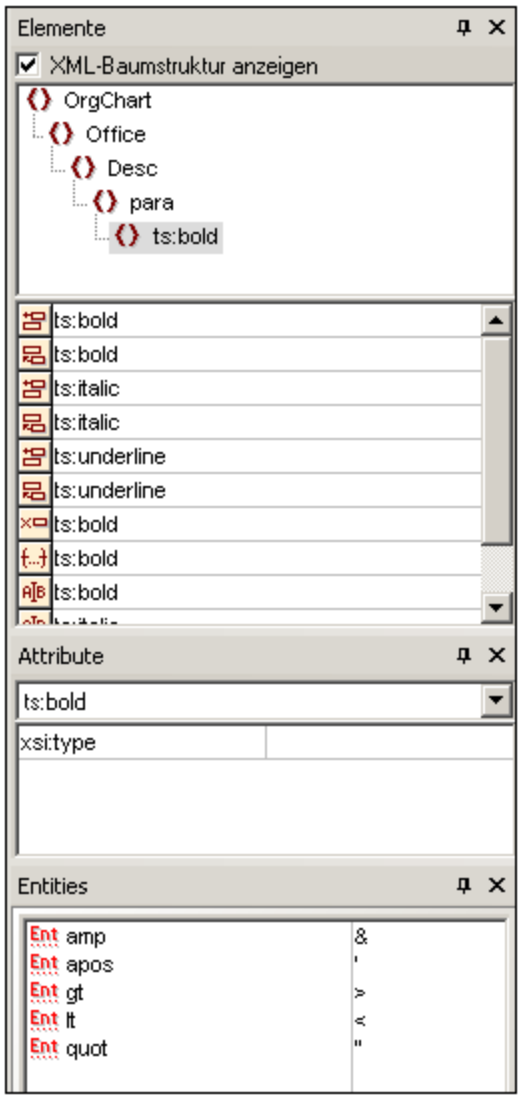
Optionale Nodes

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

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

4.4 Authentic-Ansicht Eingabehilfen

Die Authentic View-Ansicht verfügt über drei Eingabehilfen: für Elemente, Attribute und Entities. Sie werden als Fenster am rechten Rand der Authentic View-Ansicht angezeigt. (*siehe Abbildung unten*).



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

Im Folgenden werden die einzelnen Eingabehilfen beschrieben.

Element-Eingabehilfe

Die Element-Eingabehilfe besteht aus zwei Teilen:

 © 2016-2022 Altova GmbH

- dem oberen Teil mit einer XML-Baumstruktur, der über das Kontrollkästchen **Show xml tree** ein- und ausgeblendet werden kann. In der XML-Baumstruktur sehen Sie die übergeordneten Nodes bis hinauf zum Root-Element des aktuellen Elements des Dokuments. Wenn Sie auf ein Element in der XML-Baumstruktur klicken, werden die dazugehörigen Elemente (wie in nächsten Punkt dieser Liste beschrieben) im unteren Teil der Element-Eingabehilfe angezeigt.
- Im unteren Teil der Element-Eingabehilfe werden die Elemente aufgelistet, die im, vor und nach dem ausgewählten Element/Textbereich in der Authentic View-Ansicht eingefügt werden können, entfernt werden können oder darauf (d.h. durch Ersetzung) angewendet werden können. Was Sie mit einem in der Eingabehilfe aufgelisteten Element tun können, wird in der Eingabehilfe durch das Symbol auf der linken Seite des Elementnamens angezeigt. Im Folgenden finden Sie eine Liste der in der Element-Eingabehilfe vorkommenden Symbole mit einer Erklärung dazu.

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

Hinter Element einfügen

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

Vor Element einfügen

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

Element entfernen

Löscht des Element und dessen Inhalt.

Element einfügen

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

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

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

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

Element anwenden

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

Eingabehilfe auf das ausgewählte (Original-) Element angewendet werden kann. Der Befehl **Element anwenden** kann auch auf einen Textbereich innerhalb eines Elements gemischten Inhalts angewendet werden; der Textbereich wird als Inhalt des angewendeten Elements erstellt.

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

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

Element löschen

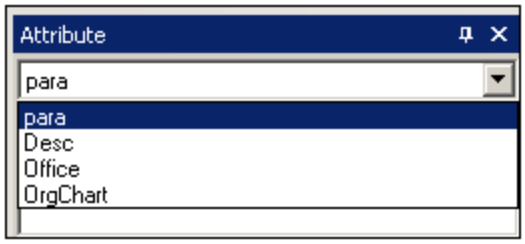
Dieses Symbol erscheint, wenn Text innerhalb eines Elements gemischten Inhalts ausgewählt wird. Wenn Sie auf dieses Symbol klicken, wird das Element, das den ausgewählten Bereich einschließt, gelöscht.

Element löschen (wenn ein Einfügepunkt ausgewählt ist)

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

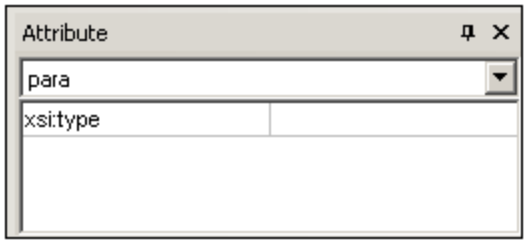
Attribut-Eingabehilfe

Die Attribut-Eingabehilfe besteht aus einem Listenfeld mit einer Dropdown-Liste von Attributen. Das ausgewählte Element (Sie können auf den Start- oder End-Tag klicken oder den Cursor irgendwo im Element-Inhalt platzieren) wird in der Dropdown-Liste angezeigt. Die in den nachfolgenden Abbildungen gezeigte Attribut-Eingabehilfe hat ein `para`-Element in der Dropdown-Liste. Wenn Sie auf den Abwärtspfeil in der Liste klicken, erscheint eine Liste aller **übergeordneten Elemente** des `para`-Elements hinauf bis zum Root-Element, in diesem Fall OrgChart.



Unterhalb der Dropdown-Liste wird eine Liste gültiger Attribute für dieses Element - in diesem Fall für das `para`-Element angezeigt. Wenn ein Attribut für ein bestimmtes Element obligatorisch ist, wird es fett angezeigt. (Im Beispiel unten gibt es keine obligatorischen Attribute mit Ausnahme des vordefinierten Attributs `xsi:type`)

 © 2016-2022 Altova GmbH



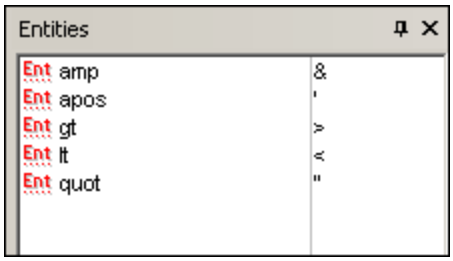
Um den Wert für ein Attribut einzugeben, klicken Sie in das Wertefeld des Attributs und geben Sie den Wert ein. Daraufhin wird das Attribut mit dem entsprechenden Wert im XML-Dokument erstellt.

Beachten Sie die folgenden Punkte:

- Im Falle des Attributs `xsi:nil`, welches in der Eingabehilfe "Attribut" erscheint, wenn ein nillable Element ausgewählt wurde, kann der Wert des Attributs `xsi:nil` nur durch Auswahl eines der zulässigen Werte (`true` oder `false`) aus der Dropdown-Liste der Werte des Attributs eingegeben werden.
- Das `xsi:type` Attribut kann durch Klicken in das Wertefeld des Attributs und entweder (i) Auswahl eines der aufgelisteten Werte aus der Dropdown-Liste oder (ii) durch Eingabe eines Werts geändert werden. Bei den in der Dropdown-Liste aufgelisteten Werten handelt es sich um die verfügbaren abstract types, die in dem XML-Schema, auf dem das Dokument in der Authentic-Ansicht basiert, definiert sind.

Entities-Eingabehilfe

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



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

Sie können in der Authentic View-Ansicht auch Ihre **eigenen Entities definieren,** die dann in der Eingabehilfe angezeigt werden. Nähere Informationen dazu finden Sie im Abschnitt "Bearbeitung in der Authentic View - Ansicht" unter Definieren von Entities 84.

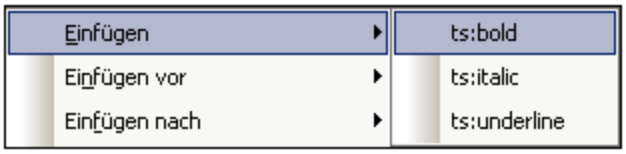
© 2016-2022 Altova GmbH

4.5 Authentic-Ansicht Kontextmenüs

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

Einfügen von Elementen

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



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

Die Befehle zum Einfügen eines Node, zum Ersetzen eines Node (**Anwenden**) und zum Entfernen von Markup-Code (**Löschen**), die im Kontextmenü zur Verfügung stehen, stehen auch in den Eingabehilfen der Authentic View-Ansicht 50 zur Verfügung und werden im entsprechenden Abschnitt ausführlich beschrieben.

Einfügen einer Entity

Wenn Sie den Cursor über den Befehl "Entity einfügen" positionieren, erscheint ein Untermenü mit einer Liste aller deklarierten Entities. Wenn Sie auf eine Entity klicken, wird sie am ausgewählten Node eingefügt. Eine Beschreibung, wie man Entities für das Dokument definiert, finden Sie unter Definieren von Entities 84.

Einfügen eines CDATA-Abschnitts

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

Entfernen eines Node

Wenn Sie den Mauszeiger über den Befehl **Entfernen** positionieren, erscheint eine Menüliste bestehend aus dem ausgewählten Node und allen seinen übergeordneten Nodes bis hinauf zum Dokument-Element, die

entfernt werden können (ohne dass das Dokument ungültig wird). Klicken Sie auf das Element, das Sie entfernen möchten. Auf diese Art lässt sich ein Element oder ein beliebiges übergeordnetes Element einfach entfernen. Beachten Sie, dass bei Entfernung eines übergeordneten Elements auch alle untergeordneten Elementen einschließlich des ausgewählten Elements entfernt werden.

Löschen

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

Übernehmen

Mit dem Befehl **Übernehmen** wird ein ausgewählten Element in Ihre Auswahl im Hauptfenster übernommen. Nähere Informationen dazu finden Sie unter Authentic View-Ansicht Eingabehilfen 50.

Kopieren, Ausschneiden, Einfügen

Dies sind die Standard-Windows-Befehle. Beachten Sie allerdings, dass Text mit dem Befehl **Einfügen** entweder als XML oder als Text eingefügt wird, je nachdem, was der Autor des Stylesheet für das SPS als Ganzes definiert hat. Informationen, wie die Befehle **Als XML-Text kopieren** und **Als Text kopieren** funktionieren, finden Sie unmittelbar unterhalb dieses Abschnitts in der Beschreibung zu **Einfügen als**.

Einfügen als

Der Befehl **Einfügen als** bietet die Option, ein in die Zwischenablage kopiertes XML-Fragment in der Authentic View-Ansicht entweder als XML oder als Text einzufügen. Wenn das kopierte Fragment als XML eingefügt wird, wird es zusammen mit dem XML-Markup-Code eingefügt. Wenn es als Text eingefügt wird, wird nur der Textinhalt des kopierten Fragments eingefügt (nicht aber der XML-Markup-Code). Möglich sind die folgenden Situationen:

- Ein ***ganzer Node*** wird in der Authentic View-Ansicht ***zusammen mit seinen Markup-Tags*** markiert und in die Zwischenablage kopiert. (i) Der Node kann als XML an jeder Stelle, an der dieser Node gültig ist, eingefügt werden. Er wird nicht an einer ungültigen Stelle eingefügt. (ii) Wenn der Node als Text eingefügt wird, wird nur der *Textinhalt* des Node eingefügt (nicht die Markup-Codes); das Textinhalt kann an jeder Stelle im XML-Dokument eingefügt werden, an der Text eingefügt werden darf.
- Ein ***Textfragment*** wird in der Authentic View-Ansicht markiert und in die Zwischenablage kopiert. (i) Wenn dieses Fragment als XML eingefügt wird, so werden die XML-Markup-Tags des Texts - obwohl diese mit dem Textfragment nicht explizit mitkopiert wurden - zusammen mit dem Text eingefügt, allerdings nur, wenn der XML-Node an der Stelle, an der das Fragment eingefügt wird, gültig ist. (ii) Wenn das Fragment als Text eingefügt wird, kann er an jeder Stelle im XML-Dokument eingefügt werden, an der Text eingefügt werden darf.

Anmerkung: Text wird in Nodes kopiert, in denen Text zulässig ist. Sie müssen daher sicher stellen, dass das Dokument durch den kopierten Text nicht ungültig wird. Der kopierte Text sollte daher:
(i) an der neuen Stelle lexikalisch gültig sein (so wären z.B. nicht numerische Zeichen in einem numerischen Node ungültig) und
(ii) den Node nicht auf andere Weise ungültig machen (so würde der Node z.B. bei Einfügen einer vierstelligen Zahl in einen Node, in dem nur dreistellige Zahlen zulässig sind, ungültig werden).

Anmerkung: Wenn das Dokument durch ein eingefügten Text auf irgendeine Weise ungültig wird, so wird der eingefügte Text rot markiert.

 © 2016-2022 Altova GmbH

Löschen

Mit dem Befehl **Löschen** wird der ausgewählte Node und sein gesamter Inhalt gelöscht. Ein Node gilt in diesem Zusammenhang als ausgewählt, wenn der Cursor in den Node platziert wurde oder indem Sie entweder auf den Start- oder End-Tag des Node klicken.

5 Bearbeitung in der Authentic-Ansicht

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

In diesem Abschnitt werden folgende Themen behandelt:

- Es gibt in der Authentic View-Ansicht drei verschiedene Tabellenarten. Im Abschnitt Tabellen in der Authentic View-Ansicht 66 werden diese drei Tabellenarten (statisches SPS, dynamisches SPS und XML) näher erläutert und deren Verwendung wird erklärt. Die Beschreibung beginnt mit einem allgemeinen Überblick. Anschließend wird näher auf die Einzelheiten eingegangen.
- Die Datumsauswahl ist ein grafischer Kalender, der Daten im korrekten XML-Format eingibt, wenn Sie auf ein Datum klicken. Siehe Datumsauswahl 81.
- Eine Entity ist ein Kürzel für ein Sonderzeichen oder einen Textstring. Sie können Ihre eigenen Entities definieren, um Sonderzeichen oder Textstrings durch Eingabe der Entity einfügen zu können. Nähere Informationen dazu finden Sie unter Definieren von Entities 84.
- In der Enterprise Edition von Altova-Produkten kann der Benutzer der Authentic View-Ansicht XML-Dokumente mit digitalen XML-Signaturen signieren und diese Signaturen überprüfen.
- Die Bildformate 89, die in der Authentic View-Ansicht angezeigt werden können.

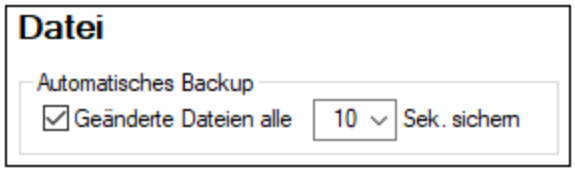
Altova Website: XML-Content-Bearbeitung, Erstellung von XML-Dokumenten

© 2016-2022 Altova GmbH

5.1 Automatisches Backup von Dateien

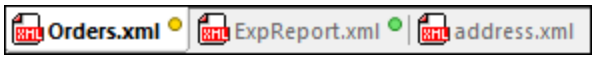
In Authentic Desktop geänderte Dateien werden in regelmäßigen Abständen automatisch gesichert. Im Dialogfeld "Optionen" (**Extras | Optionen | Datei** 268) auf dem Register *Datei* (siehe Abbildung unten) können Sie:

- automatische Backups aktivieren/deaktivieren
- die Backup-Intervalle (5 Sekunden bis 300 Sekunden) definieren



Anzeigesymbole

Die Dateiregister am unteren Rand des Hauptfensters enthalten jeweils ein Symbol rechts vom Dateinamen, das den gespeicherten/nicht gespeicherten Status und den Backup-Status der Datei angibt (*Abbildung unten*).



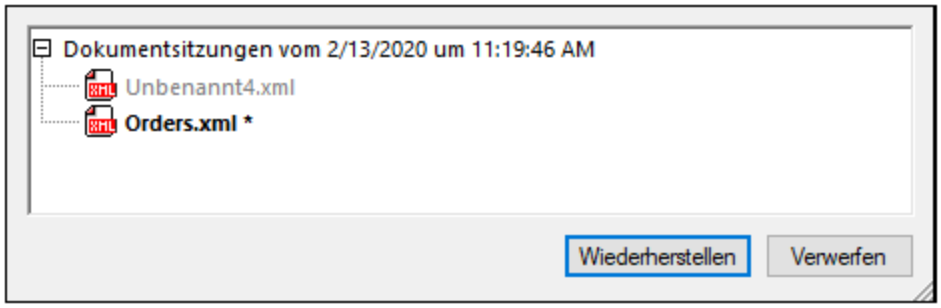
Gespeichert / nicht gespeichert
Wenn eine Datei geändert wurde, wird ein farbiges Kreissymbol angezeigt. Wenn kein solches Symbol angezeigt wird, bedeutet dies, dass die Datei seit der Öffnung oder der letzten Speicherung nicht geändert wurde. Ein Beispiel dafür in der Abbildung oben ist `address.xml`.

Backup-Status
Die Farben der Kreissymbole geben den Backup-Status der Datei an.

- *Gelb:* Die Datei wurde geändert, doch wurde die letzte Änderung nicht gesichert (oder gespeichert).
- *Grün:* Die Datei wurde gesichert und wurde seither nicht geändert. Jedoch wurde die Datei noch nicht gespeichert. (Wäre sie gespeichert worden, würde kein Kreissymbol angezeigt).
- *Rot:* Ein Backup für diese Datei wird entweder nicht unterstützt oder ist fehlgeschlagen.
- *Grau:* Die automatische Backup-Funktion wurde (über das Dialogfeld Optionen 267, *siehe oben*) deaktiviert. Dass ein Symbol angezeigt wird, gibt an, dass die Datei seit der letzten Änderung noch nicht gespeichert wurde. (Wäre sie gespeichert worden, würde kein Kreissymbol angezeigt).

Wiederherstellung anhand von Backups

Wenn Authentic Desktop unerwartet beendet wird, wird beim nächsten Applikationsstart ein "Dokument wiederherstellen"-Dialogfeld angezeigt, welches eine Liste aller zum Zeitpunkt der Applikationsbeendung geöffneten Dokumente enthält (*Abbildung unten*). Wenn Sie den Mauszeiger über eine Datei platzieren, sehen Sie ihren Pfad. Bei temporären Dateien, die noch nicht gespeichert wurden, wird als aktueller Standardpfad der Dateipfad angezeigt, der bei Öffnung des Dialogfelds "Speichern unter" angezeigt würde.



Bei jeder Datei in der Liste geben die Schriftart und das Vorhandensein oder Fehlen von Sternchen Aufschluss über die folgenden Dinge:

- Ein fetter Stil und ein Sternchen geben an, dass die Datei nicht gespeicherte Änderungen enthält. Derartige Dateien werden im Zustand der letzten Sicherung wiederhergestellt.
- Ein normaler Stil gibt an, dass die Datei gespeichert wurde und es keine nicht gespeicherten Änderungen gibt. Derartige Dateien werden zu dem Zustand der Speicherung wiederhergestellt.
- Ein ausgegrauter Stil zeigt an, dass die Datei weder gespeichert noch gesichert wurde (z.B. weil es sich um eine neue Datei handelt, die nicht bearbeitet wurde). Derartige Dateien werden nicht wiederhergestellt.

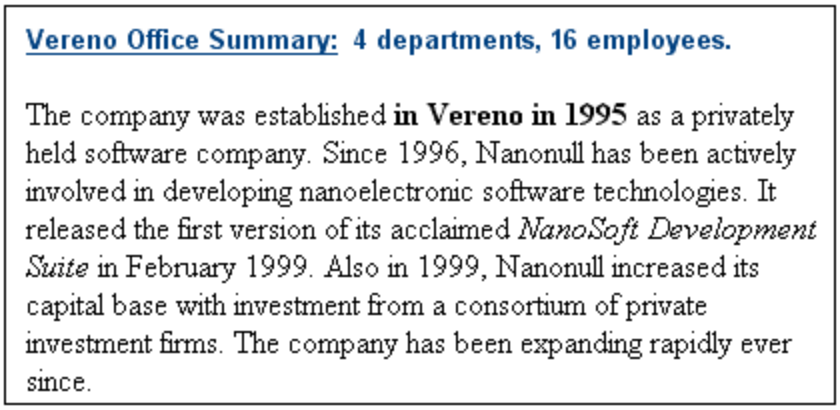
Sie haben nun eine der folgenden Möglichkeiten:

- Klicken Sie auf **Wiederherstellen**, um die Dateien auf der Benutzeroberfläche anhand ihres letzten Backups wiederherzustellen.
- Klicken Sie auf **Verwerfen**, um keine der aufgelisteten Dateien zu öffnen und alle vorhandenen Backups zu verwerfen.

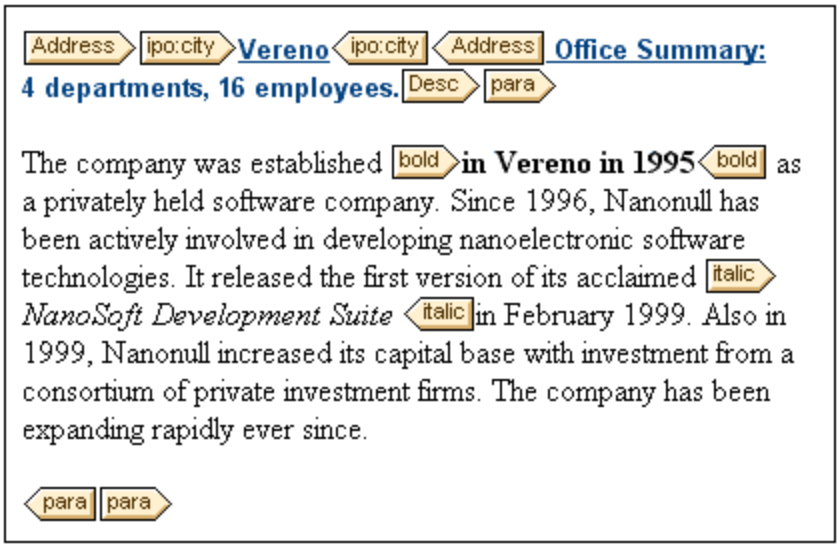
 © 2016-2022 Altova GmbH

5.2 Grundlegendes zur Bearbeitung

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



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



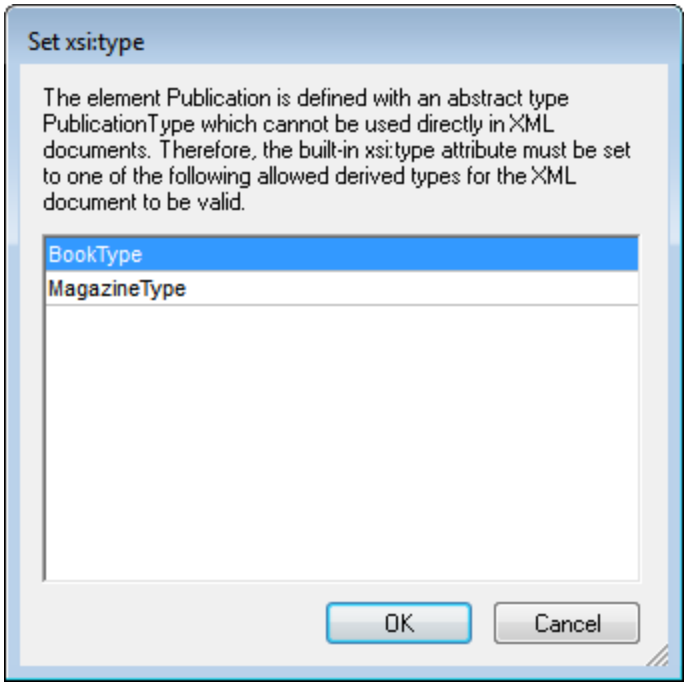
Ein editierbares Dokument in der Authentic View-Ansicht mit XML-Markup-Tags

Einfügen von Nodes

Oft kommt es vor, dass ein neuer Node in das XML-Authentic-Dokument eingefügt werden muss. So muss z.B. möglicherweise ein neues `Person` Element zu einem Dokument vom Typ Adressbuch hinzugefügt werden. In solchen Fällen würde das XML-Schema das Hinzufügen des neuen Elements gestatten. Klicken Sie einfach mit der rechten Maustaste auf den Node im Authentic-Ansichtsdokument, vor bzw. hinter dem Sie den neuen

Node hinzufügen möchten. Wählen Sie im Kontextmenü, das daraufhin angezeigt wird, je nach Bedarf den Befehl **Einfügen vor** oder **Einfügen nach** aus. Die Nodes, die an dieser Stelle in das Dokument eingefügt werden dürfen, werden in einem Untermenü aufgelistet. Klicken Sie auf den gewünschten Node, um ihn einzufügen. Der Node und alle obligatorischen untergeordneten Nodes werden eingefügt. Falls ein untergeordneter Node optional ist, wird ein Link `Add NodeName` angezeigt, auf den Sie klicken können, wenn der optionale Node eingefügt werden soll.

Wenn es sich beim hinzugefügten Node um ein Element mit einem abstract type handelt, so wird ein Dialogfeld (*wie in der Abbildung unten*) mit einer Liste von im XML-Schema verfügbaren derived types angezeigt.



Das oben gezeigte Dialogfeld wird angezeigt, wenn ein `Publication` Element hinzugefügt wird. Das `Publication` Element ist vom Typ `PublicationType`, einem abstract complexType. Die beiden complexTypes `BookType` und `MagazineType` werden vom abstract Type `PublicationType` abgeleitet. Wenn daher ein `Publication` Element zum XML-Dokument hinzugefügt wird, muss einer dieser beiden vom abstract Type `Publication` abgeleiteten concrete Types definiert werden. Das neue `Publication` Element wird mit einem `xsi:type` Attribut hinzugefügt.

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

Wenn Sie einen der verfügbaren derived types auswählen und auf **OK** klicken, geschieht Folgendes:

- Der ausgewählte derived type wird als Wert des `xsi:type` Attributs des Elements ausgewählt
- Das Element wird zusammen mit seinen im Content Model des ausgewählten derived type definierten untergeordneten Nodes eingefügt.

Der ausgewählte derived type kann in der Folge durch Ändern des Werts des `xsi:type` Attributs des Elements in der Attribut-Eingabehilfe geändert werden. Wenn der Typ des Elements auf diese Art geändert wird, werden

 © 2016-2022 Altova GmbH

alle Nodes des früheren Content Model entfernt und die Nodes des Typs des neuen Content Model werden eingefügt.

Bearbeitung von Text

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

Kopieren als XML oder als Text

Text kann als XML oder als Text kopiert und eingefügt werden.

- Wenn Text als XML eingefügt wird, so werden die XML Markup Tags zusammen mit dem Textinhalt von Nodes eingefügt. Die XML Markup Tags werden sogar dann eingefügt, wenn nur ein Teil des Node-Inhalts kopiert wurde. Damit der Markup-Code eingefügt werden kann, muss er dem Schema entsprechend an der Einfügestelle zulässig sein.
- Wenn Text als Text eingefügt wird, wird der XML Markup-Code nicht eingefügt.

Um Inhalt als XML oder als Text einzufügen, kopieren Sie zuerst den Text (**Strg+C)**, klicken Sie mit der rechten Maustaste an die Stelle, an der der Text eingefügt werden soll und wählen Sie im Kontextmenü den Befehl **Einfügen als | XML** oder **Einfügen als | Text**. Bei Verwendung des Tastaturkürzels **Strg+V** wird der Text im Standardeinfügemodus des SPS eingefügt. Der Standardeinfügemodus wird vom Ersteller des SPS definiert. Nähere Informationen dazu finden Sie im Abschnitt Kontextmenüs 55.

Alternativ dazu kann markierter Text auch mit der Maus an die gewünschte Einfügeposition gezogen werden. Wenn Sie die Maustaste loslassen, wird ein Popup-Fenster angezeigt, in dem Sie gefragt werden, ob der Text als Text oder als XML eingefügt werden soll. Wählen Sie die gewünschte Option aus.

Formatierung von Text

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

In manchen Fällen muss Inhalt speziell formatiert werden; z.B. ein Textstring der fett gedruckt werden soll. In allen diesen Fällen muss die Darstellung mit einem Strukturelement des Dokuments verknüpft sein. So wird z.B. ein Textstring, der fett dargestellt werden soll, strukturell vom restlichen Inhalt durch Markup-Code getrennt, dem der Stylesheet Designer Fettformatierung zuweisen wird. Wenn Sie als Benutzer der Authentic View-Ansicht einen solchen Textstring verwenden möchten, müssen Sie diesen in den entsprechenden Element Markup-Code einschließen. Nähere Informationen darüber finden Sie in dieser Dokumentation im Abschnitt Element-Eingabehilfe 50 unter dem Befehl "Element einfügen".

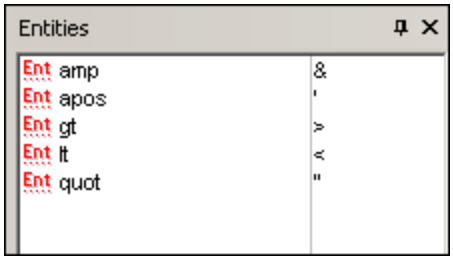
Verwendung von RichEdit in der Authentic View-Ansicht

Wenn Sie den Cursor in der Authentic View-Ansicht innerhalb von einem Element platzieren, das als RichEdit-Komponente erstellt wurde, werden die Schaltflächen und Steuerelemente in der RichEdit-Symbolleiste (*Abbildung unten*) aktiv. Anderfalls sind sie ausgegraut.

Wählen Sie den gewünschten Text aus und definieren Sie die Stile, die Sie darauf anwenden möchten, mit Hilfe der Schaltflächen und Steuerelemente der RichEdit-Symbolleiste. Mit Hilfe von RichEdit kann der Benutzer der Authentic View-Ansicht die Schriftart (font), Schriftbreite (font-weight), Schriftverzierung (font-decoration), Schriftgröße (font-size), Hintergrundfarbe und Ausrichtung des Texts definieren. Der mit Stilen versehene Text wird in die Tags des Stilelements eingeschlossen.

Einfügen von Entities

Einige Zeichen sind in XML-Dokumenten für Markup-Tags reserviert und können in normalem Text nicht verwendet werden. Dies sind die Zeichen Kaufmännisches Und (&), Apostroph ('), kleiner als (<), größer als (>) und doppeltes Anführungszeichen ("). Wenn Sie diese Zeichen in Ihren Daten verwenden möchten, müssen Sie sie als Entity-Referenzen über die Eingabehilfe "Entities" 50 (*Abbildung unten*) einfügen.



XML bietet auch die Möglichkeit benutzerdefinierte Entities zu erstellen. Dabei kann es sich um Folgendes handeln: (i) Sonderzeichen, die auf Ihrer Tastatur nicht zur Verfügung stehen, (ii) Textstrings, die Sie in Ihrem Dokumentinhalt wiederverwenden möchten, (iii) XML-Datenfragmente oder (iv) andere Ressourcen, wie z.B. Bilder. Sie können in der Applikation Authentic View-Ansicht Ihre eigenen Entities definieren 84. Nachdem Sie diese definiert haben, werden die Entities in der Eingabehilfe "Entities" 50 angezeigt und können dann in das Dokument eingefügt werden.

Einfügen von CDATA-Abschnitten

CDATA-Abschnitte sind Textabschnitte in einem XML-Dokument, die der XML-Parser nicht als XML-Daten verarbeitet. Mit Hilfe dieser Abschnitte können Sie große Textabschnitte auslassen, wenn die Ersetzung von

 © 2016-2022 Altova GmbH

Sonderzeichen durch Entity-Referenzen nicht wünschenswert ist; ein Beispiel wäre Programmcode oder ein XML-Fragment, das mit seinen Markup-Tags wiedergegeben werden soll. CDATA-Abschnitte können innerhalb von Elementinhalt vorkommen und werden am Beginn und Ende von `<![CDATA[` bzw. `]]>` eingeschlossen. Folglich sollte der Textstring `]]>` nicht in einem CDATA-Abschnitt vorkommen, da damit das Ende des Abschnitts gekennzeichnet würde. In diesem Fall kann das Zeichen "größer als" durch ihre Entity-Referenz (`>`) dargestellt werden. Um einen CDATA-Abschnitt in ein Element einzufügen, platzieren Sie den Cursor an der gewünschten Stelle, rechtsklicken Sie und wählen Sie im Kontextmenü den Befehl **CDATA-Abschnitt einfügen.** Um die CDATA-Abschnittstags in der Authentic View-Ansicht anzuzeigen, wechseln Sie in die Markup-Anzeige 44. Alternativ dazu können Sie den Text markieren, der in einen DATA-Abschnitt eingeschlossen werden soll, und den Befehl **CDATA-Abschnitt einfügen** wählen.

Anmerkung: CDATA-Abschnitte können nicht in Eingabefelder (d.h. in ein- oder mehrzeilige Textfelder) eingefügt werden. CDATA-Abschnitte können nur in Elemente eingefügt werden, die in der Authentic View-Ansicht als Komponenten des Textinhalts angezeigt werden

Bearbeiten von Links und Navigieren zum Ziel eines Links

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

5.3 Tabellen in der Authentic-Ansicht

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

SPS-Tabellen: Es gibt zwei Arten von SPS-Tabellen: statische und dynamische. SPS-Tabellen werden vom Autor des StyleVision Power Stylesheet, mit dem Ihr XML-Dokument verknüpft ist, erstellt. Sie können selbst keine SPS-Tabelle in das XML-Dokument einfügen, doch können Sie Daten in die Felder der SPS-Tabelle eingeben und Zeilen in dynamischen SPS-Tabellen hinzufügen und löschen. Im nachfolgenden Abschnitt zu SPS-Tabellen 66 werden die Funktionen dieser Tabellen näher beschrieben.

CALS/HTML-Tabellen werden vom Benutzer der Authentic View-Ansicht eingefügt. Auf diese Art können Sie auf Wunsch an jeder Stelle im Dokument, an der dies zulässig ist, Tabellen einfügen. Die Editierfeatures von CALS/HTML-Tabellen 67 und die Bearbeitungssymbole für CALS/HTML-Tabellen 71 werden im Folgenden beschrieben.

5.3.1 SPS-Tabellen

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

Statische Tabellen

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

Nanonull, Inc.

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

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

Dynamische Tabellen

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

 © 2016-2022 Altova GmbH

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

Administration

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

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

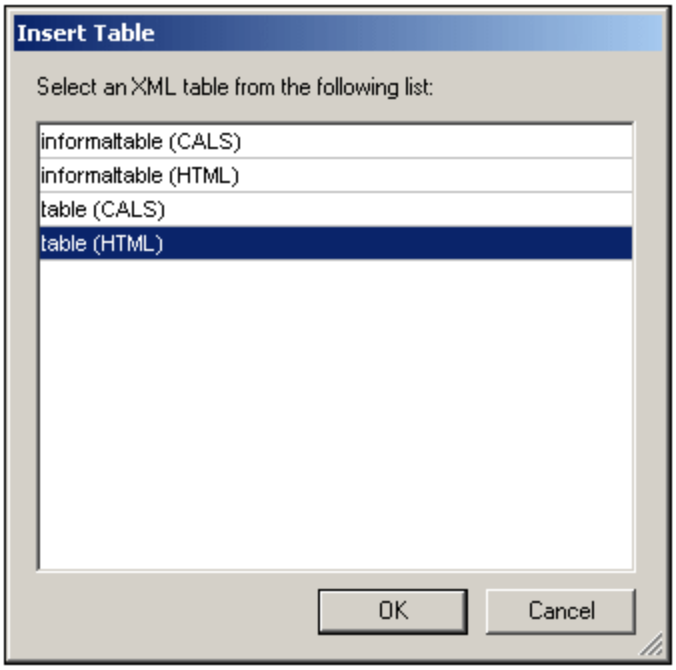
5.3.2 CALS/HTML-Tabellen

CALS/HTML-Tabellen können von Ihnen, dem Benutzer der Authentic View-Ansicht für bestimmte XML-Datenstrukturen verwendet werden, die als Tabellenformat definiert wurden. Die Arbeit mit CALS/HTML-Tabellen erfordert drei Schritte: Einfügen der Tabelle, Formatierung der Tabelle und Eingabe von Daten. Die Befehle für die Arbeit mit CALS/HTML-Tabellen stehen in Form von Schaltflächen in der Symbolleiste zur Verfügung (siehe Symbole zur Bearbeitung von CALS/HTML-Tabellen 71).

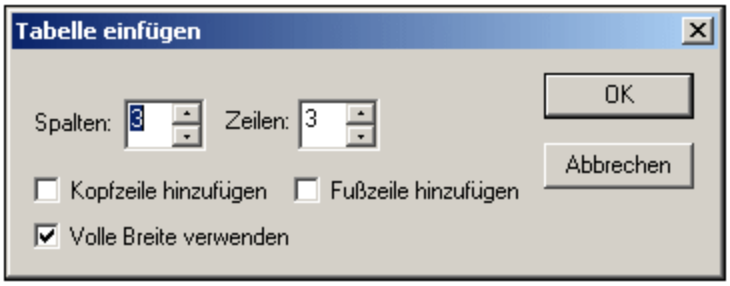
Einfügen von Tabellen

So fügen Sie eine CALS/HTML-Tabelle ein:

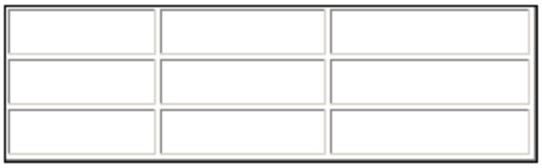
1. Setzen Sie Ihren Cursor an die Stelle, an der die Tabelle eingefügt werden soll und klicken Sie auf das Symbol . (Beachten Sie dass CALS/HTML-Tabellen nur an bestimmten im Schema definierten Stellen eingefügt werden können). Das Dialogfeld "Tabelle einfügen" (*Abbildung unten*) wird nun angezeigt. In diesem Dialogfeld werden alle XML-Element-Datenstrukturen, für die eine Tabellenstruktur definiert wurde, aufgelistet. So wurden z.B. in der Abbildung unten die Elemente `informaltable` und `table` sowohl als CALS-Tabelle als auch als HTML-Tabelle definiert.



2. Wählen Sie den Eintrag mit dem gewünschten Element und Tabellenmodell aus und klicken Sie auf OK.
3. Wählen Sie im nächsten Dialogfeld (*Abbildung unten*) die Anzahl der Spalten und Zeilen aus und geben Sie an, ob eine Kopf- und/oder Fußzeile zur Tabelle hinzugefügt werden soll und ob die Tabelle sich über die gesamte Breite erstrecken soll. Klicken Sie anschließend auf **OK.**



Daraufhin wird anhand der Angaben oben die folgende Tabelle erstellt.



Über die Befehle im Menü **Tabelle** können Sie später Spalten hinzufügen und löschen, Zeilen und Spalten zusammenfügen oder trennen. Erstellen Sie daher zuerst die grundlegende Struktur.

Formatieren von Tabellen und Eingabe von Daten

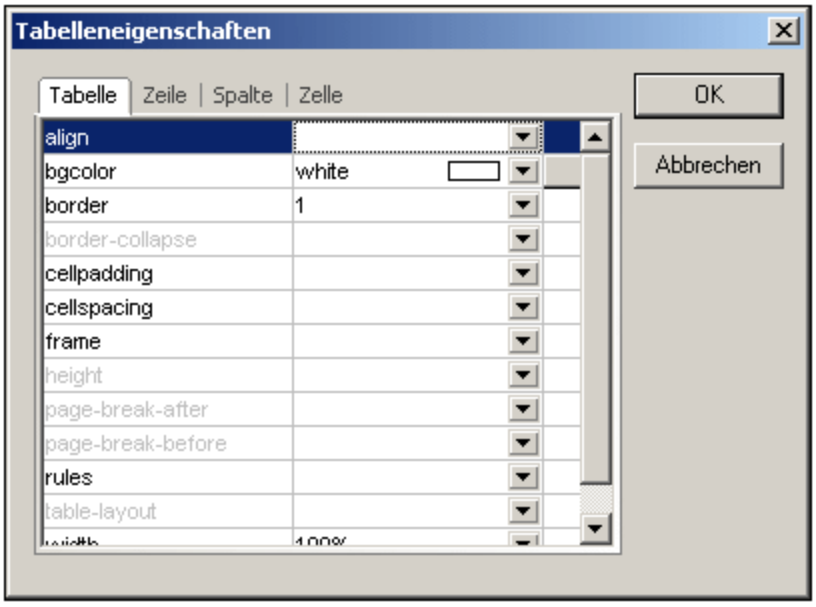
Die Tabellenformatierung wurde bereits im Dokumentdesign zugewiesen. Unter bestimmten Umständen können Sie die Tabellenformatierung jedoch ändern. Diese Umstände sind:

 © 2016-2022 Altova GmbH

- Für die Elemente, die den verschiedenen Tabellenstrukturelementen entsprechen müssen (im zugrunde liegenden XML-Schema) die entsprechenden CALS- oder HTML-Tabelleneigenschaften als Attribute definiert sein. Nur die Attribute, die definiert sind, stehen für die Formatierung zur Verfügung. Wenn im Design Werte für diese Attribute definiert wurden, können Sie diese in der Authentic View-Ansicht außer Kraft setzen.
- Im Design darf kein `style` Attribut, das CSS-Stile enthält, definiert worden sein. Wenn für ein Element ein Stilattribut mit CSS-Stilen definiert wurde, so hat das `style` Attribut Vorrang vor anderen für dieses Element gesetzten Formatierungsattributen. Folglich werden alle in der Authentic View-Ansicht definierten Formatierungen außer Kraft gesetzt.

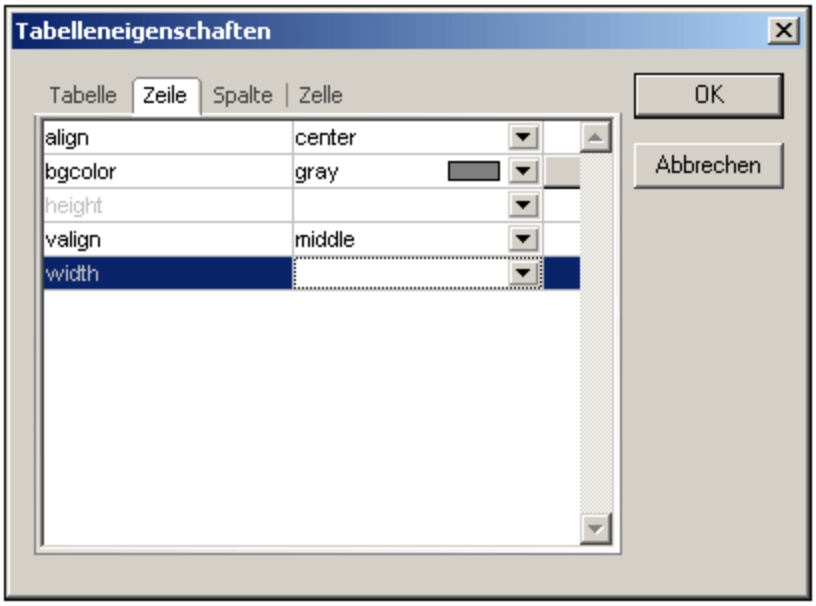
So formatieren Sie Ihre Tabelle, Zeile, Spalte oder Zelle:

1. Platzieren Sie Ihren Cursor zum Formatieren der Tabelle an eine beliebige Stelle in der Tabelle und klicken Sie auf das Symbol (Tabelleneigenschaften). Daraufhin erscheint das Dialogfeld "Tabelleneigenschaften" (siehe unten), in dem Sie die Formatierung für die Tabelle, oder für einzelne Zeilen, Spalten oder Zellen definieren können.



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

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



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

Name	Telephone	Email

Beachten Sie, dass die Ausrichtung wie angegeben mittig (centered) ist.

4. Angenommen, Sie möchten nun die Spalte "Telephone" in die Spalten "Office" und "Home" unterteilen. Um dies zu tun, müssen Sie die horizontale Breite der Spalte "Telephone" in zwei Spalten teilen. Zuerst werden wir allerdings die vertikale Größe der Kopfzeile aufteilen um eine Zeile für die Unter-Überschrift zu erstellen. Setzen Sie die Cursor in die Zelle "Telephone" und klicken Sie auf das Symbol (Vertikal teilen). Ihre Tabelle sieht nun aus wie folgt:

Name	Telephone	Email

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

 © 2016-2022 Altova GmbH

Name	Telephone		Email
	Office	Home	

Nun müssen Sie die horizontale Breite der einzelnen Zellen in der Spalte "Telephone" aufteilen.

Mit Hilfe der Schaltflächen zum Bearbeiten von Tabellen können Sie des Weiteren Spalten und Zeilen hinzufügen und löschen und Zellinhalte vertikal ausrichten. Die Schaltflächen zum Bearbeiten von CALS/HTML-Tabellen sind in der Benutzerreferenz im Abschnitt "Schaltflächen zum Bearbeiten von XML-Tabellen 71" näher beschrieben.

Navigieren innerhalb von Tabellenzellen

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

Eingabe von Daten in eine Zelle

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

Formatieren von Text

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

Name	Telephone		Email
	Office	Home	

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

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

5.3.3 Schaltflächen zum Bearbeiten von CALS/HTML-Tabellen

Die Befehle zum Bearbeiten von CALS/HTML-Tabellen stehen in Form von Symbolen in der Symbolleiste zur Verfügung. Beachten Sie, dass es zu diesen Symbolen keine Menübefehlentsprechungen gibt. Ausführliche Informationen zur Verwendung von CALS/HTML-Tabellen finden Sie unter CALS/HTML-Tabellen 67.

Tabelle einfügen

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

Tabelle löschen

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

Zeile anhängen

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

Spalte anhängen

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

Zeile einfügen

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

Spalte einfügen

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

Links zusammenfügen

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

Rechts zusammenfügen

Der Befehl "Rechts zusammenfügen" fügt die aktive Zelle (an der Cursorposition) mit der Zelle rechts davon zusammen. Die Inhalte von beiden Zellen werden in der neuen Zelle zusammengeführt.

Unten zusammenfügen

Der Befehl "Unten zusammenfügen" fügt die aktive Zelle (an der Cursorposition) mit der Zelle unterhalb davon zusammen. Die Inhalte von beiden Zellen werden in der neuen Zelle zusammengeführt.

Oben zusammenfügen

Der Befehl "Oben zusammenfügen" fügt die aktive Zelle (an der Cursorposition) mit der Zelle oberhalb davon zusammen. Die Inhalte von beiden Zellen werden in der neuen Zelle zusammengeführt.

Horizontal teilen

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

Vertikal teilen

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

© 2016-2022 Altova GmbH

Bündig oben

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

Vertikal zentrieren

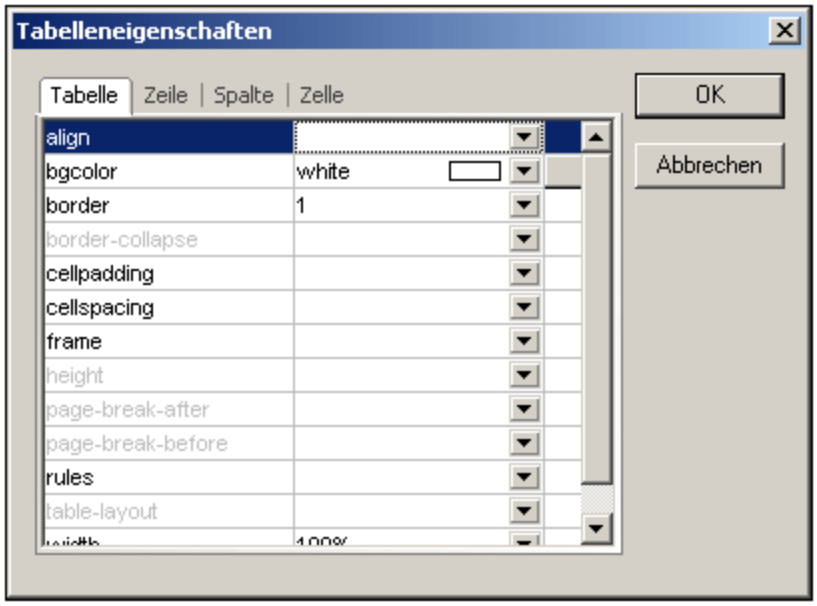
Dieser Befehl zentriert den Inhalt der Zelle.

Bündig unten

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

Tabelleneigenschaften

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



5.4 Bearbeiten einer DB

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

- Die Anzahl der Datensätze in einer DB-Tabelle, die in der Authentic View-Ansicht angezeigt werden können, wurde unter Umständen vom Autor des StyleVision Power Stylesheet absichtlich eingeschränkt, um das Design kompakter zu gestalten. In diesem Fall wird anfangs nur diese beschränkte Anzahl an Datensätzen in der Authentic View-Ansicht geladen. Sie können die restlichen Datensätze der DB-Tabelle mit Hilfe der Symbole zur Navigation in DB-Tabellenzeilen (*siehe* Navigieren in einer DB-Tabelle [74]) laden und anzeigen.
- Sie können eine Datenbankabfrage [75] durchführen, um bestimmte Datensätze anzuzeigen.
- Sie können Datensätze hinzufügen, ändern und löschen und Ihre Änderungen in der DB speichern. Siehe Ändern einer DB-Tabelle [79].

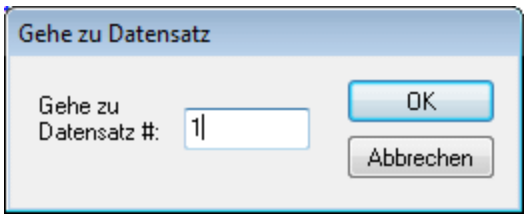
Um ein DB-basiertes StyleVision Power Stylesheet in der Authentic Ansicht zu öffnen, klicken Sie auf **Authentic | Datenbank bearbeiten**, und suchen Sie nach dem gewünschten StyleVision Power Stylesheet.

Anmerkung: Daten aus einer SQLite-Datenbank können in der Authentic-Ansicht nicht bearbeitet werden. Wenn Sie versuchen, SQLite-Daten über die Authentic-Ansicht zu speichern, erscheint eine entsprechende Meldung.

5.4.1 Navigieren in einer DB-Tabelle

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

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



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

XML-Datenbanken

Im Fall von XML-Datenbanken wie z.B. IBM DB2 enthält eine Zelle (oder Zeile) ein einziges XML-Dokument. Daher wird immer nur eine einzige Zeile auf einmal in die Authentic View-Ansicht geladen. Um ein XML-

 © 2016-2022 Altova GmbH

Dokument zu laden, das sich in einer anderen Zeile befindet, verwenden Sie die Menübefehl **Authentic | Neue Zeile mit XML-Daten für Bearbeitung auswählen**230.

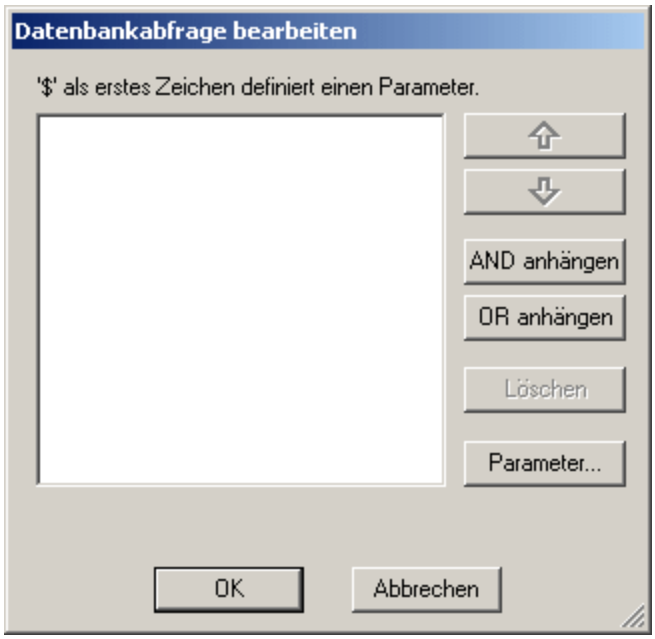
5.4.2 DB-Abfragen

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

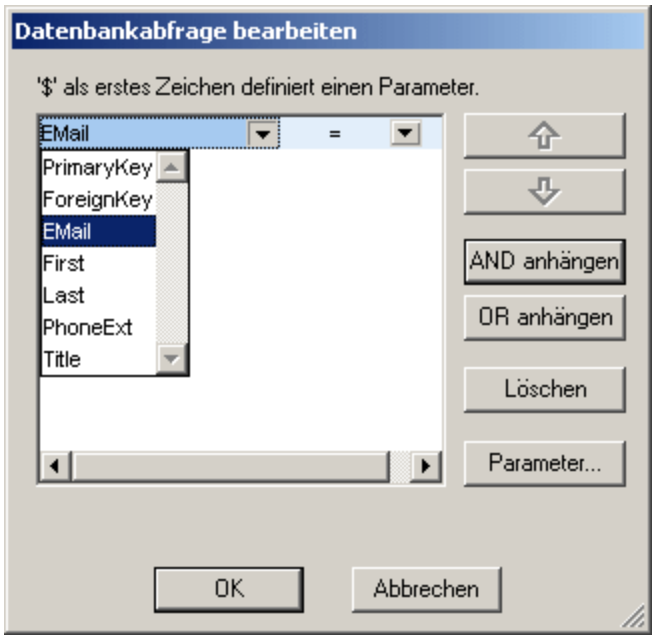
Anmerkung: Falls eine Meldung angezeigt wird, dass zu viele Tabellen geöffnet sind, können Sie die Anzahl der offenen Tabellen mit Hilfe einer Abfrage durch Filtern reduzieren.

So erstellen Sie eine Abfrage und senden diese ab:

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



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



3. Geben Sie den Ausdruck für das Kriterium ein. Ein Ausdruck besteht aus: (i) einem Feldnamen (der in der dazugehörigen Dropdown-Liste zur Verfügung steht); (ii) einem Operator (der in der zugehörigen Dropdown-Liste zur Verfügung steht); und (iii) einem Wert (direkt einzugeben). Nähere Informationen zum Erstellen von Ausdrücken finden Sie im Abschnitt Ausdrücke in Kriterien 76 weiter unten.
4. Wenn Sie ein weiteres Kriterium hinzufügen möchten, klicken Sie - je nachdem, welchen logischen Operator (AND oder OR) Sie verwenden wollen, um Ihre Kriterien zu verknüpfen - auf die Schaltfläche **AND anhängen** bzw. **OR anhängen.** Fügen Sie nun das neue Kriterium hinzu. Nähere Informationen zu logischen Operatoren finden Sie im Abschnitt Neuordnen von Kriterien in DB-Abfragen 77.

Ausdrücke in Kriterien

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

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

Wenn IS NULL oder NOT NULL ausgewählt ist, ist das Wertefeld deaktiviert. **Werte** müssen ohne Anführungszeichen (oder andere Begrenzungszeichen) eingegeben werden. Außerdem müssen Werte dieselbe Formatierung wie die des entsprechenden DB-Felds haben, sonst ist das Ergebnis des Ausdrucks `FALSE`. Wenn z.B. ein Kriterium für ein Feld des `date`-Datentyps in einer MS Access DB einen Ausdruck

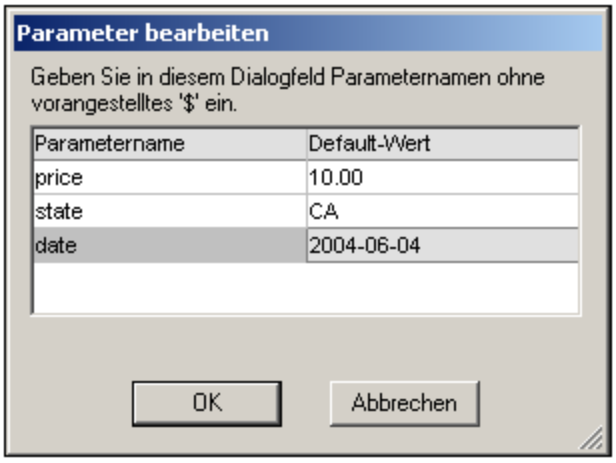
 © 2016-2022 Altova GmbH

`StartDate=25/05/2004` enthält, ist das Ergebnis der Abfrage `FALSE`, da der `date`-Datentyp in einer MS Access DB das Format `JJJJ-MM-TT` hat.

Verwendung von Parametern mit DB-Abfragen

Sie können beim Erstellen von Abfragen den Namen eines **Parameters** als Wert eines Ausdrucks eingeben. Parameter sind Variablen, die anstelle von Literal-Werten in Abfragen verwendet werden können. Wenn Sie sie in einem Ausdruck eingeben, wird ihr Wert im Ausdruck verwendet. Parameter, die verfügbar sind, sind vom SPS-Designer im SPS definiert worden und können im Dialogfeld "Parameter anzeigen" (*siehe Abbildung unten*) angezeigt werden. Parametern wurde im SPS ein Standardwert zugewiesen, der überschrieben werden kann, indem dem Parameter über die Befehlszeile ein Wert zugewiesen wird (falls und während das Ausgabedokument über die Befehlszeile kompiliert wird).

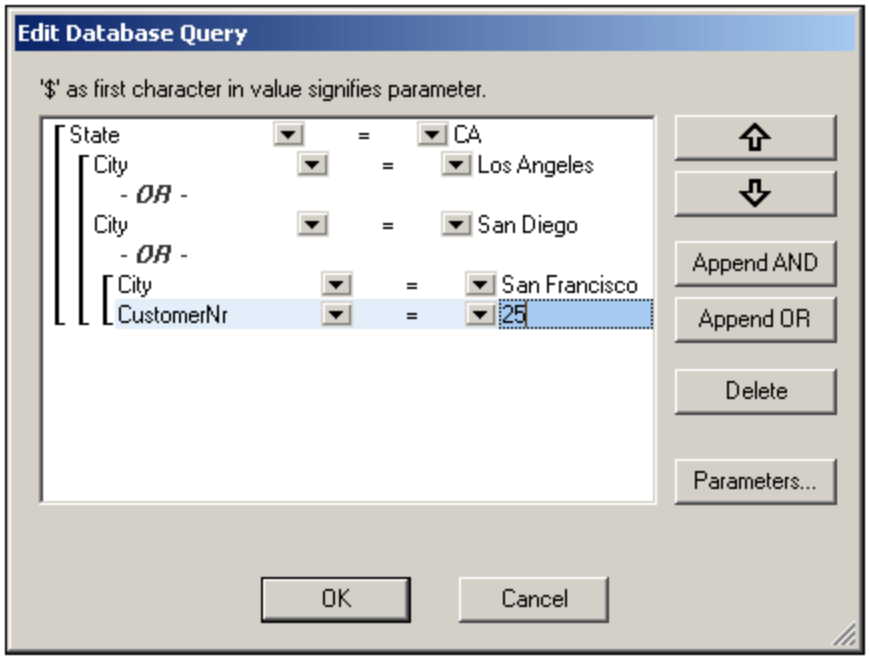
Um die Parameter, die für das SPS definiert wurden, anzuzeigen, klicken Sie im Dialogfeld **Datenbankabfrage bearbeiten** auf die Schaltfläche **Parameter**, um das unten gezeigte Dialogfeld **Parameter anzeigen** aufzurufen.



Das Dialogfeld "Parameter anzeigen" enthält **alle** Parameter, die für das Stylesheet definiert wurden und Parameter müssen im Stylesheet-Design bearbeitet werden.

Neuordnen von Kriterien in DB-Abfragen

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



Die in der obigen Abbildung gezeigte DB-Abfrage würde in Textform folgendermaßen aussehen:
`State=CA AND (City=Los Angeles OR City=San Diego OR (City=San Francisco AND CustomerNr=25))`

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

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

Beachten Sie die folgenden Punkte:

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

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

Ändern einer DB-Abfrage

So ändern Sie eine DB-Abfrage:

 © 2016-2022 Altova GmbH

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

5.4.3 Ändern einer DB-Tabelle

Hinzufügen eines Datensatzes

So fügen Sie einen Datensatz hinzu:

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

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

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

Ändern eines Datensatzes

Um einen Datensatz zu ändern, setzen Sie den Cursor an die gewünschte Stelle in der DB-Tabelle und nehmen Sie die gewünschten Änderungen vor. Wenn die Anzahl der Datensätze beschränkt ist, müssen Sie eventuell (siehe Navigieren in einer DB-Tabelle 74) zum gewünschten Datensatz navigieren.

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

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

Bitte beachten Sie die folgenden Punkte:

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

Löschen eines Datensatzes

So löschen Sie einen Datensatz:

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

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

© 2016-2022 Altova GmbH

5.5 Arbeiten mit Datumsangaben

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

- Die Eingabe oder Änderung des Datums erfolgt über die Datumswahl 81.
- Das Datum wird durch Eingabe des Werts 82 eingegeben bzw. geändert.

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

Anmerkung zu Datumsformaten

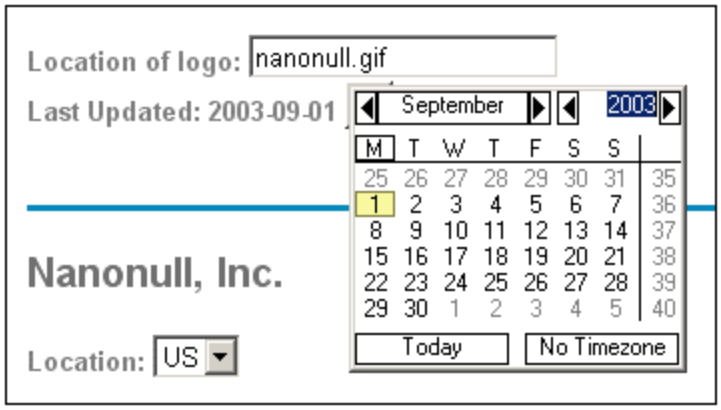
Zur Speicherung des Datums im XML-Dokument stehen mehrere Datums-Datentypen zur Verfügung. Bei jedem dieser Datums-Datentypen muss das Datum in einem bestimmten lexikalischen Format gespeichert werden, damit das XML-Dokument gültig ist. So muss das Datum z.B. beim Datentyp `xs:date` das lexikalische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

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

5.5.1 Datumswahl

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

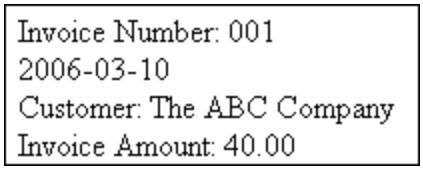
Um die Datumswahl (siehe unten) anzuzeigen, klicken Sie auf das Datumswahl-Symbol.



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

5.5.2 Texteingabe

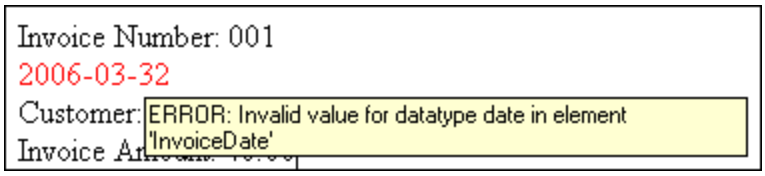
In Datumsfeldern, die keine Datumswahl haben (*siehe Abbildung*) können Sie das Datum direkt durch Eingabe des neuen Werts bearbeiten.



Fehler

Die folgenden Arten von Fehlern werden gekennzeichnet:

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



- Wenn Sie versuchen, das Datumsformat zu ändern, wird das Datum rot angezeigt. (So werden etwa in der Abbildung unten Schrägstriche anstelle von Bindestrichen verwendet).

© 2016-2022 Altova GmbH

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

5.6 Definieren von Entities

Entities

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

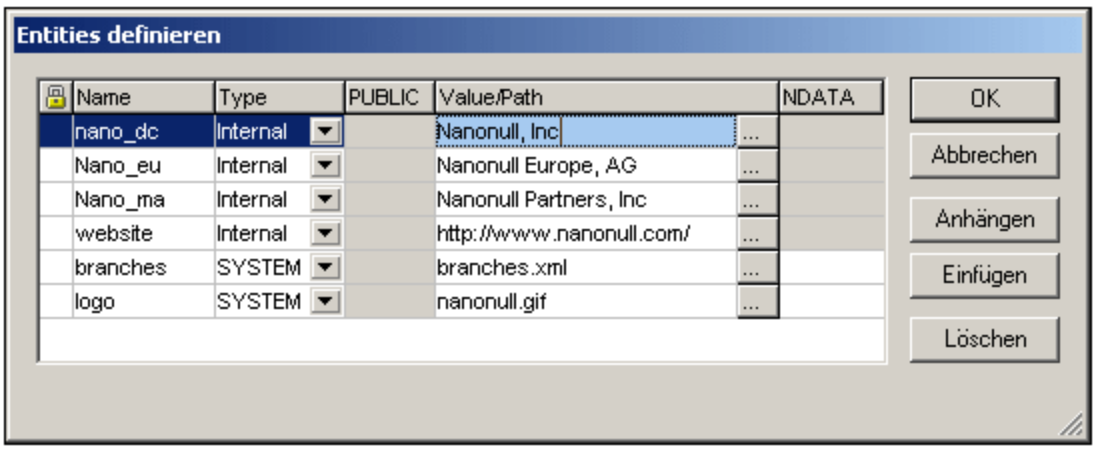
Eine Entity eignet sich v.a. dann, wenn Sie einen Textstring, ein XML-Fragment oder einige andere externe Ressourcen an mehreren Stellen in Ihrem Dokument verwenden möchten. Die Entity, bei der es sich im Grunde um eine Kurzbezeichnung für die benötigten Daten handelt, wird im Dialogfeld "Entities definieren"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

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

Definieren von Entities

So definieren Sie eine Entity:

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



2. Geben Sie den Namen Ihrer Entity im Feld "Name" ein. Dies ist der Name, der in der Entities-Eingabehilfe angezeigt werden wird.
3. Wählen Sie den Typ der Entity aus der Dropdown-Liste im Feld "Type" aus. Es stehen die folgenden Typen zur Auswahl. Eine **interne** (internal) Entity ist eine Entity, für die der zu verwendende Text im XML-Dokument selbst gespeichert ist. Wenn Sie **PUBLIC** oder **SYSTEM** auswählen, wird angegeben, dass sich die Ressource außerhalb der XML-Datei befindet und mittels eines "Public Identifier" oder

 © 2016-2022 Altova GmbH

eines "System Identifier" gefunden wird. Ein System Identifier ist eine URI, die den Pfad der Ressource definiert. Ein Public Identifier ist ein pfadunabhängiger Identifier, der es manchen Prozessoren ermöglicht, die Ressource zu identifizieren. Wenn Sie sowohl einen Public als auch einen System Identifier angeben, wird der Public Identifier zum System Identifier aufgelöst und es wird der System Identifier verwendet.

4. Wenn Sie als Type PUBLIC ausgewählt haben, geben Sie den Public Identifier der Ressource im Feld PUBLIC ein. Wenn Sie Internal oder SYSTEM als Type ausgewählt haben, ist das Feld PUBLIC deaktiviert.
5. Im Feld "Value/Path" haben Sie folgende Eingabemöglichkeiten:

   - Wenn die Entity "Internal" ist, geben Sie den Textstring ein, der als Wert der Entity verwendet werden soll. Setzen Sie den Textstring nicht in Anführungszeichen. Alle Anführungszeichen werden als Teil des Textstrings behandelt.
   - Wenn der Entity-Typ SYSTEM ist, geben Sie die URI der Ressource ein oder wählen Sie mit Hilfe der Durchsuchen-Schaltfläche eine Ressource auf Ihrem lokalen Netzwerk aus. Wenn die Ressource geparste Daten enthält, muss es sich um eine XML-Datei handeln (d.h. die Datei muss die Erweiterung `.xml` haben). Als Alternative dazu kann die Ressource eine Binärdatei sein, z.B. eine GIF-Datei.
   - Wenn der Entity-Typ PUBLIC ist, müssen Sie zusätzlich einen System Identifier in dieses Feld eingeben.

6. Der Eintrag NDATA sagt dem Prozessor, dass diese Entity nicht geparst, sondern an den entsprechenden Prozessor gesendet werden soll. Das Feld "NDATA" muss daher einen Wert enthalten, der angibt, dass es sich um eine ungeparste Entity handelt.

Dialogfeldfunktionen

Im Dialogfeld "Entities definieren" können Sie Folgendes tun:

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

Einschränkungen für Entities

- Eine Entity innerhalb einer anderen Entity wird nicht aufgelöst - weder im Dialogfeld, noch in der Authentic View-Ansicht, noch in der XSLT-Ausgabe, und das kaufmännische Und-Zeichen einer derartigen Entity wird in seiner Escape-Form, also `&` angezeigt.
- Externe ungeparste Entities, bei denen es sich nicht um Bilddateien handelt, werden in der Authentic View-Ansicht nicht aufgelöst. Wenn für ein Bild im Design definiert ist, dass eine externe ungeparste Entity gelesen werden soll und wenn als URI ein Entity-Name (z.B. `'logo'`) definiert ist, so kann dieser Entity-Name im Dialogfeld "Entities definieren" (*siehe Abbildung oben*) als externe ungeparste

Entity mit einem Wert, der zur URI der Bilddatei aufgelöst wird (wie diese für die Entity `logo` in der Abbildung oben geschehen ist), definiert werden.

© 2016-2022 Altova GmbH

5.7 XML-Signaturen

Bei der Erstellung eines SPS kann eine XML-Signatur für die Authentic View-Ansicht konfiguriert werden. Wenn XML-Signaturen im SPS aktiviert sind, kann der Benutzer der Authentic View-Ansicht die Authentic XML-Datei mit der aktivierten Signatur digital signieren. Wenn nach der Signierung des Dokuments Änderungen daran vorgenommen werden, schlägt die Überprüfung der Signatur fehl. Jedes Mal, wenn ein signiertes Authentic XML-Dokument in der Authentic View-Ansicht eines beliebigen Altova-Produkts geöffnet wird, wird die Signatur des Dokuments überprüft und das Ergebnis der Überprüfung in einem Fenster angezeigt.

Anmerkung: XML-Signaturen können in der Authentic-Ansicht von Enterprise und Professional Editions der folgenden Altova-Produkte verwendet und überprüft werden: Authentic Desktop, Authentic Browser, XMLSpy und StyleVision

XML-Signaturaktionen

In der Authentic-Ansicht sind die folgenden Benutzeraktionen möglich:

- *Wählen des Zertifikats/Passworts:* Signaturen werden entweder anhand eines Zertifikats oder Passworts authentifiziert. Das Authentifikationsobjekt (Zertifikat oder Passwort) wird bei der Erstellung der Signatur und dann wieder bei der Überprüfung der Signatur benötigt. Wenn einem Authentic XML-Dokument ein SPS, für das Signaturen aktiviert wurden, zugewiesen wurde, so wurde dem SPS möglicherweise ein Standardzertifikat oder Passwort zugewiesen. Egal, ob ein Standardzertifikat bzw. ein Standardpasswort definiert wurde oder nicht, die Signatur kann so konfiguriert werden, dass der Benutzer der Authentic View-Ansicht ein eigenes Zertifikat/Passwort auswählen kann. Er kann dies jederzeit im Dialogfeld "XML-Signatur" (*Abbildung unten*) tun und damit das Standardzertifikat/-passwort außer Kraft setzen, falls eines definiert wurde. Das eigene Zertifikat/Passwort des Benutzers wird im Arbeitsspeicher gespeichert und für die aktuelle Sitzung verwendet. Wenn der Benutzer der Authentic View-Ansicht ein eigenes Zertifikat/Passwort auswählt und die Datei oder die Applikation schließt, so wird das SPS wieder auf die Standardeinstellung für das Zertifikat/Passwort zurückgesetzt.

- *Signieren des Dokuments:* Das Authentic XML-Dokument kann entweder automatisch oder manuell signiert werden. Die automatische Signierung wird vom SPS-Designer in der Signaturkonfiguration definiert und bewirkt, dass das Authentic XML-Dokument beim Speichern automatisch signiert wird. Wenn die Option zur automatischen Signierung nicht aktiviert wurde, kann das Dokument manuell signiert werden. Klicken Sie dazu auf die Symbolleisten-Schaltfläche "XML-Signatur" oder auf den Befehl **Authentic | XML-Signatur** und klicken Sie anschließend im Dialogfeld "XML-Signatur" (*Abbildung oben*) auf die Schaltfläche **Dokument signieren**. Beachten Sie Folgendes: Damit das Dokument mit einer eingebetteten Signatur signiert werden kann, muss laut Schema das Element `Signature` als das letzte Child-Element des Root (Dokument)-Elements zugelassen sein. Andernfalls ist das Dokument laut Schema ungültig. Das Authentifizierungsobjekt und die Platzierung der Signatur werden gemäß der Signaturkonfiguration gewählt. Sie müssen sicherstellen, dass Sie Zugriff auf die Authentifizierungsinformationen haben. Für nähere Informationen wenden Sie sich an Ihren SPS-Designer.

- *Überprüfen des Authentic-XML-Dokuments:* Wenn XML-Signaturen in einem SPS aktiviert sind, wird die Signatur jedes Mal, wenn das XML-Dokument in der Authentic View-Ansicht geladen wird, überprüft. Wenn die Passwort- bzw. Zertifkat-Schlüsselinformationen nicht mit dem SPS und der Signatur gespeichert werden, wird der Benutzer der Authentic View-Ansicht aufgefordert, das Passwort einzugeben oder ein Zertifikat für die Überprüfung auszuwählen. Beachten Sie, dass eine eingebettete Signatur beim Speichern mit der XML-Datei gespeichert wird.Die generierte Signatur muss explizit

(über die Schaltfläche **Signatur entfernen** des das Dialogfelds "XML-Signatur") entfernt werden (*siehe Abbildung oben)*, wenn sie nicht mit der XML-Datei gespeichert werden soll. Ähnlich verhält es sich auch bei einer separat gespeicherten Signatur. Wenn Sie nicht benötigt wird, muss sie explizit entfernt werden.

© 2016-2022 Altova GmbH

5.8 Grafiken in der Authentic-Ansicht

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

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

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

Relative Pfade

Relative Pfade werden relativ zur SPS-Datei aufgelöst.

5.9 Verwendung von Tasten in der Authentic-Ansicht

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

Anmerkung: Bei Drücken der **Eingabetaste** wird keine neue Zeile angelegt. Dies ist auch dann der Fall, wenn sich der Cursor innerhalb eines Textnode wie z.B. eines Absatzes befindet.

Verwendung der Tastatur

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

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

 © 2016-2022 Altova GmbH

6 Authentic Skripterstellung

Durch die Funktion **Authentic-Skripterstellung** können SPS-Designs flexibler und interaktiver gestaltet werden. Diese Designs können in der StyleVision Enterprise und der Professional Edition erstellt werden und in der Authentic-Ansicht der Enterprise und Professional Edition von Altova-Produkten angezeigt werden.

In der Tabelle unten sehen Sie eine komplette Liste aller Altova-Produkte, die diese Funktion unterstützen. Beachten Sie, dass die interne Skripterstellung in der trusted Version des Authentic Browser Plug-in aus Sicherheitsgründen deaktiviert ist.

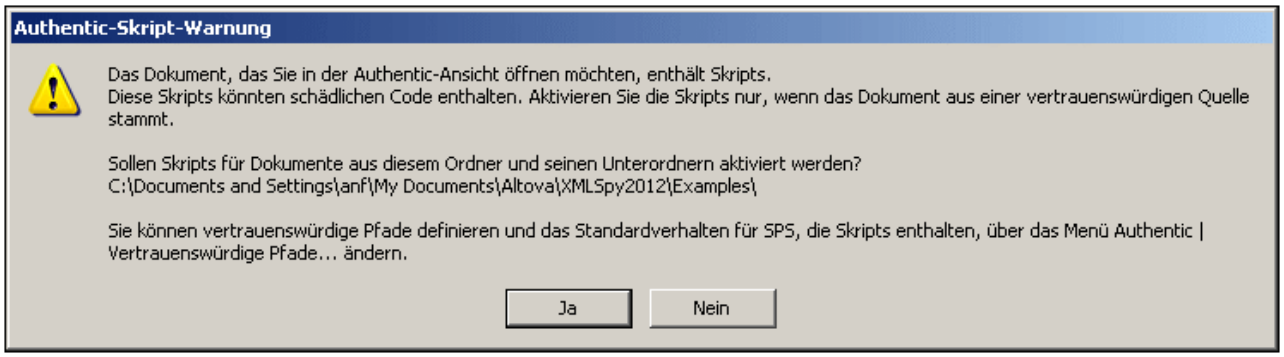
Altova-Produkt	Erstellung von Authentic Skripts	Authentic Skripts aktiv
StyleVision Enterprise	Ja	Ja
StyleVision Professional	Ja	Ja
StyleVision Basic *	Nein	Nein
XMLSpy Enterprise	Nein	Ja
XMLSpy Professional	Nein	Ja
AuthenticDesktop Enterprise	Nein	Ja
Authentic Browser Plug-in Enterprise Trusted **	Nein	Ja
Authentic Browser Plug-in Enterprise Untrusted	Nein	Ja

* *Keine Authentic-Ansicht*
** *Designs mit Skript werden angezeigt. Keine interne Makroausführung oder Event-Behandlung. Externe Events werden ausgelöst.*

Authentic-Skripts verhalten sich in allen Altova-Produkten gleich. Es ist kein produktspezifischer Code und keine produktspezifische Einstellung erforderlich.

Das Dialogfeld "Authentic-Skript-Warnung"

Wenn eine PXF-Datei oder eine mit einem SPS verknüpfte XML-Datei ein Skript enthält und die Datei geöffnet wird bzw. in die Authentic-Ansicht gewechselt wird, so wird ein Warndialogfeld (*Abbildung unten*) angezeigt.



Sie können eine der folgenden Optionen wählen:

- Klicken Sie auf **Ja,** um einen Ordner mit der Datei zur Liste der vertrauenswürdigen Pfade für Authentic-Skripts hinzuzufügen. In der Folge werden alle Dateien unter diesem Pfad in der Authentic-Ansicht geöffnet, ohne dass vorher dieses Warndialogfeld angezeigt wird. Sie können die Liste vertrauenswürdiger Pfade über den Menübefehl **Authentic | Vertrauenswürdige Pfade** 237 aufrufen und ändern.
- Klicken Sie auf **Nein**, damit der Ordner mit der Datei nicht zur Liste der vertrauenswürdigen Pfade hinzugefügt wird. Die Datei wird in der Authentic-Ansicht angezeigt, wobei die Skripts deaktiviert sind. Das Dialogfeld "Authentic-Skript-Warnung" wird jedes Mal, wenn diese Datei in der Authentic-Ansicht geöffnet wird, angezeigt. Um den Ordner der Datei zu einem späteren Zeitpunkt zur Liste der vertrauenswürdigen Pfade hinzuzufügen, öffnen Sie das Dialogfeld "Vertrauenswürdige Pfade" über den Menübefehl **Authentic | Vertrauenswürdige Pfade** 237 und fügen Sie den Ordner hinzu oder ändern Sie die Pfade nach Wunsch.

Eine Beschreibung des Dialogfelds "Vertrauenswürdige Pfade" finden Sie im Abschnitt Menübefehle unter der Beschreibung des Menübefehls **Authentic | Vertrauenswürdige Pfade** 237.

Anmerkung: Bei Aufruf von XMLSpy über die COM-Schnittstelle (Beschreibung dazu siehe Referenz für Programmierer) 295 **wird die Sicherheitsüberprüfung nicht durchgeführt und das Dialogfeld "Authentic-Skript-Warnung" wird nicht angezeigt.**

Funktionsweise der Authentic Skripterstellung

Die Authentic-Skripterstellung kann beim Design von SPS-Dateien auf zwei Arten eingesetzt werden, um Authentic-Dokumente interaktiv zu machen:

- Durch Zuweisen von Skripts für benutzerdefinierte Aktionen (Makros) zu Design-Elementen, Symbolleisten-Schaltflächen und Kontextmenübefehlen.
- Durch Hinzufügen von Event Handlern zum Design, die auf Events der Authentic-Ansicht reagieren.

Alle Skripts, die erforderlich sind, um Authentic-Dokumente interaktiv zu machen, werden über die StyleVision-Benutzeroberfläche (Enterprise und Professional Edition) erstellt. Die Formulare, Makros und Event Handler werden mit dem Skript-Editor von StyleVision erstellt und diese Skripts werden mit dem SPS gespeichert. Anschließend werden die gespeicherten Skripts in der Design-Ansicht von StyleVision Design-Elementen, Symbolleisten-Schaltflächen und Kontextmenüs zugewiesen. Wenn ein XML-Dokument, das auf dem SPS basiert, in einem Altova-Produkt, das die Authentic-Skripterstellung unterstützt (*siehe Tabelle oben),* geöffnet wird, stehen diese zusätzlichen Funktionen im Dokument zur Verfügung.

 © 2016-2022 Altova GmbH

Dokumentation zur Authentic Skripterstellung

Die Dokumentation zur Authentic Skripterstellung finden Sie in der Dokumentation zu StyleVision. Online finde Sie diese auf der Altova Website auf der Seite "Produktdokumentation".

7 Browser-Ansicht

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

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

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

Browser-Programme in der Browser-Ansicht

Standardmäßig wird in der Browser-Ansicht derzeit Microsoft Internet Explorer als Browser verwendet. Wenn Sie lieber den neueren Browser von Microsoft, Edge WebView2, für die Browser-Ansicht verwenden möchten, aktivieren Sie die entsprechende Option im Abschnitt "Ansicht" 275 des Dialogfelds "Optionen" 267.

Anmerkung: Da Microsoft Edge WebView2 das Chromium-Software-Projekt, auf dem der Browser Chrome von Google basiert, verwendet, bietet die Browser-Ansicht bei Verwendung von WebView2 auch eine gute Vorschau auf die Anzeige einer Webseite in Chrome.

Anmerkungen zu Microsoft Internet Explorer

Für die Browser-Ansicht wird Microsoft Internet Explorer 5.0 oder höher oder Microsoft Edge WebView2 benötigt (*siehe oben*).

Beachten Sie zur Verwendung von Internet Explorer in der Browser-Ansicht die folgenden Punkte:

- Zur Anzeige von XML-Dateien, die mit Hilfe eines XSLT Stylesheet transformiert wurden, sollten Sie unbedingt Internet Explorer 6.0 oder höher verwenden, da diese Versionen MSXML 3.0 verwenden, einen XML-Parser, die den XSLT 1.0 Standard vollständig unterstützt. Eventuell sollten Sie auch MSXML 4.0 installieren.
- Die Unterstützung von XSL in IE 5 ist nicht zu 100 % kompatibel mit der offiziellen XSLT Recommendation. Wenn Sie daher in der Browser-Ansicht Probleme mit IE 5 haben, sollten Sie auf IE 6 oder höher upgraden.
- Außerdem sollten Sie im Allgemeinen überprüfen, welche XSLT-Version Ihre Version von Internet Explorer unterstützt.
- Wenn bei der korrekten Anzeige von HTML in Internet Explorer Probleme auftreten, inkludieren Sie den folgenden **meta**-Tag in das **head**-Element Ihres HTML-Dokuments:

```
<head>
... <meta http-equiv="X-UA-Compatible" content="ie=edge">...
</head>
```

Developer Tools in der Browser-Ansicht

Sie können Ihren HTML-Code mit Hilfe der Developer Tools des Edge Browsers analysieren, debuggen und testen. Um die Tools aufzurufen, klicken Sie mit der rechten Maustaste in das Browser-Ansichtsfenster und wählen Sie den Befehl **Developer Tools öffnen**.

 © 2016-2022 Altova GmbH

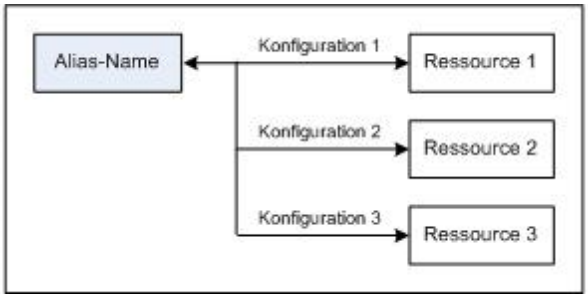
Funktionen der Browser-Ansicht

In der Browser-Ansicht stehen die folgenden Funktionen zur Verfügung. Sie können über das Menü **Browser**, **Datei** und das Menü **Bearbeiten** aufgerufen werden.

- *Öffnen in einem separaten Fenster:* Wenn die Browser-Ansicht ein separates Fenster ist, können Sie die Browser-Ansicht Seite an Seite mit der Bearbeitungsansicht desselben Dokuments anzeigen. Klicken Sie dazu auf den Menübefehl **Browser | Separates Fenster**. Dies ist ein Ein/Aus-Befehl, mit dem Sie zwischen der Browser-Ansicht in (i) einem separaten Fenster und (ii) derselben Ansicht auf einem Register im Hauptfenster wechseln können. Diese Befehle stehen auch im Dropdown-Menü der **Browser-Ansicht** (am unteren Rand des Hauptfensters) zur Verfügung.
- *Vorwärts und Zurück:* die allgemeinen Browser-Befehle zum Navigieren durch Seiten, die in der Browser-Ansicht geladen wurden. Diese Befehle befinden sich im Menü **Browser**.
- *Schriftgröße:* Kann über das Menü **Browser** angepasst werden.
- *Abbrechen, Aktualisieren, Drucken:* Weitere Browser-Standardbefehle. Diese Befehle befinden sich in den Menüs **Browser** und **Datei**.
- *Suchen:* Dient zum Suchen von Textstrings. Dieser Befehl befindet sich im Menü **Bearbeiten**.
- *Info-Fenster:* In diesem Fenster stehen Optionen zur Verfügung, um die aktive HTML-Seite mit einem beliebigen auf dem Rechner installierten Browser anzuzeigen und die installierten Browser zu öffnen oder zu entfernen.

8 Globale Altova-Ressourcen

Bei den globalen Altova-Ressourcen handelt es sich um eine Sammlung von Aliassen für Datei-, Ordner- und Datenbankressourcen. Jeder Alias kann mehrere Konfigurationen haben, wobei jede Konfiguration genau einer Ressource zugeordnet wird (*siehe Abbildung unten*). Bei Verwendung einer globalen Ressource als Input können Sie mittels Steuerelementen auf der Benutzeroberfläche jederzeit zwischen den verschiedenen Konfigurationen für die globale Ressource wechseln.



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

- Definieren globaler Ressourcen 97: Ressourcen werden definiert und die Definitionen werden in einer XML-Datei gespeichert. Diese Ressourcen können mehreren Altova-Applikationen zur Verfügung gestellt werden.
- Verwendung von globalen Ressourcen 109: In Authentic Desktop können Dateien über eine globale Ressource anstatt über einen Dateipfad zugewiesen werden. Der Vorteil ist, dass die verwendete Ressource durch Wechseln der Konfiguration in Authentic Desktop sofort gewechselt werden kann.

Globale Ressourcen in anderen Altova-Produkten

Derzeit können globale Ressourcen in den folgenden Altova-Produkten definiert und verwendet werden: XMLSpy, StyleVision, MapForce,Authentic Desktop, <%MTD%> und DatabaseSpy..

 © 2016-2022 Altova GmbH

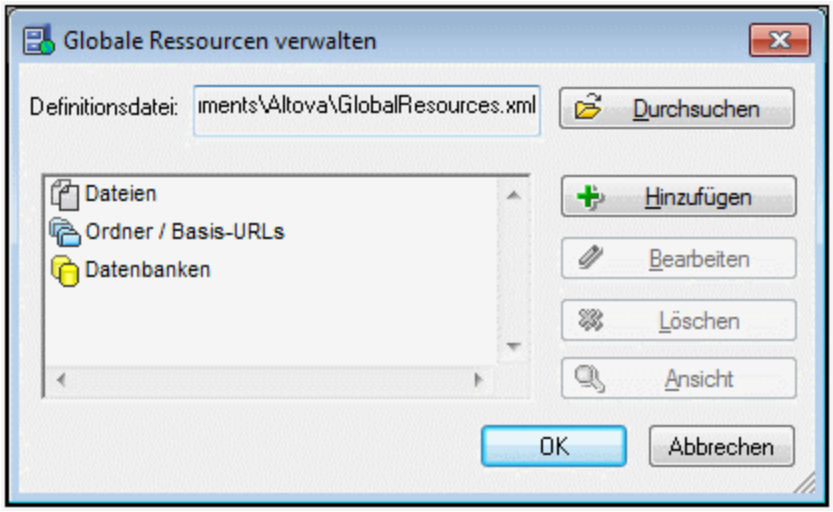
8.1 Definieren globaler Ressourcen

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

- Wählen Sie den Menübefehl **Extras | Globale Ressourcen.**
- Klicken Sie in der Symbolleiste "Globale Ressourcen" auf die Schaltfläche **Globale Ressourcen verwalten** , um das gleichnamige Dialogfeld aufzurufen. (*siehe Abbildung unten*)

Die Definitionsdatei für globale Ressourcen

Die Informationen über globale Ressourcen, werden in einer XML-Datei, der Definitionsdatei für globale Ressourcen, gespeichert. Diese Datei wird erstellt, sobald die erste globale Ressource im Dialogfeld "Globale Ressourcen verwalten" (*Abbildung unten*) definiert und gespeichert wird.



Wenn Sie das Dialogfeld "Globale Ressourcen verwalten" zum ersten Mal öffnen, wird der Standardpfad und -name der Definitionsdatei für globale Ressourcen im Textfeld *Definitionsdatei* (*siehe Abbildung oben)* definiert:

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

Diese Datei ist bei allen Altova-Applikationen als Standard-Definitionsdatei für globale Ressourcen definiert. Infolgedessen kann eine globale Ressource von einer beliebigen Altova-Applikation aus in dieser Datei gespeichert werden und steht dann allen anderen Altova-Applikationen als globale Ressource zur Verfügung. Um eine globale Ressource in der Definitionsdatei für globale Ressourcen zu definieren und zu speichern, fügen Sie die globale Ressource im Dialogfeld "Globale Ressourcen verwalten" hinzu und klicken Sie auf **OK** um sie zu speichern.

Um eine bereits vorhandene Definitionsdatei für globale Ressourcen als aktive Definitionsdatei für eine bestimmte Altova Applikation auszuwählen, suchen Sie im Dialogfeld "Globale Ressourcen verwalten" (*Abbildung unten*) im Feld *Definitionsdatei* danach und wählen Sie sie aus.

Anmerkung: Sie können der Definitionsdatei für globale Ressourcen jeden beliebigen Namen geben und sie unter jedem beliebigen Pfad, auf den Sie von Ihren Altova Applikationen aus Zugriff haben, speichern. Sie müssen diese Datei in Ihrer Applikation nur (im Textfeld *Definitionsdatei*) als die Definitionsdatei für globale Ressourcen für die jeweilige Applikation definieren. Die Ressourcen lassen sich in allen Altova-Produkten als globale Ressourcen verwenden, wenn Sie in allen Altova-Produkten eine einzige Definitionsdatei verwenden.

Anmerkung: Sie können auch mehrere Definitionsdateien für globale Ressourcen erstellen. Es kann aber immer nur eine davon in einer Altova-Applikation aktiv sein und nur die in dieser Datei enthaltenen Definitionen stehen in der Applikation zur Verfügung. Sie können dadurch je nach Bedarf festlegen, welche Ressourcen nur eingeschränkt und welche in mehreren Produkten zur Verfügung stehen sollen.

Verwalten globaler Ressourcen: Hinzufügen, Bearbeiten, Löschen, Speichern

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

Um eine ***globale Ressource hinzuzufügen***, klicken Sie auf die Schaltfläche **Hinzufügen** und definieren Sie die globale Ressource im entsprechenden **Globale Ressourcen**-Dialogfeld, das angezeigt wird (*Beschreibung zu Dateien*[99]*, Ordnern*[104] *und Datenbanken*[97] *in den Unterabschnitten dieses Abschnitts*). Nachdem Sie eine globale Ressource definiert und (durch Klicken auf die Schaltfläche **OK** im Dialogfeld "Globale Ressourcen verwalten") gespeichert haben, wird die globale Ressource zur Bibliothek der globalen Definitionen in der ausgewählten Definitionsdatei für globale Ressourcen hinzugefügt. Die globale Ressource wird durch ein Alias identifiziert.

Um eine ***globale Ressource zu bearbeiten***, wählen Sie sie aus und klicken Sie auf **Bearbeiten**. Daraufhin wird das entsprechende Dialogfeld **Globale Ressource** angezeigt, in dem Sie die nötigen Änderungen vornehmen können (*siehe dazu die Beschreibungen in den Unterabschnitten Dateien*[99]*, Ordner*[104] *in den Unterabschnitten dieses Abschnitts*).

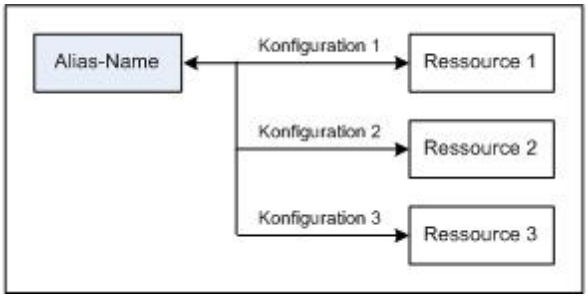
Um eine ***globale Ressource zu löschen***, wählen Sie sie aus und klicken Sie auf **Löschen**.

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

Verknüpfen von globalen Ressourcen mit Aliasnamen über Konfigurationen

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

 © 2016-2022 Altova GmbH



Sie können anschließend in einer Altova-Applikation Aliasnamen anstelle von Dateien zuweisen. Für jeden Alias können Sie zwischen den auf diesen Alias gemappten Ressourcen wechseln, indem Sie einfach die aktive Konfiguration für die globalen Ressourcen (aktive Konfiguration) wechseln. Wenn Sie z.B. in Altova XMLSpy am XML-Dokument `MyXML.xml` eine XSLT-Transformation durchführen möchten, können Sie der globalen Ressource den Alias `MyXSLT` zuweisen, damit diese als globale Ressource für XSLT-Transformationen verwendet wird. Sie können anschließend in XMLSpy die aktive Konfiguration wechseln, um andere XSLT-Dateien zu verwenden. Wenn in `Configuration-1` die Datei `First.xslt` auf `MyXSLT` gemappt ist und `Configuration-1` als aktive Konfiguration ausgewählt wird, so wird für die Transformation `First.xslt` verwendet. Auf diese Art können mehrere Konfigurationen verwendet werden, um mehrere Ressourcen über einen einzigen Alias aufzurufen. Diese Methode ist beim Testen und Vergleichen von Ressourcen nützlich. Da globale Ressourcen außerdem in mehreren Altova Produkten verwendet werden können, können die Ressourcen in mehreren Altova-Produkten getestet und verglichen werden.

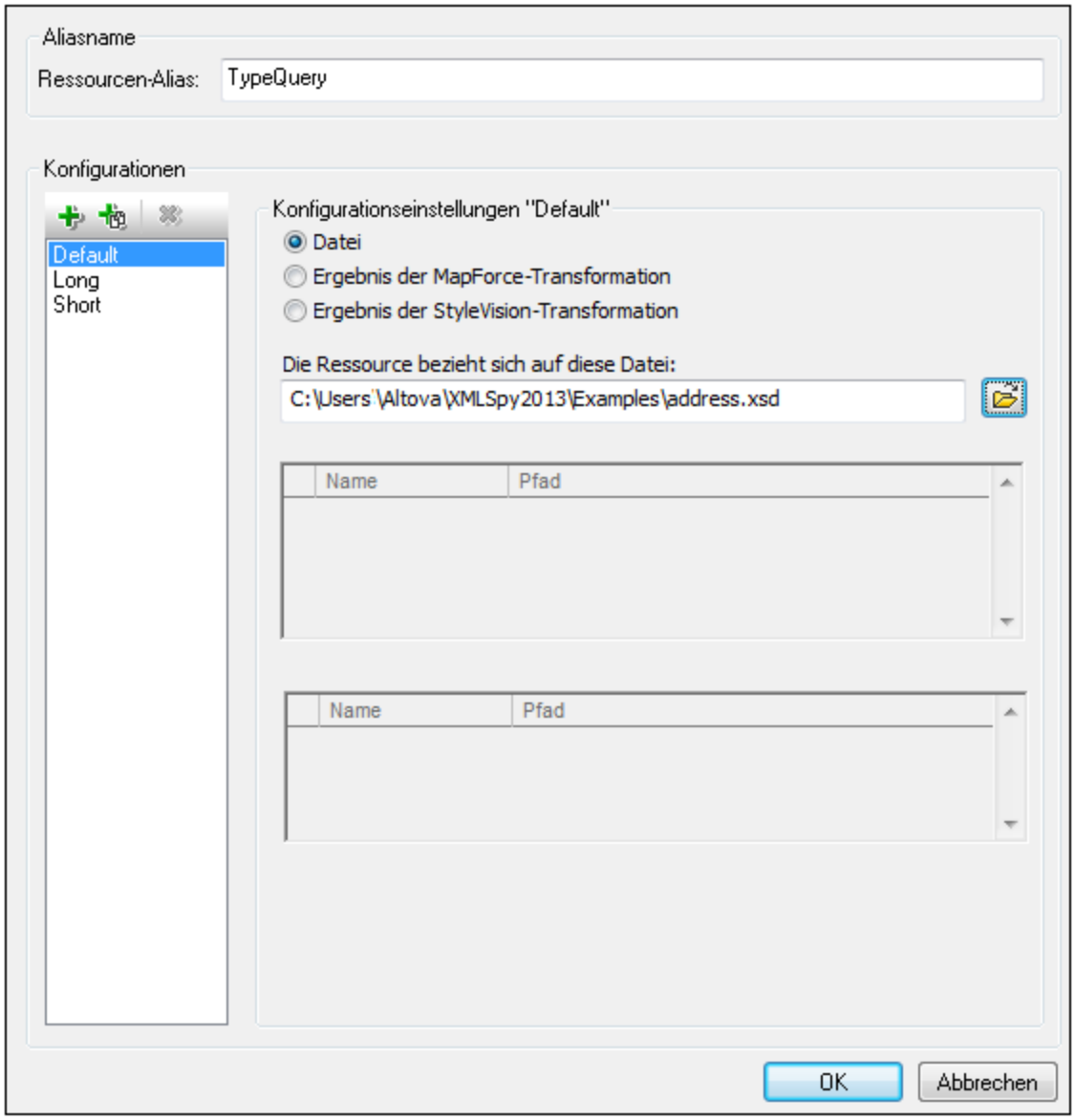
8.1.1 Dateien

Das Dialogfeld **Globale Ressource** für Dateien (*Abbildung unten*) wird über den Befehl **Hinzufügen | Datei** im Dialogfeld "Globale Ressourcen verwalten" 97 aufgerufen. In diesem Dialogfeld können Sie die Konfigurationen des im Textfeld *Ressourcen-Alias* angegebenen Alias definieren. Nachdem Sie die Eigenschaften der Konfiguration wie unten beschrieben definiert haben, speichern Sie die Aliasdefinition durch Klicken auf **OK**.

Nachdem Sie eine Aliasdefinition gespeichert haben, können Sie einen weiteren Alias hinzufügen, indem Sie die oben beschriebenen Schritte (Beginnend durch Aufrufen des Befehls **Hinzufügen | Datei** im Dialogfeld "Globale Ressourcen verwalten") 97 wiederholen.

Dialogfeld "Globale Ressource"

Ein Alias wird im Dialogfeld "Globale Ressource" (*Abbildung unten*) definiert.



Schaltflächen im Dialogfeld "Globale Ressource"

Konfiguration hinzufügen: Ruft das Dialogfeld "Konfiguration hinzufügen" auf, in das Sie den Namen der hinzuzufügenden Konfiguration eingeben.

Konfiguration als Kopie hinzufügen: Ruft das Dialogfeld "Konfiguration hinzufügen" auf, in das Sie den Namen der Konfiguration eingeben, die als Kopie der ausgewählten Konfiguration erstellt werden soll.

Löschen: Löscht die ausgewählte Konfiguration.

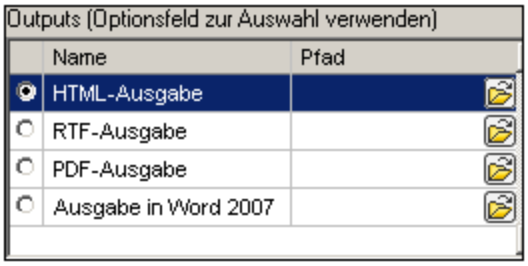
Öffnen: Damit können Sie zur Datei navigieren, die als globale Ressource erstellt werden soll.

© 2016-2022 Altova GmbH

Definieren des Alias

So definieren Sie den Alias (seinen Namen und seine Konfigurationen):

1. *Geben Sie dem Alias einen Namen*: Geben Sie den Aliasnamen in das Textfeld *Ressourcen-Alias* ein.
2. *Fügen Sie Konfigurationen hinzu*: Im Fenster "Konfigurationen" finden Sie als Standardeinstellung eine Konfiguration mit dem Namen "`Default`" (*Abbildung oben*). Diese Standardkonfiguration kann weder gelöscht noch umbenannt werden. Sie können beliebig viele zusätzliche Konfigurationen hinzufügen. (i) Klicken Sie dazu auf die Schaltfläche **Konfiguration hinzufügen** oder auf **Konfiguration als Kopie hinzufügen** und (ii) geben Sie im daraufhin angezeigten Dialogfeld einen Konfigurationsnamen ein. Alle hinzugefügten Konfigurationen werden im Fenster "Konfigurationen" aufgelistet. In der Abbildung oben wurden zwei zusätzliche Konfigurationen namens `Long` und `Short` zur Konfigurationsliste hinzugefügt. Mit Hilfe der Schaltfläche "Konfiguration als Kopie hinzufügen" können Sie die ausgewählte Konfiguration kopieren und anschließend bearbeiten.
3. *Wählen Sie für jede Konfiguration einen Ressourcentyp aus*: Wählen Sie eine Konfiguration in der Konfigurationsliste aus und definieren Sie im Bereich *Konfigurationseinstellungen* eine Ressource für die Konfiguration: (i) Datei, (ii) Ergebnis der MapForce Transformation oder (iii) Ergebnis der StyleVision Transformation. Aktivieren Sie das entsprechende Optionsfeld. Bei Auswahl einer MapForce bzw. StyleVision Transformation wird anhand der entsprechenden `.mfd` oder `.sps`-Datei und der entsprechenden Input-Datei eine Transformation in MapForce oder StyleVision durchgeführt. Die Ressource wird mit der Transformation erzeugt.
4. *Wählen Sie eine Datei für den Ressourcentyp aus:* Wenn es sich bei der Ressource um eine direkt ausgewählte Datei handelt, navigieren Sie im Textfeld *Die Ressource bezieht sich auf diese Datei*: zur gewünschten Datei. Wenn die Ressource das Ergebnis einer Transformation ist, navigieren Sie im Textfeld *Die Ressource bezieht sich auf diese Datei* zur `.mfd` Datei (bei MapForce-Transformationen) oder der `.sps` Datei (bei StyleVision-Transformationen). In Fällen, in denen mehrere Inputs oder Outputs für die Transformation möglich sind, wird eine Auswahl der Optionen angezeigt. So hängen z.B. die Ausgabeoptionen einer StyleVision-Transformationen von der installierten StyleVision Edition ab (*in der Abbildung unten sehen Sie die Output-Optionen der Enterprise Edition*).



   Aktivieren Sie das Optionsfeld für die gewünschte Option (in der Abbildung oben wurde 'HTML-Ausgabe' ausgewählt). Wenn die Ressource das Ergebnis einer Transformation ist, kann die Ausgabe als Datei oder selbst als globale Ressource gespeichert werden. Klicken Sie auf die Schaltfläche und wählen Sie entweder "Globale Ressource" aus (um die Ausgabe als globale Ressource zu speichern) oder "Durchsuchen" (um die Ausgabe als Datei zu speichern). Wenn keine dieser beiden Speicheroptionen ausgewählt ist, wird das Transformationsergebnis als temporäre Datei geladen, wenn die globale Ressource aufgerufen wird.

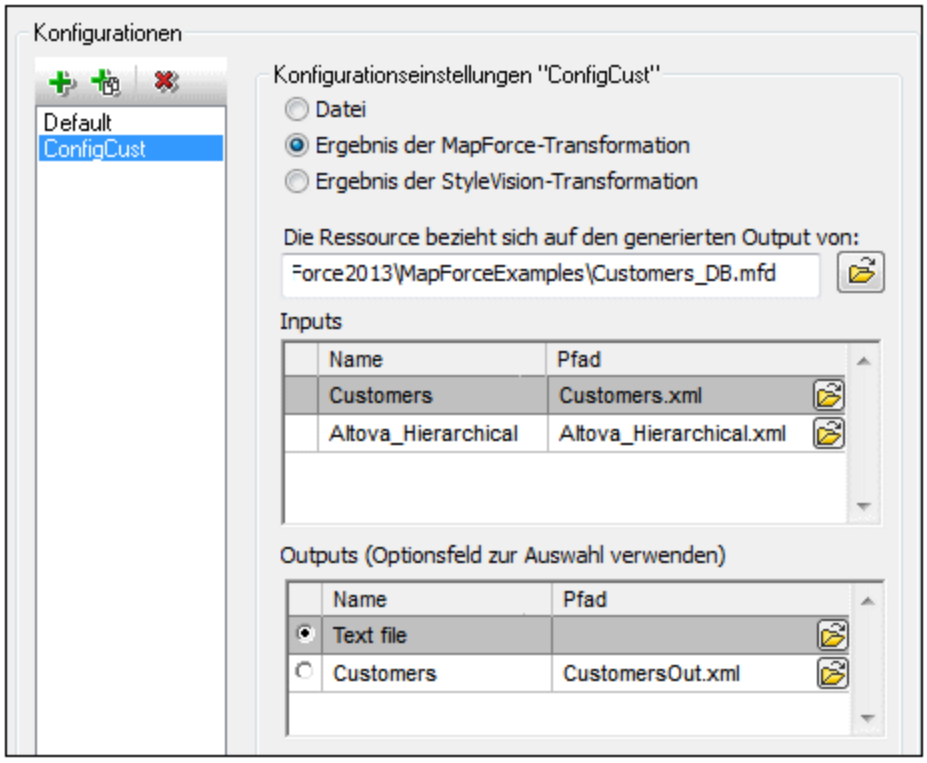
5. *Definieren Sie, falls nötig, mehrere Konfigurationen:* Sie können mehrere Konfigurationen hinzufügen und für jede davon eine Ressource definieren. Wiederholen Sie dazu für jede Konfiguration die Schritte 3 und 4. Sie können jederzeit eine neue Konfiguration zur Aliasdefinition hinzufügen.

6. *Speichern Sie die Aliasdefinition:* Klicken Sie auf **OK,** um den Alias und alle seine Konfigurationen als globale Ressource zu speichern. Die globale Ressource wird im Dialogfeld "Globale Ressourcen verwalten" 97 unter "Dateien" aufgelistet.

Ergebnis einer MapForce-Transformation

Altova MapForce mappt ein oder mehrere (bestehende) Input-Dokument-Schemas auf ein oder mehrere (neue) Output-Dokument-Schemas. Dieses Mapping, das von einem Benutzer in MapForce erstellt wird, wird als MapForce Design (MFD) bezeichnet. Als Datenquellen können XML-Dateien, Textdateien, Datenbanken usw., die den Input-Schemas entsprechen, verwendet werden. MapForce generiert XML-Ausgabedateien, die dem Output-Dokument-Schema entsprechen. Dieses generierte Ausgabedokument ist die Datei, die unter *Ergebnis der MapForce-Transformation*, als globale Ressource verwendet wird.

Wenn Sie eine von MapForce-generierte Datendatei als globale Ressource definieren möchten, müssen Sie im Dialogfeld "Globale Ressource" folgende Einstellungen definieren (*siehe Abbildung unten*):



- ***eine `.mfd` (MapForce Design) Datei.*** Sie müssen diese Datei im Textfeld *Die Ressource bezieht sich auf den generierten Output von:* (*siehe Abbildung oben*) definieren.
- ***eine oder mehrere Input-Datendateien.*** Nachdem die MFD-Datei definiert wurde, wird sie analysiert und im Bereich *Inputs (siehe Abbildung oben*) werden Standarddatendateien angezeigt. Sie können die Auswahl der Standarddatei für jedes Input-Schema ändern, indem Sie eine andere Datei definieren.
- ***eine Output-Datei.*** Wenn das MFD-Dokument mehrere Output-Schemas hat, werden diese alle im Bereich *Outputs* (*siehe Abbildung oben*) aufgelistet und Sie müssen eine davon auswählen. Wenn der Pfad der Ausgabedatei eines einzelnen Output-Schemas im MFD-Dokument definiert ist, so wird dieser Dateipfad für dieses Output-Schema im Bereich *Outputs* angezeigt. In der Abbildung oben sehen wir, dass im MFD-Dokument festgelegt ist, dass das Output-Schema `Customers` eine XML-

 © 2016-2022 Altova GmbH

Standard-Datendatei hat (`CustomersOut.xml`), während das Output-Schema `Text file` in der MFD-Datei keine Dateiverknüpfung hat. Sie können den Standard-Dateipfad im Bereich "Outputs" verwenden oder selbst einen Pfad definieren. Das Ergebnis der MapForce Transformation wird unter dem Dateipfad des ausgewählten Output-Schemas gespeichert. Dies ist die Datei, die als globale Ressource verwendet wird.

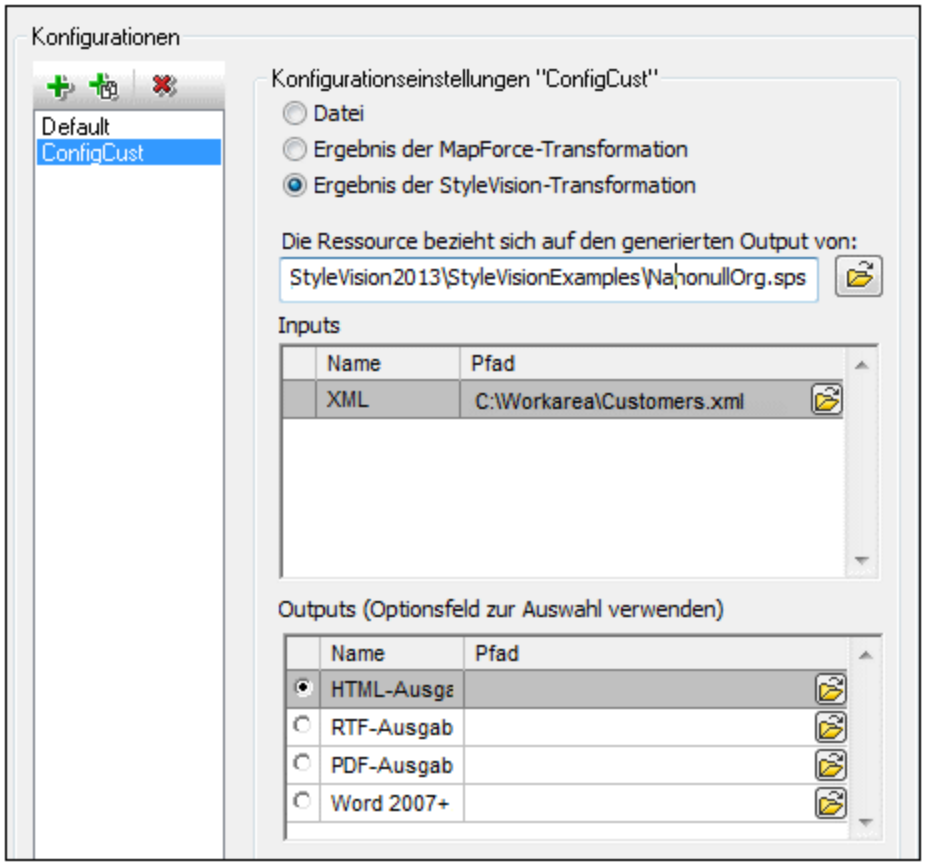
Anmerkung: Der Vorteil dieser Option (Ergebnis der MapForce-Transformation) ist, dass die Transformation zum Zeitpunkt des Aufrufs der globalen Ressource durchgeführt wird. Dadurch enthält die globale Ressource die aktuellsten Daten (aus der/den Input-Datei(en).

Anmerkung: Da zur Ausführung der Transformation MapForce benötigt wird, muss Altova MapForce installiert sein, damit diese Funktion zur Verfügung steht.

Ergebnis der StyleVision Transformation

Mit Altova StyleVision werden StyleVision Power Stylesheet (SPS)-Dateien erzeugt. Mit Hilfe dieser SPS-Dateien werden XSLT-Stylesheets generiert, mit Hilfe derer XML-Dokumente in Ausgabedokumente in verschiedenen Formaten (HTML, PDF, RTF, Word 2007+, usw) transformiert werden. Wenn Sie die Option *Ergebnis der StyleVision Transformation* auswählen, wird das von StyleVision erstellte Ausgabedokument als die globale Ressource verwendet, die mit der ausgewählten Konfiguration verknüpft ist.

Für die Option *Ergebnis der StyleVision Transformation* im Dialogfeld "Globale Ressource" (*siehe Abbildung unten*) müssen die folgenden Dateien definiert werden.



- ***eine*** *.sps* ***(SPS) Datei.*** Diese Datei muss im Textfeld *Die Ressource bezieht sich auf den generierten Output von* (*siehe Abbildung oben*) definiert werden.
- ***Input-Datei(en).*** Eventuell wurde die Input-Datei bereits in der SPS-Datei definiert. In diesem Fall wird sie automatisch im Bereich *Inputs* angezeigt, sobald die SPS-Datei ausgewählt wurde. Sie können diesen Eintrag ändern. Falls kein Input definiert ist, müssen Sie eine Input-Datei hinzufügen.
- ***Output-Datei(en).*** Wählen Sie im Bereich *Outputs* das Ausgabeformat aus und definieren Sie einen Ausgabedateipfad für dieses Format.

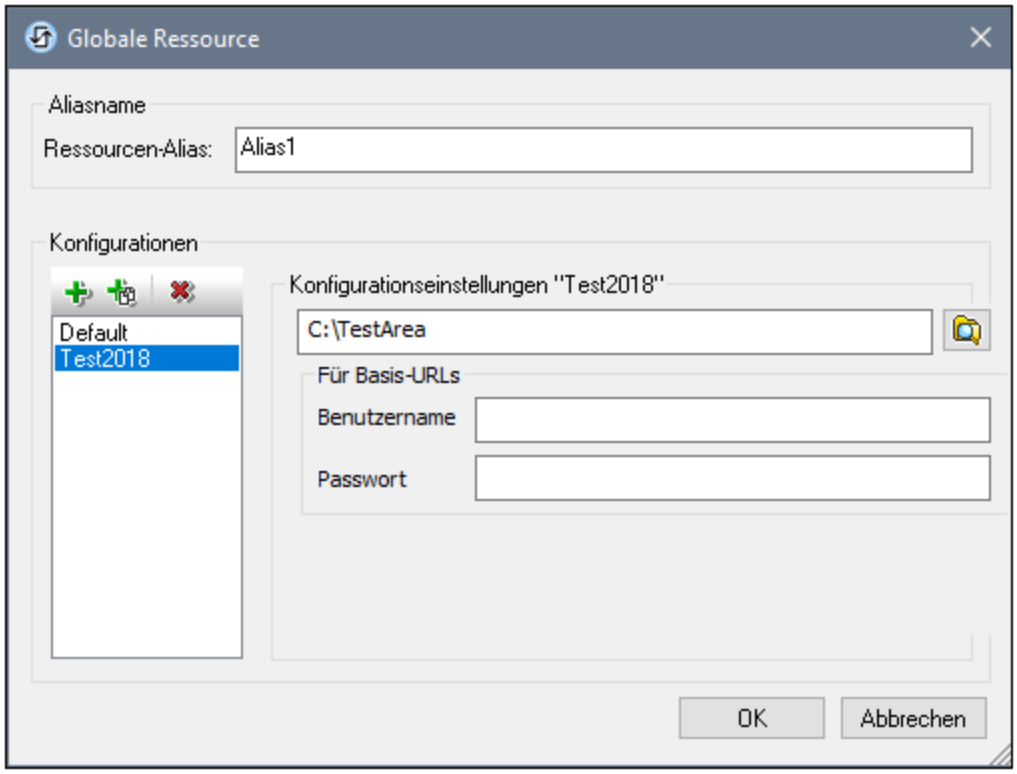
Anmerkung: Der Vorteil dieser Option (Ergebnis der StyleVision-Transformation) ist, dass die Transformation zum Zeitpunkt des Aufrufs der globalen Ressource durchgeführt wird. Dadurch enthält die globale Ressource die aktuellsten Daten (aus der/den Input-Datei(en).

Anmerkung: Da zur Ausführung der Transformation StyleVision benötigt wird, muss Altova StyleVision installiert sein, damit diese Funktion zur Verfügung steht.

8.1.2 Ordner

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

 © 2016-2022 Altova GmbH



Schaltflächen im Dialogfeld "Globale Ressource"

Konfiguration hinzufügen: Ruft das Dialogfeld "Konfiguration hinzufügen" auf, in das Sie den Namen der hinzuzufügenden Konfiguration eingeben.

Konfiguration als Kopie hinzufügen: Ruft das Dialogfeld "Konfiguration hinzufügen" auf, in das Sie den Namen der Konfiguration eingeben, die als Kopie der ausgewählten Konfiguration erstellt werden soll.

Löschen: Löscht die ausgewählte Konfiguration.

Öffnen: Damit können Sie zum Ordner navigieren, der als globale Ressource erstellt werden soll.

Definieren des Alias

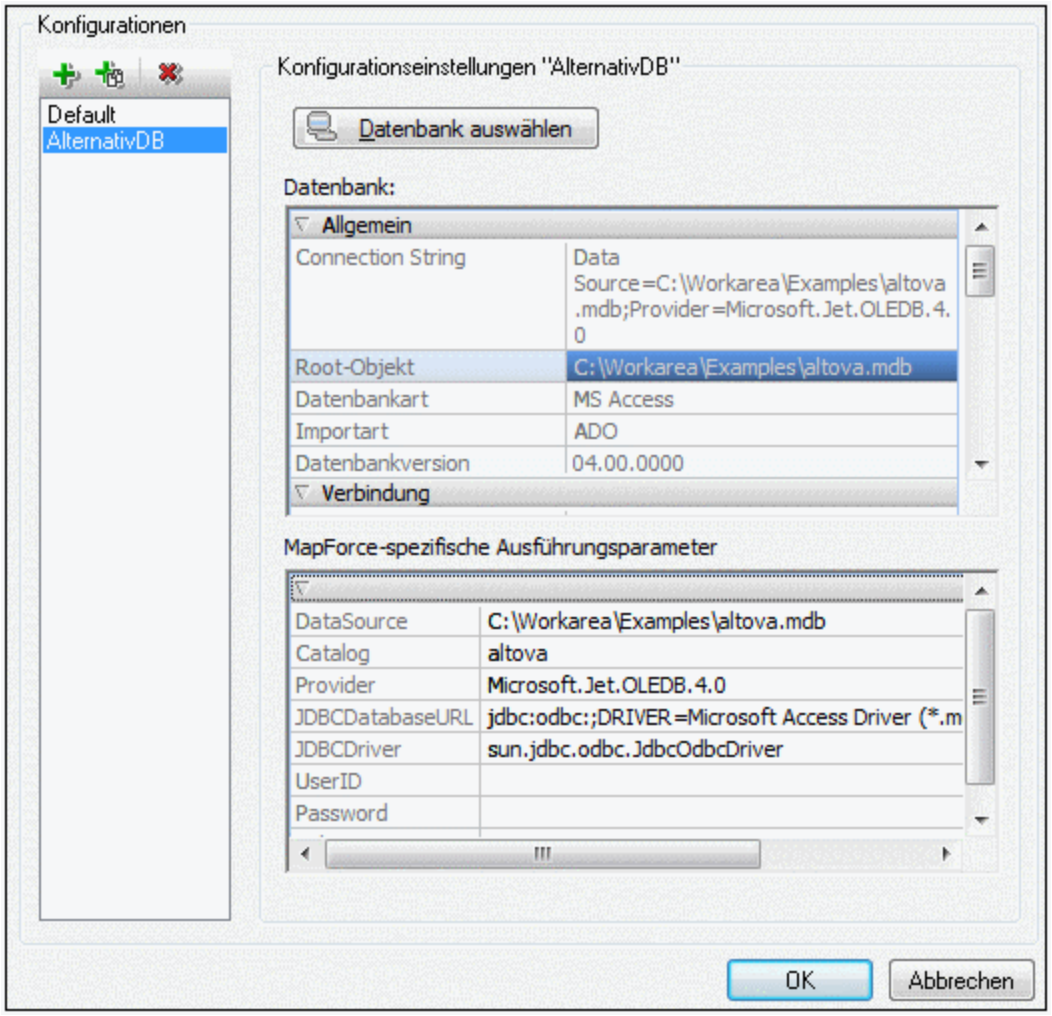
So definieren Sie den Alias (seinen Namen und seine Konfigurationen):

1. *Geben Sie dem Alias einen Namen*: Geben Sie den Aliasnamen in das Textfeld *Ressourcen-Alias* ein.
2. *Fügen Sie Konfigurationen hinzu*: Im Fenster "Konfigurationen" finden Sie eine Konfiguration mit dem Namen "Default" (*Abbildung oben*). Diese Standardkonfiguration kann nicht gelöscht werden und der Name kann nicht geändert werden. Sie können beliebig viele zusätzliche Konfigurationen für den ausgewählten Alias hinzufügen. Klicken Sie dazu auf die Schaltfläche **Konfiguration hinzufügen** oder auf **Konfiguration als Kopie hinzufügen.** Geben Sie im angezeigten Dialogfeld den Konfigurationsnamen ein. Klicken Sie auf **OK**. Die neue Konfiguration wird im Bereich "Konfigurationen" aufgelistet. Wiederholen Sie diesen Schritt um so viele Konfigurationen wie erforderlich hinzuzufügen.

3. *Wählen Sie einen Ordner als Ressource einer Konfiguration aus*: Wählen Sie im Bereich "Konfigurationen" eine der Konfigurationen aus und navigieren Sie zum Ordner, den Sie als globale Ressource erstellen möchten. Wenn für den Zugriff auf einen Ordner Anmeldeinformationen erforderlich sind, definieren Sie diese in den Feldern *Benutzername* und *Passwort*.
4. *Definieren Sie, falls nötig, mehrere Konfigurationen*: Definieren Sie eine Ordnerressource für jede Konfiguration, die Sie erstellt haben (d.h. wiederholen Sie Schritt 3 für die verschiedenen von Ihnen erstellten Konfigurationen). Sie können jederzeit eine neue Konfiguration zur Aliasdefinition hinzufügen.
5. *Speichern Sie die Aliasdefinition*: Klicken Sie im Dialogfeld **Globale Ressource** auf **OK**, um den Alias und alle seine Konfigurationen als globale Ressource zu speichern. Die globale Ressource wird im Dialogfeld Dialogfeld "Globale Ressourcen verwalten" 97 unter "Ordner" aufgelistet.

8.1.3 Datenbanken

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



Schaltflächen im Dialogfeld "Globale Ressource"

Konfiguration hinzufügen: Ruft das Dialogfeld "Konfiguration hinzufügen" auf, in das Sie den Namen der hinzuzufügenden Konfiguration eingeben.

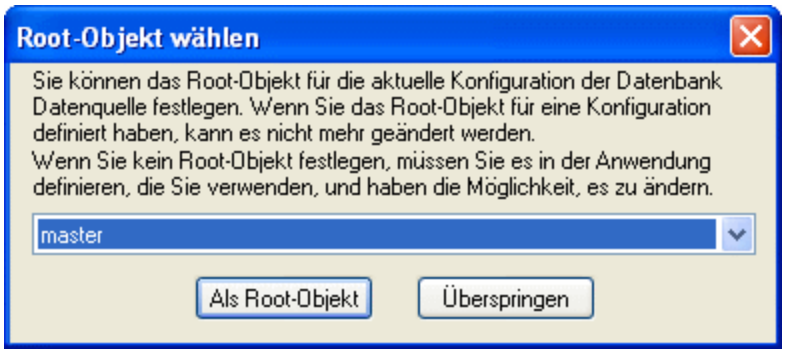
Konfiguration als Kopie hinzufügen: Ruft das Dialogfeld "Konfiguration hinzufügen" auf, in das Sie den Namen der Konfiguration eingeben, die als Kopie der ausgewählten Konfiguration erstellt werden soll.

Löschen: Löscht die ausgewählte Konfiguration.

Definieren des Aliasnamens

So definieren Sie den Alias (seinen Namen und seine Konfigurationen):

1. *Geben Sie dem Alias einen Namen*: Geben Sie den Aliasnamen in das Textfeld *Ressourcen-Alias* ein.
2. *Fügen Sie Konfigurationen hinzu*: Im Fenster "Konfigurationen" finden Sie eine Konfiguration mit dem Namen "Default" (*Abbildung oben*). Diese Standardkonfiguration kann nicht gelöscht werden und der Name kann nicht geändert werden. Sie können beliebig viele zusätzliche Konfigurationen für den ausgewählten Alias hinzufügen. Klicken Sie dazu auf die Schaltfläche **Konfiguration hinzufügen** oder auf **Konfiguration als Kopie hinzufügen.** Geben Sie im angezeigten Dialogfeld den Konfigurationsnamen ein. Klicken Sie auf **OK**. Die neue Konfiguration wird im Bereich "Konfigurationen" aufgelistet. Wiederholen Sie diesen Schritt, um so viele Konfigurationen wie erforderlich hinzuzufügen.
3. *Beginnen Sie mit der Auswahl einer Datenbank als Ressource einer Konfiguration*: Wählen Sie im Bereich "Konfigurationen" eine der Konfigurationen aus und klicken Sie auf die Schaltfläche **Datenbank wählen**. Daraufhin wird das Dialogfeld **Verbindung zu Globaler Ressource erstellen** angezeigt.
4. *Stellen Sie die Datenbankverbindung her*: Wählen Sie aus, ob Sie die Verbindung zur Datenbank mit dem Verbindungsassistenten herstellen, eine bestehende Verbindung verwenden möchten oder eine ADO-Verbindung, eine ODBC- oder eine JDBC-Verbindung herstellen möchten.
5. *Wählen Sie das Root-Objekt aus*: Wenn Sie eine Verbindung zu einem Datenbankserver herstellen, bei dem ein Root-Objekt ausgewählt werden kann, werden Sie im Dialogfeld **Root-Objekt wählen** (*Abbildung unten*) aufgefordert, ein Root-Objekt auf dem Server auszuwählen. Wählen Sie das Root-Objekt aus und klicken Sie auf **Als Root-Objekt.** Das ausgewählte Root-Objekt ist das Root-Objekt, das bei Verwendung der Konfiguration geladen wird.



Klicken Sie auf die Schaltfläche **Überspringen**, wenn alle Root-Objekte in den Anwendungen verfügbar sein sollen. Im letzteren Fall muss der Benutzer das passende Root-Objekt in der entsprechenden Altova Anwendung auswählen, wenn die globale Ressource verwendet wird.

6. *Definieren Sie, falls nötig, mehrere Konfigurationen*: Geben Sie für jede Konfiguration eine Datenbankressource an (d.h. wiederholen Sie Schritt 3 und 4 für die verschiedenen von Ihnen erstellten Konfigurationen). Sie können jederzeit eine neue Konfiguration zur Aliasdefinition hinzufügen.
7. *Speichern Sie die Aliasdefinition*: Klicken Sie im Dialogfeld **Globale Ressource** verwalten" auf **OK**, um den Alias und alle seine Konfigurationen als globale Ressource zu speichern. Die globale Ressource wird im Dialogfeld **Globale Ressourcen verwalten** unter "Datenbanken" aufgelistet.

© 2016-2022 Altova GmbH

8.2 Verwendung globaler Ressourcen

Es gibt mehrere Arten von globalen Ressourcen (Datei, Ordner und Datenbank). Einige Szenarien, in denen Sie globale Ressourcen in Authentic Desktop verwenden können, sind hier aufgelistet: Dateien und Ordner 109.

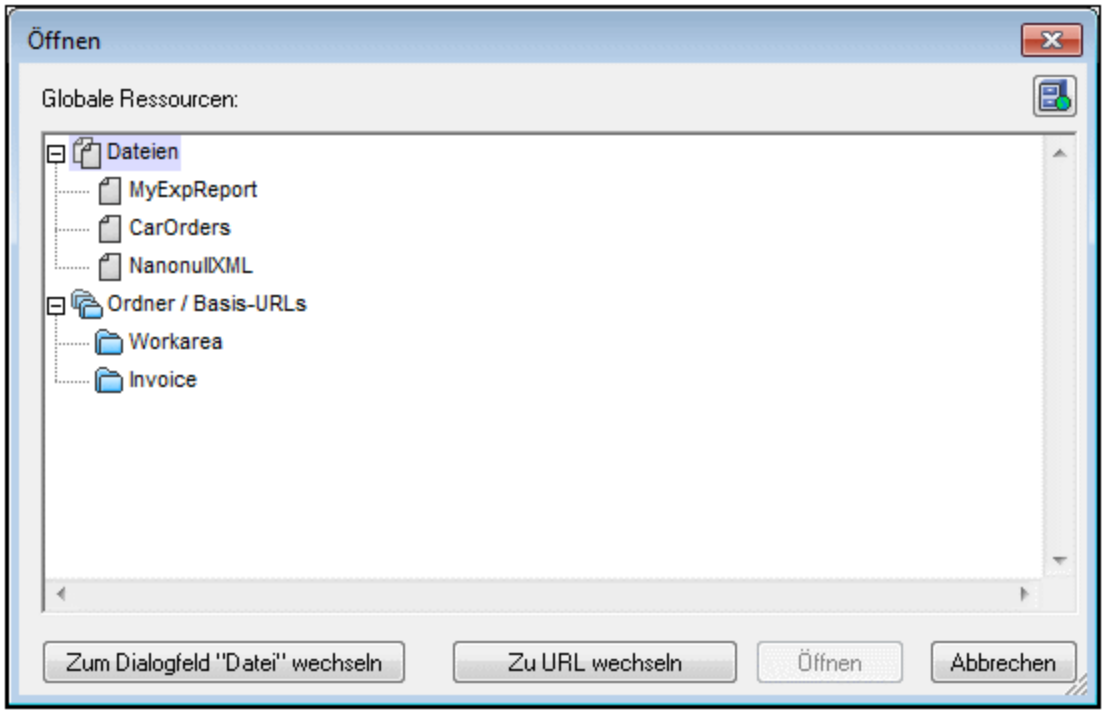
Auswahloptionen, die festlegen, welche Ressource verwendet wird

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

- *Die aktive XML-Datei für globale Ressourcen* wird im Dialogfeld "Globale Ressource" 97 ausgewählt. Die globalen Ressource-Definitionen, die in der aktiven XML-Datei für globale Ressourcen enthalten sind, stehen allen in der Applikation geöffneten Dateien zur Verfügung. Nur die Definitionen in der aktiven XML-Datei für globale Ressourcen sind verfügbar. Sie können jederzeit eine andere XML-Datei für globale Ressourcen verwenden. In diesem Fall werden die globalen Ressourcen der vorher aktiven Datei sofort durch die der neuen aktiven Datei ersetzt. Die aktive XML-Datei für globale Ressourcen legt also Folgendes fest: (i) welche globale Ressourcen zugewiesen werden können und (ii) welche globalen Ressourcen angezeigt werden können (Wenn z.B. eine globale Ressource in einer XML-Datei für globale Ressourcen zugewiesen ist, es aber in der aktiven XML-Datei für globale Ressourcen keine globale Ressource dieses Namens gibt, dann kann die zugewiesene globale Ressource (der Alias) nicht angezeigt werden).
- *Die aktive Konfiguration* wird über den Menübefehl **Extras | Aktive Konfiguration** 249 oder über die Symbolleiste "Globale Ressourcen" ausgewählt. Bei Auswahl dieses Befehls wird eine Liste der Konfigurationen für alle Aliasse angezeigt. Bei Auswahl einer Konfiguration wird diese in der gesamten Applikation aktiv. D.h. immer wenn eine globale Ressource (oder ein Alias) verwendet wird, wird die Ressource, die der aktiven Konfiguration der einzelnen verwendeten Aliasse entspricht, geladen. Die aktive Konfiguration wird auf alle verwendeten Aliasse angewendet. Wenn ein Alias keine Konfiguration mit dem Namen der aktiven Konfiguration hat, wird die Standardkonfiguration dieses Aliasses verwendet. Die aktive Konfiguration spielt beim Zuweisen von Ressourcen keine Rolle; Sie ist nur dann von Bedeutung, wenn die Ressourcen tatsächlich verwendet werden.

8.2.1 Zuweisen von Dateien und Ordnern

Die Zuweisung von globalen Ressourcen vom Typ Datei und Ordner erfolgt auf unterschiedliche Weise. In jedem der unten beschriebenen Szenarios erscheint bei Auswahl des Befehls **Globale Ressourcen** das Dialogfeld "Globale Ressource öffnen" (*Abbildung unten*).



Globale Ressourcen verwalten: Ruft das Dialogfeld Globale Ressourcen verwalten [97] auf.

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

Sie können durch Klicken auf die entsprechende Schaltfläche am unteren Rand des Dialogfelds zum Datei- bzw. URL-Dialogfeld wechseln. Mit der Schaltfläche **Globale Ressourcen verwalten** im rechten oberen Eck erscheint das Dialogfeld Globale Ressourcen verwalten [97].

Verwendungsszenarien

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

- Öffnen globaler Ressourcen [111]
- Speichern als globale Ressource [111]
- XSLT-Transformation [112]

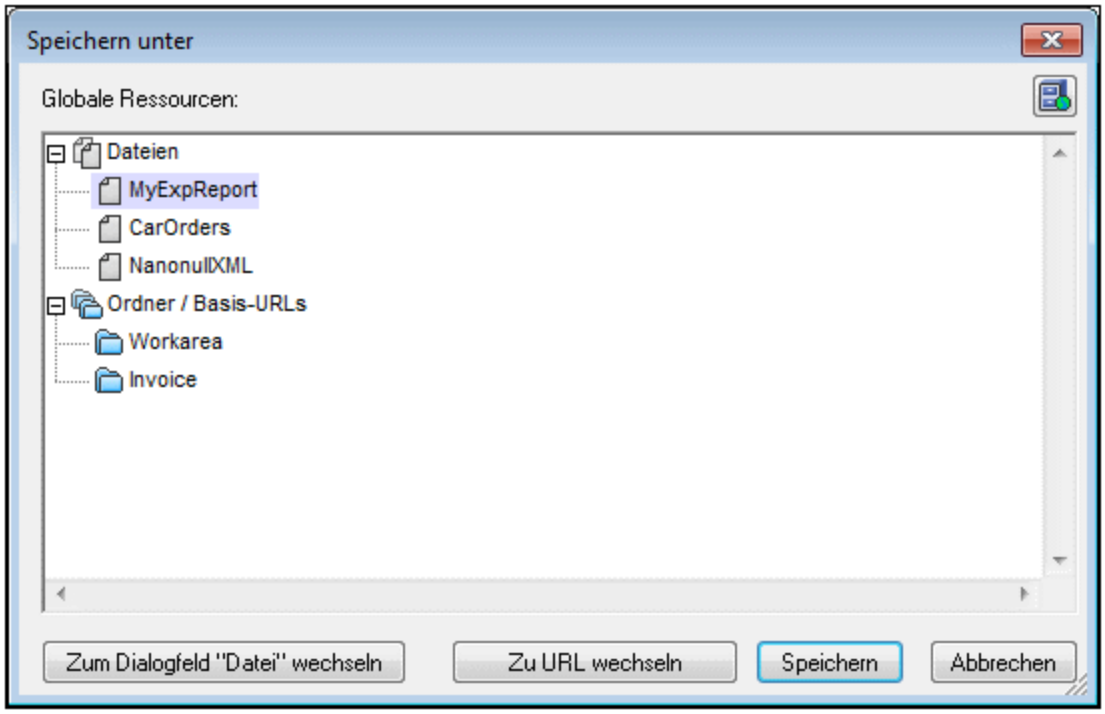
 © 2016-2022 Altova GmbH

Öffnen globaler Ressourcen

Eine globale Ressource kann in Authentic Desktop mit dem Befehl **Datei | Öffnen (Zu globaler Ressource wechseln)** 169 geöffnet und bearbeitet werden. Im Fall einer globalen Ressource vom Typ Datei wird die Datei direkt geöffnet. Im Fall einer globalen Ressource vom Typ Ordner wird ein Öffnen-Dialogfeld aufgerufen, in dem der verknüpfte Ordner ausgewählt ist. Sie können nun in den untergeordneten Ordnern nach der gewünschten Datei suchen. Ein Vorteil der Definition von Dateien mittels globaler Ressourcen für die Bearbeitung ist, dass miteinander in Zusammenhang stehende Dateien unter unterschiedlichen Konfigurationen einer einzigen globalen Ressource gespeichert und nur durch Wechseln der Konfigurationen aufgerufen werden können. Alle Bearbeitungen müssten vor dem Ändern der Konfiguration gespeichert werden.

Speichern als globale Ressource

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



Wählen Sie einen Alias aus und klicken Sie auf **Speichern**. Wenn es sich beim Alias um einen Dateialias 99 handelt, wird die Datei direkt gespeichert. Wenn es sich um einen Ordneralias 104 handelt, wird ein Dialogfeld angezeigt, in dem Sie den Namen angeben müssen, unter dem die Datei gespeichert werden soll. In beiden Fällen wird die Datei unter dem Pfad gespeichert, der für die gerade aktive Konfiguration 112 definiert wurde.

Anmerkung: Jede Konfiguration verweist auf einen bestimmten Dateipfad, der in der Definition dieser Konfiguration festgelegt ist. Wenn die Datei, die Sie als globale Ressource speichern, nicht dieselbe Dateityperweiterung wie die Datei unter dem aktuellen Pfad der Konfiguration hat, kann es beim Öffnen

der globalen Ressource in Authentic Desktop zu Bearbeitungs- und Validierungsfehlern kommen. Der Grund dafür ist, dass Authentic Desktop die Datei unter der Annahme öffnet, dass die Datei den in der Konfigurationsdefinition angegebenen Dateityp aufweist.

XSLT-Transformationen

Bei Auswahl des Befehls **XSL/XQuery | XSL-Transformation** 221 oder**XSL/XQuery | XSL:FO-Transformation** 222 wird ein Dialogfeld angezeigt, in dem Sie nach der gewünschten XSLT- oder XML-Datei suchen können. Klicken Sie auf die Schaltfläche **Durchsuchen** und anschließend auf **Globale Ressource** um das Dialogfeld "Globale Ressource öffnen" aufzurufen (*Abbildung ganz oben im Abschnitt* 109). Für die Transformation wird die Datei verwendet, die mit der aktuelle aktiven Konfiguration der ausgewählten globalen Ressource verknüpft ist.

8.2.2 Wechseln der aktiven Konfiguration

Es kann immer nur eine Konfiguration einer globalen Ressource aktiv sein. Diese Konfiguration wird als die aktive Konfiguration bezeichnet. Sie ist applikationsweit aktiv. D.h. die aktive Konfiguration ist für alle globalen Ressourcenaliasse in allen derzeit offenen Dateien und Datenquellenverbindungen aktiv. Wenn ein Alias keine Konfiguration mit dem Namen der aktiven Konfiguration hat, wird die Standardkonfiguration für diesen Alias verwendet. Betrachten wir als Beispiel, wie man eine Konfiguration wechselt, einen Fall, in dem eine Datei über eine globale Ressource mit mehreren Konfigurationen zugewiesen wurde. Jeder Konfiguration ist eine andere Datei zugewiesen. Welche Datei ausgewählt ist, hängt daher davon ab, welche Konfiguration als die aktive Konfiguration der Applikation ausgewählt ist.

Sie können die aktive Konfiguration auf folgende Arten wechseln:

- Über den Menübefehl **Extras | Aktive Konfiguration.** Wählen Sie die gewünschte Konfiguration aus dem Untermenü des Befehls aus.
- Wählen Sie in der Auswahlliste der Symbolleiste "Globale Ressourcen" (*Abbildung unten*) die gewünschte Konfiguration aus.

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

 © 2016-2022 Altova GmbH

9 Versionskontrolle

Die Versionskontrollunterstützung in Authentic Desktop steht über die Microsoft Source Control Plug-in API (vormals bekannt als MSSCCI API), Version 1.1, 1.2 sowie 1.3 zur Verfügung. Dadurch können Sie Versionskontrollbefehle wie z.B. "Einchecken" oder "Auschecken" direkt über Authentic Desktop an praktisch jedem Versionskontrollsystem ausführen, das nativen Clients oder Drittanbieter-Clients über das Microsoft Source Control Plug-in API die Verbindung zu Authentic Desktop gestattet.

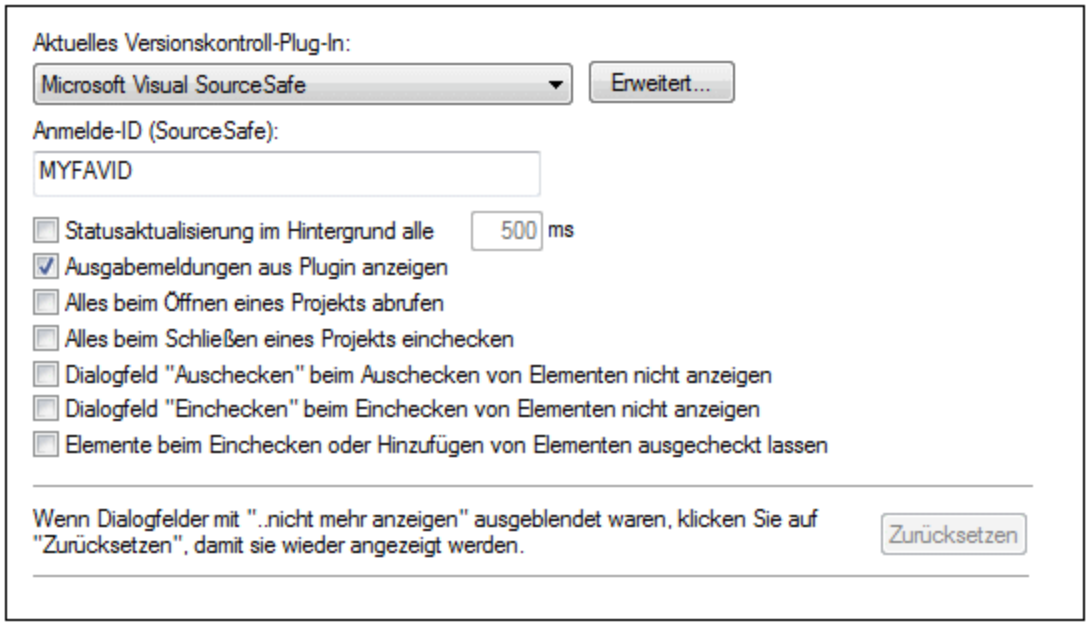
Als Versionskontrollsystemanbieter kann jedes kommerzielle oder nicht kommerzielle Plug-in, das die Microsoft Source Control Plug-in API unterstützt und mit einem kompatiblen Versionskontrollsystem verbunden werden kann, verwendet werden. Eine Liste von von Altova getesteten Versionskontrollsystemen und Plug-ins finden Sie unter Unterstützte Versionskontrollsysteme 116.

Installieren und Konfigurieren des Versionskontrollanbieters

Um die auf Ihrem System verfügbaren Versionskontrollsysteme anzuzeigen, gehen Sie folgendermaßen vor:

1. Klicken Sie im Menü **Extras** auf **Optionen**.
2. Klicken Sie auf das Register **Versionskontrolle**.

Alle mit der Microsoft Source Code Control Plug-in API kompatiblen Versionskontroll-Plug-ins werden in der Dropdown-Liste **Aktuelles Versionskontroll-Plug-in** angezeigt.



Wenn auf Ihrem System kein kompatibles Plug-in gefunden wurde, wird die folgende Meldung angezeigt:

```
"Die Registrierung des installierten Versionskontrollproviders konnte nicht gefunden werden
oder ist unvollständig."
```

Bei einigen Versionskontrollsystemen wird das Versionskontroll-Plug-in nicht automatisch installiert. In diesem Fall müssen Sie es gesondert installieren. Eine genauere Anleitung dazu finden Sie in der Dokumentation zum

jeweiligen Versionskontrollsystem. Ein Plug-in (Anbieter), das mit der Microsoft Source Code Control Plug-in API kompatibel ist, sollte auf Ihrem Betriebssystem unter dem folgenden Registrierdateieintrag zu finden sein:

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

Wenn das Plug-in korrekt installiert wurde, steht es automatisch in der Liste der Plug-ins in Authentic Desktop zur Verfügung.

Aufrufen der Versionskontrollbefehle

Die Versionskontrollbefehle stehen über das Menü **Projekt | Versionskontrolle** zur Verfügung.

Probleme im Zusammenhang mit den Ressourcen / der Geschwindigkeit

Bei sehr großen Versionskontrolldatenbanken kann die automatische Durchführung von Statusaktualisierungen im Hintergrund etwas länger dauern.

Um schneller arbeiten zu können, können Sie versuchen, auf dem Register "Versionskontrolle" (Aufruf über **Extras | Optionen**) das Feld "**Statusaktualisierung im Hintergrund alle xxx Sek**." zu deaktivieren oder das Aktualisierungsintervall zu vergrößern.

Anmerkung: Die **64-Bit**-Version Ihrer Altova-Applikation unterstützt automatisch alle in dieser Dokumentation aufgelisteten 32-Bit-Versionskontrollsysteme. Bei Verwendung einer 64-Bit-Altova-Applikation zusammen mit einem 32-Bit-Versionskontrollsystem ist die Option **Statusaktualisierung im Hintergrund alle xxx Sek** automatisch deaktiviert und kann nicht ausgewählt werden.

Vergleich mit Altova DiffDog

Sie können viele Versionskontrollsysteme (einschließlich Git und TortoiseSVN) für die Verwendung mit Altova DiffDog als Vergleichstool konfigurieren. Nähere Informationen zu DiffDog finden Sie unter https://www.altova.com/de/diffdog.html. Die Dokumentation zu DiffDog finden Sie unter https://www.altova.com/de/documentation.html.

© 2016-2022 Altova GmbH

9.1 Einrichten der Versionskontrolle

So richten Sie eine Versionskontrolle ein und stellen Dateien in einem Authentic Desktop-Projekt unter eine Versionskontrolle:

1. Installieren Sie ein Versionskontrollprogramm (siehe Unterstützte Versionskontrollsysteme 116), falls noch keines installiert ist. Richten Sie die Versionskontrolldatenbank (das Repository), in der Sie Ihre Arbeit speichern möchten, ein.
2. Erstellen Sie einen Ordner für den lokalen Arbeitsbereich, der die Arbeitsdateien enthalten soll, die unter Versionskontrolle gestellt werden sollen. Der Ordner, der alle Ihre Arbeitsbereichordner und -Dateien enthält, wird als ihr lokaler Ordner und der Pfad zum lokalen Ordner als der lokale Pfad bezeichnet. Der Ordner wird an einen bestimmten Ordner im Repository gebunden.
3. Erstellen Sie in Ihrer Altova-Applikation einen Applikationsprojektordner, zu dem Sie die unter Versionskontrolle zu stellenden Dateien hinzufügen müssen. Diese Gliederung der Dateien in einem Applikationsprojekt ist abstrakt. Die Dateien in einem Projekt referenzieren physische lokal (vorzugsweise in einem einzigen Ordner, falls nötig mit Unterordnern) gespeicherte Dateien.
4. In der Datenbank des Versionskontrollsystems (die auch als Versionskontrolle oder Repository bezeichnet wird) wird ein Ordner erstellt, der an den lokalen Ordner gebunden ist. Dieser (als gebundener Ordner bezeichnete) Ordner repliziert die Struktur des lokalen Ordners, sodass sich alle unter Versionskontrolle zu stellenden Dateien an der richtigen Stelle im gebundenen Ordner befinden. Der gebundene Ordner wird normalerweise erstellt, wenn Sie zum ersten Mal eine Datei oder ein Applikationsprojekt zur Versionskontrolle hinzufügen. Nähere Informationen zur Ordnersturktur des Repository finden Sie im Abschnitt Applikationsprojekt 119.
5. Projektdateien werden mit dem Befehl**Projekt | Versionskontrolle | Zu Versionskontrolle hinzufügen** zur Versionskontrolle hinzugefügt. Wenn Sie zum ersten Mal eine Datei oder ein Applikationsprojekt zur Versionskontrolle hinzufügen, werden die korrekten Bindings erstellt und die Ordnerstruktur wird im Repository erstellt.
6. Versionskontrollaktionen wie z.B. das Ein- und Auschecken von Dateien und das Entfernen von Dateien aus der Versionskontrolle können über Befehle im Untermenü **Projekt | Versionskontrolle** ausgeführt werden. Diese Befehle werden in der Benutzerreferenz in den Unterabschnitten des Menüs "Projekt" 191 beschrieben.

Anmerkung: Sie können den aktuellen Versionskontrollanbieter auf zwei Arten wechseln: (i) über die Versionskontrolloptionen (**Extras | Optionen | Versionskontrolle** 280)), oder (ii) im Dialogfeld "Versionskontrolle wechseln" (**Projekt | Versionskontrolle | Versionskontrolle wechseln**).

9.2 Unterstützte Versionskontrollsysteme

In der nachfolgenden Liste sind die von Authentic Desktop unterstützten Versionskontrollsysteme zusammen mit ihren entsprechenden Versionskontroll-Clients (SCCs) aufgelistet. Die Liste ist alphabetisch nach Versionskontrollsystem geordnet.

- Altova hat in Authentic Desktop die Microsoft Source Control Plug-in API (Version 1.1, 1.2 und 1.3) implementiert und die Unterstützung für die aufgelisteten Treiber und Versionskontrollsysteme getestet. Aller Voraussicht nach wird Authentic Desktop diese Produkte, falls diese aktualisiert werden, weiterhin unterstützen.
- Auch Versionskontroll-Clients, die nicht in der nachstehenden Liste erwähnt sind, die aber die Microsoft Source Control Plug-in API implementieren, sollten ebenfalls mit Authentic Desktop funktionieren.

Versionskontrollsystem	Versionskontroll-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 für 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 (siehe Versionskontrolle mit Git 134)
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

© 2016-2022 Altova GmbH

Versionskontrollsystem	Versionskontroll-Clients
Mercurial 1.0.2 für Windows	Sergey Antonov HgSCC 1.0.1
Microsoft SourceSafe 2005 mit CTP	Microsoft SourceSafe 2005 mit CTP
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 für Windows 2009.0.0	Seapine Surround SCM Client 2009.0.0
Serena Dimensions Express/CM 10.1.3 für 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

9.3 Lokaler Arbeitsbereichordner

Die Dateien, mit denen Sie arbeiten werden, sollten innerhalb eines lokalen Arbeitsbereichordners in Form einer Hierarchie gespeichert werden (*siehe Diagramm unten*).

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

Die Applikationsprojektdatei (`.spp` Datei) befindet sich normalerweise direkt innerhalb des loalen Arbeitsbereichordners (*siehe obiges Diagramm*).

Wenn eine oder mehrere Dateien dieses (Arbeitsbereichs)ordners unter Versionskontrolle gestellt werden, wird die Struktur des lokalen Arbeitzsbereichsordners zum Teil oder zur Gänze im Repository nachvollzogen. Wenn z.B. die Datei `Persons.xml` aus dem oben gezeigten lokalen Ordner unter Versionskontrolle gestellt wird, so lautet der Pfad zu dieser Datei im Repository:

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

Der Ordner `MyProject` im Repository-Ordner ist an den lokalen Ordner gebunden. Normalerweise erhält er den Namens des Projekts, kann aber auch jeden anderen Namen erhalten.

Wenn das gesamte Applikationsprojekt (durch Auswahl des Projektnamens im Fenster "Projekt" und setzen des Projekts unter Versionskontrolle) unter Versionskontrolle gestellt wird, so wird die gesamte lokale Ordnerstruktur auch im Repository angelegt.

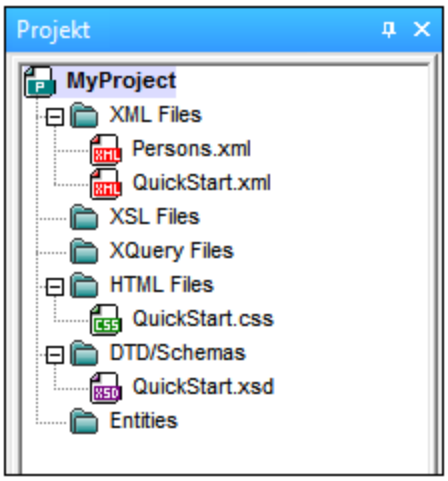
Anmerkung: Es können auch Dateien, die außerhalb des lokalen Arbeitsbereichsordners gespeichert sind, zum Applikationsprojekt hinzugefügt werden. Ob Sie eine solche Datei unter Versionskontrolle stellen können, hängt vom verwendeten Versionskontrollsystem ab und lässt sich nicht bei allen Versionskontrollsystemen bewerkstelligen. Es wird daher empfohlen, dass sich alle Projektdateien, die unter Versionskontrolle gestellt werden sollen, im lokalen Arbeitsbereichsordner befinden sollten.

© 2016-2022 Altova GmbH

9.4 Applikationsprojekt

Erstellen oder laden Sie das Altova Applikationsprojekt, das Sie unter Versionskontrolle stellen möchten. Wenn Sie eine einzige Datei unter Versionskontrolle stellen möchten, sollten Sie sie in ein Projekt inkludieren, da die Versionskontrolle nur über ein Projekt aufgerufen werden kann.

Betrachten wir z.B. ein Projekt in der Altova Applikation XMLSpy. Die Eigenschaften des Projekts sind in einer `.spp` Datei gespeichert. Das Projekt wird in der Applikation im Fenster "Projekt" (*siehe Abbildung unten*) angezeigt. Das Projekt in der Abbildung unten hat den Namen `MyProject` und die Eigenschaften des Projekts sind in der Datei `MyProject.spp` gespeichert.



Sie können das gesamte Projekt (alle Dateien im Projekt) oder nur einige Projektdateien unter Versionskontrolle stellen. **Es können nur Dateien, die sich im Projekt befinden, unter Versionskontrolle gestellt werden.** Sie müssen daher Dateien zum Projekt hinzufügen, bevor Sie sie unter Versionskontrolle stellen können. Die Projektdatei (`.spp` Datei) wird automatisch unter Versionskontrolle gestellt, sobald eine Datei aus dem Projekt unter Versionskontrolle gestellt wird.

Mit dem Befehl **Projekt | Versionskontrolle | Zu Versionskontrolle hinzufügen** (*siehe nächster Abschnitt weiter unten*) wird das gesamte Projekt oder eine oder mehrere Projektdateien unter Versionskontrolle gestellt.

Beachten Sie jedoch, dass die Ordnerstruktur des Repository nicht der Ordnerstruktur des Projekts (*Abbildung oben*), sondern der Struktur des lokalen Arbeitsbereichordners 118 (*siehe Ordnerdiagramm unten*) entspricht. Beachten Sie, dass der Ordner `MyProject` im Diagramm unten im Repository eine Ordnerstruktur hat, die der des lokalen Arbeitsbereichordners entspricht. Beachten Sie außerdem, dass sich der gebundene Ordner innerhalb des Repository-Ordners befindet.

```
Lokaler Arbeitsbereichsordner        Repository
|                                    |
|-- MyProject.spp                    |-- MyProject (gebunden an lokalen
                                     Arbeitsbereich)
|-- QuickStart                       ||-- MyProject.spp
|   |-- QuickStart.css               ||-- QuickStart
|   |-- QuickStart.xml               ||   |-- QuickStart.css
```

```
|   |-- QuickStart.xsd
|-- Grouping
|   |-- Persons
|   |   |-- Persons.xml
```

```
||   |-- QuickStart.xml
||   |-- QuickStart.xsd
||-- Grouping
||   |-- Persons
||   |   |-- Persons.xml
```

Anmerkung: Ein Applikationsprojekt kann Projektordner (grün) und externe Ordner (gelb) enthalten. Nur Dateien in (grünen) Projektordnern können unter Versionskontrolle gestellt werden. Dateien in (gelben) externen Ordnern können nicht unter Versionskontrolle gestellt werden.

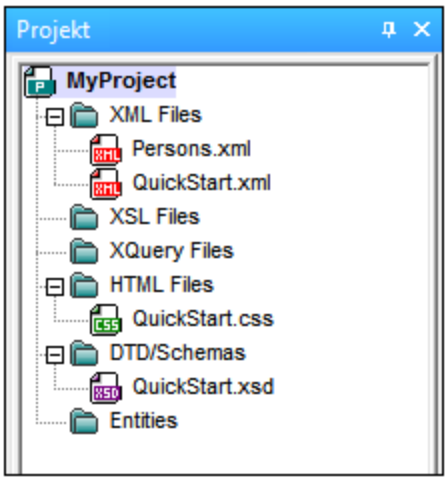
Anmerkung: Es können auch Dateien, die außerhalb des lokalen Arbeitsbereichsordners gespeichert sind, zum Applikationsprojekt hinzugefügt werden. Ob Sie eine solche Datei unter Versionskontrolle stellen können, hängt vom verwendeten Versionskontrollsystem ab und lässt sich nicht bei allen Versionskontrollsystemen bewerkstelligen. Es wird daher empfohlen, dass sich alle Projektdateien, die unter Versionskontrolle gestellt werden sollen, im lokalen Arbeitsbereichsordner befinden sollten.

© 2016-2022 Altova GmbH

9.5 Zu Versionskontrolle hinzufügen

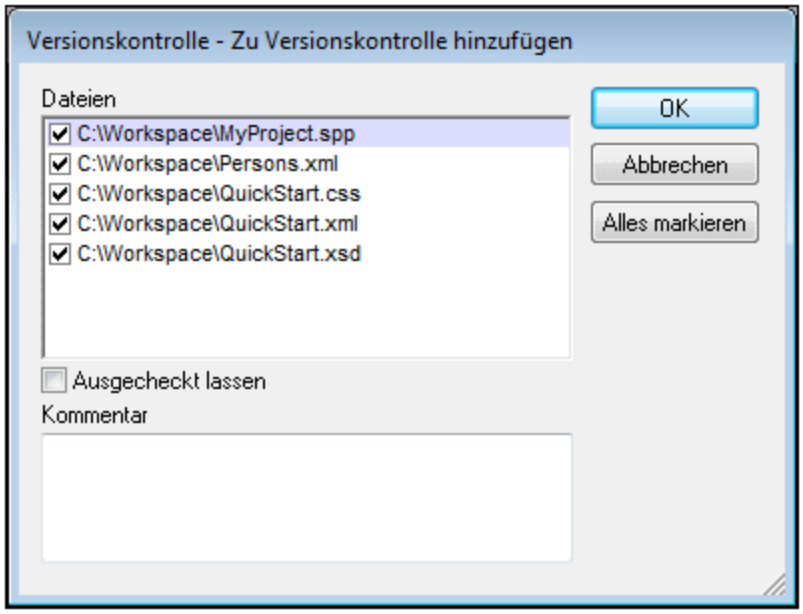
Wenn Sie das Projekt zur Versionskontrolle hinzufügen, werden automatisch die richtigen Bindings und die korrekte Repository-Struktur erstellt, bevor die Projektdatei (`.spp` Datei) oder einzelne Dateien zur Versionskontrolle hinzugefügt werden. Um ein Projekt zur Versionskontrolle hinzuzufügen, gehen Sie folgendermaßen vor.

Wählen Sie das Projekt im Fenster "Projekt" aus (in der Abbildung unten `MyProject`), so dass es markiert erscheint (*wie in der Abbildung unten*). Wählen Sie alternativ dazu eine einzige Datei aus oder halten Sie beim Anklicken der Dateien die **Strg**-Taste gedrückt, um mehrere auszuwählen. Wenn Sie eine einzige Datei zur Versionskontrolle hinzufügen, wird auch automatisch die Projektdatei (`.spp` Datei) zur Versionskontrolle hinzugefügt.



Wählen Sie als nächstes den Menübefehl **Projekt | Versionskontrolle | Zu Versionskontrolle hinzufügen**. Daraufhin werden die Dialogfelder zur Herstellung und Konfigurierung der Verbindung zum aktuell ausgewählten Versionskontrollsystem angezeigt. (Sie können das Versionskontrollsystem über das Dialogfeld "Versionskontrolle wechseln" (**Projekt | Versionskontrolle | Versionskontrolle wechseln**) wechseln).

Befolgen Sie die Anweisungen des Versionskontrollsystems um die Verbindung herzustellen und das System zu konfigurieren. Anschließend werden alle zum Hinzufügen ausgewählten Dateien plus die Projektdatei (`.spp` Datei) im Dialogfeld "Zu Versionskontrolle hinzufügen" (*Abbildung unten*) angezeigt. Wählen Sie die gewünschten Dateien aus und klicken Sie auf **OK**.



Die Dateien werden zum Repository hinzugefügt und je nachdem, ob das Kontrollkästchen *Ausgecheckt lassen* aktiviert wurde oder nicht, entweder ein- oder ausgecheckt [125].

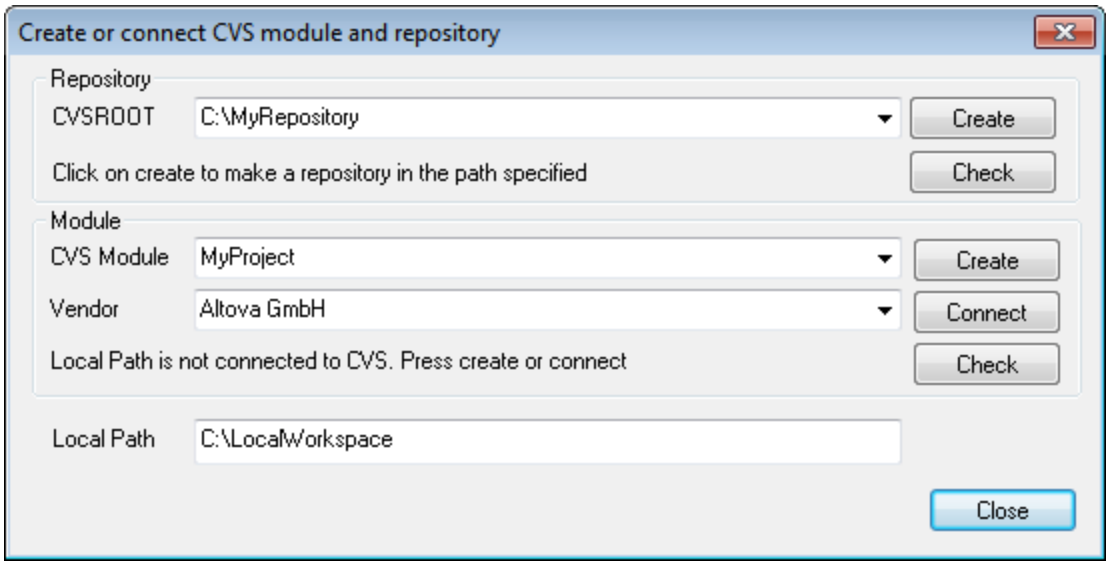
Anmerkungen zur Konfiguration

Eventuell werden Sie aufgefordert, im Repository einen Ordner für das Projekt zu erstellen, falls dies noch nicht geschehen ist. Nachdem Sie diesen erstellt haben, wird der lokale Arbeitsbereichordner [118] an diesen im Repository erstellten Ordner gebunden (*siehe Diagramme unten*).

```
Lokaler Arbeitsbereichsordner        Repository
|                                    |
|-- MyProject.spp                    |-- MyProject (gebunden an lokalen
                                     Arbeitsbereich)
|-- QuickStart                       ||-- MyProject.spp
|   |-- QuickStart.css               ||-- QuickStart
|   |-- QuickStart.xml               ||   |-- QuickStart.css
|   |-- QuickStart.xsd               ||   |-- QuickStart.xml
|-- Grouping                         ||   |-- QuickStart.xsd
|   |-- Persons                      ||-- Grouping
|   |   |-- Persons.xml              ||   |-- Persons
                                     ||   |   |-- Persons.xml
```

Unten sehen Sie das Konfigurationsdialogfeld für Jalindi Igloo. Das Feld "CVSROOT" ist der Pfad zum Repository-Ordner.

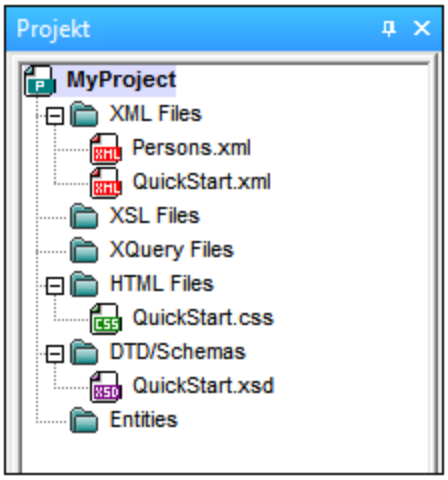
 © 2016-2022 Altova GmbH



In der Abbildung oben entspricht der lokale Pfad dem lokalen Arbeitsbereichordner, der dem CVS-Modul `MyProject` entspricht und an diesen gebunden ist.

9.6 Arbeiten mit Versionskontrolle

Um mit der Versionskontrolle zu arbeiten, wählen Sie das Projekt, einen Projektordner oder eine Projektdatei im Fenster "Projekt" aus (*Abbildung unten*) und wählen Sie anschließend im Menü **Projekt | Versionskontrolle** den gewünschten Befehl aus. Die Befehle **Einchecken** und **Auschecken** stehen im Fenster "Projekt" als Kontextmenübefehle zur Verfügung.



In diesem Abschnitt beschreiben wir die wichtigsten Versionskontrollfunktionen im Detail:

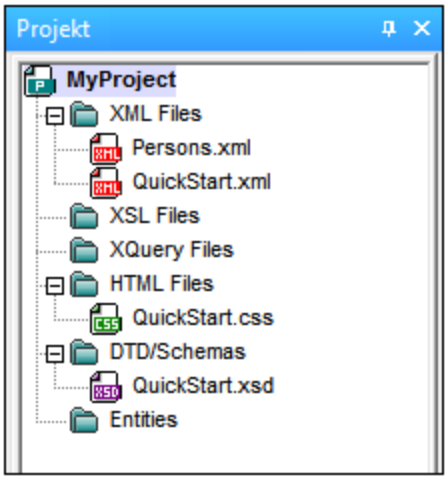
- Hinzufügen zur und Entfernen aus der Versionskontrolle 124
- Einchecken, Auschecken 125
- Dateien schreibgeschützt abrufen 128
- Kopieren und Freigeben über die Versionskontrolle 129
- Wechseln der Versionskontrolle 132

Weitere Befehle im Menü **Projekt | Versionskontrolle** werden im Abschnitt Menübefehle 191 dieses Handbuchs beschrieben. Informationen zu einem spezifischen Versionskontrollsystem finden Sie in der Dokumentation zu diesem System.

9.6.1 Hinzufügen zur, Entfernen aus der Versionskontrolle

Hinzufügen
Nachdem ein Projekt zu einer Versionskontrolle hinzugefügt wurde, können Sie Dateien entweder einzeln oder in Gruppen unter Versionskontrolle stellen. Dies wird auch als Hinzufügen der Dateien zur Versionskontrolle bezeichnet. Wählen Sie die Datei im Fenster "Projekt" aus und klicken Sie auf den Befehl **Projekt | Versionskontrolle | Zu Versionskontrolle hinzufügen**. Um mehrere Dateien auszuwählen, halten Sie beim Anklicken der Dateien die **Strg**-Taste gedrückt. Wenn Sie den Befehl an einem (grünen) Projektordner (*siehe Abbildung unten*) ausführen, werden alle Dateien im Ordner und seinen Unterordnern zur Versionskontrolle hinzugefügt.

 © 2016-2022 Altova GmbH



Wenn Dateien zur Versionskontrolle hinzugefügt werden, wird die lokale Ordnerhierarchie im Repository repliziert 119 (jedoch nicht die Projektordnerhierarchie). Wenn eine Datei sich also im lokalen Ordner in einem Unterordner mehrere Ebenen unterhalb des lokalen Ordners befindet, so werden im Repository automatisch auch der der Datei übergeordnete Ordner und alle darüber liegenden erstellt.

Wenn die erste Datei aus einem Projekt zur Versionskontrolle hinzugefügt wird, werden im Repository die korrekten Bindings erstellt und die Projektdatei (`.spp` Datei) wird automatisch hinzugefügt. Nähere Informationen dazu finden Sie im Abschnitt Zur Versionskontrolle hinzufügen 121.

Versionskontrollsymbole

Zu Dateien und dem Projektordner können bestimmte Symbole angezeigt werden, deren Bedeutung unten erklärt ist.

	Eingecheckt. Für das Auschecken verfügbar.
	Von einem anderen Benutzer ausgecheckt. Kann nicht ausgecheckt werden.
	Lokal ausgecheckt. Kann bearbeitet und eingecheckt werden.

Entfernen

Um eine Datei aus der Versionskontrolle zu entfernen, wählen Sie sie aus und klicken Sie auf den Befehl **Projekt | Versionskontrolle | Von Versionskontrolle ausgliedern**. Außerdem können Sie: (i) Dateien in einem Projektordner durch Ausführen des Befehls am Ordner entfernen und (ii) das gesamte Projekt durch Ausführen des Befehls am Projekt entfernen.

9.6.2 Auschecken, Einchecken

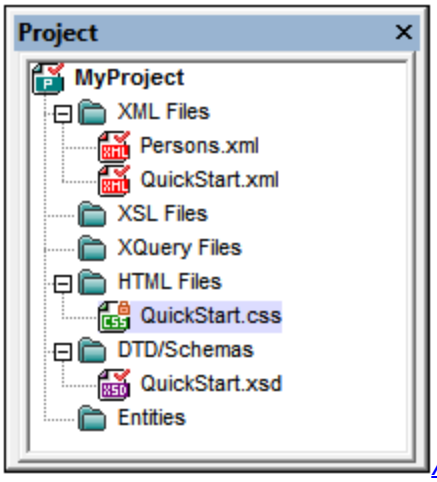
Nachdem Sie eine Projektdatei unter Versionskontrolle gestellt haben, kann sie durch Auswahl der Datei (im Projektfenster) und Klicken auf den entsprechenden Befehl im Menü **Projekt | Versionskontrolle** ein- und ausgecheckt werden.

Wenn eine Datei ausgecheckt wird, wird eine Kopie aus dem Repository in den lokalen Ordner platziert. Eine ausgecheckte Datei kann bearbeitet werden. Wenn eine Datei, die sich unter Versionskontrolle befindet, nicht ausgecheckt ist, kann sie nicht bearbeitet werden. Nachdem eine Datei bearbeitet wurde, können die Änderungen durch Einchecken der Datei im Repository gespeichert werden. Selbst wenn die Datei nicht in der Applikation gespeichert wurde, werden die Änderungen an der Datei durch Einchecken im Repository gespeichert. Der Ein- und Auscheckstatus einer Datei wird durch ein Häkchen bzw. ein Schloßsymbol im Symbol der jeweiligen Datei im Projektfenster angezeigt.

Zu Dateien und dem Projektordner können bestimmte Symbole angezeigt werden, deren Bedeutung unten erklärt ist.

	Eingecheckt. Für das Auschecken verfügbar.
	Von einem anderen Benutzer ausgecheckt. Kann nicht ausgecheckt werden.
	Lokal ausgecheckt. Kann bearbeitet und eingecheckt werden.

Durch Auswahl des Projekts oder eines Ordners im Projekt werden alle Dateien im ausgewählten Objekt ausgewählt. Um mehrere Objekte (Dateien und Ordner) auszuwählen, halten Sie beim Anklicken der Objekte die Strg-Taste gedrückt. In der Abbildung unten sehen Sie ein ausgechecktes Projekt. Die Datei `QuickStart.css` wurde im Anschluss daran eingecheckt.



Abrufen von Dateien als schreibgeschützte Dateien 128

Speichern und Ablehnen von Bearbeitungen

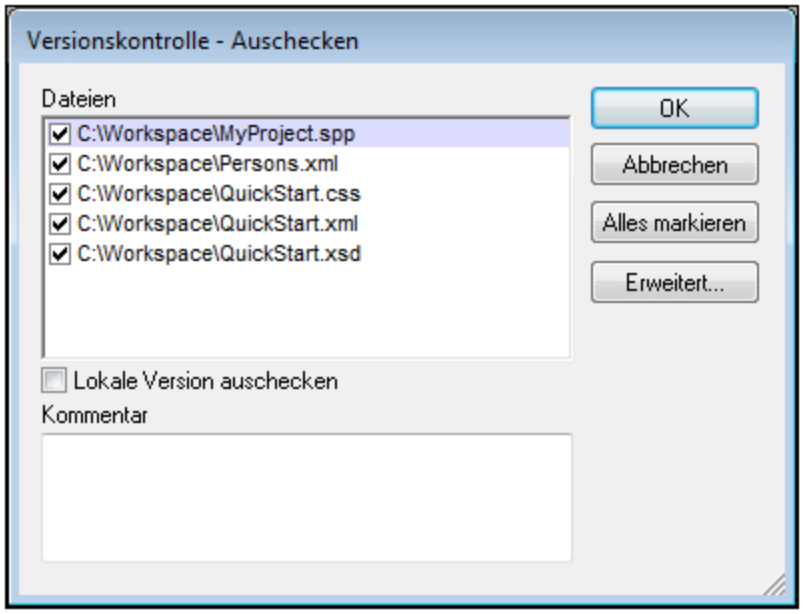
Beachten Sie, dass Sie eine Datei beim Einchecken ausgecheckt lassen können. Dabei werden die Änderungen im Repository gespeichert, während die Datei weiterhin ausgecheckt bleibt. Diese Option ist nützlich, wenn Sie Ihre Änderungen regelmäßig im Repository speichern und mit der Bearbeitung fortfahren möchten.

Wenn Sie eine Datei ausgecheckt haben, diese bearbeitet haben und die Änderungen anschließend aber nicht übernehmen möchten, können Sie die im Repository gespeicherte Dokumentversion in Ihrem lokalen Ordner wiederherstellen, indem Sie den Befehl **Projekt | Versionskontrolle | Auschecken rückgängig machen** auswählen.

 © 2016-2022 Altova GmbH

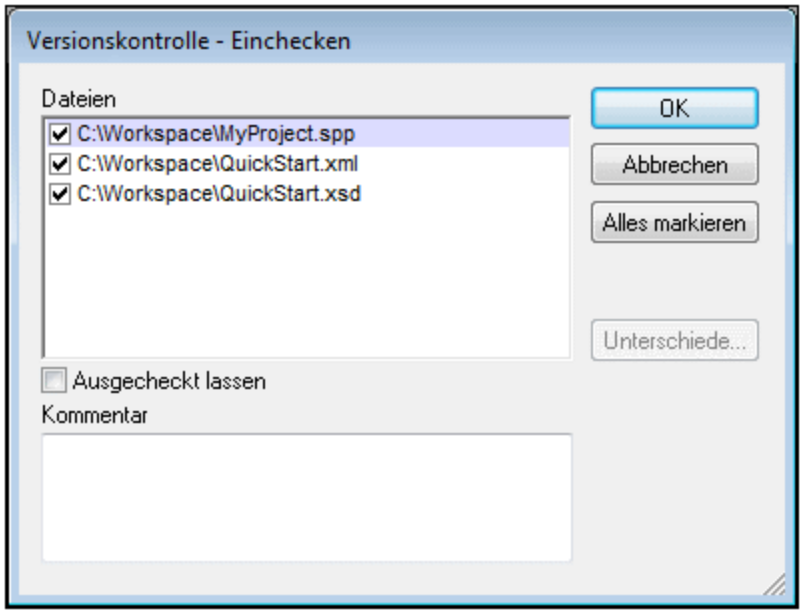
Auschecken

Im Dialogfeld "Auschecken" (*Abbildung unten*) können Sie (i) die auszucheckenden Dateien auswählen und (ii) wählen, ob die Repository-Version oder die lokale Version ausgecheckt werden soll.



Einchecken

Im Dialogfeld "Einchecken" (*Abbildung unten*) können Sie (i) die einzucheckenden Dateien auswählen und (ii) wählen, ob die Datei ausgecheckt bleiben soll.

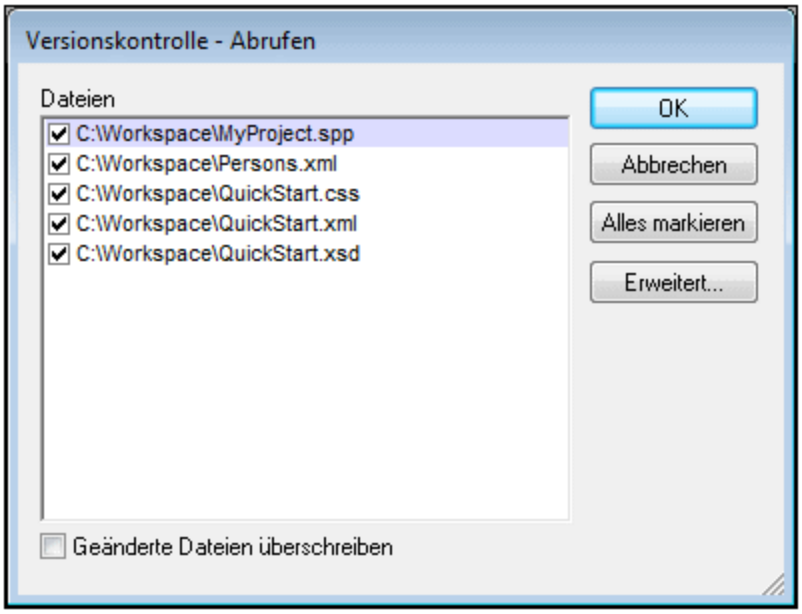


Anmerkung: In beiden Dialogfeldern (Einchecken und Auschecken) werden mehrere Dateien angezeigt, wenn das ausgewählte Objekt (Projekt oder Projektordner) mehrere Dateien enthält.

9.6.3 Dateien schreibgeschützt abrufen

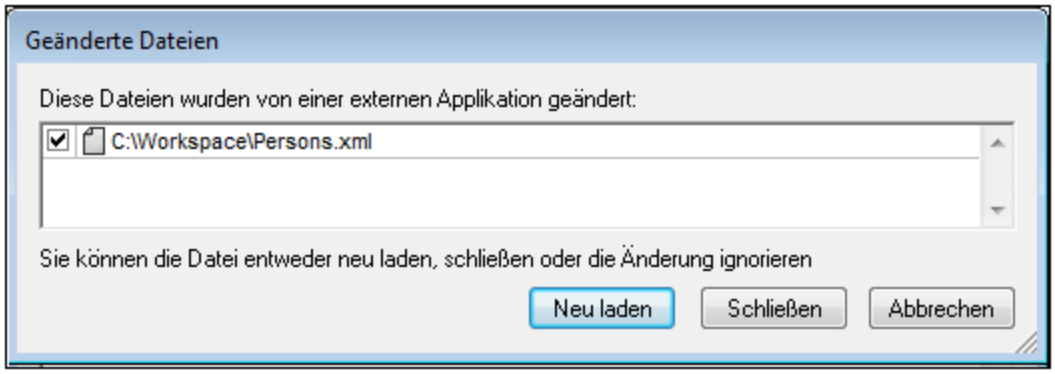
Mit dem Befehl **Abrufen** (im Menü **Projekt | Versionskontrolle**) werden Dateien aus dem Repository als schreibgeschützte Dateien abgerufen. (Um eine Datei bearbeiten zu können, müssen Sie sie auschecken 125.) Im Dialogfeld "Abrufen" werden die Dateien im Objekt (Projekt oder Ordner), an denen der Befehl "Abrufen" ausgeführt wurde, aufgelistet (*siehe Abbildung unten*). Sie können die abzurufenden Dateien markieren, indem Sie sie im Dialogfeld "Abrufen" mit einem Häkchen versehen.

Anmerkung: Mit dem Befehl **Ordner abrufen** können Sie einzelne Unterordner im Repository auswählen, falls dies Ihr Versionskontrollsystem gestattet.



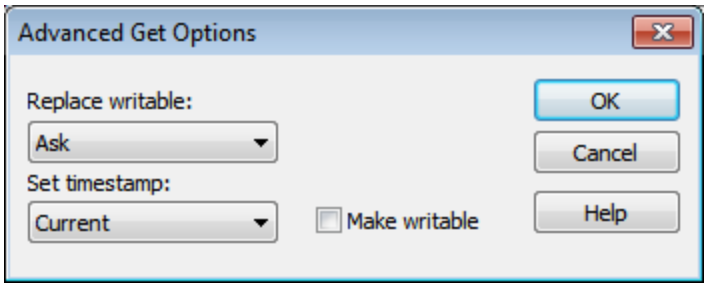
Sie können auswählen, ob geänderte ausgecheckte Dateien überschrieben werden sollen, indem Sie die entsprechende Option am unteren Rand des Dialogfelds "Abrufen" aktivieren. Wenn Sie auf **OK** klicken, werden die Dateien überschrieben. Wenn eine der überschriebenen Dateien gerade geöffnet ist, wird ein Dialogfeld angezeigt (*Abbildung unten*), in dem Sie gefragt werden, ob die Datei(en) neu geladen (Schaltfläche **Neu laden**), geschlossen (Schaltfläche **Schließen**) werden sollen oder ob die aktuelle Ansicht der Datei beibehalten werden soll (**Abbrechen**).

 © 2016-2022 Altova GmbH



Erweiterte Abrufoptionen

Das Dialogfeld "Erweiterte Abrufoptionen" (*Abbildung unten*) wird über die Schaltfläche **Erweitert** im Dialogfeld "Abrufen" (*siehe erste Abbildung in diesem Abschnitt*) aufgerufen.



Hier können Sie Optionen zum (i) Ersetzen ausgecheckter beschreibbarer Dateien (ii) zum Zeitstempel und (iii) ob die Eigenschaft "Schreibgeschützt" der abgerufenen Datei geändert werden soll, damit die Datei beschreibbar wird, einstellen.

Aktuellste Version holen

Mit dem Befehl **Aktuellste Version holen** (im Menü **Projekt | Versionskontrolle**) wird die neueste Version der ausgewählten Datei(en) abgerufen und in das Arbeitsverzeichnis platziert. Die Dateien werden schreibgeschützt abgerufen und nicht ausgecheckt. Dieser Befehl funktioniert wie der Befehl **Abrufen** (*siehe oben*), doch wird das Dialogfeld "Abrufen" nicht angezeigt.

Wenn die ausgewählten Dateien derzeit ausgecheckt sind, hängt es von Ihrem Versionskontrollsystem ab, was geschieht. Normalerweise werden Sie gefragt, ob die ausgecheckte Datei ersetzt, zusammengeführt oder so belassen werden soll.

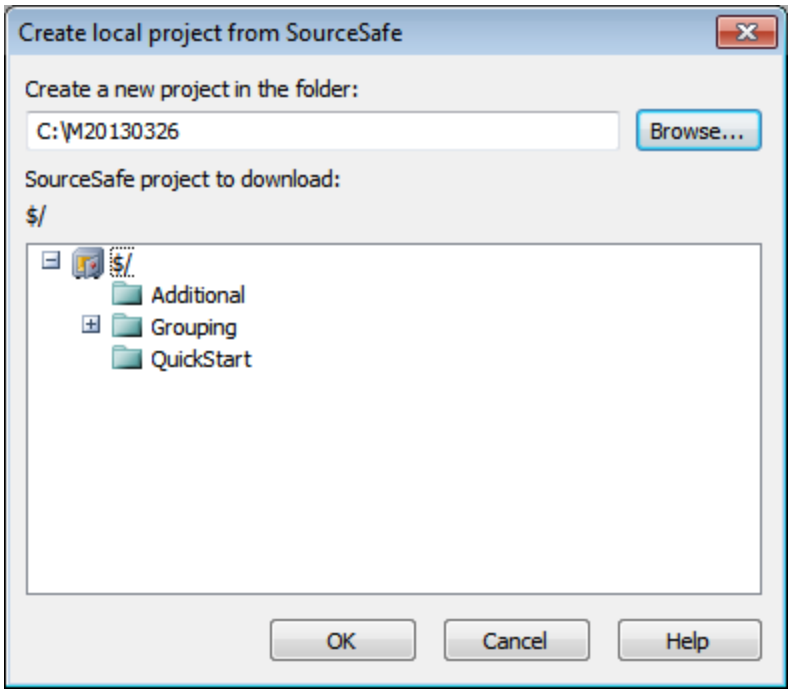
Anmerkung: Dieser Befehl ist bei Ausführung an einem Ordner rekursiv, d.h. er wird auf alle Dateien unterhalb des aktuellen Ordners in der Ordnerhierarchie angewendet.

9.6.4 Kopieren und Freigeben über die Versionskontrolle

Mit dem Befehl **Aus Versionskontrolle öffnen** erstellen Sie ein neues Applikationsprojekt von einem unter Versionskontrolle gestellten Projekt.

Um ein neues Projekt zu erstellen, gehen Sie folgendermaßen vor:

1. Je nach verwendetem Versionskontrollsystem müssen Sie vor dem Erstellen eines neuen Projekts über die Versionskontrolle sicherstellen, dass keine Datei aus dem unter Versionskontrolle stehenden Projekt ausgecheckt ist.
2. In der Applikation kann ein Projekt geöffnet sein, dies muss aber nicht der Fall sein.
3. Wählen Sie den Befehl **Projekt | Versionskontrolle | Aus Versionskontrolle öffnen**.
4. Das aktuell eingestellte Versionskontrollsystem wird mit den jeweiligen Überprüfungs- und Verbindungsdialogfeldern angezeigt. Stellen Sie eine Verbindung zum gewünschten Repository, d.h. zum gebundenen Ordner im Repository, der kopiert werden soll, 119 her.
5. Navigieren Sie im Dialogfeld, das daraufhin angezeigt wird (*Abbildung unten*), zum lokalen Ordner, in den der Inhalt des gebundenen Ordners im Repository (zu dem Sie soeben eine Verbindung hergestellt haben) kopiert werden soll. In der Abbildung unten hat der gebundene Ordner den Namen `MyProject` und wird durch das Zeichen `$` repräsentiert; der lokale Ordner ist `C:\M20130326`.



6. Klicken Sie auf **OK**. Der Inhalt des gebundenen Ordners (`MyProject`) wird in den lokalen Ordner `C:\M20130326`. kopiert und ein Dialogfeld wird aufgerufen, in dem Sie aufgefordert werden, die Projektdatei (`.spp` Datei) auszuwählen, die als das neue Projekt erstellt werden soll.
7. Wählen Sie die `.spp` Datei aus, die in den lokalen Ordner kopiert worden sein sollte. In unserem Beispiel handelt es sich um `MyProject.spp` im Ordner `C:\M20130326`. In der Applikation wird ein neues Projekt namens `MyProject` erstellt und im Fenster "Projekt" angezeigt. Die Dateien des Projekts werden im Ordner `C:\M20130326` gespeichert.

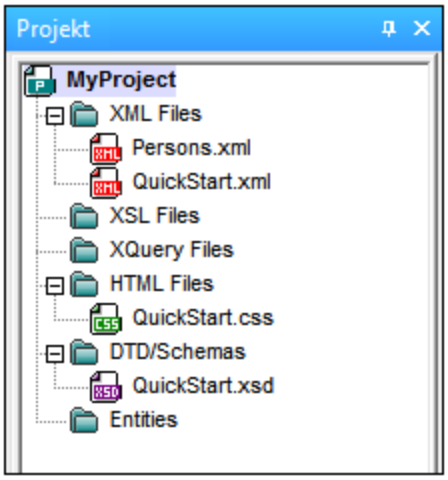
Freigeben über die Versionskontrolle

Der Befehl **Aus Versionskontrolle freigeben** wird unterstützt, wenn das verwendete Versionskontrollsystem die Freigabe von Dateien unterstützt. Sie können eine Datei freigeben, sodass sie auf mehreren lokalen

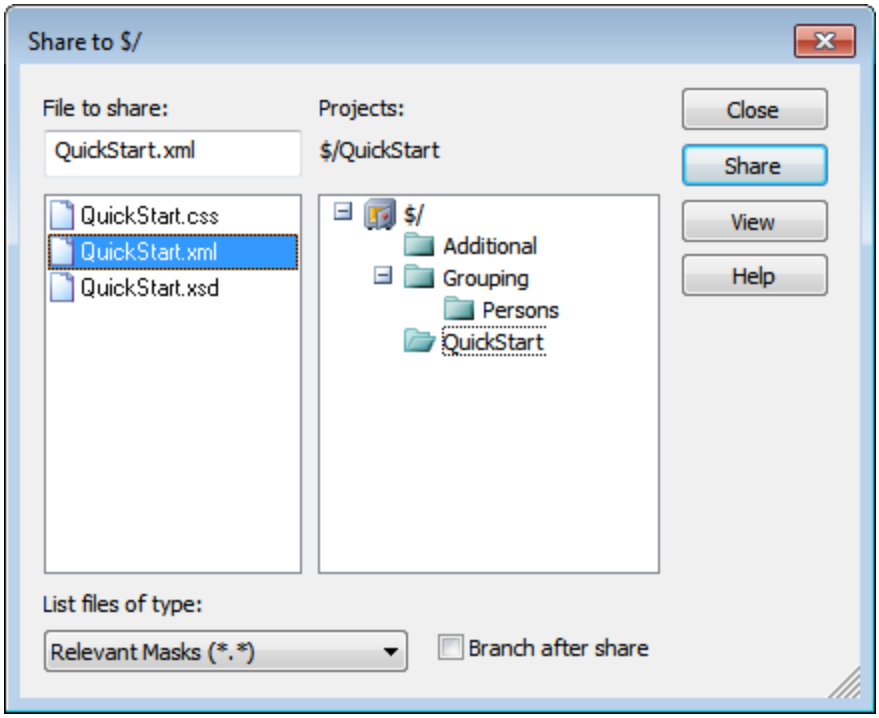
 © 2016-2022 Altova GmbH

Rechnern verfügbar ist. Wenn an einer dieser Dateien eine Änderung vorgenommen wird, wird dies in allen anderen freigegebenen Versionen übernommen.

Wählen Sie das (*in der Abbildung unten markierte)* Projekt im Fenster "Projekt" der Applikation aus und klicken Sie anschließend auf den Befehl **Aus Versionskontrolle freigeben**.



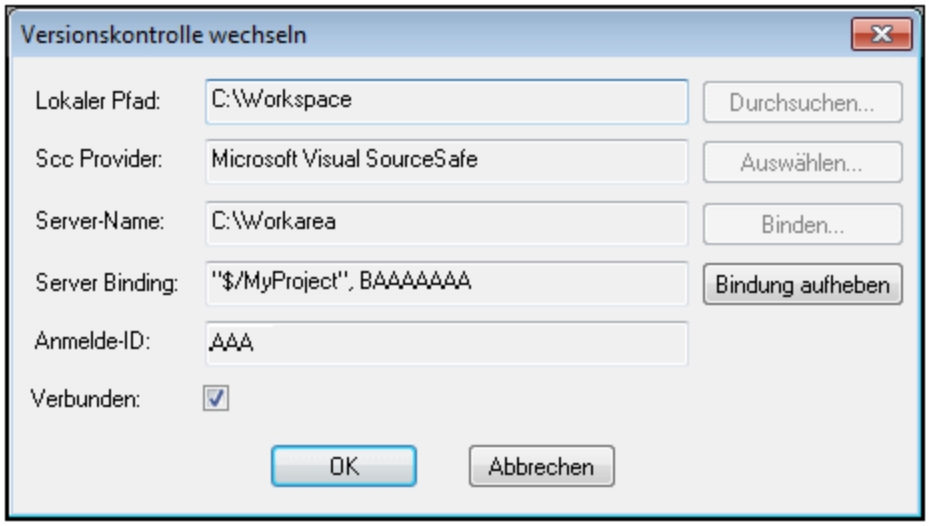
Daraufhin wird das Dialogfeld "Share To [Ordner]" (*Abbildung unten*) angezeigt.



Um die gewünschten Dateien auszuwählen, wählen Sie zuerst den Ordner, in dem sich die Dateien befinden, in der Projektstruktur auf der rechten Seite des Dialogfelds (*siehe Abbildung oben*) aus. Die Dateien im ausgewählten Ordner werden im Fenster auf der linken Seite angezeigt. Wählen Sie die freizugebende Datei aus (um mehrere Dateien auszuwählen, halten Sie beim Anklicken die **Strg**-Taste gedrückt). Die ausgewählten Datei(en) werden im Textfeld *Files to Share* (*links oben*) angezeigt. Die Dateien verschwinden aus dem Fenster

auf der linken Seite. Klicken Sie auf **Freigeben** und anschließend **Schließen,** um die ausgewählte(n) Datei(en) in den ausgewählten lokalen Ordner zu kopieren.

Der freigegebene Ordner wird im Namen des Dialogfelds "Share to [Ordner]" angegeben. In der Abbildung oben ist es der lokale Ordner (da sich das Zeichen `$` in dem Ordner im Repository, an den der lokale Ordner gebunden ist, befindet). Sie können den freigegebenen Ordner im Dialogfeld "Versionskontrolle wechseln" (*Abbildung unten,* **Versionskontrolle wechseln***)* sehen und den Ordner durch Ändern des lokalen Pfads und Binding einstellen.



Nähere Informationen zum Freigeben von Dateien über Ihr Versionskontrollsystem finden Sie in der Benutzerdokumentation zu Ihrem Versionskontrollsystem.

9.6.5 Wechseln der Versionskontrolle

Die Versionskontrolleinstellungen können über zwei Befehle im Menü **Projekt | Versionskontrolle** geändert werden:

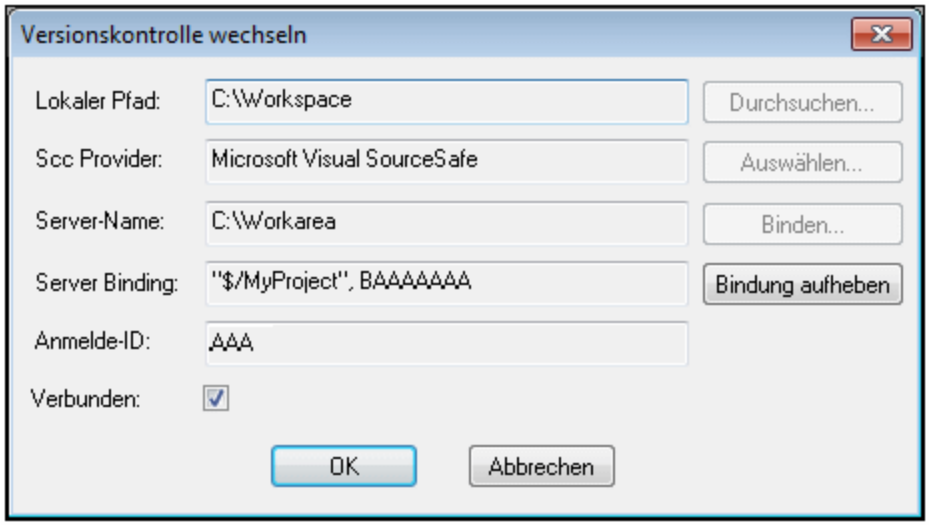
- **Versionskontrollmanager:** Öffnet die Versionskontrollsystemapplikation und ermöglicht Ihnen, Datenbanken einzurichten und Bindings zu konfigurieren.
- **Versionskontrolle wechseln:** Ruft das Dialogfeld "Versionskontrolle wechseln" auf, in dem Sie das von der Altova Applikation verwendete Versionskontrollsystem und das aktuelle Binding wechseln können. Dieses Dialogfeld ist im Folgenden beschrieben.

Über das aktuelle Binding stellt das aktive Applikationsprojekt eine Verbindung zur Versionskontrolldatenbank her. Das aktuelle Binding ist korrekt sein, wenn sich die Applikationsprojektdatei (`.spp` Datei) im lokalen Ordner und der gebundene Ordner in dem Repository befindet, in dem die Projektdateien gespeichert werden sind. Normalerweise entsprechen der gebundene Ordner und seine Substruktur dem lokalen Arbeitsbereichordner und dessen Substruktur.

Im Dialogfeld "Versionskontrolle wechseln" (*Abbildung unten*) können Sie das Versionskontrollsystem (*SCC Provider*), den lokalen Ordner (*Lokaler Pfad*) und das Repository-Binding (*Server-Name* und *Server Binding*) wechseln.

 © 2016-2022 Altova GmbH

Die Einstellungen können erst geändert werden, wenn Sie die aktuelle Bindung aufgehoben haben. Verwenden Sie dazu die Schaltfläche **Bindung aufheben**. Alle Einstellungen können nun bearbeitet werden.



Ändern Sie die Versionskontrolleinstellungen folgendermaßen:

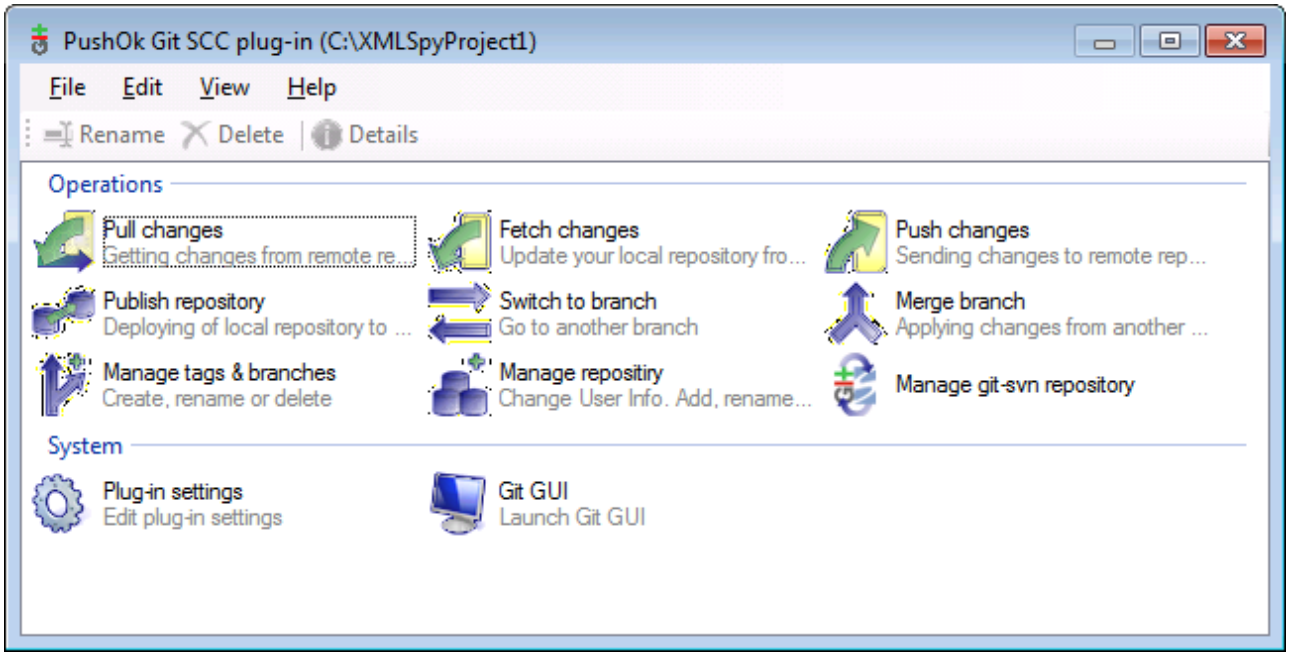
1. Navigieren Sie über die Schaltfläche **Durchsuchen** zum lokalen Ordner und klicken Sie auf die Schaltfläche **Auswählen,** um eines der installierten Versionskontrollsysteme auszuwählen.
2. Anschließend können Sie den lokalen Ordner an eine Repository-Datenbank binden. Klicken Sie dazu auf die Schaltfläche **Binden**. Daraufhin wird das Verbindungsdialogfeld Ihres Versionskontrollsystems aufgerufen.
3. Wenn Sie eine *Anmelde-ID* eingegeben haben, wird diese an das Versionskontrollsystem übergeben; andernfalls müssen Sie Ihre Anmeldeinformationen im Verbindungsdialogfeld eingeben.
4. Wählen Sie im Repository die Datenbank aus, an die Sie diesen lokalen Ordner binden möchten. Eventuell müssen diese Einstellungen in mehreren Dialogfeldern vorgenommen werden.
5. Nachdem Sie alle Einstellungen vorgenommen haben, klicken Sie im Dialogfeld "Versionskontrolle wechseln" auf **OK**.

9.7 Versionskontrolle mit Git

Unterstützung für Git als Versionskontrollsystem in Authentic Desktop steht in Form eines Drittanbieter-Plug-in namens **GIT SCC Plug-in** (http://www.pushok.com/software/git.html) zur Verfügung.

> Zum Zeitpunkt der Verfassung dieser Dokumentation steht das **GIT SCC Plug-in** zum Experimentieren zur Verfügung. Um das Plug-in verwenden zu können, müssen Sie beim Plug-in-Anbieter registriert sein.

Mit Hilfe des GIT SCC Plug-in können Sie über die Befehle im Menü **Projekt | Versionskontrolle** von Authentic Desktop mit einem Git Repository arbeiten. Beachten Sie, dass die Befehle im Menü **Projekt | Versionskontrolle** von Authentic Desktop von der Microsoft Source Control Plug-in API (MSSCCI API), für die eine andere Art von Design als von Git verwendet wird, bereitgestellt werden. Daher bildet das Plug-in eine Zwischenschaltung zwischen "Visual Source Safe"-Funktionalitäten und Git-Funktionalitäten. Das bedeutet einerseits, dass ein Befehl wie z.B. **Aktuellste Version holen** für Git eventuell so nicht verwendet werden kann. Andererseits gibt es einige neue Git-spezifische Aktionen, die über das Plug-in im Dialogfeld "Versionskontrollmanager" zur Verfügung gestellt werden (Menü **Projekt | Versionskontrolle | Versionskontrollmanager** von Authentic Desktop).



Das Dialogfeld "Versionskontrollmanager"

Andere häufig benötigte Versionskontrollbefehle stehen direkt im Menü **Projekt | Versionskontrolle** zur Verfügung.

In den folgenden Abschnitten wird die Anfangskonfiguration des Plug-in sowie der grundlegende Arbeitsablauf beschrieben:

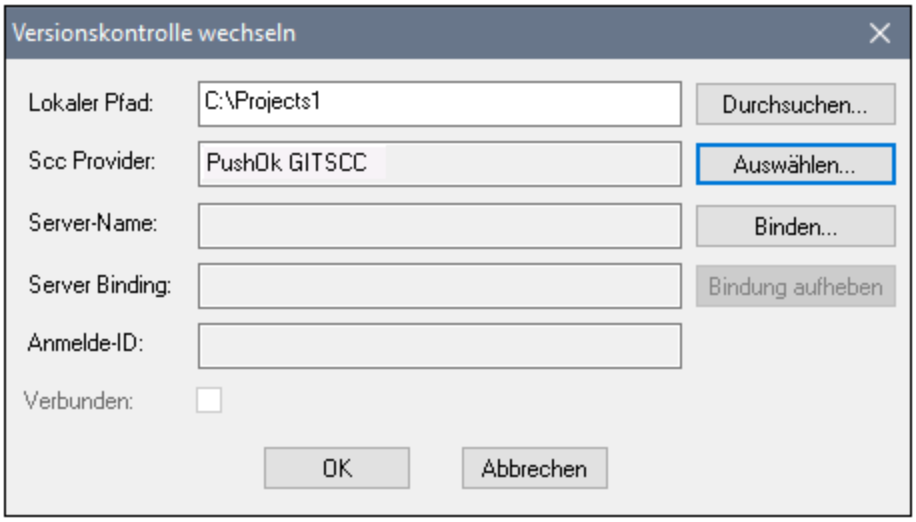
- Aktivieren der Git-Versionskontrolle mit dem Git SCC Plug-in 135
- Hinzufügen eines Projekts zur Git-Versionskontrolle 135

- Klonen eines Projekts anhand der Git-Versionskontrolle 137

9.7.1 Aktivieren der Git-Versionskontrolle mit dem Git SCC Plug-in

Um die Git-Versionskontrolle mit Authentic Desktop zu aktivieren, muss zuerst das Drittanbieter-**PushOK GIT SCC Plug-in** installiert, registriert und als Versionskontrollanbieter ausgewählt werden. Dies geschieht folgendermaßen:

1. Laden Sie die Installationsdatei für das Plug-in von der Website des Anbieters (http://www.pushok.com) herunter, starten Sie sie und befolgen Sie die Installationsanweisungen.
2. Klicken Sie im Menü **Projekt** von Authentic Desktop auf **Versionskontrolle wechseln** und vergewissern Sie sich, dass **PushOk GITSCC** als Versionskontroll-Provider ausgewählt ist. Wenn **Push Ok GITSCC** in der Liste der Provider nicht angezeigt wird, war die Installation des Plug-in wahrscheinlich nicht erfolgreich. Lesen Sie in diesem Fall in der Dokumentation des Anbieters nach, wie Sie das Problem lösen können.



3. Wenn ein Dialogfeld angezeigt wird, in dem Sie aufgefordert werden, das Plug-in zu registrieren, klicken Sie auf **Registrierung** und befolgen Sie die Anweisungen des Assistenten, um die Registrierung abzuschließen.

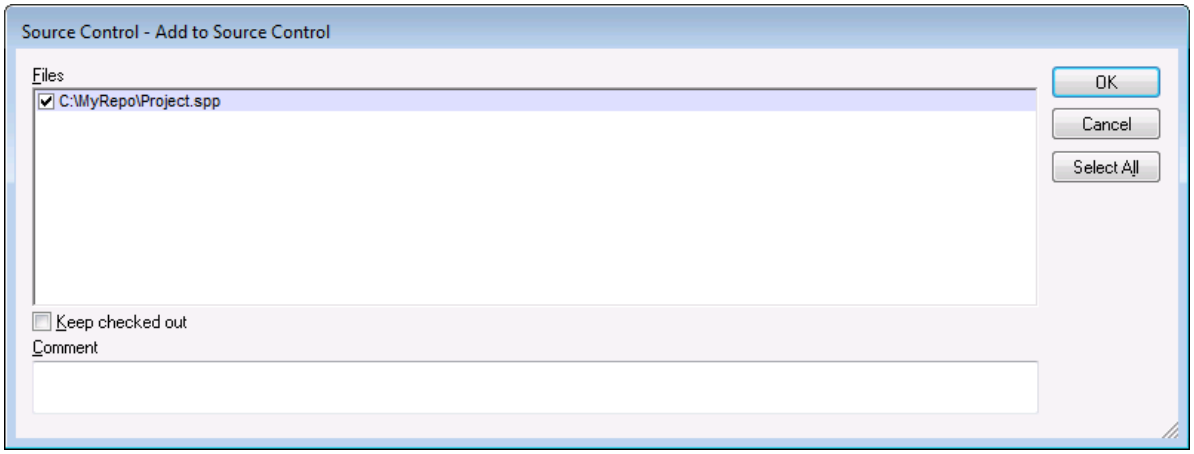
9.7.2 Hinzufügen eines Projekts zur Git-Versionskontrolle

Sie können Authentic Desktop -Projekte als Git Repositories speichern. Die Struktur der zum Projekt hinzugefügten Dateien oder Ordner würde anschließend der Struktur des Git Repository entsprechen.

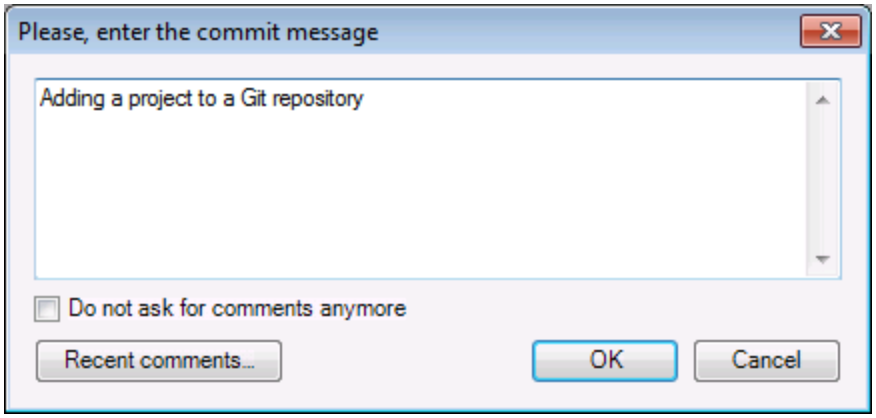
So fügen Sie ein Projekt zu einer Git-Versionskontrolle hinzu:

1. Stellen Sie sicher, dass das **PushOK GIT SCC Plug-in** als Versionskontrollanbieter ausgewählt ist (siehe Aktivieren der Git-Versionskontrolle mit dem Git SCC Plug-in 135).
2. Erstellen Sie mit dem Menübefehl **Projekt | Projekt erstellen** ein neues Projekt.
3. Speichern Sie das Projekt in einem lokalen Ordner, z.B. unter `C:\MyRepo\Project.spp`

4. Klicken Sie im Menü **Projekt** unter **Versionskontrolle** auf **Zu Versionskontrolle hinzufügen**.

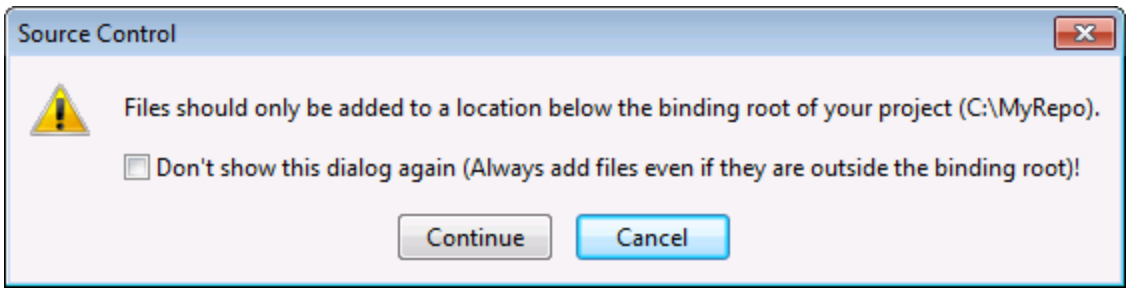


5. Klicken Sie auf **OK**.



6. Geben Sie den Text Ihrer Commit-Meldung ein und klicken Sie auf **OK**.

Sie können jetzt Dateien und Ordner zu Ihrem Projekt hinzufügen. Beachten Sie, dass alle Projektdateien und -ordner sich unter dem Root-Ordner des Projekts befinden müssen. Wenn das Projekt z.B. im Ordner `C:\MyRepo` angelegt wurde, so sollten nur Dateien unter `C:\MyRepo` zur Projekt hinzugefügt werden. Wenn Sie versuchen, Dateien, die sich außerhalb des Projekt-Root-Ordners befinden, zum Projekt hinzuzufügen, wird eine Warnmeldung angezeigt:

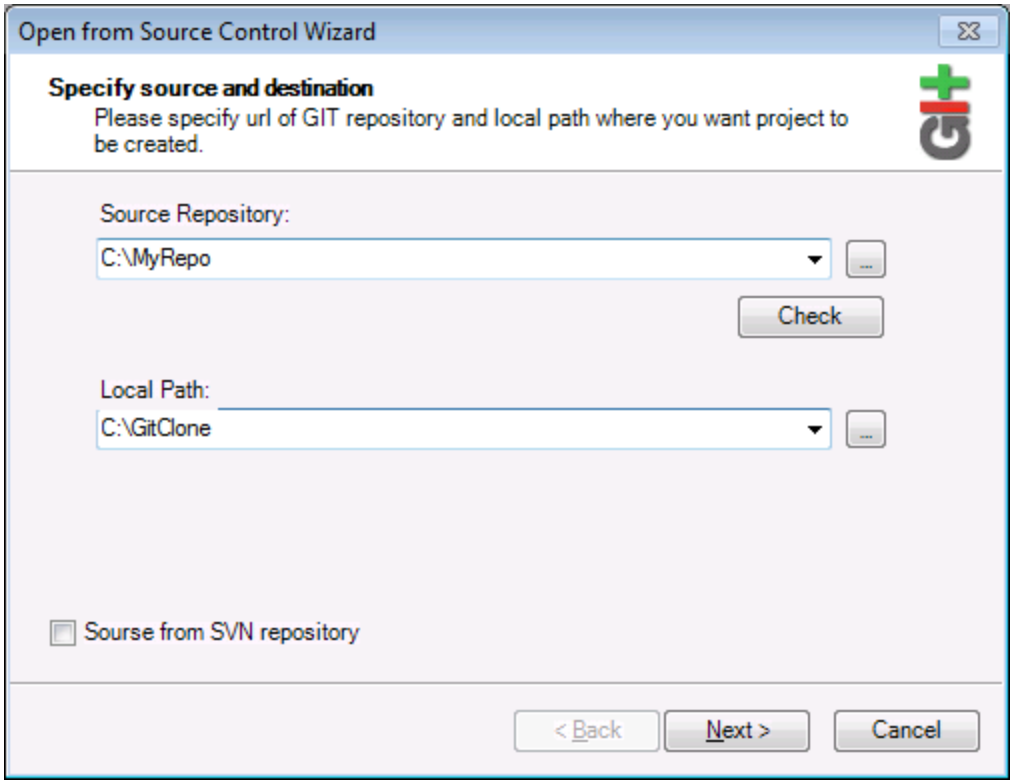


 © 2016-2022 Altova GmbH

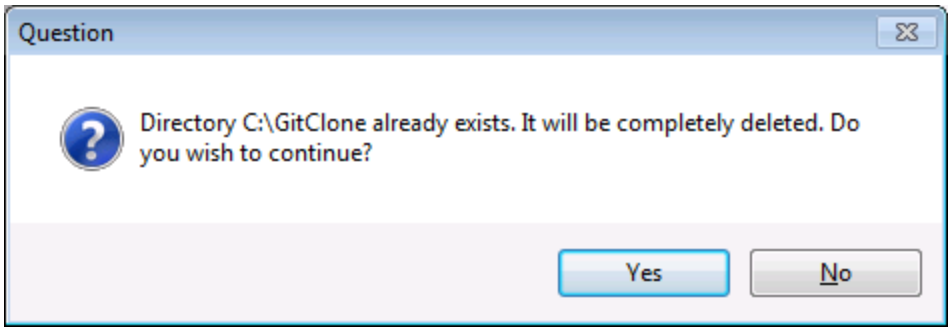
9.7.3 Klonen eines Projekts anhand der Git-Versionskontrolle

Projekte, die bereits zur Git-Versionskontrolle hinzugefügt wurden (siehe Hinzufügen eines Projekts zur Git-Versionskontrolle 135) können folgendermaßen über das Git Repository geöffnet werden:

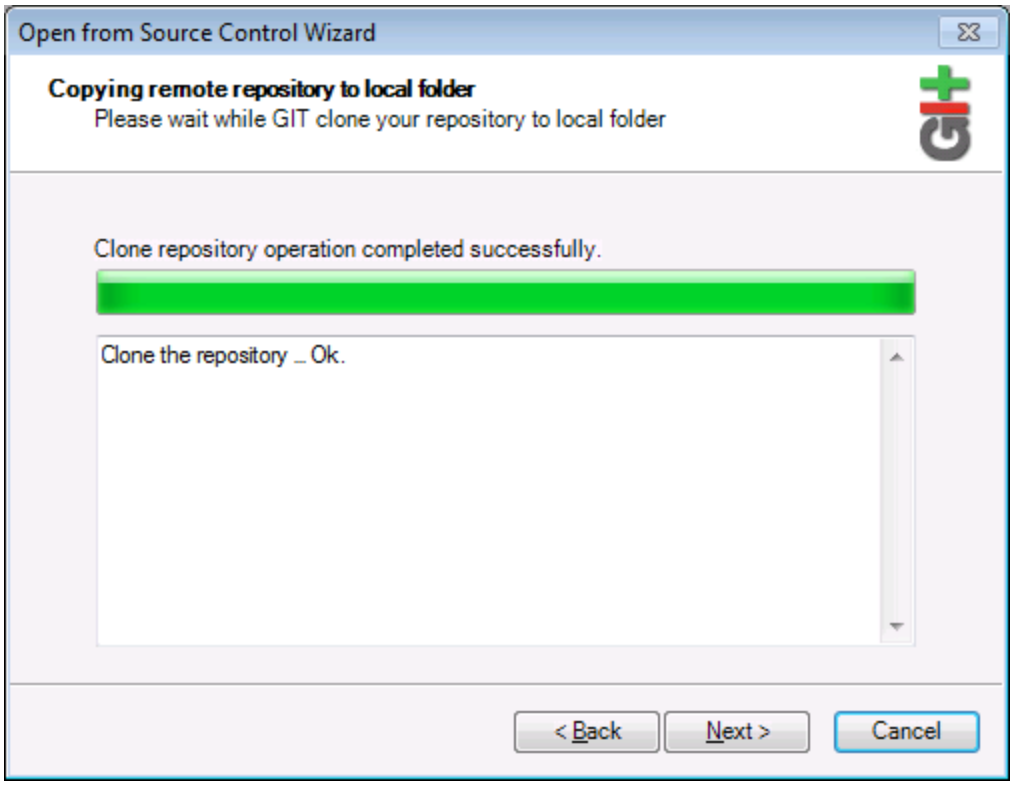
1. Vergewissern Sie sich, dass das **PushOK GIT SCC Plug-in** als Versionskontroll-Provider ausgewählt ist (siehe Aktivieren der Git-Versionskontrolle mit dem Git SCC Plug-in 135).
2. Klicken Sie im Menü **Projekt** auf **Versionskontrolle | Aus Versionskontrolle öffnen**.
3. Geben Sie den Pfad oder die URL des Versionskontroll-Repository ein. Klicken Sie auf **Überprüfen,** um die Gültigkeit des Pfads oder der URL zu überprüfen.



4. Geben Sie unter **Lokaler Pfad** den Pfad zu dem lokalen Ordner ein, in dem das Projekt erstellt werden soll und klicken Sie auf **Weiter**. Wenn der lokale Ordner vorhanden ist, wird (auch wenn er leer ist) das folgende Dialogfeld aufgerufen:



5. Klicken Sie zur Bestätigung auf **Ja** und anschließend auf **Weiter**.



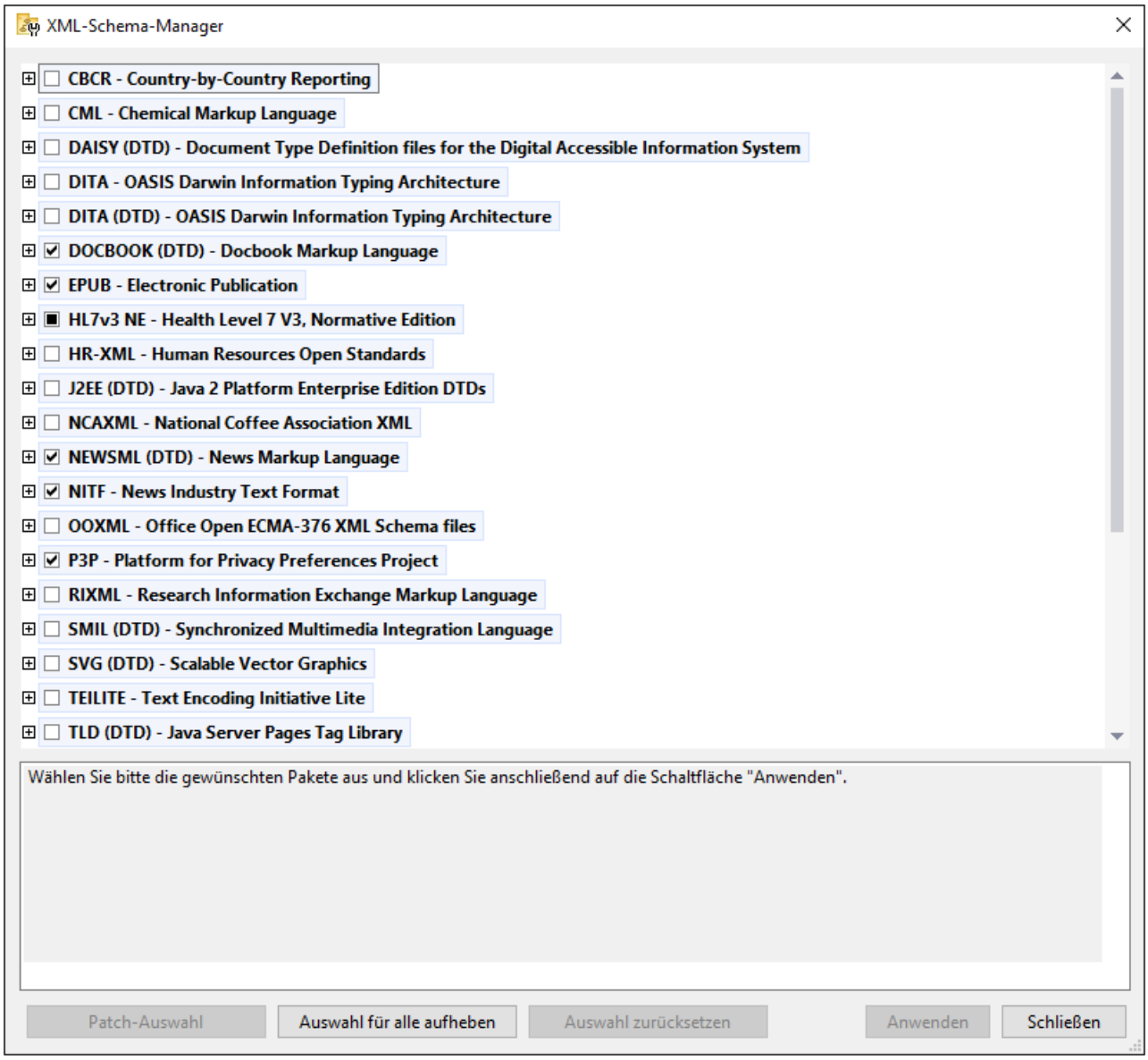
6. Stellen Sie die restlichen Schritte des Assistenten fertig, wie für Ihr Projekt erforderlich.
7. Nach der Fertigstellung wird ein Durchsuchen-Dialogfeld angezeigt, in dem Sie aufgefordert werden, das Authentic Desktop-Projekt (*.spp)-Datei zu öffnen. Wählen Sie die Projektdatei aus, um den Projektinhalt in Authentic Desktop zu laden.

© 2016-2022 Altova GmbH

10 Schema-Manager

Der XML-Schema-Manager ist ein Altova-Tool, mit dem Sie XML-Schemas (DTDs für XML-Dateien und XML-Schemas) zentral installieren und verwalten können, um diese in allen XML-fähigen Applikationen von Altova einschließlich Authentic Desktop verwenden zu können.

- Unter Windows hat der Schema-Manager eine grafische Benutzeroberfläche (*siehe Abbildung unten*) und steht auch über die Befehlszeile zur Verfügung. (Die Desktop-Applikationen von Altova stehen nur unter Windows zur Verfügung; *siehe Liste unten*).
- Unter Linux und MacOS steht der Schema-Manager nur über die Befehlszeile zur Verfügung. (Die Server-Applikationen von Altova stehen unter Windows, Linux und macOS zur Verfügung; *siehe Liste unten*).



Altova-Applikationen, die mit Schema-Manager arbeiten:

Desktop-Applikationen (nur Windows)	Server-Applikationen (Windows, Linux, macOS)
XMLSpy (alle Editionen)	RaptorXML Server, RaptorXML+XBRL Server
MapForce (alle Editionen)	StyleVision Server
StyleVision (alle Editionen)	
Authentic Desktop Enterprise Edition	

Installation und Deinstallation des Schema-Managers

Der Schema-Manager wird bei der ersten Installation einer neuen Version der Altova Mission Kit Enterprise Edition oder einer der XML-Schema-fähigen Applikationen von Altova (*siehe Tabelle oben*) automatisch installiert.

Ebenso wird er auch automatisch entfernt, wenn Sie die letzte XML-Schema-fähige Applikation von Altova auf Ihrem Rechner deinstallieren.

Schema-Manager-Funktionalitäten

Im Schema-Manager stehen die folgenden Funktionalitäten zur Verfügung:

- Anzeigen der auf Ihrem Rechner installierten XML-Schemas und Überprüfung, ob neue Versionen zum Download zur Verfügung stehen.
- Download neuer Versionen von XML-Schemas unabhängig vom Altova Produkt-Release-Zyklus. (Die Schemas werden von Altova online bereitgestellt und können über den Schema-Manager heruntergeladen werden).
- Installation oder Deinstallation jeder beliebigen (oder ggf. aller) der zahlreichen Versionen eines bestimmten Schemas.
- Ein XML-Schema kann Abhängigkeiten von anderen Schemas aufweisen. Bei der Installation oder Deinstallation eines bestimmten Schemas informiert Sie der Schema-Manager über davon abhängige Schemas und installiert bzw. entfernt diese ebenfalls automatisch.
- Der Schema-Manager ordnet Schema-Referenzen mit Hilfe des XML-Katalogs lokalen Dateien zu. Dadurch lassen sich große XML-Schemas schneller verarbeiten, als wenn sie sich unter einem entfernten Pfad befinden.
- Alle wichtigen Schemas werden über den Schema-Manager bereitgestellt und regelmäßig auf die jeweils neuesten Version aktualisiert. Dadurch können alle Ihre Schemas zentral verwaltet werden und stehen allen XML-Schema-fähigen Applikationen von Altova jederzeit zur Verfügung.
- Im Schema-Manager vorgenommene Änderungen werden für alle auf dem Rechner installierten Altova-Produkte wirksam.

Funktionsweise

Alle in Altova-Produkten verwendeten XML-Schemas werden von Altova online bereitgestellt. Dieser Speicher wird bei Veröffentlichung neuer Versionen der Schemas aktualisiert. Im Schema-Manager werden sowohl bei Aufruf über die Benutzeroberfläche als auch über das CLI Informationen über die neuesten verfügbaren Schemas angezeigt. Sie können die gewünschten Schemas dann über den Schema-Manager installieren, aktualisieren oder deinstallieren.

 © 2016-2022 Altova GmbH

Schemas können vom Schema-Manager auch auf eine weitere Art installiert werden. Sie können ein Schemas und die davon abhängigen Schemas auf der Altova Website (https://www.altova.com/de/schema-manager) auswählen. Daraufhin wird auf der Website eine Datei des Typs **.altova_xmlschemas** mit Informationen über Ihre ausgewählten Schemas zum Download vorbereitet. Bei Doppelklick auf diese Datei oder bei Übergabe an den Schema-Manager über das CLI als Argument des Befehls install[152] installiert der Schema-Manager die ausgewählten Schemas.

Lokaler Cache: Überprüfung Ihrer Schemas

Alle Informationen über installierte Schemas werden in einem zentralen Cache-Verzeichnis auf Ihrem Rechner aufgezeichnet. Das Verzeichnis befindet sich hier:

Windows	`C:\ProgramData\Altova\pkgs\.cache`
Linux	`/var/opt/Altova/pkgs\.cache`
macOS	`/var/Altova/pkgs`

Dieses Cache-Verzeichnis wird regelmäßig mit dem neuesten Status der Schemas aus dem Online-Speicher von Altova aktualisiert. Diese Aktualisierungen finden unter den folgenden Bedingungen statt:

- bei jedem Start von Schema-Manager.
- Wenn Sie Authentic Desktop zum ersten Mal an einem bestimmten Kalendertag starten.
- Wenn Authentic Desktop länger als 24 Stunden geöffnet ist, findet alle 24 Stunden eine Aktualisierung des Cache statt.
- Sie können den Cache auch durch Ausführung des update[155]-Befehls über die Befehlszeilenschnittstelle aktualisieren.

Der Schema-Manager kann somit Ihre installierten Schemas über den Cache ständig anhand der online verfügbaren Schemas auf der Altova Website überprüfen.

Nehmen Sie keine manuellen Änderungen am Cache vor!

Das lokale Cache-Verzeichnis wird automatisch auf Basis der installierten oder deinstallierten Schemas verwaltet; es sollte nicht manuell geändert oder gelöscht werden. Falls Sie den Schema-Manager je in seinen Originalzustand zurücksetzen möchten, (i) führen Sie den CLI-Befehl reset[153] der Befehlszeilenschnittstelle und (ii) anschließend den Befehl initialize[151] aus. (Führen Sie alternativ dazu den Befehl **reset** mit der Option **-i** aus).

10.1 Ausführen des Schema-Managers

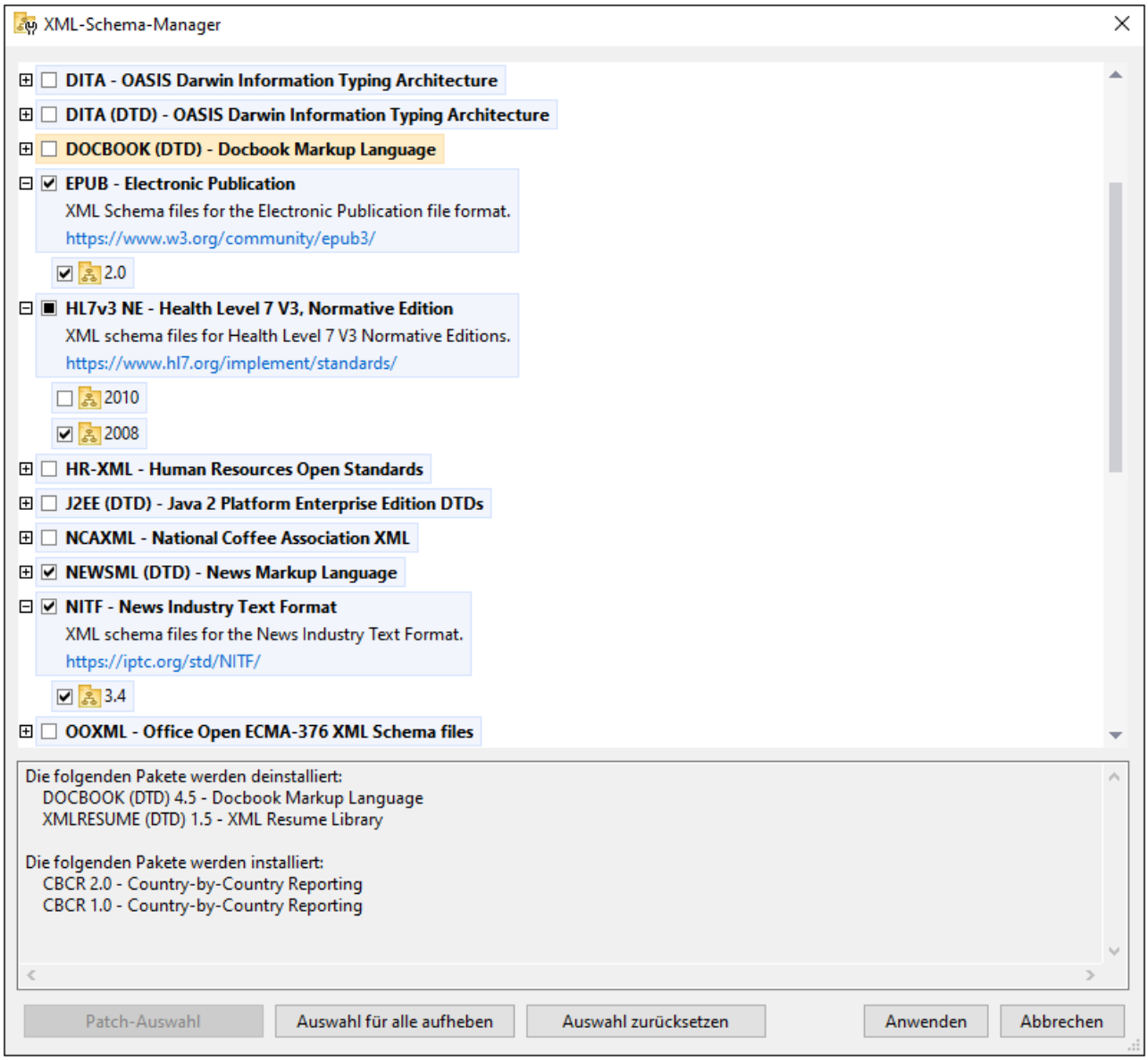
Grafische Benutzeroberfläche

Sie können die Benutzeroberfläche des Schema-Managers auf eine der folgenden Arten aufrufen:

- *Bei der Installationen von Authentic Desktop:* Aktivieren Sie gegen Ende der Installation das Kontrollkästchen *Altova-Schema-Manager aufrufen*, wodurch Sie die Benutzeroberfläche des Schema-Managers direkt aufrufen können. Auf diese Art können Sie Schemas während der Installation Ihrer Altova-Applikation installieren.
- *Nach der Installation von Authentic Desktop:* Nachdem die Applikation installiert wurde, können Sie die Benutzeroberfläche des Schema-Managers jederzeit über den Menübefehl **Tools | XML-Schemas-Manager** aufrufen.
- über die von der Altova Webseite heruntergeladene `.altova_xmlschemas`-Datei: Doppelklicken Sie auf die heruntergeladene Datei, um den Schema-Manager zu starten, der daraufhin die (auf der Website) ausgewählten Schemas installiert.

Nachdem die Benutzeroberfläche des Schema-Managers geöffnet wurde (*Abbildung unten*), werden bereits installierte Schemas markiert angezeigt. Wenn ein zusätzliches Schema installiert werden soll, aktivieren Sie dieses. Wenn ein bereits installiertes Schema deinstalliert werden soll, deaktivieren Sie dieses. Nachdem Sie Ihre Auswahl getroffen haben, können Ihre Änderungen angewendet werden. Die Schemas, die installiert bzw. deinstalliert werden, werden markiert und im Fenster "Meldungen" am unteren Rand des Schema-Manager-Fensters (*siehe Abbildung*) erscheint eine Meldung über die bevorstehenden Änderungen.

© 2016-2022 Altova GmbH



Befehlszeilenschnittstelle

Sie können den Schema-Manager über eine Befehlszeilenschnittstelle starten, indem Sie Befehle an die ausführbare Datei **`xmlschemamanager.exe`** senden.

Die Datei **`xmlschemamanager.exe`** steht im folgenden Ordner zur Verfügung:

- *unter Windows:* `C:\ProgramData\Altova\SharedBetweenVersions`
- *Unter Linux oder macOS (nur Server-Applikationen):* `%INSTALLDIR%/bin`, wobei `%INSTALLDIR%` das Installationsverzeichnis des Programms ist.

Anschließend können Sie jeden der im Abschnitt zur CLI-Befehlsreferenz 150 aufgelisteten Befehle verwenden.

Um die Hilfe zu den Befehlen anzuzeigen, führen Sie den folgenden Befehl aus:

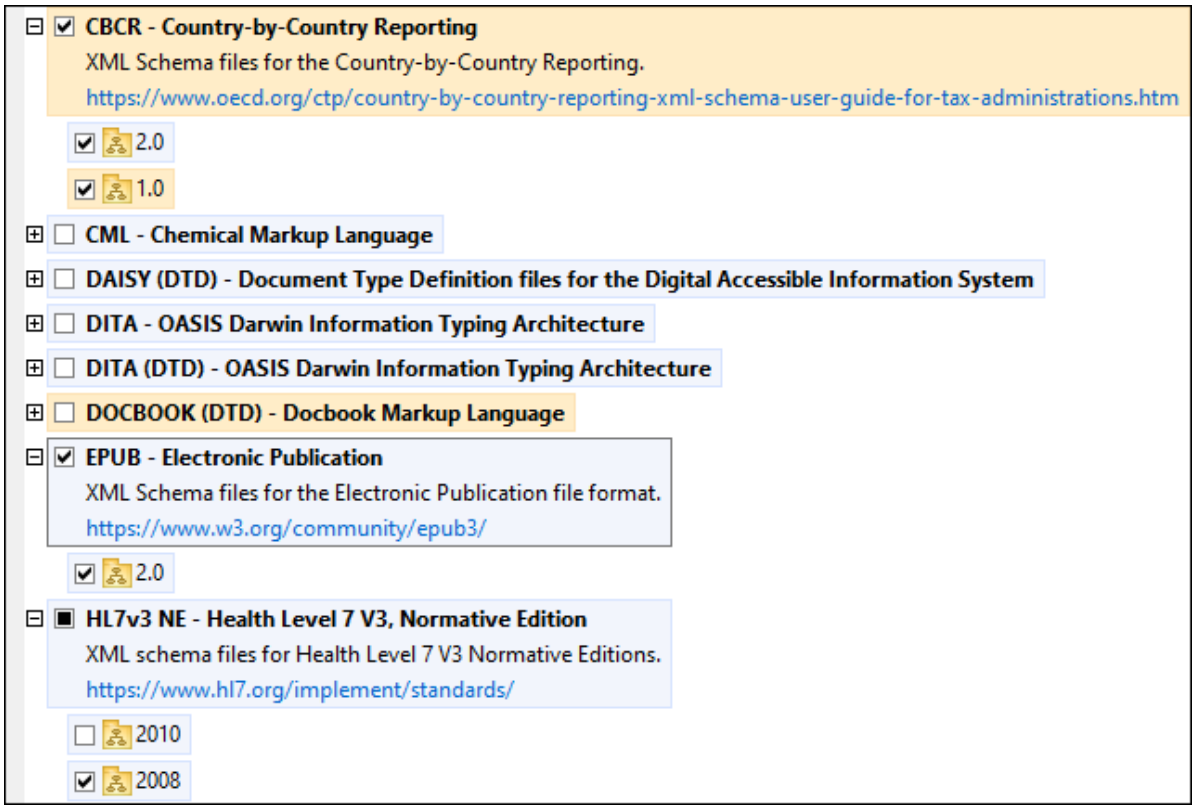
- *unter Windows:* `xmlschemamanager.exe --help`
- *Unter Linux oder macOS (nur Server-Applikationen):* `sudo ./xmlschemamanager --help`

© 2016-2022 Altova GmbH

10.2 Statuskategorien

Der Schema-Manager unterscheidet folgendermaßen zwischen den von ihm verwalteten Schemas:

- *installierte Schemas.* Diese werden auf der Benutzeroberfläche mit einem Häkchen angezeigt (*in der Abbildung unten sind die mit einem Häkchen versehenen und blau angezeigten Versionen der EPUB- und HL7v3 NE-Schemas installiert*). Wenn alle Versionen eines Schemas ausgewählt sind, wird ein Häkchen angezeigt. Wenn zumindest eine Version nicht ausgewählt ist, wird ein gefülltes Quadrat angezeigt. Sie können das Kontrollkästchen für ein installiertes Schema deaktivieren, um es zu **deinstallieren**; (*in der Abbildung unten ist die DocBook DTD installiert und wurde deaktiviert; sie wird daher für die Deinstallation vorbereitet*).
- *Nicht installierte verfügbare Schemas.* Diese werden auf der Benutzeroberfläche mit einem deaktivierten Kontrollkästchen angezeigt. Sie können die Schemas, die **installiert** werden sollen, auswählen.



- *Schemas, für die ein Upgrade zur Verfügung steht* sind diejenigen, die seit ihrer Installation vom Herausgeber überarbeitet wurden. Sie werden auf der Benutzeroberfläche durch ein -Symbol gekennzeichnet. Sie können für ein installiertes Schema ein **Patch** der verfügbaren überarbeiteten Version installieren.

Wichtige Punkte

- In der Abbildung oben sind beide CBCR-Schemas ausgewählt. Dasjenige mit einem blauen Hintergrund ist bereits installiert. Dasjenige mit dem gelben Hintergrund ist nicht installiert und wurde für die Installation ausgewählt. Beachten Sie, dass das Schema "HL7v3 NE 2010" nicht installiert ist und nicht für die Installation ausgewählt wurde.

- Ein gelber Hintergrund bedeutet, dass das Schema auf irgendeine Art geändert wird, wenn Sie auf die Schaltfläche **Anwenden** klicken. Wenn ein Schema deaktiviert ist und einen gelben Hintergrund hat, bedeutet dies, dass es bei Klick auf die Schaltfläche **Anwenden** deinstalliert wird. In der Abbildung oben ist dies bei der DocBook DTD der Fall.
- Bei Ausführung des Schema-Managers über die Befehlszeile wird der Befehl `list` 152 mit verschiedenen Optionen verwendet, um verschiedene Schemakategorien aufzulisten:

`xmlschemamanager.exe list`	Listet alle installierten und verfügbaren Schemas auf; auch verfügbare Upgrades werden angezeigt.
`xmlschemamanager.exe list -i`	Listet nur installierte Schemas auf; auch verfügbare Upgrades werden angezeigt.
`xmlschemamanager.exe list -u`	Listet Schemas auf, für die Upgrades zur Verfügung stehen

Anmerkung: Verwenden Sie unter Linux und macOS **`sudo ./xmlschemamanager list`**

 © 2016-2022 Altova GmbH

10.3 Anwenden eines Patch oder Installation eines Schemas

Anwenden eines Patch auf ein installiertes Schema

Von Zeit zu Zeit werden von den Herausgebern der XML-Schemas Patches (Upgrades oder Überarbeitungen) veröffentlicht. Wenn der Schema-Manager erkennt, dass Patches zur Verfügung stehen, werden diese in der Schemaliste des Schema-Managers angezeigt und Sie können diese Patches schnell installieren.

Über die Benutzeroberfläche

Patches werden mit dem Symbol gekennzeichnet. (*Siehe auch vorhergehendes Kapitel über Statuskategorien* 145*).* Falls Patches zur Verfügung stehen, ist die Schaltfläche **Patch-Auswahl** aktiv. Klicken Sie darauf, um alle Patches für die Installation auszuwählen und vorzubereiten. Auf der Benutzeroberfläche ändert sich das Symbol von Schemas, für die ein Patch installiert wird von in , und im Fenster "Meldungen" am unteren Rand des Dialogfelds werden die Patches, die angewendet werden, aufgelistet. Sobald Sie mit der Auswahl fertig sind, klicken Sie auf **Anwenden**. Alle Patches werden gemeinsam angewendet. Beachten Sie, dass ein für die Installation eines Patch markiertes Schema deinstalliert wird, wenn Sie die Auswahl aufheben.

Über das CLI

So wenden Sie einen Patch über die Befehlszeilenschnittstelle an:

1. Führen Sie den Befehl `list -u` 152 aus. Daraufhin werden alle Schemas, für die Upgrades zur Verfügung stehen, aufgelistet.
2. Führen Sie den Befehl `upgrade` 155 aus, um alle Patches zu installieren.

Installieren eines verfügbaren Schemas

Sie können Schemas entweder über die Benutzeroberfläche des Schema-Managers oder durch Senden der Schema-Manager-Installationsbefehle über die Befehlszeile installieren.

Anmerkung: Wenn das aktuelle Schema andere Schemas referenziert, werden auch die referenzierten Schemas installiert.

Über die Benutzeroberfläche

Um Schemas über die Benutzeroberfläche des Schema-Managers zu installieren, wählen Sie die gewünschten Schemas aus und klicken Sie auf **Anwenden**.

Sie können die gewünschten Schemas auch auf der Altova Website auswählen und eine herunterladbare `.altova_xmlschemas`-Datei generieren. Bei Doppelklick auf diese Datei wird der Schema-Manager aufgerufen, in dem die gewünschten Schemas bereits vorausgewählt sind. Sie müssen nur mehr auf **Anwenden** klicken.

Über das CLI

Um Schemas über die Befehlszeile zu installieren, rufen Sie den Befehl `install` 152 auf:

```
xmlschemamanager.exe install [options] Schema+
```

wobei es sich bei `Schema` um das/die gewünschte(n) Schema(s) bzw. eine `.altova_xmlschemas`-Datei handelt. Ein Schema wird von einem Identifier im Format `<name>-<version>` referenziert. (Die Identifier von

Schemas werden angezeigt, wenn Sie den Befehl `list` 152 ausführen.) Sie können beliebig viele Schemas eingeben. Nähere Informationen dazu finden Sie unter der Beschreibung des Befehls `install` 152.

Anmerkung: Verwenden Sie unter Linux oder macOS den Befehl **`sudo ./xmlschemamanager`**.

Installation eines benötigten Schemas

Wenn Sie einen XML-Schema-Befehl in Authentic Desktop ausführen und Authentic Desktop erkennt, dass ein zur Ausführung des Befehls erforderliches Schema nicht vorhanden oder unvollständig ist, wird der Schema-Manager mit Informationen über das/die fehlende(n) Schema(s) aufgerufen. Sie können die gewünschten Schemas dann über den Schema-Manager direkt installieren.

Alle bereits installierten Schemas können jederzeit durch Aufruf des Schema-Managers über **Extras | Schema-Manager** über die Benutzeroberfläche des Schema-Managers angezeigt werden.

© 2016-2022 Altova GmbH

10.4 Deinstallieren eines Schemas, Zurücksetzen

Deinstallieren eines Schemas

Sie können Schemas entweder über die Benutzeroberfläche des Schema-Managers oder durch Senden der Schema-Manager-Deinstallationsbefehle über die Befehlszeile deinstallieren.

Anmerkung: Wenn das gewünschte Schema andere Schemas referenziert, so werden auch die referenzierten Schemas deinstalliert.

Über die Benutzeroberfläche

Um Schemas über die Benutzeroberfläche des Schema-Managers zu deinstallieren, deaktivieren Sie die Kontrollkästchen der entsprechenden Schemas und klicken Sie auf **Anwenden**. Daraufhin werden die ausgewählten Schemas und die davon referenzierten Schemas deinstalliert.

Um alle Schemas zu deinstallieren, klicken Sie auf **Auswahl für alle aufheben** und anschließend auf **Anwenden**.

Über das CLI

Um Schemas über die Befehlszeile zu deinstallieren, rufen Sie den Befehl `uninstall` 154 auf:

```
xmlschemamanager.exe uninstall [options] Schema+
```

wobei es sich beim Argument `Schema` ein zu deinstallierendes Schema oder eine `.altova_xmlschemas`-Datei handelt. Ein Schema wird von einem Identifier im Format `<name>-<version>` definiert. (Die Identifier von Schemas werden angezeigt, wenn Sie den Befehl `list` 152 ausführen.) Sie können beliebig viele Schemas eingeben. Nähere Informationen dazu finden Sie unter der Beschreibung des Befehls `uninstall` 154.

Anmerkung: Verwenden Sie unter Linux oder macOS den Befehl `sudo ./xmlschemamanager`.

Zurücksetzen des Schema-Managers

Sie können den Schema-Manager zurücksetzen. Damit werden alle installierten Schemas und das Cache-Verzeichnis entfernt.

- Klicken Sie auf der Benutzeroberfläche auf **Auswahl zurücksetzen**.
- Führen Sie über die Benutzeroberfläche den Befehl `reset` 153 aus.

Nachdem Sie diesen Befehl ausgeführt haben, muss der Befehl `initialize` 151 ausgeführt werden, um das Cache-Verzeichnis neu zu erstellen. Führen Sie alternativ dazu den Befehl `reset` 153 mit der Option `-i` aus.

Beachten Sie, dass mit `reset-i` 153 die Originalinstallation des Produkts wiederhergestellt wird, daher wird empfohlen, nach dem Zurücksetzen auch den Befehl `update` 155 auszuführen. Führen Sie alternativ dazu den Befehl `reset` 153 mit den Optionen `-i` und `-u` aus.

10.5 Befehlszeilenschnittstelle (CLI)

Um den Schema-Manager über die Befehlszeile aufzurufen, müssen Sie den Pfad zur ausführbaren Datei kennen. Standardmäßig befindet sich die ausführbare Schema-Manager-Datei hier:

```
C:\ProgramData\Altova\SharedBetweenVersions\XMLSchemaManager.exe
```

Anmerkung: Nachdem Sie auf Linux- und macOS-Systemen das Verzeichnis in dasjenige, das die ausführbare Datei enthält, geändert haben, können Sie die ausführbare Datei mit **`sudo ./xmlschemamanager`** aufrufen. Das Präfix **`./`** gibt an, dass sich die ausführbare Datei im aktuellen Verzeichnis befindet. Das Präfix **`sudo`** gibt an, dass der Befehl mit Root-Rechten ausgeführt werden muss.

Befehlszeilensyntax

Die allgemeine Syntax zur Verwendung der Befehlszeile lautet folgendermaßen:

```
<exec> -h | --help | --version | <command> [options] [arguments]
```

Der senkrechte Balken **`|`** im Codefragment oben trennt eine Gruppe einander gegenseitig ausschließender Elemente. Optionale Elemente stehen innerhalb von eckigen Klammern **`[]`**. Im Prinzip können Sie den Pfad zur ausführbaren Datei, gefolgt von entweder `--h`, `--help` oder `--version`-Optionen oder gefolgt von einem Befehl eingeben. Jeder Befehl kann Optionen und Argumente haben. Die Liste der Befehle wird in den folgenden Abschnitten beschrieben.

10.5.1 help

Mit diesem Befehl erhalten Sie Hilfe zu Befehlen zur ausführbaren Schema-Manager-Datei.

Syntax

```
<exec> help [Befehl]
```

`[Befehl]` ist hierbei ein optionales Argument zur Angabe jedes beliebigen gültigen Befehlsnamens.

Beachten Sie dazu Folgendes:

- Sie können die Hilfe zu einem Befehl auch durch Eingabe des Befehls, gefolgt von **`-h`** oder **`--help`** aufrufen, z.B: **`<exec> list -h`**
- Wenn Sie **`-h`** oder **`--help`** direkt nach dem Namen der ausführbaren Datei und vor einem Befehl eingeben, wird die allgemeine Hilfe (und nicht die Hilfe zu einem bestimmten Befehl) angezeigt, z.B: **`<exec> -h list`**

Beispiel

Mit dem folgenden Befehl wird Hilfe zum Befehl `list` angezeigt:

```
xmlschemamanager help list
```

 © 2016-2022 Altova GmbH

10.5.2 info

Mit diesem Befehl werden ausführliche Informationen über die einzelnen als `Schema`-Argument angegebenen Schemas angezeigt. Darin enthalten sind Titel, Version, Beschreibung, Herausgeber der jeweils angegebenen Schemas und davon referenzierte Schemas sowie die Information, ob das Schema installiert ist oder nicht.

Syntax

```
<exec> info [options] Schema+
```

- Das Argument **`Schema`** ist der Name eines Schemas oder Teil eines Schemanamens. (Die Paket-ID eines Schemas und detaillierte Informationen über ihren Installationsstatus erhalten Sie mit dem Befehl `list` 152.)
- Mit **`<exec> info -h`** können Sie die Hilfe zum Befehl anzeigen.

Beispiel

Mit dem folgenden Befehl werden Informationen über das jeweils neueste **`DocBook-DTD`**- und **`NITF`**-Schemas angezeigt.

```
xmlschemamanager info doc nitf
```

10.5.3 initialize

Mit diesem Befehl wird die Schema-Manager-Umgebung initialisiert. Sie erstellen damit ein Cache-Verzeichnis, in dem Informationen über alle Schemas lokal gespeichert werden. Die Initialisierung erfolgt automatisch bei der ersten Installation einer Schema-fähigen Altova-Applikation. Normalerweise muss dieser Befehl nicht ausgeführt werden. Nach Ausführung des **`reset`**-Befehls ist dies allerdings erforderlich.

Syntax

```
<exec> initialize | init [options]
```

Optionen

Für den Befehl **`initialize`** stehen die folgenden Optionen zur Verfügung:

`--silent, --s`	Nur Fehlermeldungen anzeigen. Der Standardwert ist **`false`**.
`--verbose, --v`	Anzeige detaillierter Informationen während der Ausführung. Der Standardwert ist **`false`**.
`--help, --h`	Anzeige der Hilfe zum Befehl.

Beispiel

Mit dem folgenden Befehl wird der Schema-Manager initialisiert:

```
xmlschemamanager initialize
```

10.5.4 install

Mit diesem Befehl installieren Sie ein oder mehrere Schemas.

Syntax

```
<exec> install [options] Schema+
```

Um mehrere Schemas zu installieren, fügen Sie das Argument **Schema** mehrmals hinzu.

Als **Schema**-Argument kann eines der folgenden verwendet werden:

- Ein Schema-Identifier (im Format **<name>-<version>**, z.B: **cbcr-2.0**). Um die Schema-Identifier der gewünschten Schemas zu eruieren, führen Sie den Befehl list 152 aus. Sie können auch einen abgekürzten Identifier verwenden, sofern dieser eindeutig ist, z.B. **docbook**. Falls Sie einen abgekürzten Identifier verwenden, wird die neueste Version dieses Schemas installiert.
- Der Pfad zu einer von der Altova-Website heruntergeladenen **.altova_xmlschemas**-Datei. Informationen zu diesen Dateien finden Sie in der *Einführung zu Schema-Manager: Funktionsweise* 139.

Optionen
Für den Befehl **install** stehen die folgenden Optionen zur Verfügung:

`--silent, --s`	Nur Fehlermeldungen anzeigen. Der Standardwert ist **false**.
`--verbose, --v`	Anzeige detaillierter Informationen während der Ausführung. Der Standardwert ist **false**.
`--help, --h`	Anzeige der Hilfe zum Befehl.

Beispiel

Mit dem folgenden Befehl werden das CBCR 2.0 (Country-By-Country Reporting)-Schema und die neueste DocBook-DTD installiert:

```
xmlschemamanager install cbcr-2.0 docbook
```

10.5.5 list

Mit diesem Befehl werden vom Schema-Manager verwaltete Schemas aufgelistet. In der Liste wird eine der folgenden Informationen angezeigt:

- alle verfügbaren Schemas
- Schemas, die im Namen den im Argument **Schema** angegebenen String enthalten
- nur installierte Schemas
- Nur Schemas, für die ein Upgrade installiert werden kann

 © 2016-2022 Altova GmbH

Syntax

```
<exec> list | ls [options] Schema?
```

Wenn kein **Schema**-Argument angegeben wird, werden alle verfügbaren Schemas aufgelistet. Andernfalls werden die durch die angegebenen Optionen definierten Schemas aufgelistet (*siehe Beispiel unten*). Beachten Sie, dass Sie das Argument **Schema** mehrfach angeben können.

Optionen
Für den Befehl **list** stehen die folgenden Optionen zur Verfügung:

`--installed, --i`	Auflisten nur der installierten Schemas. Der Standardwert ist **false**.
`--upgradeable, --u`	Auflisten nur derjenigen Schemas, für die Upgrades (Patches) zur Verfügung stehen. Der Standardwert ist **false**.
`--help, --h`	Anzeige der Hilfe zum Befehl.

Beispiele

- Um alle verfügbaren Schemas aufzulisten, führen Sie den folgenden Befehl aus: **xmlschemamanager list**
- Um nur installierte Schemas aufzulisten, führen Sie **xmlschemamanager list -i** aus.
- Um Schemas, die in ihrem Namen entweder "doc" oder "nitf" enthalten, aufzulisten, führen Sie **xmlschemamanager list doc nitf** aus.

10.5.6 reset

Mit diesem Befehl werden alle installierten Schemas und das Cache-Verzeichnis entfernt. Ihre Schemaumgebung wird vollständig zurückgesetzt. Nachdem Sie diesen Befehl ausgeführt haben, muss der Befehl initialize 151 ausgeführt werden, um das Cache-Verzeichnis neu zu erstellen. Führen Sie alternativ dazu den Befehl **reset** mit der Option **-i** aus. Da mit **reset-i** die Originalinstallation des Produkts wiederhergestellt wird, wird empfohlen, nach dem Zurücksetzen und Initialisieren auch den Befehl update 155 auszuführen. Führen Sie alternativ dazu den Befehl **reset** mit den Optionen **-i** und **-u** aus.

Syntax

```
<exec> reset [Optionen]
```

Optionen
Für den Befehl **reset** stehen die folgenden Optionen zur Verfügung:

`--init, --i`	Initialisierung des Schema-Managers nach dem Zurücksetzen. Der Standardwert ist **false**.
`--update, --u`	Aktualisiert die Liste der verfügbaren Schemas im Cache. Der Standardwert ist **false**.

`--silent, --s`	Nur Fehlermeldungen anzeigen. Der Standardwert ist **`false`**.
`--verbose, --v`	Anzeige detaillierter Informationen während der Ausführung. Der Standardwert ist **`false`**.
`--help, --h`	Anzeige der Hilfe zum Befehl.

Beispiele

- Um den Schema-Manager zurückzusetzen, führen Sie den folgenden Befehl aus: **`xmlschemamanager reset`**
- Um den Schema-Manager zurückzusetzen und ihn zu initialisieren, führen Sie **`xmlschemamanager reset -i`** aus.
- Um den Schema-Manager zurückzusetzen, ihn zu initialisieren und seine Schemaliste zu aktualisieren, führen Sie **`xmlschemamanager reset -i -u`** aus.

10.5.7 uninstall

Mit diesem Befehl deinstallieren Sie ein oder mehrere Schemas. Standardmäßig werden auch alle Schemas, die vom der aktuellen Schema referenziert werden, deinstalliert. Um nur das aktuelle Schema zu deinstallieren und die referenzierten Schemas beizubehalten, setzen Sie die Option `--k`.

Syntax

```
<exec> uninstall [options] Schema+
```

Um mehrere Schemas zu deinstallieren, fügen Sie das Argument **`Schema`** mehrmals hinzu.

Als **`Schema`**-Argument kann eines der folgenden verwendet werden:

- Ein Schema-Identifier (im Format **`<name>-<version>`**, z.B: **`cbcr-2.0`**). Um die Schema-Identifier der installierten Schemas zu eruieren, führen Sie den Befehl `list -i` 152 aus. Sie können auch einen abgekürzten Schemanamen verwenden, sofern dieser eindeutig ist, z.B. **`docbook`**. Falls Sie einen abgekürzten Namen verwenden, werden alle Schemas, die die Abkürzung in ihrem Namen enthalten, deinstalliert.
- Der Pfad zu einer von der Altova-Website heruntergeladenen **`.altova_xmlschemas`**-Datei. Informationen zu diesen Dateien finden Sie in der *Einführung zu Schema-Manager: Funktionsweise* 139.

Optionen
Für den Befehl **`uninstall`** stehen die folgenden Optionen zur Verfügung:

`--keep-references, --k`	Definieren Sie diese Option, um referenzierte Schemas beizubehalten. Der Standardwert ist **`false`**.
`--silent, --s`	Nur Fehlermeldungen anzeigen. Der Standardwert ist **`false`**.
`--verbose, --v`	Anzeige detaillierter Informationen während der Ausführung. Der Standardwert ist **`false`**.

 © 2016-2022 Altova GmbH

`--help, --h`	Anzeige der Hilfe zum Befehl.

Beispiel

Mit dem folgenden Befehl werden die Schemas CBCR 2.0 und EPUB 2.0 und deren Abhängigkeiten deinstalliert:

```
xmlschemamanager uninstall cbcr-2.0 epub-2.0
```

Mit dem folgenden Befehl wird das **`eba-2.10`**-Schema, nicht aber die davon referenzierten Schemas deinstalliert:

```
xmlschemamanager uninstall --k cbcr-2.0
```

10.5.8 update

Mit diesem Befehl wird die Liste der über den Online-Speicher verfügbaren Schemas abgefragt und das lokale Cache-Verzeichnis wird aktualisiert. Normalerweise muss dieser Befehl nur ausgeführt werden, wenn Sie `reset` 153 und `initialize` 151 ausgeführt haben.

Syntax

```
<exec> update [options]
```

Optionen

Für den Befehl **`update`** stehen die folgenden Optionen zur Verfügung:

`--silent, --s`	Nur Fehlermeldungen anzeigen. Der Standardwert ist **`false`**.
`--verbose, --v`	Anzeige detaillierter Informationen während der Ausführung. Der Standardwert ist **`false`**.
`--help, --h`	Anzeige der Hilfe zum Befehl.

Beispiel

Mit dem folgenden Befehl wird der lokale Cache mit der Liste der neuesten Schemas aktualisiert:

```
xmlschemamanager update
```

10.5.9 upgrade

Mit diesem Befehl werden alle installierten Schemas, für die ein Upgrade installiert werden kann, auf die neueste verfügbare *Patch*-Version aktualisiert. Um herauszufinden, welche Schemas aktualisiert werden können, starten Sie den Befehl `list-u` 152.

Anmerkung: Der Befehl **`upgrade`** entfernt ein veraltetes Schema, falls keine neuere Version zur Verfügung steht.

Syntax

```
<exec> upgrade [Optionen]
```

Optionen

Für den Befehl **upgrade** stehen die folgenden Optionen zur Verfügung:

`--silent, --s`	Nur Fehlermeldungen anzeigen. Der Standardwert ist **false**.
`--verbose, --v`	Anzeige detaillierter Informationen während der Ausführung. Der Standardwert ist **false**.
`--help, --h`	Anzeige der Hilfe zum Befehl.

© 2016-2022 Altova GmbH

11 Authentic Desktop in Visual Studio

Sie können Authentic Desktop in die Microsoft Visual Studio IDE-Versionen 2012/2013/2015/2017/2019/2022 integrieren. Auf diese Weise stehen Ihnen sowohl die XML-Bearbeitungsfunktionen als auch die professionelle Entwicklungsumgebung von Visual Studio zur Verfügung:

In diesem Abschnitt wird Folgendes beschrieben:

- der allgemeine Installationsvorgang 158 und die Integration des Authentic Desktop Plugin in Visual Studio.
- Unterschiede 159 zwischen der Visual Studio -Version und der Standalone-Version.

11.1 Installieren des Authentic Plugin für Visual Studio

Um das Authentic Desktop Plugin für Visual Studio zu installieren, gehen Sie folgendermaßen vor:

1. Installieren Sie Microsoft Visual Studio 2012/2013/2015/2017/2019/2022. Beachten Sie, dass Visual Studio ab Visual Studio 2022 nur als 64-Bit-Applikation zur Verfügung steht.
2. Installieren Sie Authentic Desktop. Wenn Sie Visual Studio 2022+ installiert haben, müssen Sie die 64-Bit-Version von Authentic Desktop installieren.
3. Downloaden Sie das Authentic Desktop Integrationspaket für Microsoft Visual Studio. Dieses Paket steht auf der Authentic Desktop Download-Seite unter www.altova.com zur Verfügung.

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

Achtung

Sie müssen das richtige Integrationspaket für Ihre jeweilige Authentic Desktop Version (die aktuelle Version ist 2023) verwenden.

© 2016-2022 Altova GmbH

11.2 Unterschiede zur Standalone-Version

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

Eingabehilfen (Tool-Fenster in Visual Studio)

Die Eingabehilfen von Authentic Desktop stehen in Visual Studio als Tools-Fenster zur Verfügung. Beachten Sie dazu die folgenden Punkte. (Eine Beschreibung der Eingabehilfen und der Benutzeroberfläche von Authentic Desktop finden Sie im Abschnitt Benutzeroberfläche und Umgebung 13).

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

Authentic Desktop Befehle als Visual Studio-Befehle

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

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

Anmerkung: In Versionen ab Visual Studio 2019 können Authentic Desktop-Funktionalitäten über das Menü **Erweiterungen** von Visual Studio aufgerufen werden. In früheren Versionen von Visual Studio, Authentic Desktop stehen die Funktionen in Visual Studio-Menüs der obersten Ebene zur Verfügung.

Anmerkung: Symbolleisten-Schaltflächen werden nicht unterstützt. Wenn Sie in Authentic Desktop eine Symbolleisten-Schaltfläche zur Ausführung eines Befehls oder Skripts definiert haben, steht diese Schaltfläche im Plug-in nicht zur Verfügung.

Zusätzliche Anmerkungen

Hier finden Sie einige zusätzliche Anmerkungen und Tipps:

- Um eine XML-Datei mit dem Authentic-Plugin zu bearbeiten, wählen Sie den Befehl **Datei | Öffnen**. Wählen Sie anschließend im Dialogfeld "Datei Öffnen" aus, ob Sie eine globale Authentic-Ressource oder eine Authentic-Datei über eine URL öffnen möchten.

12 Authentic Desktop in Eclipse

Eclipse ist eine Open-Source-Umgebung, in die verschiedene Arten von Applikationen in Form von Plug-Ins integriert werden können. Mit Hilfe des Authentic Desktop-Integrationspakets für Eclipse können Sie die Funktionalitäten von Authentic Desktop in die Eclipse-Plattform für Windows integrieren und aufrufen. Die folgenden Eclipse-Versionen werden unterstützt: 2022-09, 2022-06, 2022-03, 2021-12.

In diesem Abschnitt werden folgende Punkte erläutert:

- Installation des Integrationspakets für Eclipse und Integration von Authentic Desktop in Eclipse 161
- Die Authentic Desktop-Perspektive in Eclipse 163
- Andere Authentic Desktop-Einstiegspunkte in Eclipse 166

 © 2016-2022 Altova GmbH

12.1 Installation des Integrationspakets für Eclipse

Voraussetzungen

- Eclipse 2022-09, 2022-06, 2022-03, 2021-12 (http://www.eclipse.org), 64-Bit.
- Java Runtime Environment (JRE) oder Java Development Kit (JDK) für die 64-Bit-Plattform.
- Authentic Desktop 64-Bit.

Anmerkung: Alle oben aufgelisteten Programme müssen die 64-Bit-Plattform haben. Die Integration mit älteren Eclipse 32-Bit-Plattformen wird nicht mehr unterstützt, funktioniert aber eventuell noch.

Mit den obigen Voraussetzungen können Sie das Authentic Desktop Integrationspaket (64-Bit) installieren, um Authentic Desktop in Eclipse zu integrieren. Die Integration kann entweder während der Installation des Installationspakets oder nach Installation des Integrationspakets manuell von Eclipse aus durchgeführt werden. Das Authentic Desktop Integrationspaket kann von https://www.altova.com/de/components/download heruntergeladen werden.

Anmerkung: Eclipse muss während der Installation bzw. Deinstallation des Authentic Desktop Integrationspakets geschlossen sein.

Integration von Authentic Desktop während der Installation des Integrationspakets

Sie können Authentic Desktop während der Installation des Authentic Desktop-Integrationspakets in Eclipse integrieren. Gehen Sie dazu folgendermaßen vor:

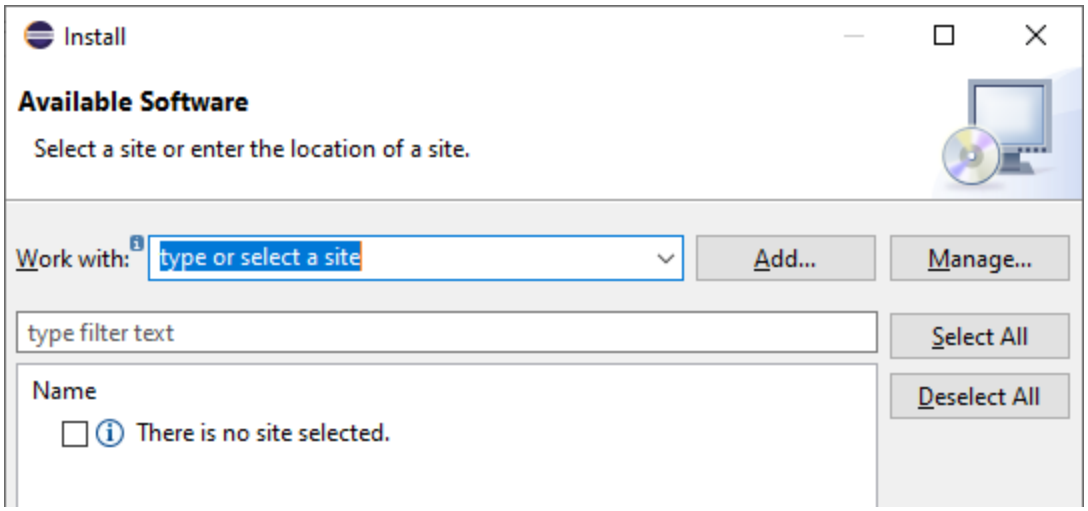
1. Starten Sie das Authentic DesktopIntegrationspaket, um den Installationsassistenten aufzurufen.
2. Führen Sie mit dem Assistenten die ersten Schritte der Installation durch.
3. Aktivieren Sie beim Integrationsschritt die Option *Altova Authentic Desktop Plug-in mit diesem Assistenten in Eclipse integrieren* und suchen Sie nach dem Ordner, in dem sich die ausführbare Datei von Eclipse befindet (`eclipse.exe`).
4. Klicken Sie auf **Weiter** und schließen Sie die Installation ab.

Die Authentic Desktop-Perspektive und die entsprechenden Menüs stehen daraufhin beim nächsten Start von Eclipse zur Verfügung.

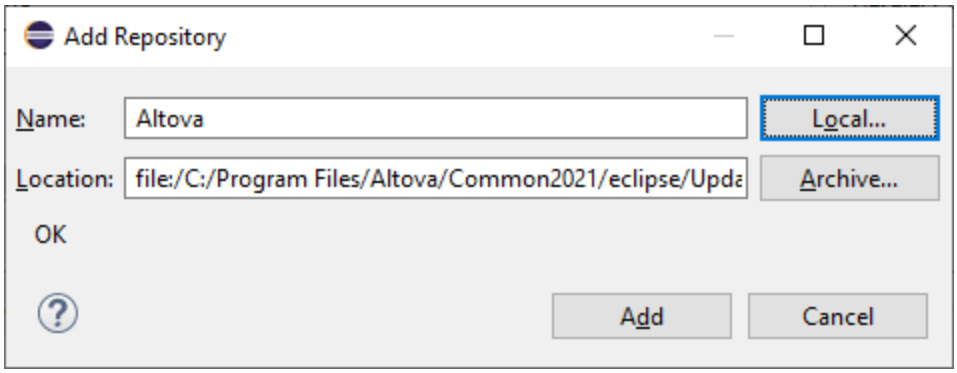
Manuelle Integration von Authentic Desktop in Eclipse

Nach Installation des Authentic Desktop-Integrationspakets können Sie Authentic Desktop folgendermaßen manuell in Eclipse integrieren:

1. Klicken Sie in Eclipse auf den Menübefehl **Help | Install new Software**.
2. Klicken Sie im Dialogfeld "Install", das daraufhin angezeigt wird, auf die Schaltfläche **Add**.



3. Klicken Sie im Dialogfeld "Add Repository" auf **Local**. Navigieren Sie zum Ordner `C:\Programme\Altova\Common2023\eclipse\UpdateSite`, wählen Sie ihn aus und klicken Sie auf OK. Geben Sie einen Namen dafür ein (z.B. "Altova").



4. Wiederholen Sie die Schritte 2 bis 3 und wählen Sie diesmal den Ordner `C:\Programme\Altova\Authentic\eclipse\UpdateSite`aus. Geben Sie einen Namen wie z.B. "Altova Authentic Desktop" ein.
5. Wählen Sie im Installationsdialogfeld *Only Local Sites* aus. Wählen Sie als nächstes den "Altova category"-Ordner aus und klicken Sie auf **Next**.
6. Überprüfen Sie die zu installierenden Objekte und klicken Sie zum Fortfahren auf **Next**.
7. Aktivieren Sie das entsprechende Kontrollkästchen, um die Lizenzvereinbarung zu akzeptieren.
8. Klicken Sie anschließend auf **Finish**, um die Installation fertig zu stellen.

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

 © 2016-2022 Altova GmbH

12.2 Authentic Desktop-Perspektive in Eclipse

Eine Perspektive in Eclipse ist eine Ansicht der Benutzeroberfläche, die mit den Funktionalitäten einer bestimmten Applikation konfiguriert ist. Nachdem Authentic Desktop in Eclipse integriert wurde, steht in Eclipse eine neue Perspektive namens Authentic Desktop, zur Verfügung. Diese Perspektive ähnelt der Benutzeroberfläche von Authentic Desktop und enthält eine Reihe ihrer Komponenten.

Wenn eine Datei geöffnet wird, deren Dateityp (`.xml`) (z.B.`.mfd`) mit Authentic Desktop verknüpft ist, kann diese Datei in der Authentic Desktop Perspektive bearbeitet werden. Ebenso kann eine Datei eines anderen Dateityps in einer anderen Perspektive von Eclipse geöffnet werden. Zusätzlich dazu können Sie die Perspektive für jeden Dateityp wechseln (*siehe unten*) und so die Datei in einer anderen Umgebung bearbeiten bzw. verarbeiten.

Perspektiven haben daher zwei wichtige Vorteile:

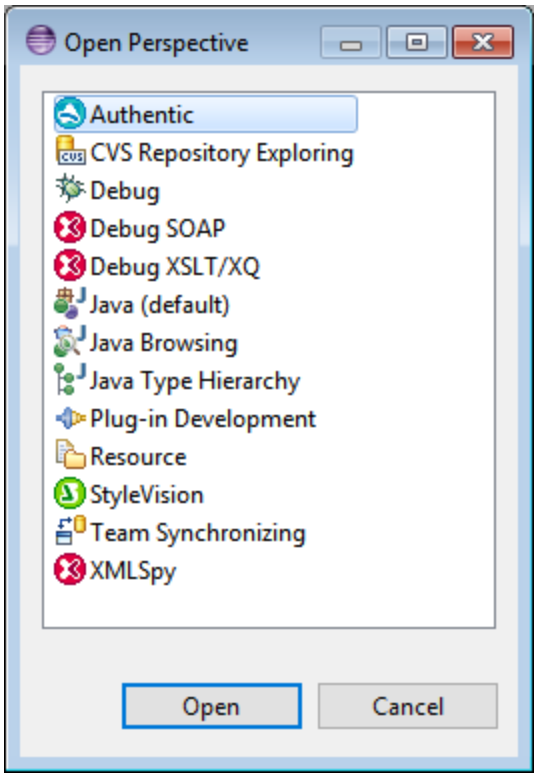
1. Die Arbeitsumgebung für die aktive Datei kann schnell gewechselt werden und
2. Sie können zwischen Dateien wechseln, ohne eine neue Entwicklungsumgebung öffnen zu müssen (die mit der Datei verknüpfte Umgebung steht in einer Perspektive zur Verfügung).

Beim Arbeiten mit der Authentic Desktop-Perspektive kommen die folgenden unten beschriebenen Verfahren zum Einsatz:

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

Wechseln in die Authentic Desktop Perspektive

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



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

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

Einstellungen für die Authentic Desktop Perspektive

Wählen Sie den Menübefehl **Window | Preferences**, um das Dialogfeld "Preferences" aufzurufen. Wählen Sie in der Liste der Perspektiven auf der linken Seite Authentic Desktop aus und wählen Sie dann die gewünschten Einstellungen aus. Klicken Sie anschließend auf **OK**.

Zu den Einstellungen einer Perspektive gehören:

- Der automatische Wechsel in die Authentic Desktop-Perspektive, wenn eine Datei eines verknüpften Dateityps (*siehe oben*) geöffnet wird
- Optionen zum Inkludieren oder Exkludieren einzelner Authentic Desktop-Symbolleisten
- Zugriff auf die Authentic Desktop-Optionen.

 © 2016-2022 Altova GmbH

Anpassen der Authentic Desktop-Perspektive

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

- Auf den Registern *Toolbar Visibility* und *Menu Visibility* können Sie festlegen, welche Symbolleisten und Menüs angezeigt werden sollen.
- Auf dem Register *Action Set Availability* können Sie Aktionsgruppen zu den übergeordneten Menüs und zur Symbolleiste hinzufügen. Um eine Aktionsgruppe zu aktivieren, klicken Sie auf das entsprechende Kontrollkästchen.
- Auf dem Register *Shortcuts* des Dialogfelds "Customize Perspective" können Sie die Shortcuts für Untermenüs festlegen. Aktivieren Sie in der Untermenü-Auswahlliste das gewünschte Untermenü. Wählen Sie anschließend eine Shortcut-Kategorie aus und aktivieren Sie die Shortcuts, die in der Perspektive inkludiert werden sollen.

Klicken Sie auf **Apply and Close**, um die Anpassung abzuschließen und damit die Änderungen wirksam werden.

12.3 Andere Authentic Desktop-Einstiegspunkte in Eclipse

Zusätzlich zur Authentic Desktop-Perspektive können die Authentic Desktop-Funktionalitäten über zwei weitere Einstiegspunkte in Eclipse aufgerufen werden:

- Authentic Desktop-Menü
- Authentic Desktop-Symbolleiste

Authentic Desktop Menü in Eclipse

Das **Authentic Desktop**-Menü von Eclipse enthält Authentic Desktop-Befehle, über die wichtige Authentic Desktop-Funktionalitäten zur Verfügung stehen. Diese Befehle kommen in verschiedenen Menüs der Standalone-Version von Authentic Desktop vor.

Authentic Desktop-Symbolleiste in Eclipse

Die Authentic Desktop-Symbolleiste in Eclipse (*Abbildung unten*) enthält zwei Schaltflächen.

Diese Schaltflächen haben die folgenden Funktionen:

- Aufruf der Authentic Desktop-Hilfe
- Aufruf der Authentic Desktop-Befehle (anstatt diese über das **Authentic Desktop**-Menü, *siehe oben*) aufzurufen.

Anmerkung: Symbolleisten-Schaltflächen werden nicht unterstützt. Wenn Sie in Authentic Desktop eine Symbolleisten-Schaltfläche zur Ausführung eines Befehls oder Skripts definiert haben, steht diese Schaltfläche im Plug-in nicht zur Verfügung.

© 2016-2022 Altova GmbH

13 Menübefehle

In diesem Abschnitt finden Sie eine Beschreibung aller Authentic Desktop-Menübefehle. Die Standard-Windows-Befehle wie (**Öffnen**, **Speichern**, **Ausschneiden**, **Kopieren** und **Einfügen**) befinden sich in den Menüs Datei 168 und Bearbeiten 184.

Nachstehend finden Sie eine Liste der Menüs von Authentic Desktop.

- Menü "Datei" 168
- Menü "Bearbeiten" 184
- Menü "Projekt" 187
- Menü "XML" 219
- Menü "XSL/XQuery" 221
- Menü "Authentic" 228
- Menü "Ansicht" 239
- Menü "Browser" 240
- Menü "Extras" 241
- Menü "Fenster" 285
- Menü "Hilfe" 287

Außerdem enthält dieser Abschnitt eine Beschreibung der Befehle, die über die Befehlszeile 294 verwendet werden können.

13.1 Menü "Datei"

Das Menü **Datei** enthält Befehle zu Dateioperationen, die wie in den meisten Windows-Applikationen geordnet sind. Diese Befehle sind:

- Neu 168
- Öffnen 169
- Neu laden 174
- Kodierung 174
- Schließen, Alle schließen, Alle inaktiven schließen 175
- Speichern, Speichern unter, Alles speichern 175
- Als Mail senden 181
- Drucken 182
- Druckvorschau 182
- Druckereinrichtung 182
- Zuletzt verwendete Dateien 183
- Beenden 183

13.1.1 Neu

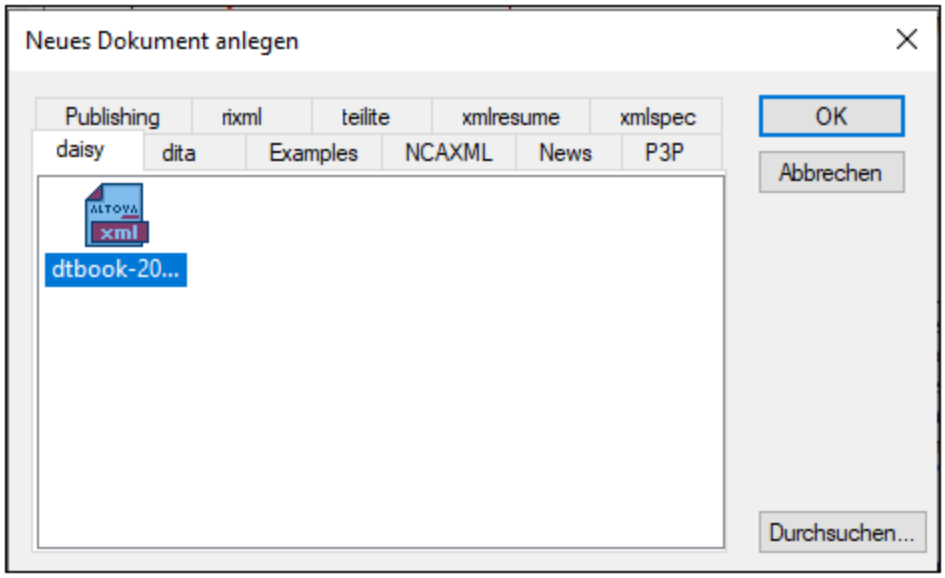
Schaltfläche und Tastenkürzel

Symbol:	
Tastenkürzel:	**Strg+N**

Beschreibung

Mit diesem Befehl öffnen Sie eine neue XML-Dokumentvorlage in der Authentic View-Ansicht. Die XML-Dokumentvorlage basiert auf einem StyleVision Power Stylesheet (.sps-Datei) und wird durch Auswahl des StyleVision Power Stylesheet im Dialogfeld "Neues Dokument anlegen" (*Abbildung unten*) geöffnet. Nachdem Sie ein SPS ausgewählt und mit OK bestätigt haben, wird die für diese SPS-Datei definierte XML-Dokumentvorlage in der Authentic View-Ansicht geöffnet.

 © 2016-2022 Altova GmbH



Im Dialogfeld "Neues Dokument anlegen" steht eine Reihe von XML-Dokumentvorlagen, die auf beliebten DTDs oder Schemas basieren, zur Auswahl zur Verfügung. Alternativ dazu können Sie eine benutzerdefinierte SPS-Datei, der eine XML-Vorlagendatei zugewiesen wurde, aussuchen. SPS-Dateien werden mit Altova StyleVision, einer Applikation zum Erstellen von XML-Dokumentvorlagen anhand einer DTD oder eines XML-Schemas, erstellt. Nachdem Sie die erforderliche SPS-Datei in StyleVision erstellt haben, wird der SPS-Datei (in StyleVision) eine XML-Datei als XML-Vorlagendatei zugewiesen. Die Daten in dieser XML-Datei liefern die Ausgangsdaten der neuen, in der Authentic View-Ansicht von Authentic Desktop geöffneten Dokumentvorlage.

Die neue XML-Dokumentvorlage enthält daher die im SPS definierten Layout-Eigenschaften und die Daten der als XML-Vorlagendatei ausgewählten XML-Datei. Der Benutzer der Authentic View -Ansicht kann die XML-Dokumentvorlage nun über eine grafische WYSIWYG-Oberfläche bearbeiten und als XML-Dokument speichern.

13.1.2 Öffnen

Schaltfläche und Tastenkürzel

Schaltfläche:	
Tastenkürzel:	**Strg+O**

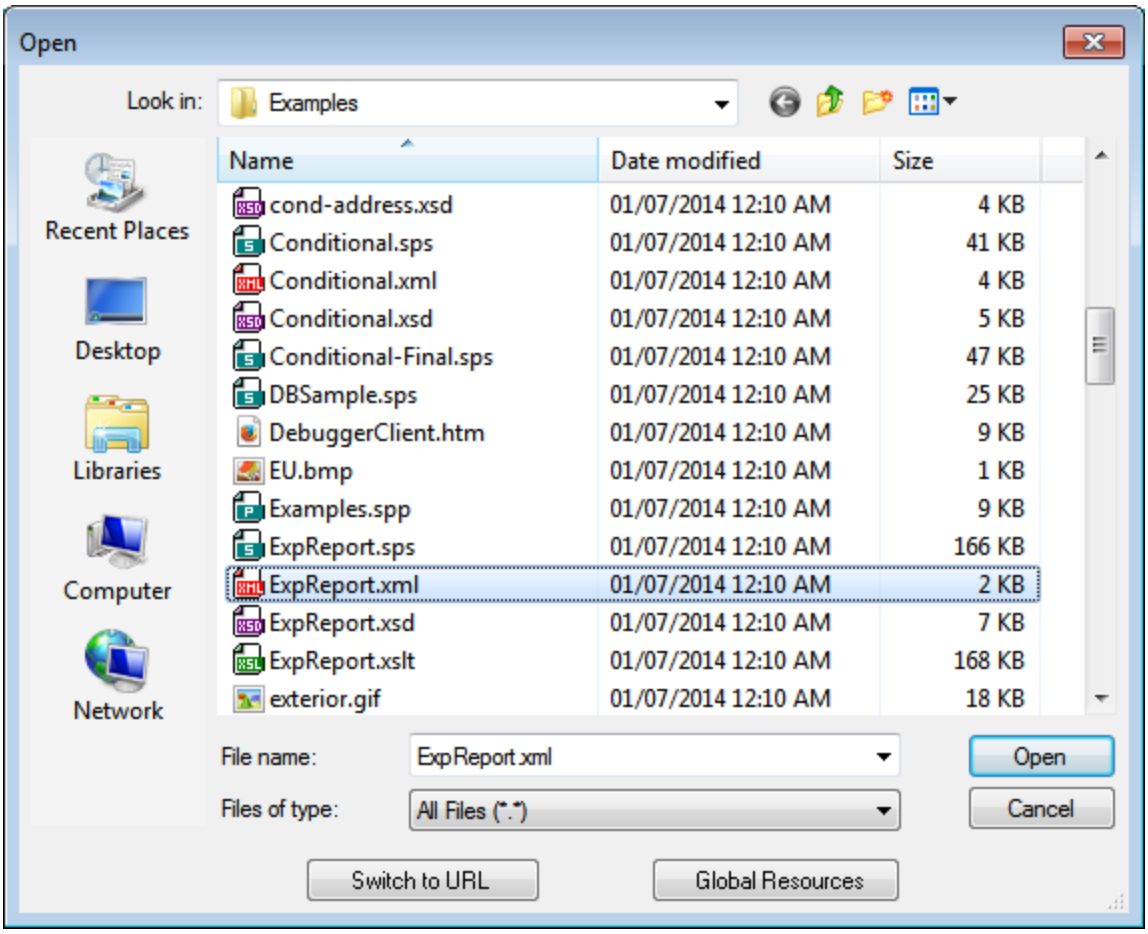
Beschreibung

Mit **Öffnen** rufen Sie das bekannte Windows-Dialogfeld "Öffnen" auf, über das Sie jedes XML-Dokument oder jede Textdatei öffnen können. Sie haben auch die Möglichkeit, mehrere Dateien auszuwählen. Im Listenfeld "Dateityp" können Sie auswählen, welche Dateien im Dialogfeld angezeigt werden sollen (die Liste der verfügbaren Dateiarten kann auf dem Register "Dateiarten" unter 271 **Extras | Optionen** 271 eingestellt werden). Beim Öffnen einer XML-Datei wird diese auf Wohlgeformtheit überprüft. Ist sie nicht wohlgeformt, wird eine Fehlermeldung angezeigt. Beheben Sie den Fehler und wählen Sie den Menübefehl **XML | Wohngeformtheit prüfen (F7)** 219, um die Datei erneut zu überprüfen. Wenn Sie die Option Validierung beim Öffnen der Datei 268

aktiviert haben und die Datei ungültig ist, wird ebenfalls eine Fehlermeldung angezeigt. Beheben Sie den Fehler und klicken Sie auf **XML | XML validieren (F8)** 219, um das Dokument erneut zu validieren.

▼ Auswählen und Speichern von Dateien über URLs und globale Ressourcen

Es gibt mehrere Dialogfelder zum Öffnen und Speichern von Dateien, über die Sie die gewünschte Datei entweder über eine URL oder eine globale Ressource (*siehe Abbildung unten*) auswählen bzw. speichern können. Wählen Sie den Befehl **Zu URL wechseln** oder **Globale Ressourcen**, um das entsprechende Dialogfeld aufzurufen.

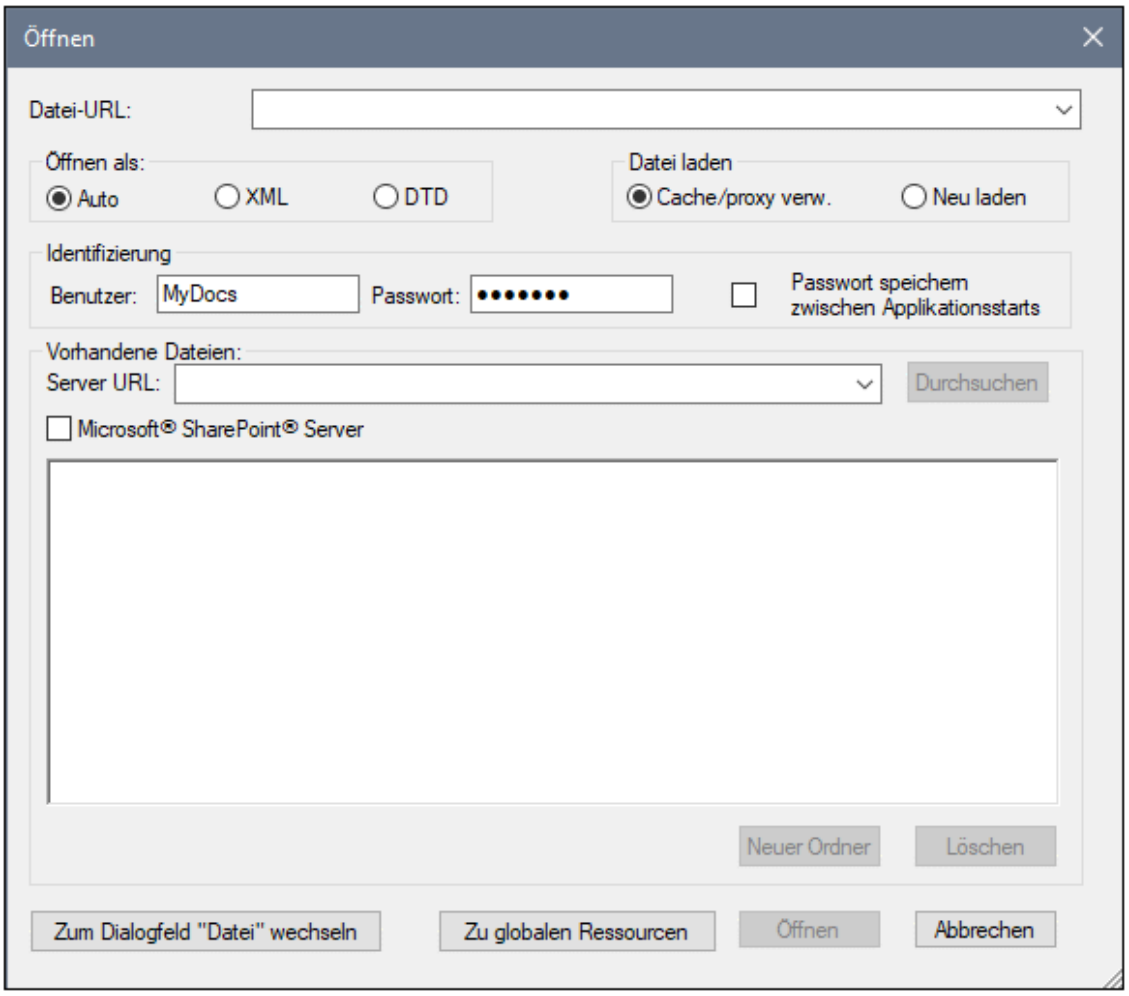


Auswählen von Dateien über URLs

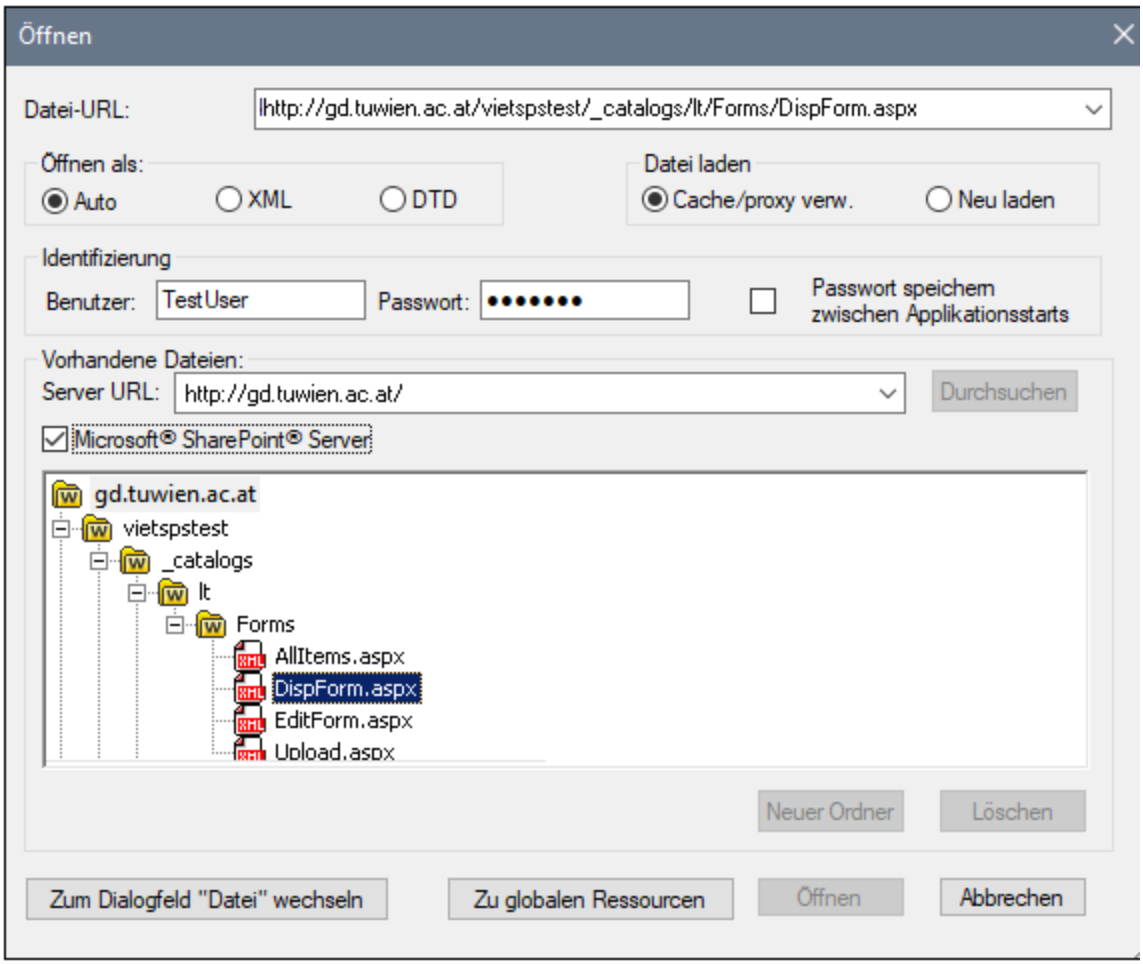
Um eine Datei über eine URL auszuwählen, gehen Sie folgendermaßen vor:

1. Klicken Sie auf den Befehl **Zu URL wechseln**. Daraufhin wechselt das Dialogfeld "Öffnen" bzw. "Speichern" in den URL-Modus (*In der Abbildung unten sehen Sie das Dialogfeld "Öffnen"*).

 © 2016-2022 Altova GmbH



2. Geben Sie die gewünschte URL in das Feld *Server URL* ein (*Abbildung oben*). Wenn es sich beim Server um einen Microsoft® SharePoint® Server handelt, aktivieren Sie das Kontrollkästchen *Microsoft® SharePoint® Server.* Nähere Informationen zum Arbeiten mit Dateien dieses Servertyps finden Sie unter den Anmerkungen weiter unten zu Microsoft® SharePoint® Server.
3. Geben Sie Ihre Benutzer-ID und Ihr Passwort in die Felder *Benutzer* und *Passwort* ein, wenn der Server passwortgeschützt ist.
4. Klicken Sie auf **Durchsuchen**, um die Verzeichnisstruktur des Servers zu sehen und darin zu navigieren.
5. Suchen Sie in die Ordnerstruktur Sie die gewünschte Datei und klicken Sie darauf.



Die Datei-URL wird im Feld "Datei-URL" angezeigt (*Abbildung oben*). Die Schaltflächen **Öffnen** oder **Speichern** werden erst zu diesem Zeitpunkt aktiv.

6. Klicken Sie auf **Öffnen,** um die Datei zu laden bzw. auf **Speichern**, um sie zu speichern.

Beachten Sie bitte die folgenden Punkte:

- Die Durchsuchen-Funktion steht nur auf Servern zur Verfügung, die WebDAV und Microsoft SharePoint Server unterstützen. Die unterstützten Protokolle sind FTP, HTTP und HTTPS.
- Damit Sie mehr Kontrolle über den Ladevorgang haben, können Sie auswählen, ob die Datei über den lokalen Cache-Speicher oder über einen Proxy-Server (wodurch der Ladevorgang beträchtlich schneller geht, wenn die Datei bereits einmal geladen war) geladen werden soll. Alternativ dazu müssen Sie die Datei eventuell neu laden, wenn Sie z.B. mit einem EPS oder einem Datenbanksystem arbeiten; wählen Sie in diesem Fall die Option **Neu laden**.

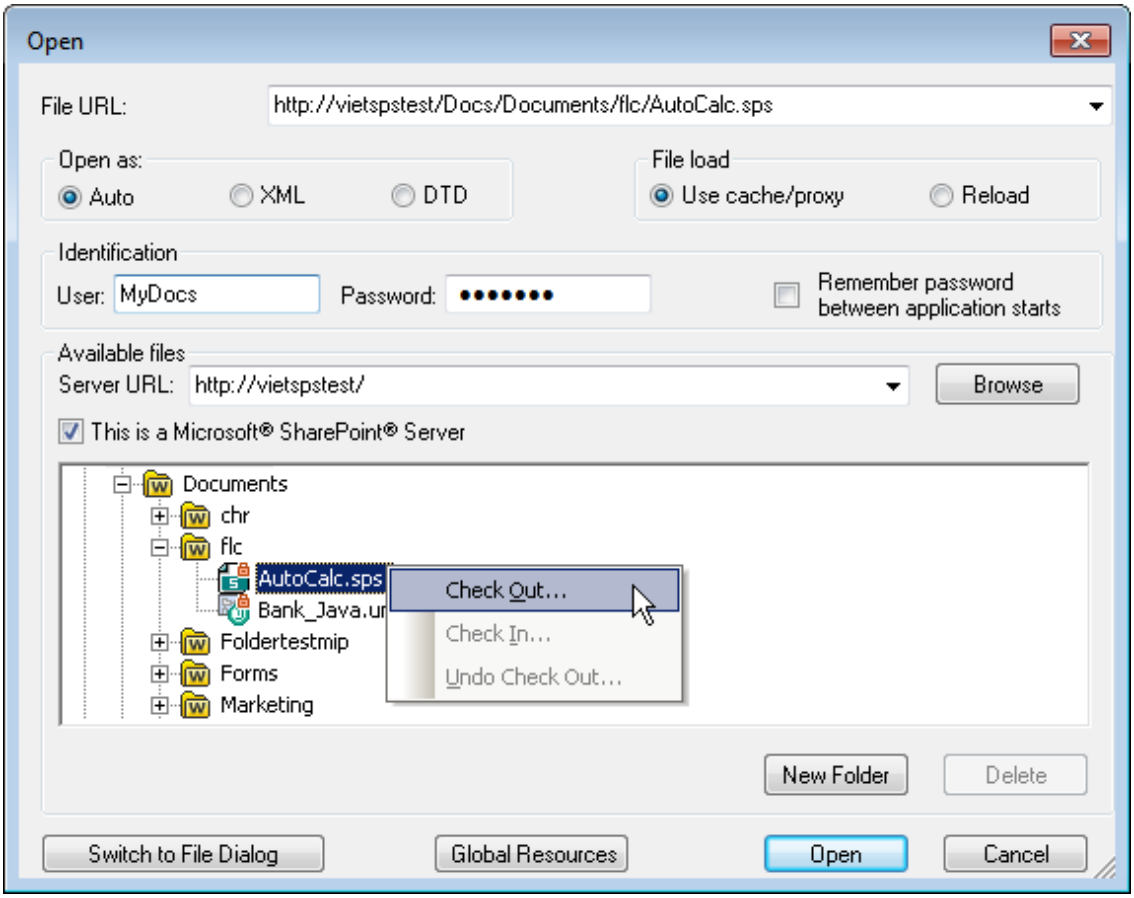
.

Anmerkungen zu Microsoft® SharePoint® Server

Beachten Sie bei Dateien auf Microsoft® SharePoint® Servern die folgenden Punkte:

 © 2016-2022 Altova GmbH

- In der Verzeichnisstruktur, die im Fenster der verfügbaren Dateien angezeigt wird (*Abbildung unten*), sind die Dateisymbole mit Symbolen versehen, die den Status der Datei (Eingecheckt/Ausgecheckt) kennzeichnen.



Wenn Sie mit der rechten Maustaste auf eine Datei klicken, erscheint ein Kontextmenü mit Befehlen, die für diese Datei zur Verfügung stehen (*Abbildung oben*).

- Im Folgenden sehen Sie die verschiedenen Dateisymbole:

	Eingecheckt. Die Datei kann ausgecheckt werden.
	Von einem anderen Benutzer ausgecheckt. Kann nicht ausgecheckt werden.
	Lokal ausgecheckt. Kann bearbeitet und eingecheckt werden.

- Nachdem Sie eine Datei ausgecheckt haben, können Sie sie in Ihrer Altova-Applikation bearbeiten und mit **Datei | Speichern (Strg+S)** speichern.
- Sie können die bearbeitete Datei über das Kontextmenü im Dialogfeld "URL öffnen" (*siehe Abbildung oben*) oder über das Kontextmenü, das angezeigt wird, wenn Sie mit der rechten Maustaste auf das Dateiregister im Hauptfenster Ihrer Applikation klicken (*Abbildung unten*), einchecken.

- Wenn eine Datei von einem anderen Benutzer ausgecheckt wurde, kann sie nicht auscheckt werden.
- Wenn eine Datei lokal von Ihnen ausgecheckt wurde, können Sie dies mit dem Kontextmenübefehl "Auschecken rückgängig" rückgängig machen. In diesem Fall wird die Datei dem Server unverändert zurückgegeben.
- Wenn Sie eine Datei in einer Altova-Applikation auschecken, können Sie sie nicht in einer anderen Altova-Applikation auschecken. Die Datei gilt bereits als für Sie ausgecheckt. Daher stehen in diesem Fall in jeder Altova-Applikation, die Microsoft® SharePoint® Server unterstützt, nur die Befehle **Einchecken** und **Auschecken rückgängig** zur Verfügung.

Öffnen und Speichern von Dateien über globale Ressourcen

Um eine Datei über eine globale Ressource zu öffnen oder zu speichern, klicken Sie auf **Globale Ressourcen**. Daraufhin wird ein Dialogfeld angezeigt, in dem Sie die globale Ressource auswählen können. Diese Dialogfelder werden im Abschnitt Verwenden globaler Ressourcen 109 näher beschrieben. Eine allgemeine Beschreibung globaler Ressourcen finden Sie in der Dokumentation im Abschnitt Globale Ressourcen 96.

13.1.3 Neu laden

Schaltfläche

Schaltfläche:	

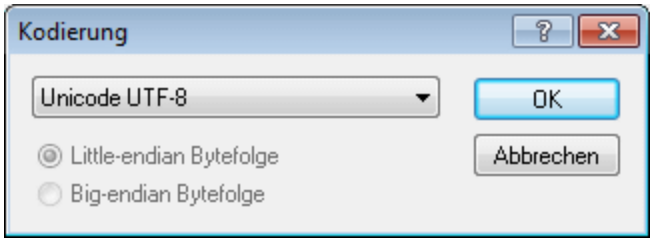
Beschreibung

Mit **Neu laden** können Sie geöffnete Dokumente, die außerhalb von Authentic Desktop geändert wurden, neu laden. Wenn ein oder mehrere Dokumente außerhalb von Authentic Desktop geändert wurden, werden Sie von Authentic Desktop gefragt, ob die Datei erneut geladen werden soll. Wenn Sie dies tun, gehen alle Änderungen, die Sie an der Datei seit der letzten Speicherung vorgenommen haben, verloren.

13.1.4 Kodierung

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

 © 2016-2022 Altova GmbH



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

Die Standardkodierung für vorhandene und neue XML- und nicht-XML-Dokumente kann im Dialogfeld "Optionen" im Abschnitt "Kodierung" 272 festgelegt werden.

Anmerkung: Beim Speichern eines Dokumentes prüft Authentic Desktop automatisch die Kodierungsspezifikation. Über das Dialogfeld "Kodierung" können Sie die gewünschte Kodierung auswählen. Sollte Ihr Dokument Zeichen enthalten, die in der gewählten Kodierung nicht angezeigt werden können, erhalten Sie eine Warnmeldung beim Speichern der Datei.

13.1.5 Schließen, Alle schließen, Alle inaktiven schließen

Schließen

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

Alle schließen

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

Alle inaktiven schließen

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

13.1.6 Speichern, Speichern unter, Alles speichern

Schaltflächen und Tastenkürzel

Befehl	*Schaltfläche*	*Tastenkürzel*
Speichern		**Strg+S**
Alles speichern		

Speichern

Mit dem Befehl **Speichern** (**Strg+S)** wird der Inhalt des aktiven Dokuments in der geöffneten Datei gespeichert. Beim Speichern wird das Dokument automatisch auf Wohlgeformtheit geprüft 219. Außerdem wird das Dokument auch automatisch validiert, wenn diese Option im Dialogfeld "Optionen" (**Extras | Optionen** 268) im Abschnitt "Datei" aktiviert wurde. Die XML Deklaration wird auch auf Kodierung 272sspezifikationen geprüft. Diese Kodierung wird beim Speichern des Dokuments verwendet.

Speichern unter

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

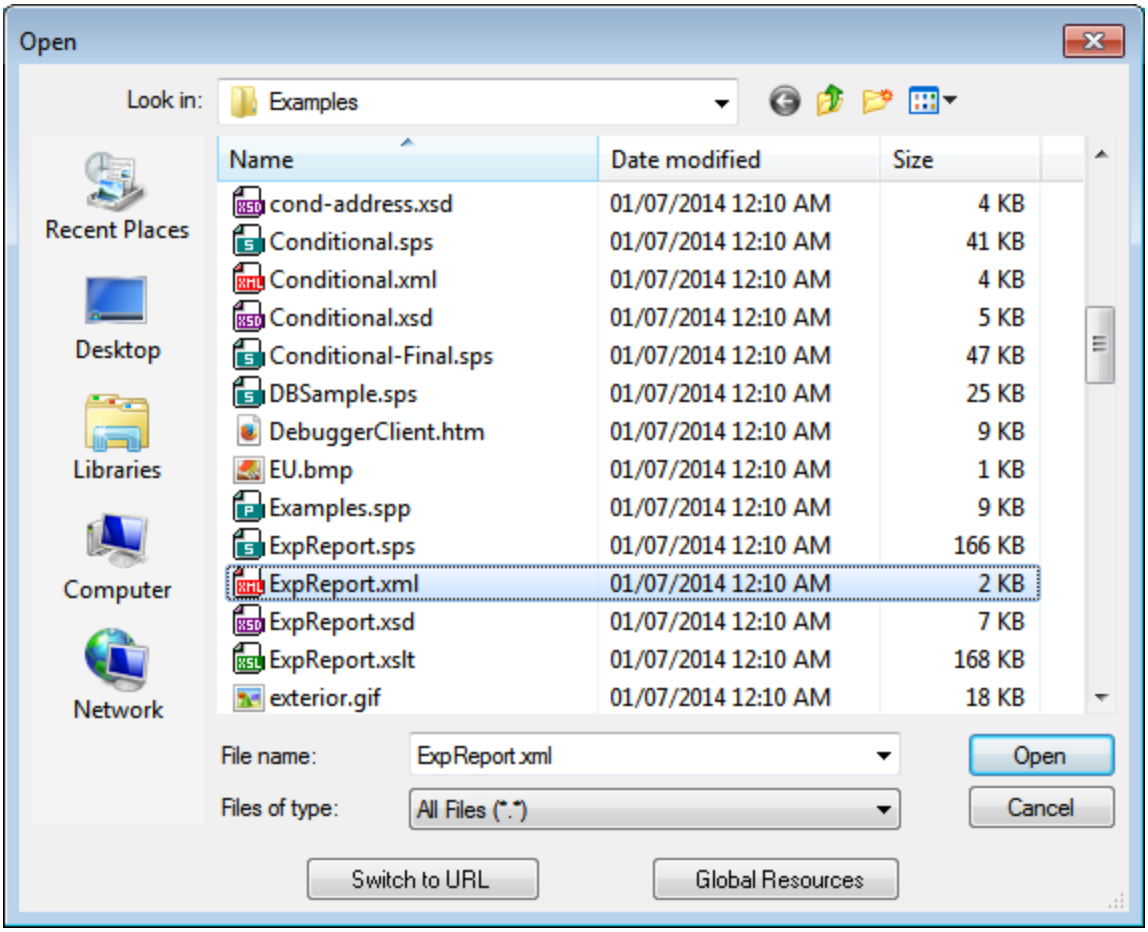
Alles speichern

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

- Auswählen und Speichern von Dateien über URLs und globale Ressourcen

  Es gibt mehrere Dialogfelder zum Öffnen und Speichern von Dateien, über die Sie die gewünschte Datei entweder über eine URL oder eine globale Ressource (*siehe Abbildung unten*) auswählen bzw. speichern können. Wählen Sie den Befehl **Zu URL wechseln** oder **Globale Ressourcen**, um das entsprechende Dialogfeld aufzurufen.

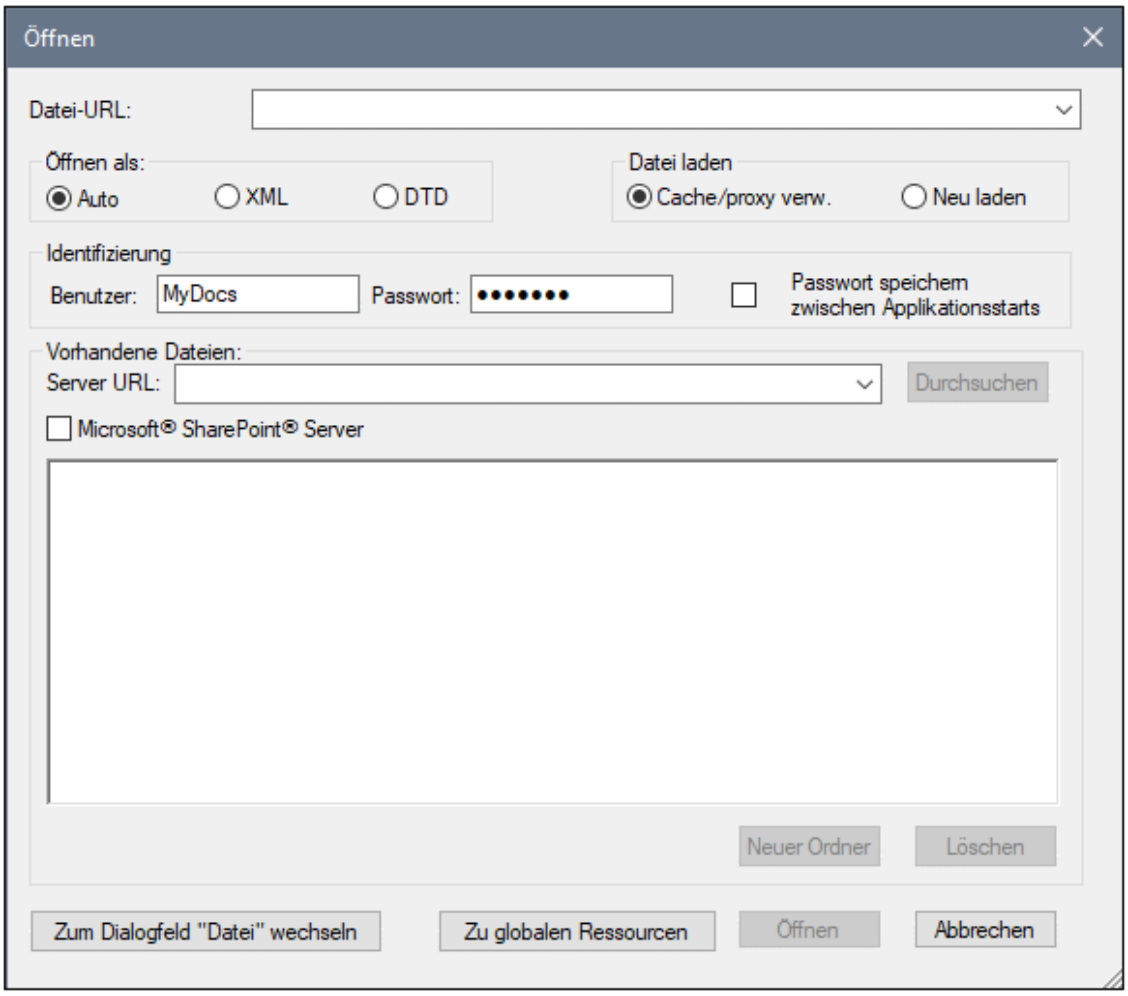
 © 2016-2022 Altova GmbH



Auswählen von Dateien über URLs

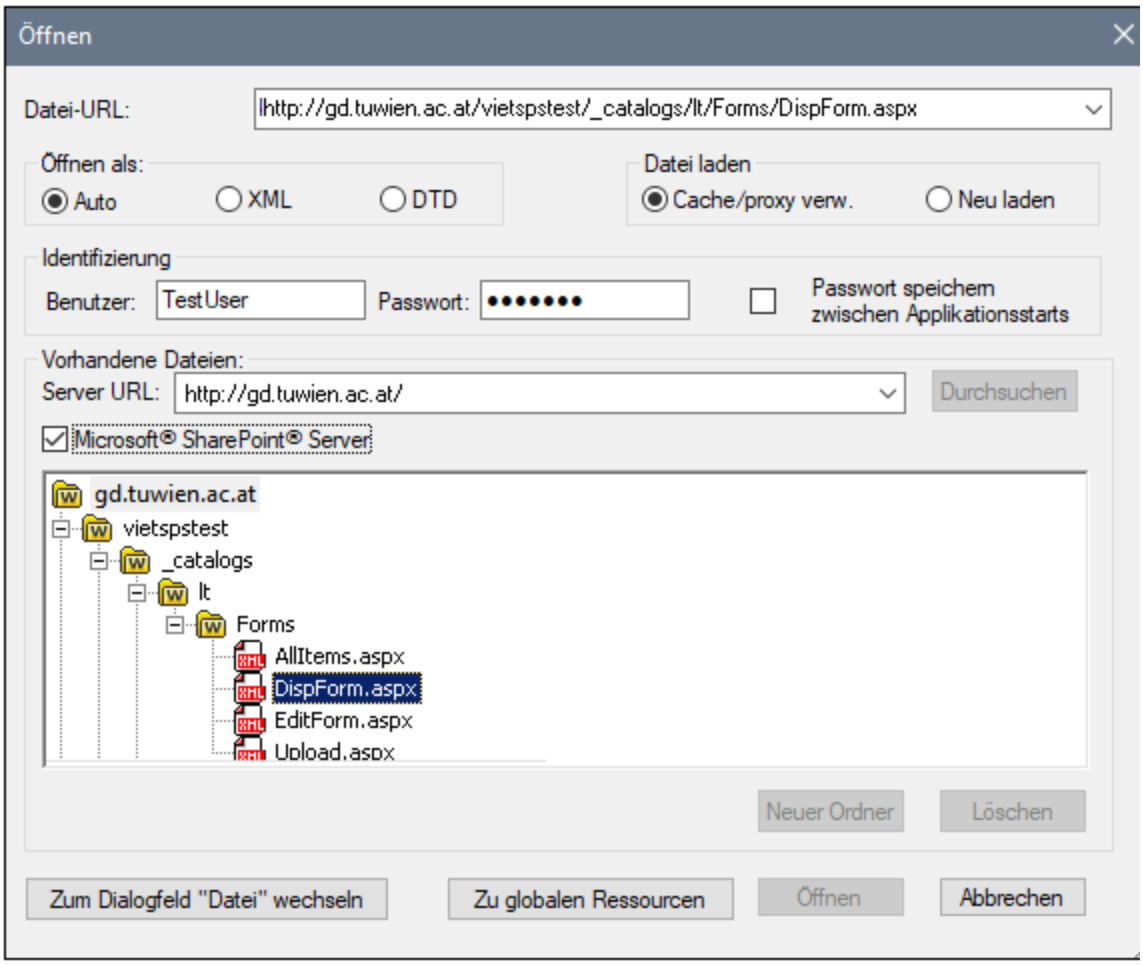
Um eine Datei über eine URL auszuwählen, gehen Sie folgendermaßen vor:

1. Klicken Sie auf den Befehl **Zu URL wechseln**. Daraufhin wechselt das Dialogfeld "Öffnen" bzw. "Speichern" in den URL-Modus (*In der Abbildung unten sehen Sie das Dialogfeld "Öffnen"*).



2. Geben Sie die gewünschte URL in das Feld *Server URL* ein (*Abbildung oben*). Wenn es sich beim Server um einen Microsoft® SharePoint® Server handelt, aktivieren Sie das Kontrollkästchen *Microsoft® SharePoint® Server.* Nähere Informationen zum Arbeiten mit Dateien dieses Servertyps finden Sie unter den Anmerkungen weiter unten zu Microsoft® SharePoint® Server.
3. Geben Sie Ihre Benutzer-ID und Ihr Passwort in die Felder *Benutzer* und *Passwort* ein, wenn der Server passwortgeschützt ist.
4. Klicken Sie auf **Durchsuchen**, um die Verzeichnisstruktur des Servers zu sehen und darin zu navigieren.
5. Suchen Sie in die Ordnerstruktur Sie die gewünschte Datei und klicken Sie darauf.

 © 2016-2022 Altova GmbH



Die Datei-URL wird im Feld "Datei-URL" angezeigt (*Abbildung oben*). Die Schaltflächen **Öffnen** oder **Speichern** werden erst zu diesem Zeitpunkt aktiv.

6. Klicken Sie auf **Öffnen,** um die Datei zu laden bzw. auf **Speichern**, um sie zu speichern.

Beachten Sie bitte die folgenden Punkte:

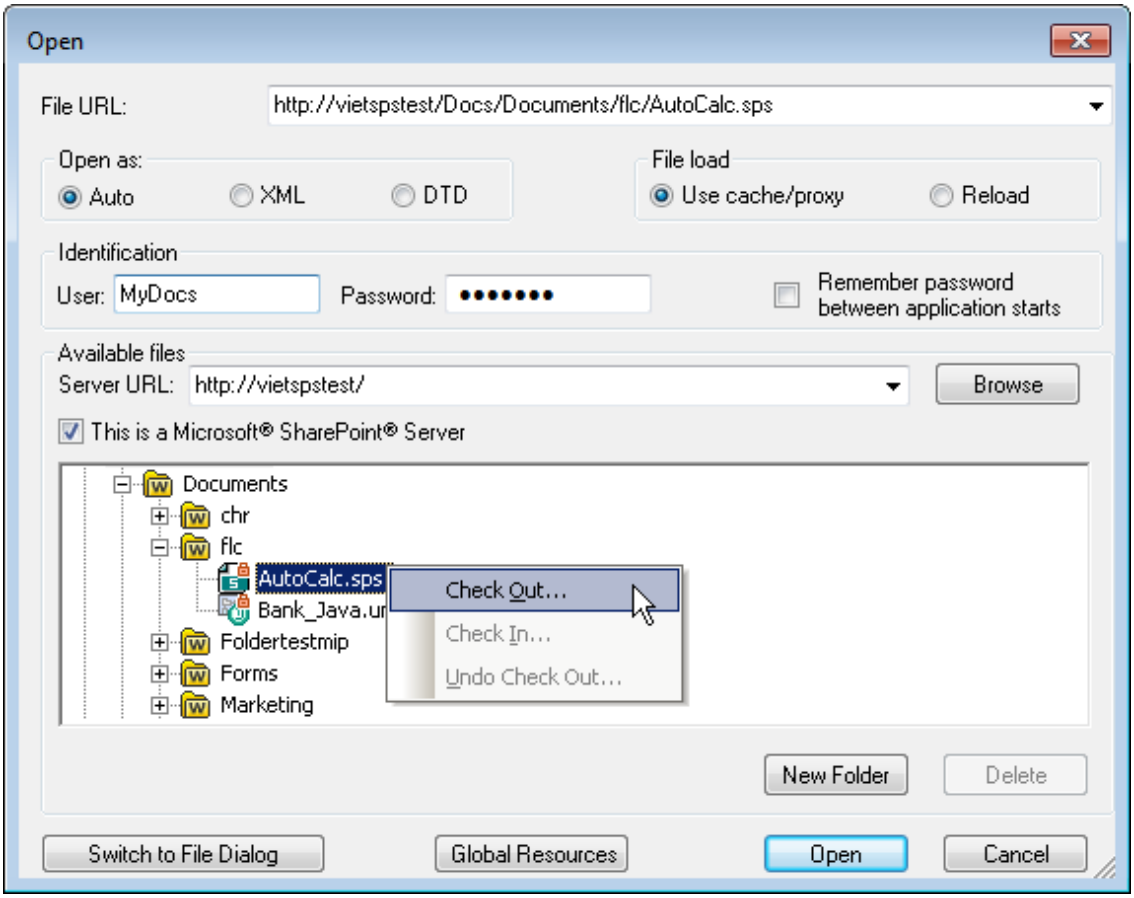
- Die Durchsuchen-Funktion steht nur auf Servern zur Verfügung, die WebDAV und Microsoft SharePoint Server unterstützen. Die unterstützten Protokolle sind FTP, HTTP und HTTPS.
- Damit Sie mehr Kontrolle über den Ladevorgang haben, können Sie auswählen, ob die Datei über den lokalen Cache-Speicher oder über einen Proxy-Server (wodurch der Ladevorgang beträchtlich schneller geht, wenn die Datei bereits einmal geladen war) geladen werden soll. Alternativ dazu müssen Sie die Datei eventuell neu laden, wenn Sie z.B. mit einem EPS oder einem Datenbanksystem arbeiten; wählen Sie in diesem Fall die Option **Neu laden**.

.

Anmerkungen zu Microsoft® SharePoint® Server

Beachten Sie bei Dateien auf Microsoft® SharePoint® Servern die folgenden Punkte:

- In der Verzeichnisstruktur, die im Fenster der verfügbaren Dateien angezeigt wird (*Abbildung unten*), sind die Dateisymbole mit Symbolen versehen, die den Status der Datei (Eingecheckt/Ausgecheckt) kennzeichnen.



Wenn Sie mit der rechten Maustaste auf eine Datei klicken, erscheint ein Kontextmenü mit Befehlen, die für diese Datei zur Verfügung stehen (*Abbildung oben*).

- Im Folgenden sehen Sie die verschiedenen Dateisymbole:

	Eingecheckt. Die Datei kann ausgecheckt werden.
	Von einem anderen Benutzer ausgecheckt. Kann nicht ausgecheckt werden.
	Lokal ausgecheckt. Kann bearbeitet und eingecheckt werden.

- Nachdem Sie eine Datei ausgecheckt haben, können Sie sie in Ihrer Altova-Applikation bearbeiten und mit **Datei | Speichern (Strg+S)** speichern.
- Sie können die bearbeitete Datei über das Kontextmenü im Dialogfeld "URL öffnen" (*siehe Abbildung oben*) oder über das Kontextmenü, das angezeigt wird, wenn Sie mit der rechten Maustaste auf das Dateiregister im Hauptfenster Ihrer Applikation klicken (*Abbildung unten*), einchecken.

 © 2016-2022 Altova GmbH

- Wenn eine Datei von einem anderen Benutzer ausgecheckt wurde, kann sie nicht auscheckt werden.
- Wenn eine Datei lokal von Ihnen ausgecheckt wurde, können Sie dies mit dem Kontextmenübefehl "Auschecken rückgängig" rückgängig machen. In diesem Fall wird die Datei dem Server unverändert zurückgegeben.
- Wenn Sie eine Datei in einer Altova-Applikation auschecken, können Sie sie nicht in einer anderen Altova-Applikation auschecken. Die Datei gilt bereits als für Sie ausgecheckt. Daher stehen in diesem Fall in jeder Altova-Applikation, die Microsoft® SharePoint® Server unterstützt, nur die Befehle **Einchecken** und **Auschecken rückgängig** zur Verfügung.

▼ Öffnen und Speichern von Dateien über globale Ressourcen

Um eine Datei über eine globale Ressource zu öffnen oder zu speichern, klicken Sie auf **Globale Ressourcen**. Daraufhin wird ein Dialogfeld angezeigt, in dem Sie die globale Ressource auswählen können. Diese Dialogfelder werden im Abschnitt Verwenden globaler Ressourcen 109 näher beschrieben. Eine allgemeine Beschreibung globaler Ressourcen finden Sie in der Dokumentation im Abschnitt Globale Ressourcen 96.

13.1.7 Als Mail senden

Symbol

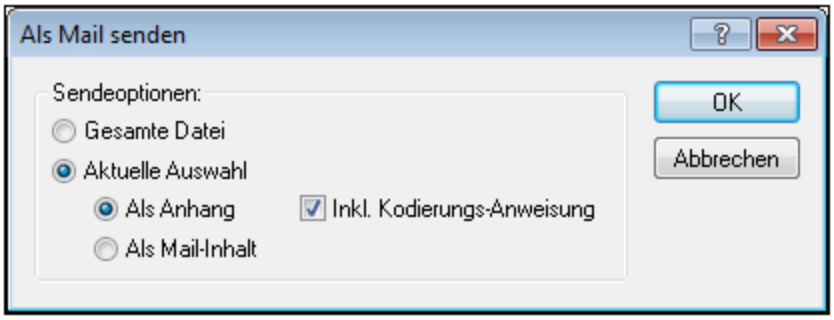
Schaltfläche:	

Beschreibung

Mit dem Befehl **Als Mail senden** können Sie die aktive XML-Datei oder das aktive PXF-Dokument als E-Mail-Anhang senden. Sie können im Fenster "Projekt" auch mehrere Dateien auswählen, um diese als E-Mail-Anhänge zu senden. Je nach Dokumentart kann ein Dokument oder eine Auswahl als Anhang, Inhalt oder Link gesendet werden. Nähere Informationen sehen Sie in der nachstehenden Tabelle.

Was kann gesendet werden	Wie kann es versendet werden
Aktives XML- oder PXF-Dokument	Als E-Mail-Anhang
Eine oder mehrere Dateien im Projektfenster	Als E-Mail-Anhang
Eine oder mehrere URLs im Projektfenster	Als E-Mail-Anhang oder Link

- Wenn der Befehl **Als Mail senden** auf einen ausgewählten Bereich im aktiven XML-Dokument angewendet wird, wird das Dialogfeld "Als Mail senden" (*Abbildung unten*) aufgerufen, wobei die Option *Gesamte Datei* die einzige aktivierte Option ist; die anderen Optionen sind deaktiviert. Klicken Sie auf **OK**, um eine E-Mail zu öffnen, an die die ausgewählte Datei angehängt wird.



- Wenn Sie Dateien über das Projektfenster versenden, wird eine E-Mail geöffnet, an die die ausgewählten Dateien in Form von Anhängen hinzugefügt werden.
- URLs im Fenster "Projekt" können als Anhang oder Verknüpfung (Link) versendet werden. Aktivieren Sie die gewünschten Optionen und klicken Sie auf **OK**.

13.1.8 Drucken

Schaltfläche und Tastenkürzel

Schaltfläche:	
Tastenkürzel:	**Strg+P**

Beschreibung

Der Befehl **Drucken** öffnet das Dialogfeld "Drucken". Hier können Sie Optionen zum Drucken des gerade aktiven Dokuments festlegen.

13.1.9 Druckvorschau, Druckereinrichtung

Druckvorschau

Der Befehl **Druckvorschau** steht in der Authentic-Ansicht zur Verfügung. Er öffnet eine Druckvorschau des aktuell aktiven Dokuments.

Im Modus "Druckvorschau" finden Sie in der Durckvorschau-Symbolleiste links oben im Vorschaufenster Druck- und Druckvorschauoptionen. Am unteren Rand des Vorschaufenster finden Sie Navigationsschaltflächen.

Anmerkung: Um Hintergrundfarben und Bilder in der Druckvorschau zu aktivieren, gehen Sie folgendermaßen vor: (i) Klicken Sie in Internet Explorer im Menü **Extras** auf **Internetoptionen** und

 © 2016-2022 Altova GmbH

anschließend auf das Register "Erweitert"; (ii) aktivieren Sie im Kästchen "Einstellungen" unter "Drucken" das Kontrollkästchen *Hintergrundfarben und -bilder drucken* und (iii) klicken Sie anschließend auf **OK**.

Druckereinrichtung

Mit dem Befehl **Druckereinrichtung** wird das druckerspezifische Dialogfeld "Druckereinrichtung" aufgerufen, in dem Sie Druckereinstellungen wie z.B. Papierformat und Seitenausrichtung definieren können. Diese Einstellungen werden auf alle nachfolgenden Druckaufträge angewendet.

13.1.10 Zuletzt verwendete Dateien, Beenden

Zuletzt verwendete Dateien

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

Beenden

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

13.2 Menü "Bearbeiten"

In diesem Menü befinden sich Befehle, die beim Editieren mit Authentic Desktop verwendet werden. Darunter befinden sich die Standardbefehle Rückgängig 184, Wiederherstellen 184, Ausschneiden 184, Kopieren 184, Löschen 184, Alles markieren 185, Suchen 185, Weitersuchen 185 und Ersetzen 186.

13.2.1 Rückgängig, Wiederherstellen

Schaltflächen und Tastenkürzel

Befehl	*Schaltfläche*	*Tastenkürzel*
Rückgängig		**Strg+Z**
Wiederherstellen		**Strg+Y**

Rückgängig

Der Befehl **Rückgängig** unterstützt unbegrenztes Rückgängigmachen aller Aktionen. Jede durchgeführte Aktion kann in Einzelschritten rückgängig gemacht werden. Der Verlauf der durchgeführten Aktionen wird beim Schließen des Dokuments gespeichert, sodass Aktionen auch zu einem späteren Zeitpunkt zurückverfolgt werden können. Mit Hilfe der Befehle **Rückgängig** und **Wiederherstellen** (*siehe Befehl **Wiederherstellen** unten*) können Sie sich Schritt für Schritt vorwärts und rückwärts durch den Verlauf bewegen.

Wiederherstellen

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

13.2.2 Ausschneiden, Kopieren, Einfügen, Löschen

Schaltflächen und Tastenkürzel

Befehl	*Schaltfläche*	*Tastenkürzel*
Ausschneiden		**Strg+X** oder **Umschalt+Entf**
Kopieren		**Strg+C**
Einfügen		**Strg+V**
Löschen		**Entf**

© 2016-2022 Altova GmbH

Ausschneiden

Mit dem Befehl **Ausschneiden** wird der markierte Text bzw. das markierte Objekt in die Zwischenablage kopiert und aus der aktuellen Position im Dokument gelöscht.

Kopieren

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

Einfügen

Mit dem Befehl **Einfügen** fügen Sie den Inhalt der Zwischenablage an der aktuellen Cursorposition ein.

Löschen

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

13.2.3 Alles markieren

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

13.2.4 Suchen, Weitersuchen...

Schaltflächen und Tastenkürzel

Befehl	*Schaltfläche*	*Tastenkürzel*
Suchen		**Strg+F**
Weitersuchen		**F3**

Suchen

Mit dem Befehl **Suchen** rufen Sie das Dialogfeld "Suchen" auf, in dem Sie den zu suchenden String sowie weitere Suchoptionen spezifizieren können. Um nach Text zu suchen, geben Sie diesen in das Suchfeld ein oder wählen Sie einen der letzten 10 Einträge aus der Dropdown-Liste aus und legen Sie anschließend die Suchoptionen fest.

Mit den Befehlen **Suchen** und **Weitersuchen** können Sie auch nach Datei- und Ordnernamen suchen, wenn ein Projekt im Fenster "Projekt" ausgewählt ist.

Weitersuchen

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

13.2.5 Ersetzen...

Schaltflächen und Tastenkürzel

Befehl	*Schaltfläche*	*Tastenkürzel*
Ersetzen		**Strg+H**

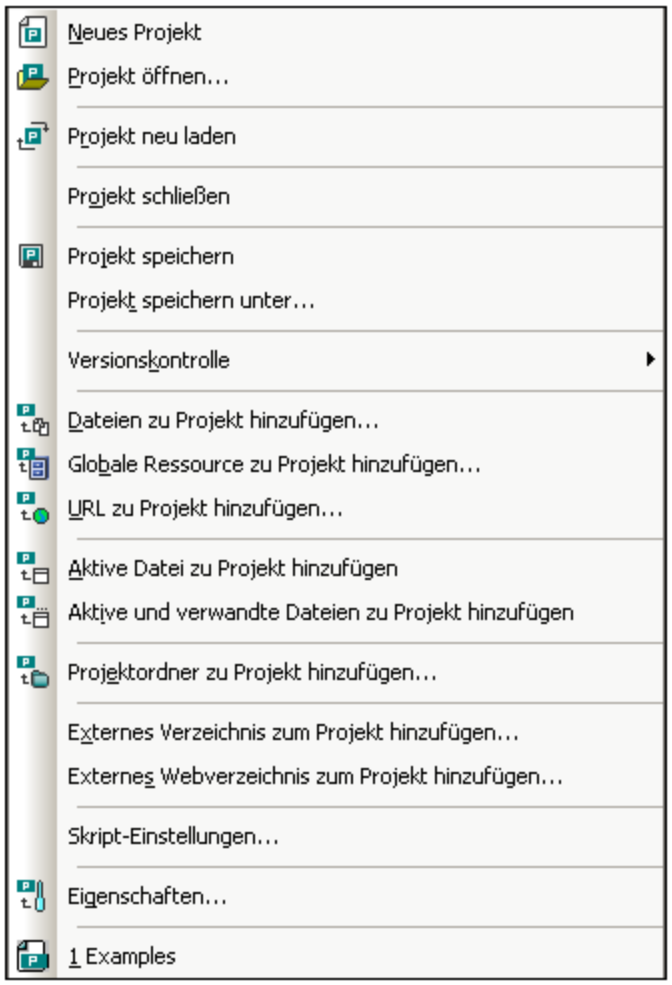
Beschreibung

Mit dem Befehl **Ersetzen** (in Visual Studio **Suchen und Ersetzen**) können Sie einen Textstring durch einen anderen ersetzen. Der Befehl verfügt über dieselben Optionen wie der Befehl **Suchen** 185, 185. Sie können diese Funktion Schritt für Schritt verwenden, oder alle Begriffe auf einmal ersetzen, indem Sie auf die Schaltfläche "**Alle ersetzen**" klicken.

 © 2016-2022 Altova GmbH

13.3 Menü "Projekt"

Authentic Desktop verwendet die bekannte Baumstruktur, um Dateien und URLs in einem XML-Projekt zu gliedern. Dateien[206] und URLs[207] mit gleicher Erweiterung können in Ordnern[208] gesammelt werden, wodurch eine einfache Strukturierung und Batchmanipulation gegeben ist.



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

Absolute und relative Pfade

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

```
|-Folder1
```

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

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

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

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

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

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

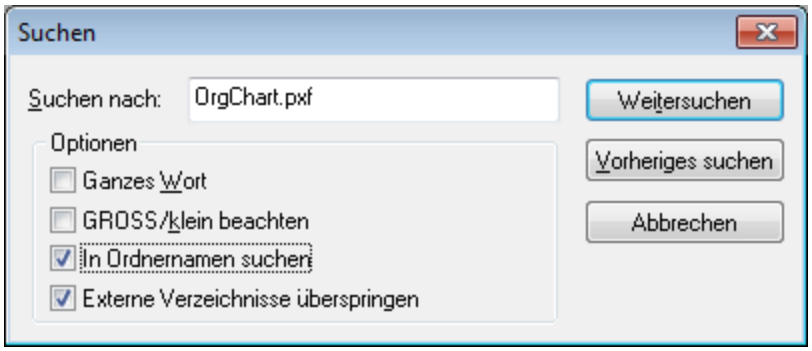
Drag-and-Drop

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

In Projekt suchen

Sie können Projektdateien und -ordner nach ihrem Namen bzw. einem Teil ihres Namens suchen. Wurden Treffer gefunden, so werden die gefundenen Dateien bzw. Ordner der Reihe nach markiert.

Um eine Suche zu starten, wählen Sie den zu durchsuchenden Projektordner in der Seitenleiste "Projekt" aus und klicken Sie auf den Befehl **Bearbeiten | Suchen** (oder das Tastenkürzel **Strg+F**). Geben Sie nun im Dialogfeld "Suchen" (*Abbildung unten*) den gewünschten Such-String ein und aktivieren bzw. deaktivieren Sie die Suchoptionen nach Bedarf (*Erklärung weiter unten*)



Es stehen die folgenden Suchoptionen zur Verfügung:

 © 2016-2022 Altova GmbH

- Ganzes Wort: Hierbei ist die Suche stärker eingeschränkt, da der Suchstring mit einem ganzen Wort im Datei- bzw. Ordnernamen übereinstimmen muss. Die Teile vor und nach dem Punkt in den Dateinamen werden (ohne Berücksichtigung des Punkts) jeweils als Wort behandelt.
- Sie können festlegen, dass die Groß- und Kleinschreibung im Suchstring genau mit dem Textstring im Datei- oder Ordnernamen übereinstimmen muss.
- Ordnernamen können in die Suche einbezogen werden. Andernfalls werden nur Dateinamen durchsucht.
- Externe Ordner 208 können in die Suche einbezogen oder davon ausgeschlossen werden. Bei externen Ordnern handelt es sich um tatsächliche Ordner auf Ihrem Rechner bzw. Netzwerk und nicht um Projektordner, die innerhalb des Projekts und nicht auf dem Rechner angelegt wurden.

Wurden Treffer gefunden, wird der erste Treffer in der Seitenleiste "Projekt" markiert. Sie können anschließend im Dialogfeld **Suchen** mit Hilfe von **Weitersuchen** und **Vorh. suchen** der Reihe nach zu allen gefundenen Suchtreffern springen.

Aktualisieren von Projekten

Wenn eine Änderung an einem externen Ordner vorgenommen wird, wird diese Änderung erst im Projektfenster angezeigt, wenn das Projekt aktualisiert wurde.

Globale Ressourcen im Kontextmenü

Wenn Sie mit der rechten Maustaste im Projektfenster auf einen Ordner klicken, können Sie im daraufhin angezeigten Kontextmenü den Befehl **Globale Ressource hinzufügen** auswählen, um eine globale Ressource 96 hinzuzufügen. Daraufhin wird das Dialogfeld "Globale Ressource wählen" angezeigt, in dem alle globalen Ressourcen vom Typ Datei und Ordner in der derzeit aktiven XML-Datei für globale Ressourcen aufgelistet sind. Wählen Sie die gewünschte globale Ressource und sie wird zum ausgewählten Projektordner hinzugefügt.

Projekte und Versionskontrolle

Wenn Sie vorhaben, ein Authentic Desktop-Projekt zu einem Versionskontroll-Repository hinzuzufügen, stellen Sie bitte sicher, dass die Projektdateien in der hierarchischen Dateisystemstruktur in einem Ordner abgelegt werden, von dem aus Dateien nur aus diesem untergeordneten Ordnern hinzugefügt werden können (Nehmen Sie das Root-Verzeichnis als oberste Ebene der Verzeichnisstruktur)

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

Nehmen wir als Beispiel die oben gezeigte Verzeichnisstruktur:
Wenn eine Projektdatei in `Folder3` gespeichert wird und einer Versionskontrolle unterliegt:

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

13.3.1 Neues Projekt

Mit dem Befehl **Neues Projekt** erstellen Sie ein **neues** Projekt in Authentic Desktop. Wenn Sie gerade an einem anderen Projekt arbeiten, werden Sie gefragt, ob Sie alle Dokumente schließen wollen, die zu dem aktuellen Projekt gehören. Der Name des Projekts wird beim Speichern des Projekts als `.spp`-Datei zugewiesen.

13.3.2 Projekt öffnen.

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

13.3.3 Projekt neu laden

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

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

13.3.4 Projekt schließen

Mit diesem Befehl schließen Sie das **aktuelle** Projekt. Wurde das Projekt geändert, werden Sie vor dem Schließen gefragt, ob die Änderungen gespeichert werden sollen.
Wenn ein Projekt oder eine XML-Datei geändert wurde, wird der Name der Datei im Projektfenster mit einem Stern "*" versehen.

 © 2016-2022 Altova GmbH

13.3.5 Projekt speichern, Projekt speichern unter

Mit dem Befehl **Projekt speichern** wird das aktuelle Projekt **gespeichert**. Sie können ein Projekt auch speichern, indem Sie das Projektfenster zum aktiven machen und auf das Symbol klicken.
Mit dem Befehl **Projekt speichern unter** wird das aktuelle Projekt mit einem neuen Namen **gespeichert**, den Sie eingeben können, wenn Sie vom Programm dazu aufgefordert werden.

13.3.6 Versionskontrolle

Ihre Altova-Applikation unterstützt Microsoft Source-Safe und andere kompatible Speicherschnittstellen. Die von Altova-Applikationen unterstützten Versionskontrollsysteme sind im Abschnitt Unterstützte Versionskontrollsysteme 116 aufgelistet. In diesem Abschnitt werden die Befehle aus dem Untermenü **Projekt | Versionskontrolle** beschrieben, mit denen Sie von Ihrer Altova-Applikation aus auf Ihr Versionskontrollsystem zugreifen können.

Übersicht über die Funktion "Versionskontrolle"

Um Dateien in einem Applikationsprojekt unter Versionskontrolle zu stellen, gehen Sie folgendermaßen vor:

1. In Authentic Desktop wird ein Projektordner mit den unter Versionskontrolle zu stellenden Dateien erstellt. Normalerweise entspricht der Projektordner einem lokalen Ordner, in dem sich die Projektdateien befinden. Der Pfad zum lokalen Ordner wird als lokaler Pfad referenziert.
2. In der Datenbank des Versionskontrollsystems (auch als Versionskontrolle oder Repository bezeichnet) wird ein Ordner für die unter Versionskontrolle zu stellenden Dateien erstellt.
3. Die Projektdateien werden mit Hilfe des Befehls **Projekt | Versionskontrolle | Zu Versionskontrolle hinzufügen** 198 zur Versionskontrolle hinzugefügt.
4. Versionskontrollaktionen wie z.B. Einchecken, Auschecken und Entfernen von Dateien aus der Versionskontrolle können mit Hilfe der Befehle im **Projekt | Versionskontrolle** 191 Untermenü durchgeführt werden. Die Befehle in diesem Untermenü sind in den Unterabschnitten dieses Abschnitts beschrieben.

Anmerkung: Der aktuelle Versionskontroll-Provider kann auf zwei Arten gewechselt werden: (i) über die Versionskontrolloptionen (**Extras | Optionen | Versionskontrolle** 280) oder (ii) im Dialogfeld "Versionskontrolle wechseln" (**Projekt | Versionskontrolle | Versionskontrolle wechseln** 205).

Anmerkung: Beachten Sie, dass ein Versionskontrollprojekt nicht dasselbe wie ein Applikationsprojekt ist. Versionskontrollprojekte sind verzeichnisabhängig, während Authentic Desktop-Projekte logische Konstruktionen ohne direkte Verzeichnisabhängigkeit sind.

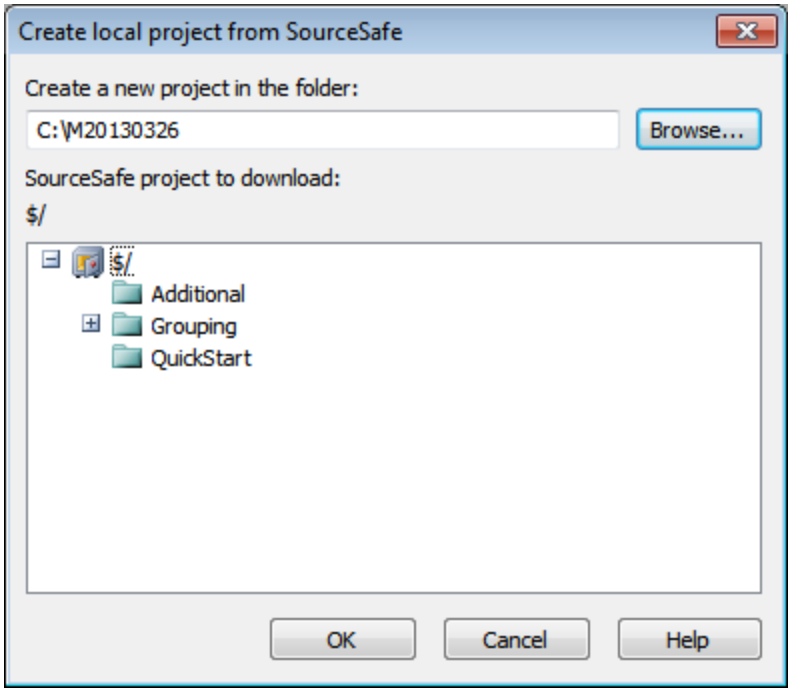
Nähere Informationen finden Sie im Abschnitt Versionskontrolle 113.

13.3.6.1 Aus Versionskontrolle öffnen

Mit dem Befehl **Aus Versionskontrolle öffnen** erstellen Sie ein neues Applikationsprojekt von einem unter Versionskontrolle gestellten Projekt.

Um ein neues Projekt zu erstellen, gehen Sie folgendermaßen vor:

1. Je nach verwendetem Versionskontrollsystem müssen Sie vor dem Erstellen eines neuen Projekts über die Versionskontrolle sicherstellen, dass keine Datei aus dem Projekt ausgecheckt ist.
2. In der Applikation kann ein Projekt geöffnet sein, dies muss aber nicht der Fall sein.
3. Wählen Sie den Befehl "Projekt | Versionskontrolle | Aus Versionskontrolle öffnen".
4. Das aktuell eingestellte Versionskontrollsystem wird mit den jeweiligen Überprüfungs- und Verbindungsdialogfeldern angezeigt. Stellen Sie eine Verbindung zum gewünschten Repository, d.h. zum gebundenen Ordner im Repository, der dem lokalen Ordner entspricht, her.
5. Navigieren Sie im Dialogfeld, das daraufhin angezeigt wird (*Abbildung unten*), zum lokalen Ordner, in den der Inhalt des gebundenen Ordners im Repository (zu dem Sie soeben eine Verbindung hergestellt haben) kopiert werden soll. In der Abbildung unten hat der gebundene Ordner den Namen `MyProject` und wird durch das Zeichen `$` repräsentiert; der lokale Ordner ist `C:\M20130326`.



6. Klicken Sie auf **OK**. Der Inhalt des gebundenen Ordners (`MyProject`) wird in den lokalen Ordner `C:\M20130326`. kopiert und ein Dialogfeld wird aufgerufen, in dem Sie aufgefordert werden, die Projektdatei (`.spp` Datei) auszuwählen, die als das neue Projekt erstellt werden soll.
7. Wählen Sie die `.spp` Datei aus, die in den lokalen Ordner kopiert worden sein sollte. In unserem Beispiel handelt es sich um `MyProject.spp` im Ordner `C:\M20130326`. In der Applikation wird ein neues Projekt namens `MyProject` erstellt und im Fenster "Projekt" angezeigt. Die Dateien des Projekts werden im Ordner `C:\M20130326` gespeichert.

 © 2016-2022 Altova GmbH

Symbole der Versionskontrolle:

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

	Eingecheckt. Die Datei kann ausgecheckt werden.
	Die Datei wurde von einem anderen Benutzer ausgecheckt und kann nicht ausgecheckt werden.
	Lokal ausgecheckt. Die Datei kann bearbeitet und eingecheckt werden.

13.3.6.2 Versionskontrolle aktivieren

Mit Hilfe dieses Befehls können Sie die Versionskontrolle für ein Applikationsprojekt aktivieren oder deaktivieren. Bei Auswahl dieser Option für eine Datei oder einen Ordner, wird die Versionskontrolle für das gesamte Projekt aktiviert/deaktiviert. Nach Aktivierung der Versionskontrolle wird der Ein- und Auscheckstatus der verschiedenen Dateien im Projektfenster angezeigt.

	Eingecheckt. Die Datei kann ausgecheckt werden.
	Die Datei wurde von einem anderen Benutzer ausgecheckt und kann nicht ausgecheckt werden.
	Lokal ausgecheckt. Die Datei kann bearbeitet und eingecheckt werden.

13.3.6.3 Aktuellste Version holen

Mit dem Befehl "Aktuellste Version holen" (Menü **Projekt | Versionskontrolle**) wird die aktuellste Version der ausgewählten Datei(en) aus dem Versionskontrollspeicher in das Arbeitsverzeichnis geholt. Die Dateien werden als schreibgeschützte Dateien und nicht ausgecheckt abgerufen. Dieser Befehl funktioniert wie der **Befehl Abrufen** 194, aber das Dialogfeld "Abrufen" wird nicht angezeigt.

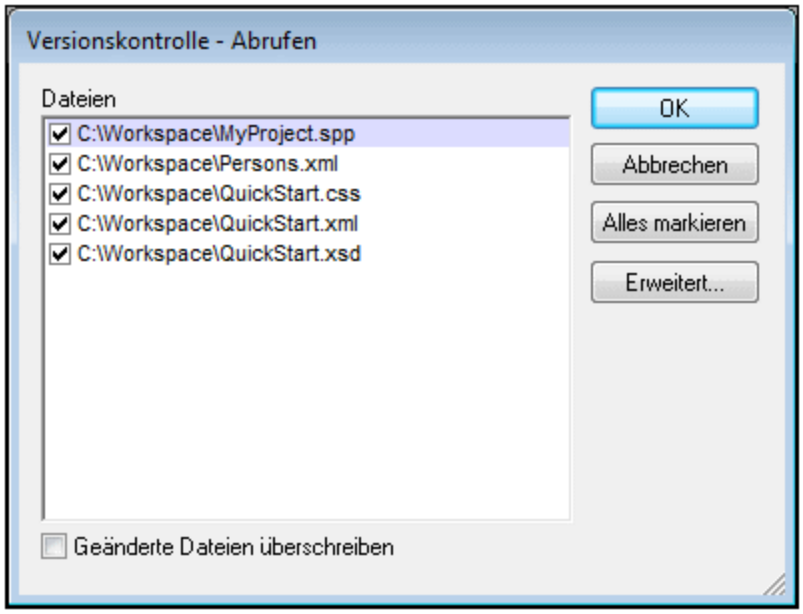
Wenn die ausgewählten Dateien derzeit ausgecheckt sind, hängt es von Ihrem Versionskontrollsystem ab, was geschieht. Normalerweise werden Sie gefragt, ob die ausgecheckte Datei ersetzt, zusammengeführt oder so belassen werden soll.

Anmerkung: Dieser Befehl ist bei Ausführung an einem Ordner rekursiv, d.h. er wird auf alle Dateien unterhalb des aktuellen Ordners in der Ordnerhierarchie angewendet.

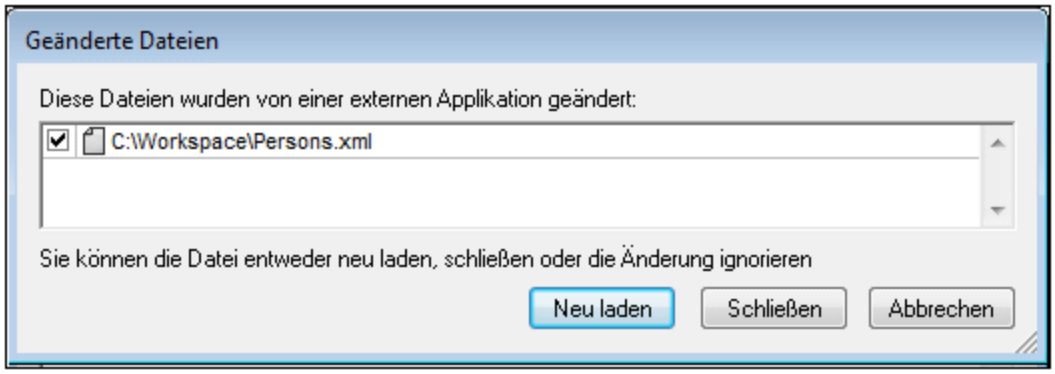
13.3.6.4 Abrufen, Ordner abrufen

Mit dem Befehl **Abrufen** (im Menü **Projekt | Versionskontrolle**) werden Dateien aus dem Repository als schreibgeschützte Dateien abgerufen. (Um eine Datei bearbeiten zu können, müssen Sie sie auschecken.) Im Dialogfeld "Abrufen" werden die Dateien im Objekt (Projekt oder Ordner), an denen der Befehl "Abrufen" ausgeführt wurde, aufgelistet (*siehe Abbildung unten*). Sie können die abzurufenden Dateien markieren, indem Sie sie mit einem Häkchen versehen.

Anmerkung: Mit dem Befehl **Ordner abrufen** können Sie einzelne Unterordner im Repository auswählen, falls dies Ihr Versionskontrollsystem gestattet.



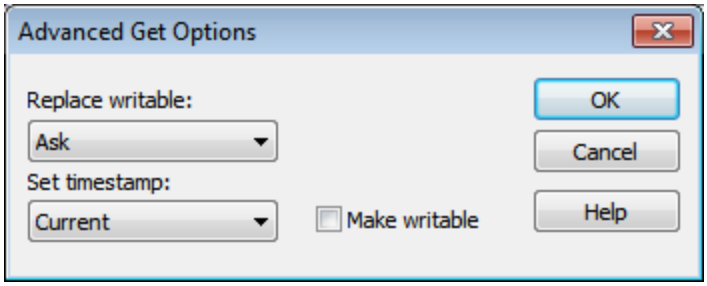
Sie können auswählen, ob geänderte ausgecheckte Dateien überschrieben werden sollen, indem Sie die entsprechende Option am unteren Rand des Dialogfelds "Abrufen" aktivieren. Wenn Sie auf **OK** klicken, werden die Dateien überschrieben. Wenn eine der überschriebenen Dateien gerade geöffnet ist, wird ein Dialogfeld angezeigt (*Abbildung unten*), in dem Sie gefragt werden, ob die Datei(en) neu geladen (Schaltfläche **Neu laden**), geschlossen (Schaltfläche **Schließen**) werden sollen oder ob die aktuelle Ansicht der Datei beibehalten werden soll (**Abbrechen**).



 © 2016-2022 Altova GmbH

Erweiterte Abrufoptionen

Das Dialogfeld "Erweiterte Abrufoptionen" (*Abbildung unten*) wird über die Schaltfläche **Erweitert** im Dialogfeld "Abrufen" (*siehe erste Abbildung in diesem Abschnitt*) aufgerufen.



Hier können Sie Optionen zum (i) Ersetzen ausgecheckter beschreibbarer Dateien (ii) zum Zeitstempel und (iii) ob die Eigenschaft "Schreibgeschützt" der abgerufenen Datei geändert werden soll, damit die Datei beschreibbar wird, einstellen.

13.3.6.5 Auschecken, Einchecken

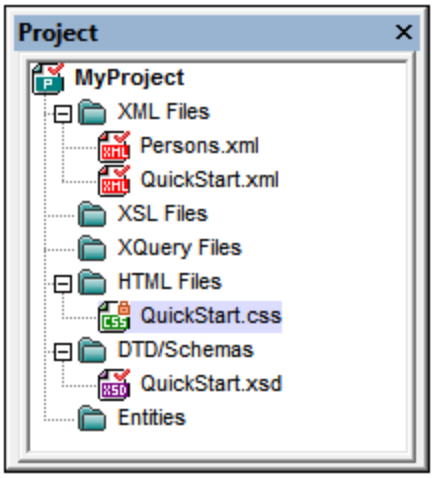
Nachdem Sie eine Projektdatei unter Versionskontrolle gestellt haben, kann sie durch Auswahl der Datei (im Projektfenster) und Klicken auf den entsprechenden Befehl (**Einchecken/Auschecken**) im Menü **Projekt | Versionskontrolle** ein- und ausgecheckt werden.

Wenn eine Datei ausgecheckt wird, wird eine Kopie aus dem Repository in den lokalen Ordner platziert. Eine ausgecheckte Datei kann bearbeitet werden. Wenn eine Datei, die sich unter Versionskontrolle befindet, nicht ausgecheckt ist, kann sie nicht bearbeitet werden. Nachdem eine Datei bearbeitet wurde, können die Änderungen durch Einchecken der Datei im Repository gespeichert werden. Selbst wenn die Datei nicht gespeichert wurde, werden die Änderungen an der Datei durch Einchecken im Repository gespeichert. Der Ein- und Auscheckstatus einer Datei wird durch ein Häkchen bzw. ein Schloßsymbol im Dateisymbol angezeigt.

Zu Dateien und dem Projektordner können bestimmte Symbole angezeigt werden, deren Bedeutung unten erklärt ist.

	Eingecheckt. Für das Auschecken verfügbar.
	Von einem anderen Benutzer ausgecheckt. Kann nicht ausgecheckt werden.
	Lokal ausgecheckt. Kann bearbeitet und eingecheckt werden.

Durch Auswahl des Projekts oder eines Ordners im Projekt werden alle Dateien im ausgewählten Objekt ausgewählt. Um mehrere Objekte (Dateien und Ordner) auszuwählen, halten Sie beim Anklicken der Objekte die Strg-Taste gedrückt. In der Abbildung unten sehen Sie ein ausgechecktes Projekt. Die Datei `QuickStart.css` wurde im Anschluss daran eingecheckt.



Speichern und Ablehnen von Bearbeitungen

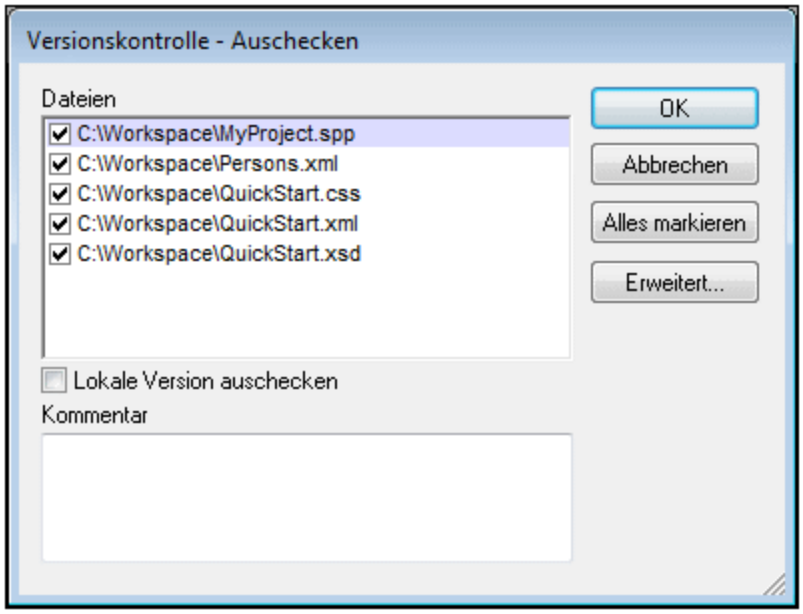
Beachten Sie, dass Sie eine Datei beim Einchecken ausgecheckt lassen können. Dabei werden die Änderungen im Repository gespeichert, während die Datei weiterhin ausgecheckt bleibt. Diese Option ist nützlich, wenn Sie Ihre Änderungen regelmäßig im Repository speichern und mit der Bearbeitung fortfahren möchten.

Wenn Sie eine Datei ausgecheckt haben, diese bearbeitet haben und die Änderungen anschließend aber nicht übernehmen möchten, können Sie die im Repository gespeicherte Dokumentversion in Ihrem lokalen Ordner wiederherstellen, indem Sie den Befehl **Projekt | Versionskontrolle | Auschecken rückgängig machen** auswählen.

Auschecken

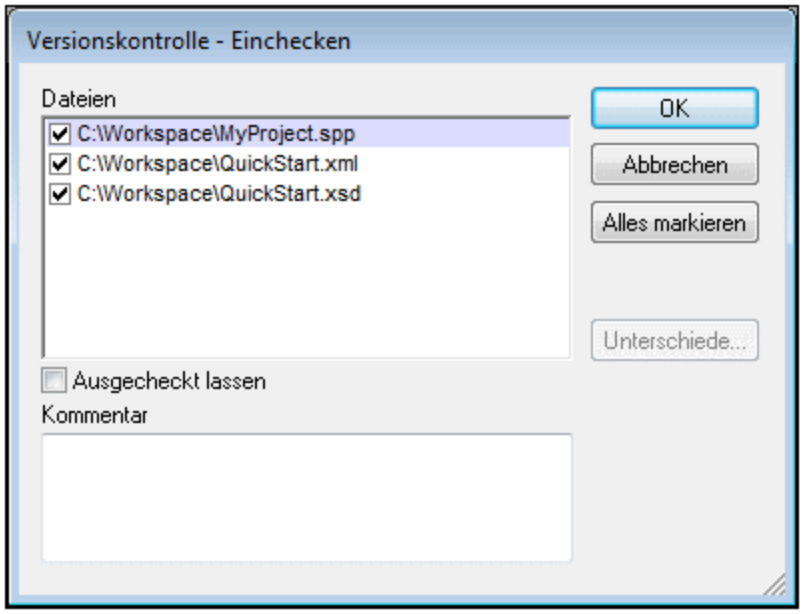
Im Dialogfeld "Auschecken" (*Abbildung unten*) können Sie (i) die auszucheckenden Dateien auswählen und (ii) wählen, ob die Repository-Version oder die lokale Version ausgecheckt werden soll.

 © 2016-2022 Altova GmbH



Einchecken

Im Dialogfeld "Einchecken" (*Abbildung unten*) können Sie (i) die einzucheckenden Dateien auswählen und (ii) wählen, ob die Datei ausgecheckt bleiben soll.



Anmerkung: In beiden Dialogfeldern (Einchecken und Auschecken) werden mehrere Dateien angezeigt, wenn das ausgewählte Objekt (Projekt oder Projektordner) mehrere Dateien enthält.

13.3.6.6 Auschecken rückgängig...

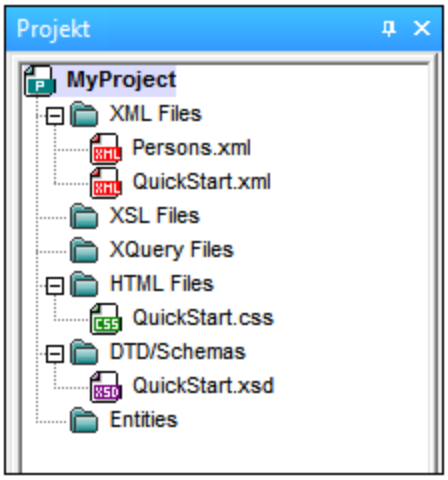
Wenn Sie eine Datei ausgecheckt und bearbeitet haben und diese Änderung nicht beibehalten möchten, können Sie die im Repository gespeicherte Dokumentversion wiederherstellen. Wählen Sie dazu den Befehl **Projekt | Versionskontrolle | Auchecken rückgängig...**.

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

	Eingecheckt. Die Datei kann ausgecheckt werden.
	Die Datei wurde von einem anderen Benutzer ausgecheckt und kann nicht ausgecheckt werden.
	Lokal ausgecheckt. Die Datei kann bearbeitet und eingecheckt werden.

13.3.6.7 Zu Versionskontrolle hinzufügen

Nachdem ein Projekt zu einer Versionskontrolle hinzugefügt wurde, können Sie Dateien entweder einzeln oder in Gruppen zur Versionskontrolle hinzufügen. Wählen Sie die Datei im Fenster "Projekt" aus und klicken Sie auf den Befehl **Projekt | Versionskontrolle | Zu Versionskontrolle hinzufügen**. Um mehrere Dateien auszuwählen, halten Sie beim Anklicken der Dateien die **Strg**-Taste gedrückt. Wenn Sie den Befehl an einem (grünen) Projektordner (*siehe Abbildung unten*) ausführen, werden alle Dateien im Ordner und seinen Unterordnern zur Versionskontrolle hinzugefügt.



Wenn Dateien zur Versionskontrolle hinzugefügt werden, wird die lokale Ordnerhierarchie im Repository repliziert (jedoch nicht die Projektordnerhierarchie). Wenn eine Datei sich also im lokalen Ordner in einem Unterordner mehrere Ebenen unterhalb des lokalen Ordners befindet, so werden im Repository automatisch auch der der Datei übergeordnete Ordner und alle darüber liegenden erstellt.

Wenn die erste Datei aus einem Projekt zur Versionskontrolle hinzugefügt wird, werden im Repository die korrekten Bindings erstellt und die Projektdatei (`.spp` Datei) wird automatisch hinzugefügt. Nähere Informationen dazu finden Sie im Abschnitt Zur Versionskontrolle hinzufügen 121.

 © 2016-2022 Altova GmbH

Versionskontrollsymbole

Zu Dateien und dem Projektordner können bestimmte Symbole angezeigt werden, deren Bedeutung unten erklärt ist.

	Eingecheckt. Für das Auschecken verfügbar.
	Von einem anderen Benutzer ausgecheckt. Kann nicht ausgecheckt werden.
	Lokal ausgecheckt. Kann bearbeitet und eingecheckt werden.

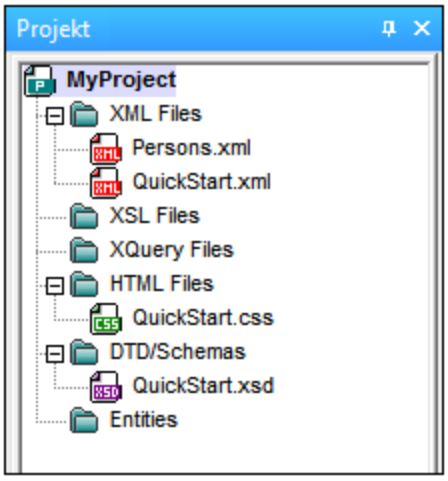
13.3.6.8 Von Versionskontrolle ausgliedern

Um eine Datei aus der Versionskontrolle zu entfernen, wählen Sie sie aus und klicken Sie auf den Befehl **Projekt | Versionskontrolle | Von Versionskontrolle ausgliedern**. Außerdem können Sie: (i) Dateien in einem Projektordner durch Ausführen des Befehls am Ordner entfernen, (ii) durch Drücken der **Strg**-Taste mehrere Dateien auswählen und entfernen und (iii) das gesamte Projekt durch Ausführen des Befehls am Projekt entfernen.

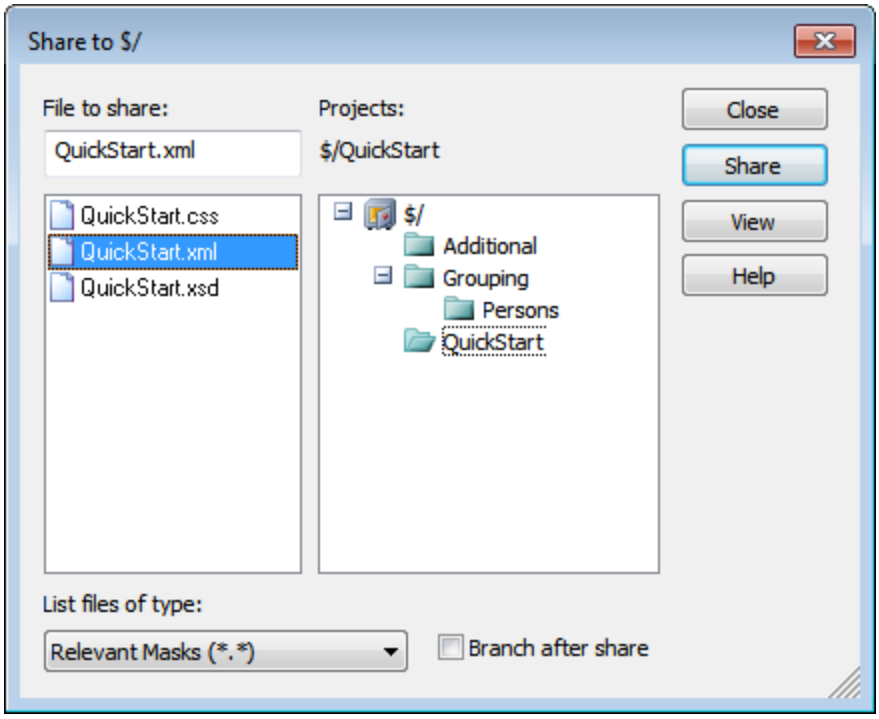
13.3.6.9 Aus Versionskontrolle freigeben

Der Befehl **Aus Versionskontrolle freigeben** wird unterstützt, wenn das verwendete Versionskontrollsystem die Freigabe von Dateien unterstützt. Sie können eine Datei freigeben, sodass sie auf mehreren lokalen Rechnern verfügbar ist. Wenn an einer dieser Dateien eine Änderung vorgenommen wird, wird dies in allen anderen freigegebenen Versionen übernommen.

Wählen Sie das (*in der Abbildung unten markierte)* Projekt im Fenster "Projekt" der Applikation aus und klicken Sie anschließend auf den Befehl **Aus Versionskontrolle freigeben**.



Daraufhin wird das Dialogfeld "Share To [Ordner]" (*Abbildung unten*) angezeigt.

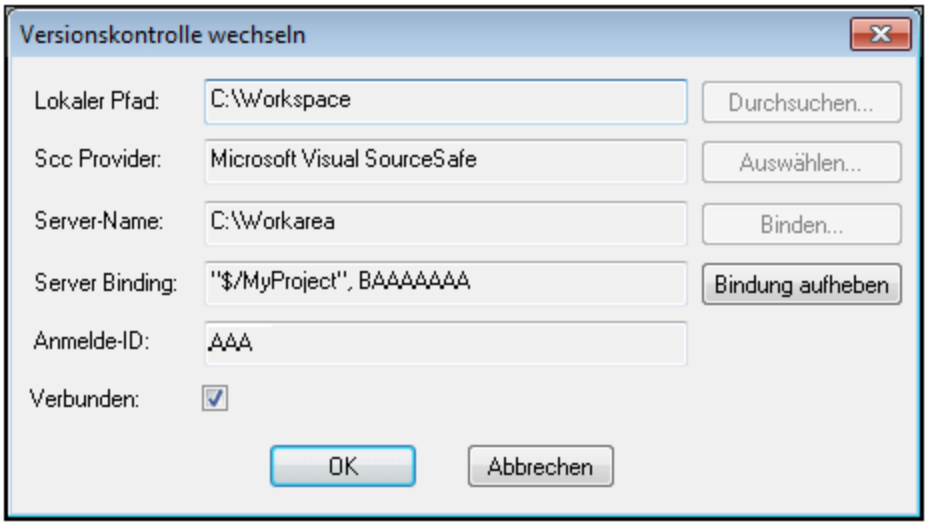


Um die gewünschten Dateien auszuwählen, wählen Sie zuerst den Ordner, in dem sich die Dateien befinden, in der Projektstruktur auf der rechten Seite aus. Die Dateien im ausgewählten Ordner werden im Fenster auf der linken Seite angezeigt. Wählen Sie die freizugebende Datei aus (um mehrere Dateien auszuwählen, halten Sie beim Anklicken die **Strg**-Taste gedrückt). Die ausgewählten Datei(en) werden im Textfeld *Files to Share* (*links oben*) angezeigt. Die Dateien verschwinden aus dem Fenster auf der linken Seite. Klicken Sie auf **Freigeben** und anschließend **Schließen,** um die ausgewählte(n) Datei(en) in den lokalen freigegebenen Ordner zu kopieren.

Der freigegebene Ordner wird im Namen des Dialogfelds "Share to [Ordner]" angegeben. In der Abbildung oben ist es der lokale Ordner (da sich das Zeichen `$` in dem Ordner im Repository, an den der lokale Ordner

 © 2016-2022 Altova GmbH

gebunden ist, befindet). Sie können den freigegebenen Ordner im Dialogfeld "Versionskontrolle wechseln" (*Abbildung unten,* **Versionskontrolle wechseln***)* sehen und den Ordner durch Ändern des lokalen Pfads und Binding einstellen.

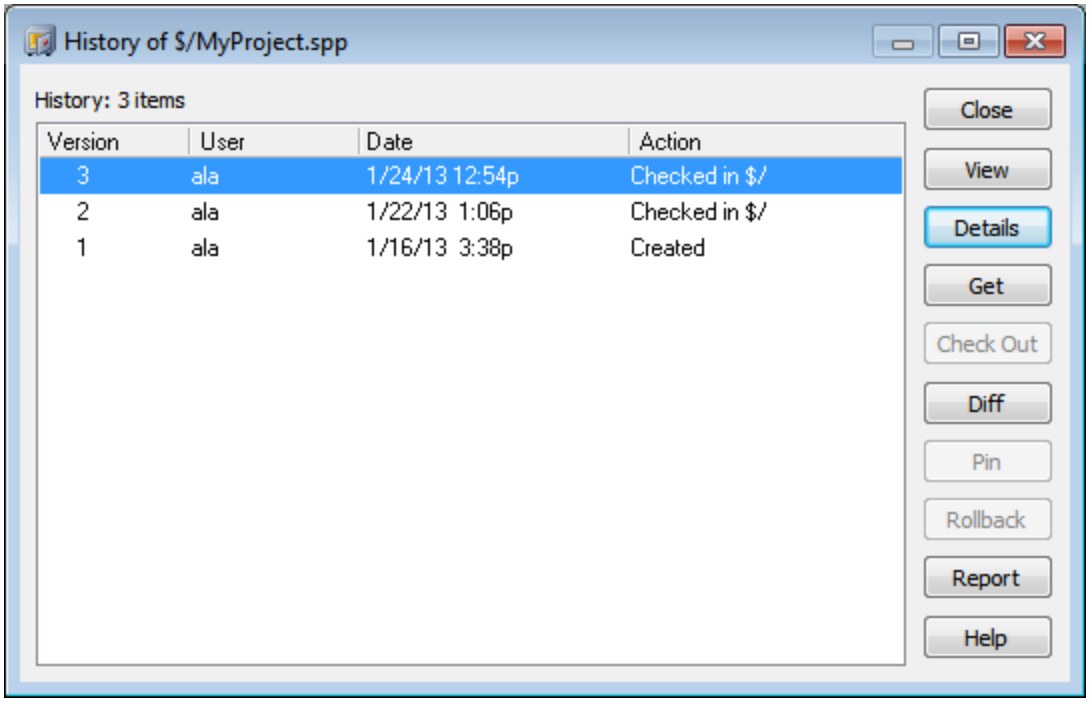


Nähere Informationen zum Freigeben von Dateien über Ihr Versionskontrollsystem finden Sie in der Benutzerdokumentation zu Ihrem Versionskontrollsystem.

13.3.6.10 Verlauf anzeigen

Mit dem Befehl **Verlauf anzeigen** aktivieren Sie die Verlaufsfunktion des aktiven Versionskontrollsystems. Damit zeigen Sie den Verlauf der im Projektfenster ausgewählten Datei an. Um den Verlauf der Projektdatei (`.spp` Datei) anzuzeigen, wählen Sie den Projekttitel aus. Sie können Informationen über die früheren Versionen einer Datei und die Unterschiede anzeigen und frühere Versionen der Datei abrufen.

In der Abbildung unten sehen Sie das Dialogfeld "Verlauf" des Versionskontrollsystems Visual SourceSafe. Darin werden die verschiedenen Versionen der Datei `MyProject.spp` aufgelistet.



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

- *Schließen*: Schließt dieses Dialogfeld.
- *Ansicht*: Öffnet ein weiteres Dialogfeld, in dem Sie die Art des Ansichtsprogramms auswählen können, in dem Sie die Datei anzeigen möchten.
- *Details*: Öffnet ein Dialogfeld, in dem Sie die Eigenschaften 204 der gerade aktiven Datei sehen.
- *Abrufen*: Damit können Sie eine der vorherigen Versionen der Datei aus der Versionsliste in das Arbeitsverzeichnis holen.
- *Auschecken*: Damit können eine frühere Version der Datei auschecken.
- *Diff*: Öffnet das Dialogfeld Vergleichsoptionen 202, in dem Sie die Vergleichsoptionen zur Anzeige der Unterschiede zwischen den beiden Dateiversionen definieren können. Mit Hilfe von **Strg + Klick** können Sie zwei Dateiversionen in diesem Fenster markieren. Klicken Sie anschließend auf Diff, um die Unterschiede zwischen den beiden Dateien anzuzeigen.
- *Pin*: Markiert eine Version der Datei mit einem Pin bzw. hebt die Marierung auf. Damit können Sie die Dateiversion definieren, die für den Dateivergleich verwendet werden soll.
- *Rollback*: Führt ein Rollback für die ausgewählte Version der Datei durch.
- *Report*: Generiert einen Verlaufsbericht, den Sie an den Drucken, die Datei oder die Zwischenablage senden können.
- *Hilfe*: Öffnet die Online-Hilfe des Versionskontrollanbieters für das Plugin.

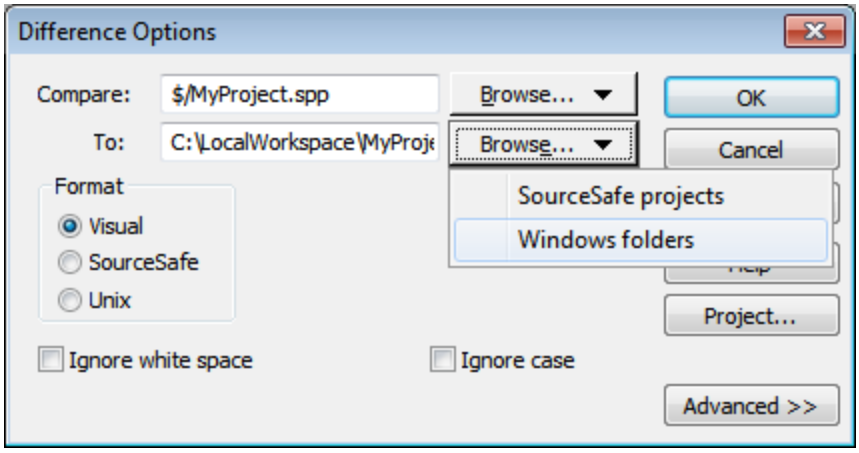
13.3.6.11 Unterschiede anzeigen

Der Befehl **Unterschiede anzeigen** ist verfügbar, wenn eine Datei im Projektfenster ausgewählt ist. Um die Projektdatei (`.spp` Datei) auszuwählen, wählen Sie den Titel des Projekts im Projektfenster aus. Mit dem

 © 2016-2022 Altova GmbH

Befehl **Unterschiede anzeigen** wird das Vergleichstool des Versionskontrollsystems gestartet, sodass Sie die Unterschiede zwischen den Dateien direkt in Ihrer Altova-Applikation überprüfen können.

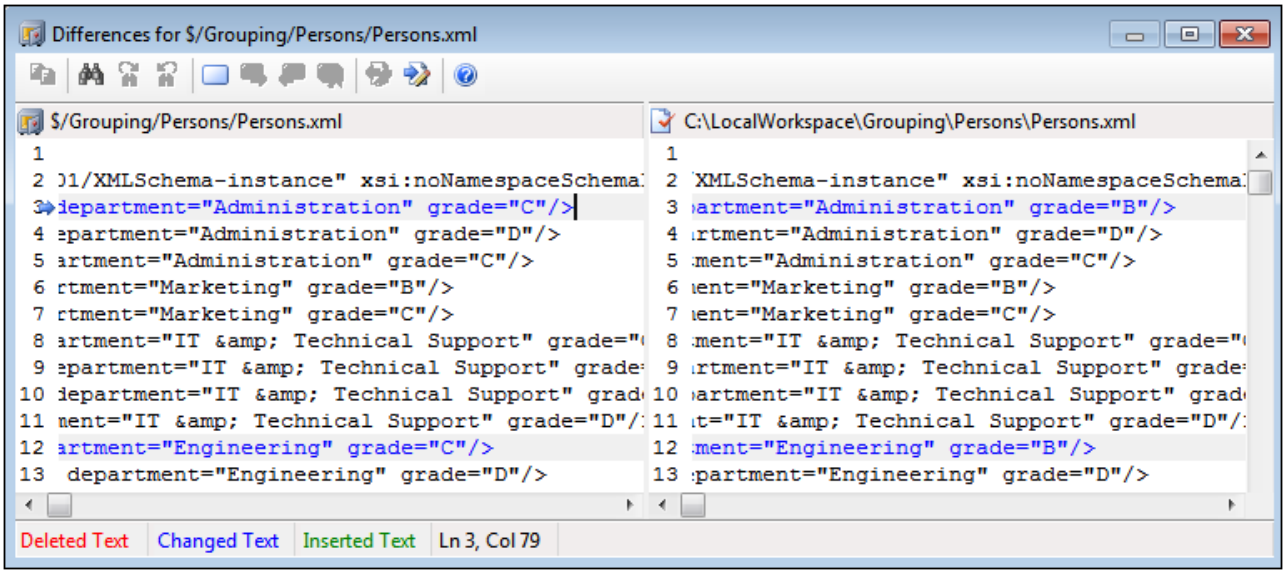
In der Abbildung unten sehen Sie das Vergleichstool des Versionskontrollsystems Visual SourceSafe.



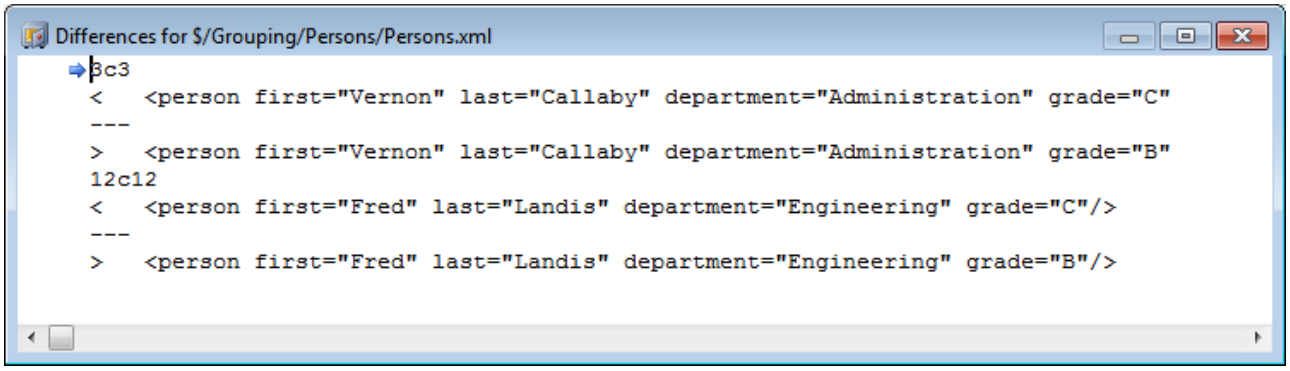
Standardmäßig werden in den Textfeldern *Compare* und *To* die Repository-Version bzw. die lokale Version angezeigt. Folgendermaßen können Sie zu anderen Dateien navigieren:

1. Über die Dropdown-Liste der Schaltfläche **Durchsuchen** können Sie SourceSafe Projekte (zum Durchsuchen der Repository-Dateien) oder Windows-Ordner (zum Durchsuchen lokaler Ordner) auswählen.
2. Navigieren Sie zu den gewünschten Dateien und wählen Sie sie aus.

Wählen Sie die gewünschten Optionen aus und klicken Sie auf **OK,** um die Dateien zu vergleichen. Die Vergleichsergebnisse werden in einem separaten Fenster angezeigt. In den Abbildung unten sehen Sie die Ergebnisse des Vergleichs in zwei Formaten.



In der Abbildung oben sehen Sie das Visual SourceSafe-Vergleichsergebnis (*siehe Dialogfeld "Optionen" oben*), während die Ergebnisse in der Abbildung unten im Unix-Format angezeigt werden. In beiden Abbildungen gibt es zwei Unterschiede - in beiden Fällen eine Änderung des "grade" von `C` in `B`.



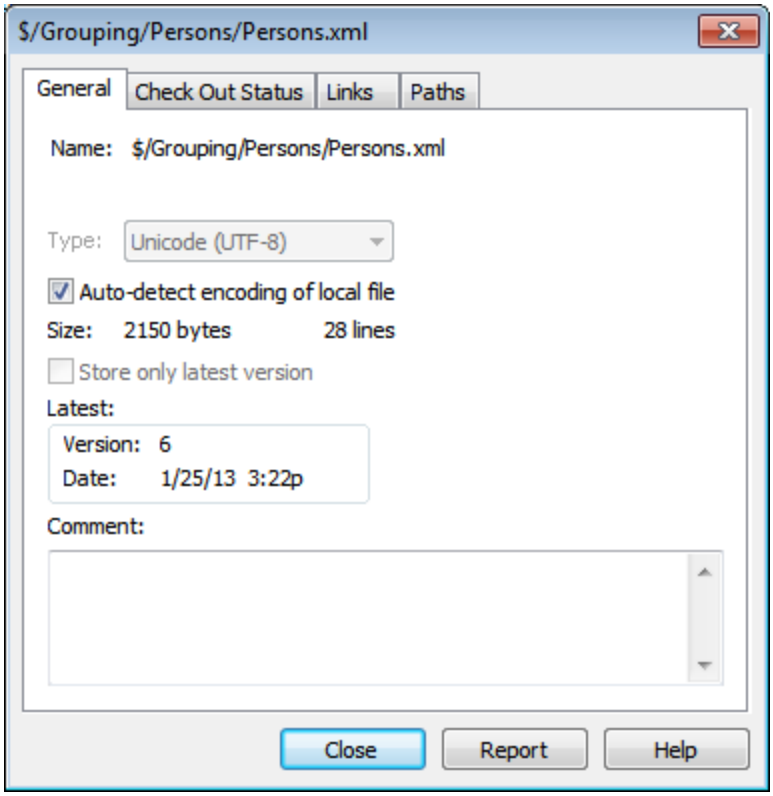
Eine ausführliche Beschreibung dazu, wie Ihr Versionskontrollsystem Unterschiede behandelt, finden Sie in der jeweiligen Produktdokumentation.

13.3.6.12 Eigenschaften anzeigen

Mit dem Befehl **Eigenschaften anzeigen** werden die Eigenschaften der gerade ausgewählten Datei angezeigt (*Abbildung unten*). Welche Eigenschaften angezeigt werden, hängt vom verwendeten Versionskontrollsystem ab. In der Abbildung unten sehen Sie die von Visual SourceSafe angezeigten Eigenschaften.

Beachten Sie, dass dieser Befehl nur für einzelne Dateien aktiv ist.

 © 2016-2022 Altova GmbH



Nähere Informationen dazu finden Sie in der Dokumentation zum jeweiligen Versionskontrollsystem.

13.3.6.13 Status aktualisieren

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

13.3.6.14 Versionskontrollmanager

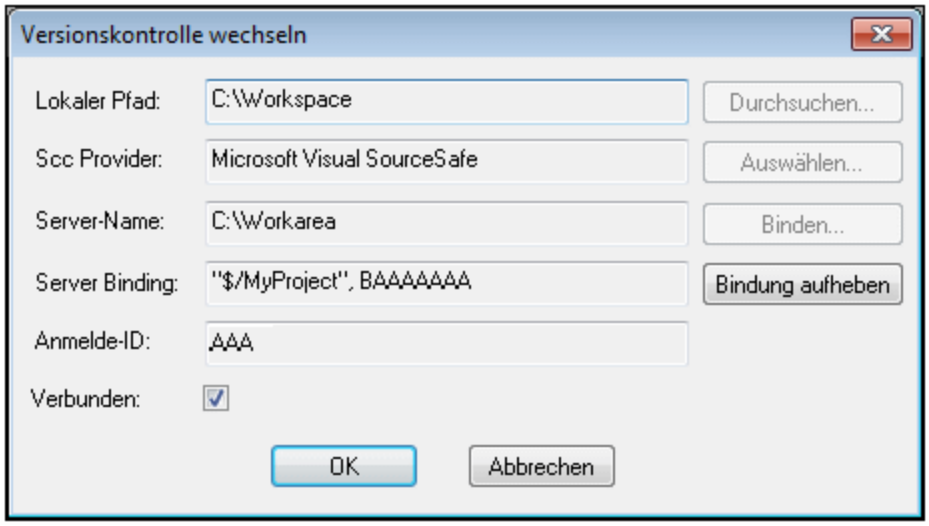
Mit diesem Befehl **starten** Sie die native Benutzeroberfläche Ihrer Versionskontroll-Software.

13.3.6.15 Versionskontrolle wechseln

Über das aktuelle Binding stellt das aktive Applikationsprojekt eine Verbindung zur Versionskontrolldatenbank her, daher muss das aktuelle Binding korrekt sein, d.h. die Applikationsprojektdatei (`.spp` Datei) muss sich im Ordner unter dem lokalen Pfad befinden und der gebundene Ordner im Repository muss die Datenbank sein, in der die Dateien dieses Projekts gespeichert sind. Normalerweise entsprechen der gebundene Ordner und seine Substruktur dem lokalen Arbeitsbereichordner und dessen Substruktur.

Im Dialogfeld "Versionskontrolle wechseln" (*Abbildung unten*) können Sie das Versionskontrollsystem (*SCC Provider*), den lokalen Ordner (*Lokaler Pfad*) und das Repository-Binding (*Server-Name* und *Server Binding*) wechseln.

Die Einstellungen können erst geändert werden, wenn Sie die aktuelle Bindung aufgehoben haben. Verwenden Sie dazu die Schaltfläche **Bindung aufheben**. Alle Einstellungen können nun bearbeitet werden.



Ändern Sie die Versionskontrolleinstellungen folgendermaßen:

1. Navigieren Sie über die Schaltfläche **Durchsuchen** zum lokalen Ordner und klicken Sie auf die Schaltfläche **Auswählen,** um eines der installierten Versionskontrollsysteme auszuwählen.
2. Anschließend können Sie den lokalen Ordner an eine Repository-Datenbank binden. Klicken Sie dazu auf die Schaltfläche **Binden**. Daraufhin wird das Verbindungsdialogfeld Ihres Versionskontrollsystems aufgerufen.
3. Wenn Sie eine *Anmelde-ID* eingegeben haben, wird diese an das Versionskontrollsystem übergeben; andernfalls müssen Sie Ihre Anmeldeinformationen im Verbindungsdialogfeld eingeben.
4. Wählen Sie im Repository die Datenbank aus, an die Sie diesen lokalen Ordner binden möchten. Eventuell müssen diese Einstellungen in mehreren Dialogfeldern vorgenommen werden.
5. Nachdem Sie alle Einstellungen vorgenommen haben, klicken Sie im Dialogfeld "Versionskontrolle wechseln" auf **OK**.

13.3.7 Dateien zu Projekt hinzufügen

Mit diesem Befehl fügen Sie Dateien in das aktuelle Projekt ein. Verwenden Sie diesen Befehl um Dateien in einen beliebigen Ordner Ihres Projekts einzufügen. Sie können einzelne Dateien oder Dateigruppen (Auswahl mit **Strg + Klick**) im Dialogfeld "Öffnen" hinzufügen. Wenn Sie Dateien in ein Projekt einfügen, werden diese automatisch - nach Erweiterung sortiert - in die entsprechenden Projektordner verteilt (Einstellungen hierzu im Dialogfeld Projekteigenschaften[215]).

 © 2016-2022 Altova GmbH

13.3.8 Globale Ressource zu Projekt hinzufügen

Mit dem Befehl **Projekt | Globale Ressource zu Projekt hinzufügen** wird das Dialogfeld "Globale Ressource auswählen" aufgerufen, in dem Sie eine globale Ressource vom Typ Datei oder Ordner auswählen können, um diese zum Projekt hinzuzufügen. Bei Auswahl einer globalen Ressource vom Typ Datei wird eine Datei auf Basis der im Dialogfeld Projekteigenschaften 215 definierten Dateierweiterung zum entsprechenden Ordner hinzugefügt. Bei Auswahl einer globalen Ressource vom Typ Ordner wird ein Ordner in einem "Datei öffnen" Dialogfeld geöffnet und Sie werden aufgefordert, eine Datei auszuwählen; die ausgewählte Datei wird auf Basis der im Dialogfeld Projekteigenschaften 215 definierten Dateierweiterung zum entsprechenden Ordner hinzugefügt. Eine Beschreibung globaler Ressourcen finden Sie im Abschnitt "Globale Ressourcen" in dieser Dokumentation.

13.3.9 URL zu Projekt hinzufügen

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

13.3.10 Aktive Datei zu Projekt hinzufügen

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

13.3.11 Aktive und verwandte Dateien zu Projekt hinzufügen

Mit diesem Befehl fügen Sie das aktive XML-Dokument und alle verwandten Dateien in das Projekt ein. Beim Arbeiten mit einem XML-Dokument, das auf einem Schema oder einer DTD basiert, fügt dieser Befehl nicht nur

das XML-Dokument zum Projekt hinzu sondern auch alle verwandten Dateien (z.B. DTDs und sämtliche geparste Entities, auf die die DTD referenziert).

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

13.3.12 Projektordner zu Projekt hinzufügen

Mit diesem Befehl fügen Sie einen neuen Ordner in das aktuelle Projekt ein. Verwenden Sie diesen Befehl, um einen neuen Ordner oder Unterordner im aktuellen Projekt anzulegen. Sie können diesen Befehl auch über das Kontextmenü (Rechtsklick auf einen Ordner im Projektfenster) aufrufen.

Anmerkung: Ein Projektordner in einen anderen Projektordner oder an eine andere Stelle im selben Projekt gezogen werden. Zusätzlich dazu kann ein Ordner aus dem Windows-Datei-Explorer in jeden beliebigen Projektordner gezogen werden.

Anmerkung: Projektordner sind grün, während externe Ordner 208 gelb sind.

13.3.13 Externes Verzeichnis zum Projekt hinzufügen

Mit dem Befehl **Projekt | Externes Verzeichnis zum Projekt hinzufügen** fügen Sie einen neuen externen Ordner in das aktuelle Projekt ein. Verwenden Sie diesen Befehl, um einen lokalen Ordner oder einen Netzwerkordner zum aktuellen Projekt hinzuzufügen. Sie können diesen Befehl auch über das Kontextmenü (Rechtsklick auf ein Verzeichnis im Projektfenster) aufrufen.

Anmerkung: Externe Ordner sind gelb, während Projektordner 208 grün sind.

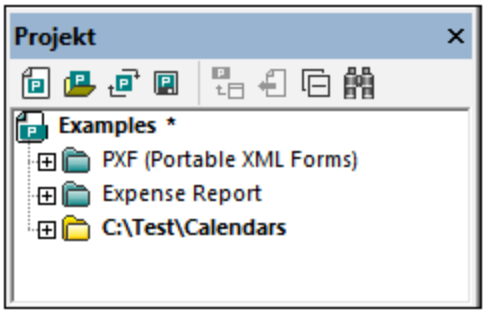
Anmerkung: In externen Ordnern enthaltene Dateien können nicht unter Versionskontrolle gestellt werden.

Hinzufügen von externen Ordnern zu Projekten

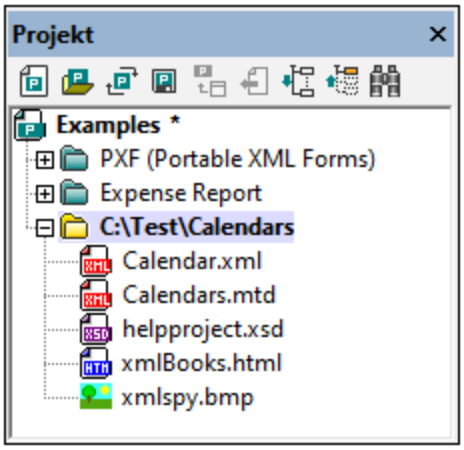
So fügen Sie einen externen Ordner zum Projekt hinzu:

1. Wählen Sie den Menübefehl **Projekt | Externes Verzeichnis zum Projekt hinzufügen**.
2. Wählen Sie den gewünschten Ordner aus und klicken Sie auf **OK**. Das ausgewählte Verzeichnis wird nun im Projektfenster angezeigt.

© 2016-2022 Altova GmbH



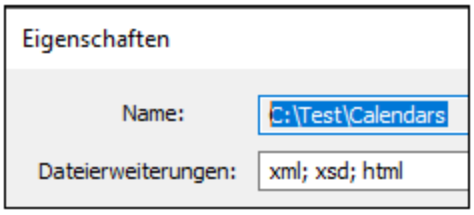
3. Klicken Sie auf das Plus-Symbol, um den Ordnerinhalt zu sehen.



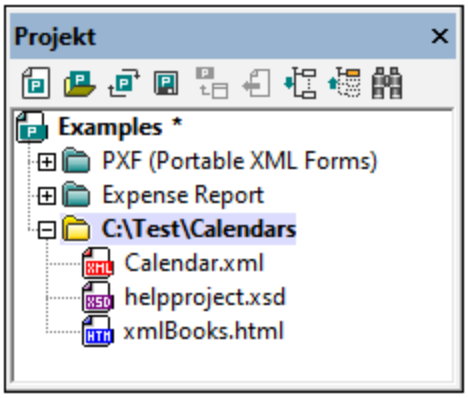
Filtern von Ordnerinhalt

So filtern Sie den Inhalt eines Ordners:

1. Rechtsklicken Sie auf den externen von Ihnen hinzugefügten Ordner. Wählen Sie den Befehl **Eigenschaften**. Daraufhin wird das Dialogfeld "Eigenschaften" geöffnet.



2. Klicken Sie in das Feld *Dateierweiterungen* und geben Sie die Erweiterungen der Dateietypen an, die angezeigt werden sollen. Bei mehreren Erweiterungen trennen Sie diese durch ein Semikolon (*siehe Abbildung oben*).
3. Klicken Sie zur Bestätigung auf **OK**.



Im Projektfenster werden jetzt ausschließlich die ausgewählten Dateiarten angezeigt.

Validieren von externen Ordnern

So validieren Sie ein externes Verzeichnis und prüfen es auf Wohlgeformtheit:

1. Wählen Sie aus, welche Dateiarten angezeigt werden sollen oder markieren Sie diese im externen Ordner.
2. Wählen Sie den Ordner aus und klicken Sie anschließend auf den Befehl **XML | Wohlgeformtheit prüfen** oder **XML validieren** (Kürzel **F7** bzw. **F8**). Alle im Ordner angezeigten Dateien werden nun überprüft. Falls eine Datei nicht wohlgeformt oder ungültig ist, wird diese im Hauptfenster geöffnet. Hier kann der Fehler korrigiert werden.
3. Korrigieren Sie den Fehler, und starten Sie die Überprüfung erneut, um den Ordner erneut zu überprüfen.

Aktualisieren eines Projektverzeichnisses

In lokalen oder netzwerkgebundenen Verzeichnissen können jederzeit Dateien gelöscht oder hinzugefügt werden. Um die Verzeichnisansicht zu aktualisieren, rechtsklicken Sie auf den externen Ordner und wählen Sie im Kontextmenü die Option **Aktualisieren**.

Löschen von externen Verzeichnissen und Dateien

Um ein Verzeichnis aus dem Projektfenster zu löschen, klicken Sie auf das **Verzeichnis** und drücken Sie die **Entf**-Taste. Klicken Sie alternativ dazu mit der rechten Maustaste auf den externen Ordner und wählen Sie den Befehl **Löschen**. In beiden Fällen wird der externe Ordner nur aus dem Projektfenster gelöscht und nicht von der Festplatte bzw. im Netzwerk.

Um eine Datei in einem externen Ordner zu löschen, müssen Sie diese physisch von der Festplatte bzw. vom Netzwerk löschen. Um die Änderungen im Projekt zu sehen, müssen Sie den Inhalt des externen Ordners (durch Rechtsklick auf den externen Ordner und Auswahl des Befehls **Aktualisieren**) aktualisieren.

Anmerkung: Ein externer Ordner kann in einen Projektordner oder an eine andere Stelle im selben Projekt (aber nicht in einen anderen externen Ordner) gezogen werden. Zusätzlich dazu kann ein Ordner aus dem Windows-Datei-Explorer an jede beliebige Stelle im Projektfenster (mit Ausnahme eines anderen externen Ordners) gezogen werden.

 © 2016-2022 Altova GmbH

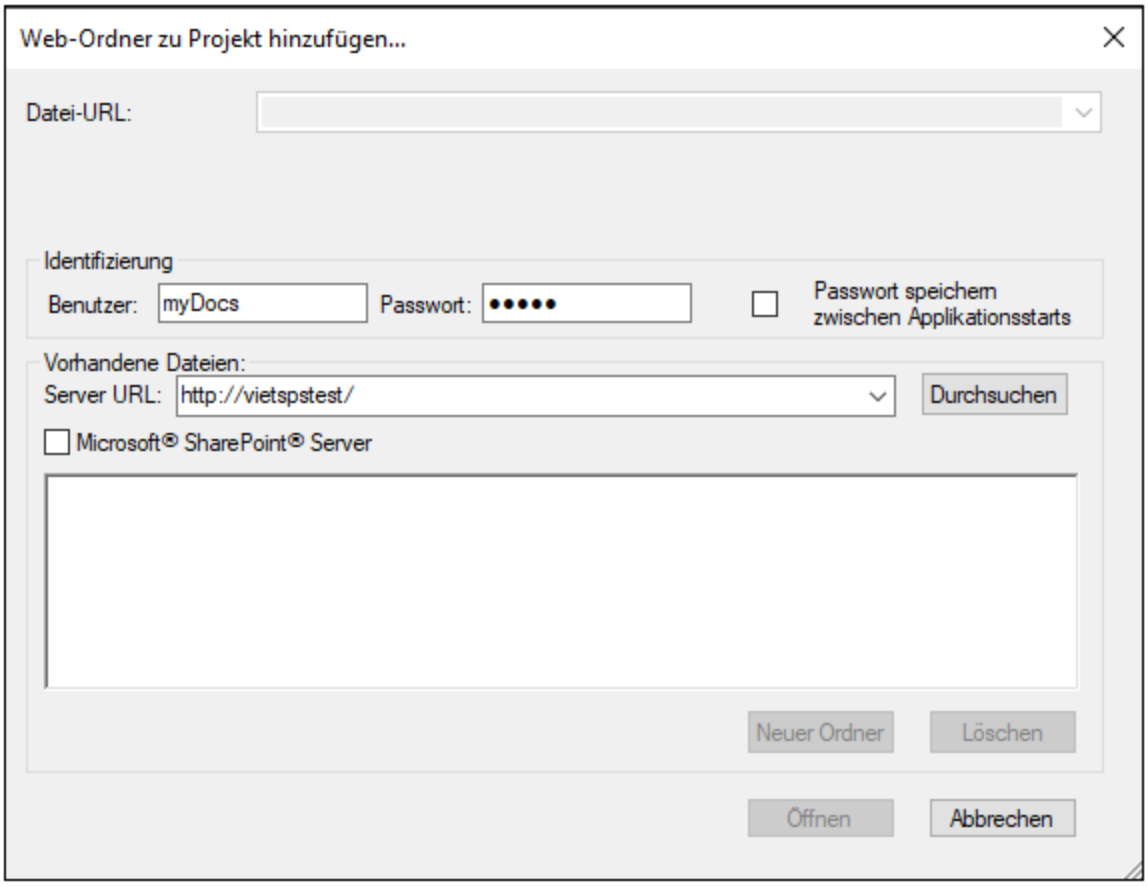
13.3.14 Externes Webverzeichnis zum Projekt hinzufügen

Verwenden Sie diese Funktion, um ein neues Webverzeichnis in das aktuelle Projekt einzufügen. Sie können diesen Befehl auch über das Kontextmenü (Rechtsklick auf ein Verzeichnis im Projektfenster) aufrufen. Bitte beachten Sie, dass Dateien in externen Verzeichnissen nicht unter Versionskontrolle gestellt werden können.

Hinzufügen eines externen Webverzeichnisses zum Projekt

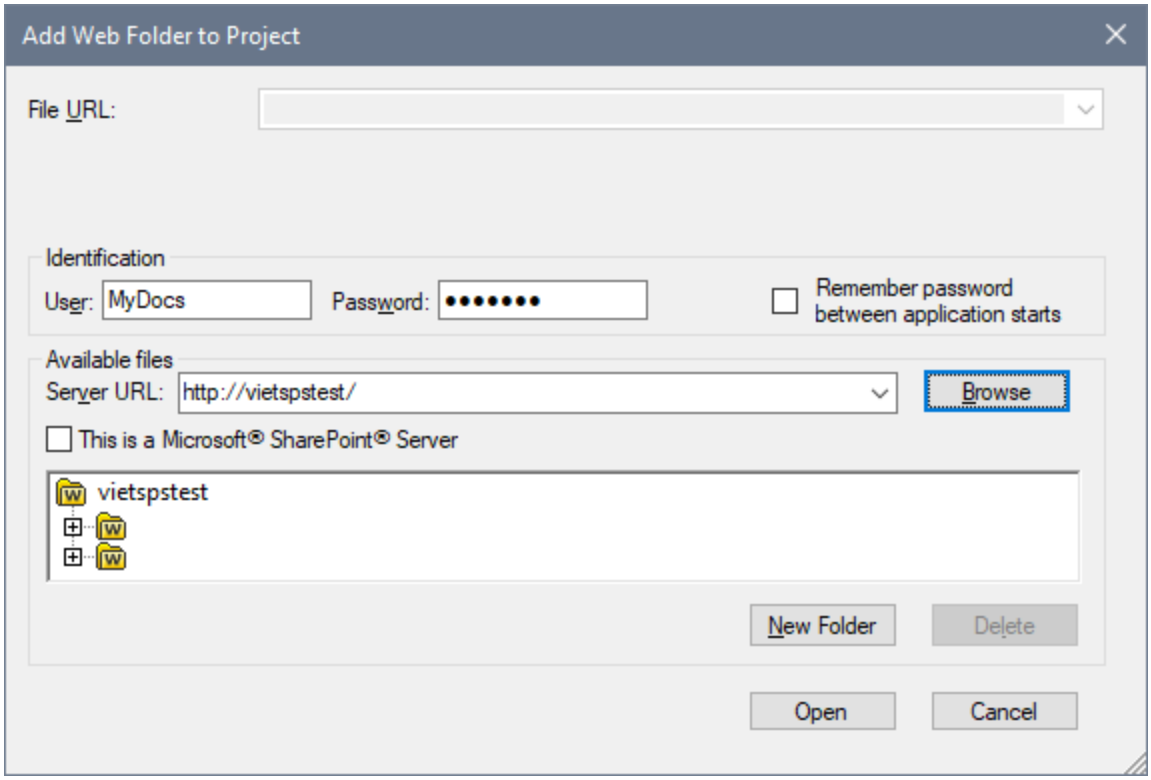
Um ein externes Webverzeichnis zum Projekt hinzuzufügen, gehen Sie folgendermaßen vor:

1. Wählen Sie die Menüoption **Projekt | Externes Webverzeichnis zum Projekt hinzufügen**. Daraufhin wird das Dialogfeld "Webordner in Projekt einfügen" angezeigt (*Abbildung unten*).

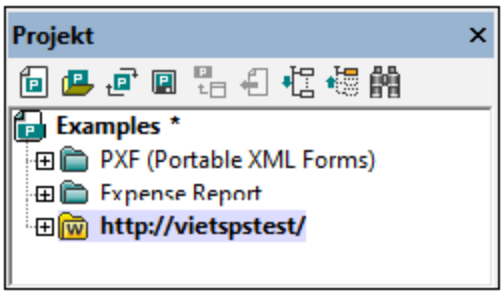


2. Klicken Sie in das Feld "Server URL" und geben Sie die URL des Servers ein. Wenn es sich beim Server um einen Microsoft® SharePoint® Server handelt, aktivieren Sie das Kontrollkästchen *Microsoft® SharePoint® Server*. Nähere Informationen zum Arbeiten mit Dateien dieses Servertyps finden Sie im Abschnitt *Ordner auf einem Microsoft® SharePoint® Server* weiter unten.
3. Geben Sie Ihre Benutzer-ID und Ihr Passwort in die Felder *Benutzer* und *Passwort* ein, wenn der Server passwortgeschützt ist.

4. Klicken Sie auf "**Durchsuchen**" um die Verbindung mit dem Server herzustellen, und die verfügbaren Ordner anzuzeigen.

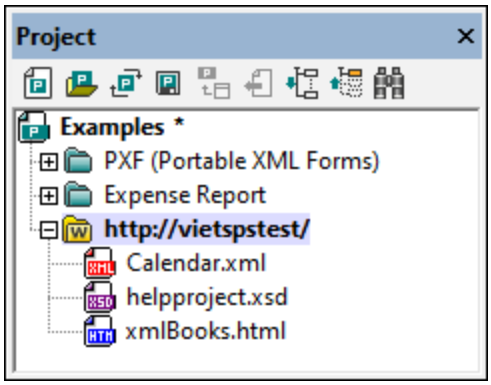


5. Klicken Sie auf das Verzeichnis, welches Sie in die Projektansicht übernehmen möchten. Die Schaltfläche **Öffnen** wird erst danach aktiv. Die URL des Ordners wird nun im Feld Datei-URL angezeigt.
6. Klicken Sie auf **Öffnen**, um das Verzeichnis in das Projekt zu übernehmen.



7. Klicken Sie auf das Symbol "+" um den Inhalt des Verzeichnisses anzuzeigen.

 © 2016-2022 Altova GmbH



Filtern von Verzeichnisinhalt

Um den Inhalt eines Ordners zu filtern, rechtsklicken Sie auf den Ordner und wählen Sie im Kontextmenü die Option **Eigenschaften**. Klicken Sie im Dialogfeld "Eigenschaften", das daraufhin geöffnet wird, auf das Feld *Dateierweiterungen* und geben Sie die Erweiterungen der Dateien an, die angezeigt werden sollen (z.B. XML- und XSD-Dateien). Trennen Sie die einzelnen Dateitypen durch ein Semikolon (z.B. `xml; xsd; sps`). Im Projektfenster wird jetzt nur mehr dieser Ordner mit Dateien mit der definierten Erweiterung angezeigt.

Validierung und Überprüfen der Wohlgeformtheit eines Ordners

Um die Dateien in einem Ordner auf Wohlgeformtheit zu prüfen oder sie zu validieren, klicken Sie auf den Ordner und anschließend auf den Menübefehl **XML | Wohlgeformtheit prüfen** bzw. **XML | XML validieren** (Kürzel F7 bzw. F8). Alle angezeigten Dateien im Ordner werden nun überprüft. Falls eine Datei nicht wohlgeformt oder ungültig ist, wird diese im Hauptfenster geöffnet. Hier können Sie den Fehler korrigieren. Nach erfolgter Korrektur, starten Sie die Prüfung erneut, um das restliche Verzeichnis zu kontrollieren. Bitte beachten Sie: Sie können mehrere nicht nebeneinander liegende Dateien gleichzeitig im Verzeichnis markieren, wenn Sie die Strg-Taste gedrückt halten, und die gewünschten Dateien anklicken. In diesem Fall werden beim Betätigen der Funktionstasten **F7** oder **F8** nur die markierten Dateien überprüft.

Aktualisieren des Inhalts des Projektordners

Sie können jederzeit Dateien aus dem Webverzeichnis hinzufügen oder löschen. Um die Verzeichnis-Ansicht zu aktualisieren, rechtsklicken Sie auf den externen Ordner und wählen Sie im Kontextmenü die Option **Aktualisieren**.

Löschen von Dateien oder Verzeichnissen

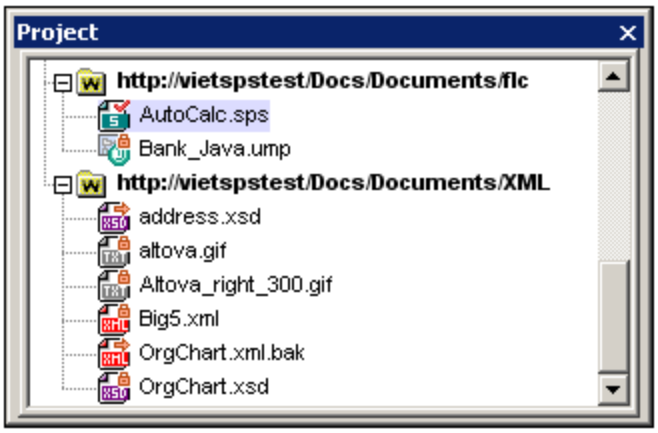
Da es sich um einen Webordner handelt, der zum Projekt hinzugefügt wurde, kann nur der Webordner (und nicht die Dateien darin) aus dem Projekt gelöscht werden. Um einen Webordner aus dem Projektfenster zu löschen, klicken Sie entweder (i) mit der rechten Maustaste auf den Ordner und wählen Sie den Befehl **Löschen** oder (ii) wählen Sie den Ordner aus und drücken Sie die **Entf**-Taste. Der Ordner wird nur aus der Projekt-Ansicht und nicht vom Webserver entfernt.

Anmerkung: Durch Klick auf eine **einzelne Datei** und Betätigen der **Entf**-Taste wird die Datei **nicht aus dem Projektfenster gelöscht**. Sie müssen sie physisch löschen, und dann den Inhalt des externen Ordners aktualisieren.

Ordner auf einem Microsoft® SharePoint® Server

Wenn ein Ordner auf einem Microsoft® SharePoint® Server zum Projekt hinzugefügt wurde, können Dateien im Ordner über Befehle im Kontextmenü zur Dateiliste im Projektfenster (*siehe Abbildung unten*) ein- und ausgecheckt werden. Um diese Befehle aufzurufen, klicken Sie mit der rechten Maustaste auf die gewünschte Datei und wählen Sie den entsprechenden Befehl aus (**Auschecken**, **Einchecken**, **Auschecken rückgängig**).

Die Benutzer-ID und das Passwort können in den Eigenschaften der einzelnen Ordner im Projekt 215 gespeichert werden, sodass Sie diese Zugriffsdaten nicht jedes Mal eingeben müssen, wenn Sie auf den Server zugreifen.



Die Dateien im Projektfenster (*Abbildung unten*) sind mit Symbolen versehen, die den Status der Dateien (Eingecheckt/Ausgecheckt) kennzeichnen. Hier sehen Sie die verschiedenen Dateisymbole:

	Eingecheckt. Die Datei kann ausgecheckt werden.
	Die Datei wurde von einem anderen Benutzer ausgecheckt und kann nicht ausgecheckt werden.
	Lokal ausgecheckt. Die Datei kann bearbeitet und eingecheckt werden.

Beachten Sie bitte die folgenden Punkte:

- Nachdem Sie eine Datei ausgecheckt haben, können Sie sie in Ihrer Altova-Applikation bearbeiten und mit **Datei | Speichern (Strg+S)** speichern.
- Sie können die bearbeitete Datei über das Kontextmenü im Projektfenster (*siehe Abbildung oben*) oder über das Kontextmenü, das angezeigt wird, wenn Sie mit der rechten Maustaste auf das Dateiregister im Hauptfenster Ihrer Applikation klicken (*Abbildung unten*), einchecken.

- Wenn eine Datei von einem anderen Benutzer ausgecheckt wurde, kann sie nicht auscheckt werden.

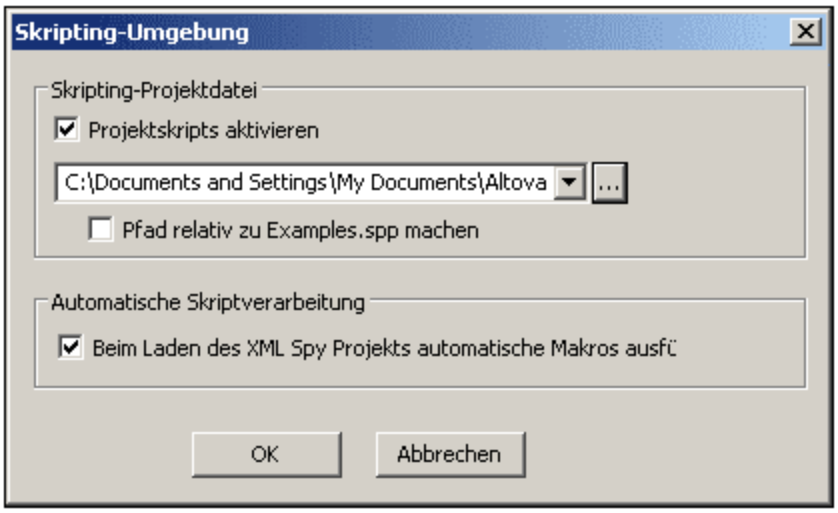
 © 2016-2022 Altova GmbH

- Wenn eine Datei lokal von Ihnen ausgecheckt wurde, können Sie dies mit dem Kontextmenübefehl "Auschecken rückgängig" rückgängig machen. In diesem Fall wird die Datei dem Server unverändert zurückgegeben.
- Wenn Sie eine Datei in einer Altova-Applikation auschecken, können Sie sie in einer anderen Altova-Applikation nicht auschecken. Die Datei gilt bereits als für Sie ausgecheckt. Daher stehen in diesem Fall in jeder Altova-Applikation, die Microsoft® SharePoint® Server unterstützt, nur die Befehle **Einchecken** und **Auschecken rückgängig** zur Verfügung.

13.3.15 Skript-Einstellungen

Ein Skripting-Projekt wird einem Authentic Desktop-Projekt auf folgende Weise zugewiesen:

1. Öffnen Sie auf der Authentic Desktop-Benutzeroberfläche das gewünschte Applikationsprojekt.
2. Wählen Sie den Menübefehl **Projekt | Skript-Einstellungen**. Daraufhin wird das Dialogfeld "Skripting-Umgebung" (*Abbildung unten*) geöffnet.



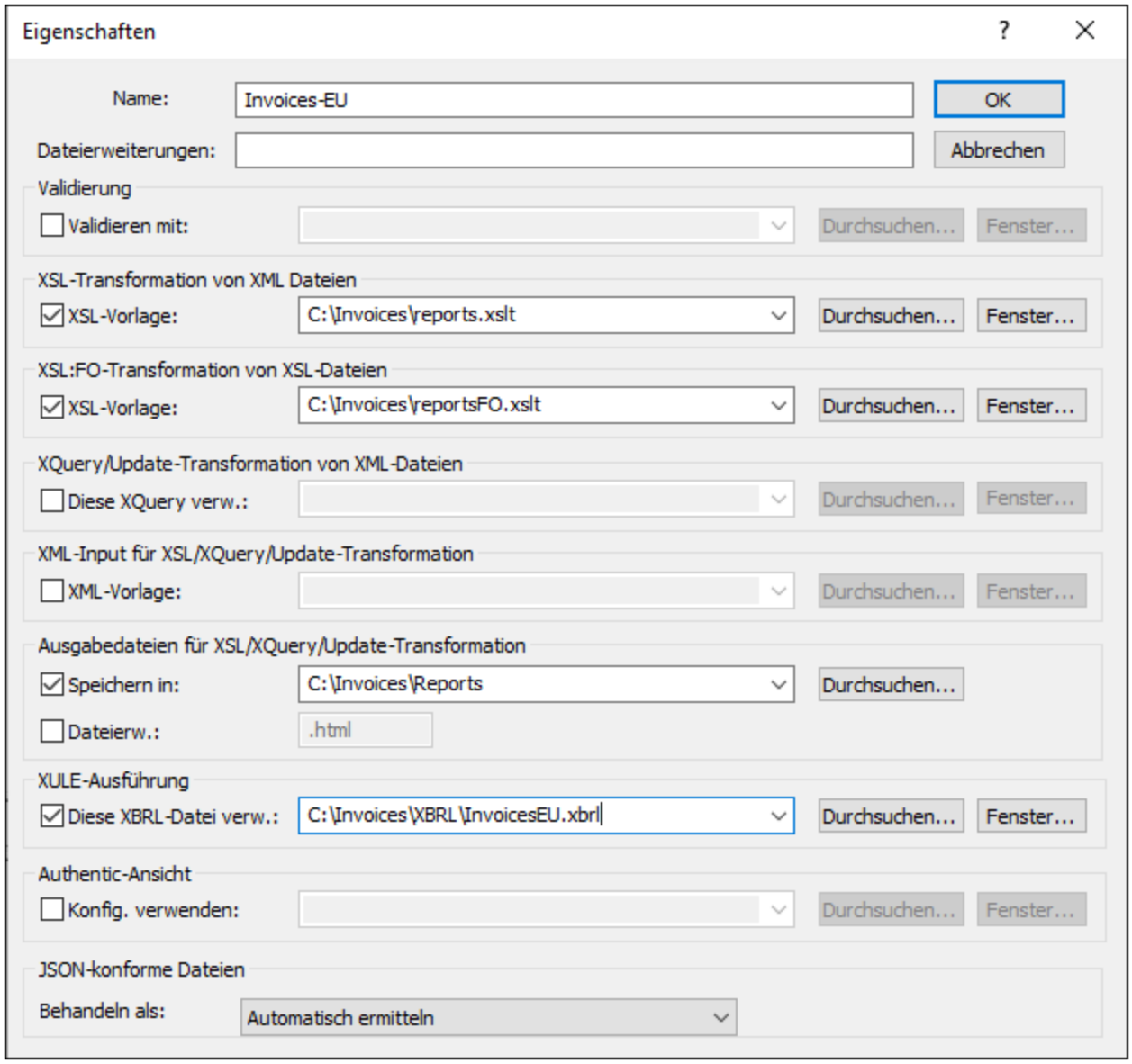
3. Aktivieren Sie das Kontrollkästchen *Projektskritps aktivieren* und wählen Sie das gewünschte Skripting-Projekt (`.asprj`-Datei) aus. Wenn beim Laden des Authentic Desktop-Projekts automatische Makros ausgeführt werden sollen, aktivieren Sie das Kontrollkästchen *Automatische Makros ausführen*.
4. Klicken Sie zum Fertigstellen auf **OK**.

Anmerkung: Um das Skripting-Projekt eines Authentic Desktop-Projekts zu deaktivieren, d.h. die Zuweisung aufzuheben, deaktivieren Sie das Kontrollkästchen *Projektskritps aktivieren*.

13.3.16 Eigenschaften

Mit dem Befehl **Projekt | Eigenschaften** wird das Dialogfeld "Eigenschaften" (*Abbildung unten*) des aktiven Projekts geöffnet. Wenn Sie mit der rechten Maustaste im Projektfenster auf einen Ordner klicken (und nicht auf den Projektordner selbst) und den Befehl **Eigenschaften** wählen, wird das Eigenschaftsdialogfeld für diesen Ordner geöffnet. Im Folgenden finden Sie eine Beschreibung der Einstellungen des Dialogfelds.

Anmerkung: Falls die Projektdatei unter Versionskontrolle steht, werden Sie gefragt, ob die Projektdatei (`.spp`) ausgecheckt werden soll. Klicken Sie auf **OK**, wenn Sie Einstellungen bearbeiten und diese speichern möchten.



Einstellungen

 © 2016-2022 Altova GmbH

Dateierweiterungen
Die Einstellung *Dateierweiterungen* ist nur bei Ordnern, nicht aber für den Projektordner aktiv. Wenn eine Datei zu einem Projekt hinzugefügt wird, wird sie zu dem Ordner hinzugefügt, für den ihre Dateierweiterungen definiert wurden. Wenn z.B. eine Datei namens `MyReport.xml` zum Projekt hinzugefügt wird und die Dateierweiterung `.xml` für den Ordner `Invoices-EU` definiert wurde (siehe Abbildung oben), so wird `MyReport.xml` zum Ordner `Invoices-EU` hinzugefügt. Wenn mehrere Ordner vorhanden sind, zu denen Sie XML-Dateien hinzufügen möchten, so sollten Sie die einzelnen XML-Dateien direkt zum Ordner (und nicht zum Projekt) hinzufügen.

Benutzer-ID und Passwort für externe Ordner
Für externe Ordner (auch externe Webverzeichnisse) können Sie die Benutzer-ID und das Passwort speichern, die unter Umständen für den Zugriff auf den Server benötigt werden.

Validierung
Die DTD, das XML-Schema oder JSON-Schema, die/das für die Validierung 219 der Dateien im aktuellen Ordner (oder im gesamten Projekt, wenn es sich um die Eigenschaften des Projekts handelt) verwendet wird.

XSL-Transformation von XML-Dateien
Das XSLT-Stylesheet, das für die XSLT-Transformation 221 von XML-Dateien im Ordner verwendet wird.

XSL-FO-Transformation von XML-Dateien
Das XSLT-Stylesheet, das für die Transformation von XML-Dateien in XSL-FO in diesem Ordner verwendet wird.

XQuery/Update-Transformation von XML-Dateien
Die XQuery- oder XQuery Update-Datei, die in diesem Ordner für XQuery- oder XQuery Update-Ausführungen von XML-Dateien verwendet wird.

XML-Input für XSL/XQuery-Update-Transformation von XML-Dateien
Die XML-Datei, die in diesem Ordner für die XSLT-Transformation oder die XQuery/XQuery Update-Ausführung mit der entsprechenden XSLT-, XQuery- oder XQuery Update-Datei in diesem Ordner verwendet wird.

Ausgabedateien für XSL/XQuery/Update-Transformation
Das Zielverzeichnis von Transformationen und optional dazu die Dateierweiterung des Ergebnisdokuments.

XULE-Ausführung
Die XBRL-Instanzdatei, die mit dem XULE-Dokument, das im XMLSpy-Applikationsfenster aktiv ist, verarbeitet werden soll.

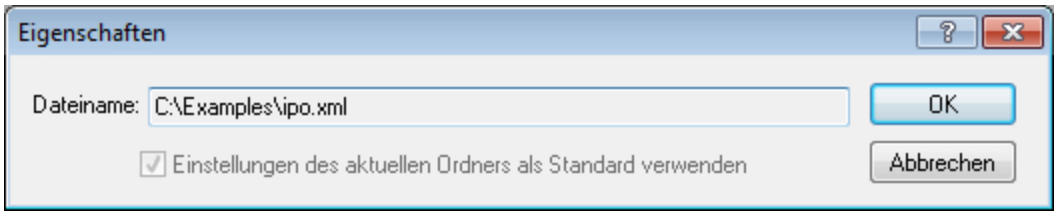
Authentic-Ansicht
Mit der Option *Konfig. verw.* wird die StyleVision Power Stylesheet-Datei (SPS-Datei) definiert, die für die Anzeige von XML-Dateien aus diesem Ordner in der Authentic View verwendet werden soll. Beachten Sie, dass die XML-Arbeitsdatei gemäß dem für das SPS ausgewählten Schema gültig sein muss.

JSON-konforme Dateien
Mit dieser Eigenschaft wird definiert, ob ein Projektordner JSON-Schema-Dateien oder JSON-Instanzdateien enthält. Sie ist sehr nützlich, wenn es darum geht, JSON-Schema-Dateien zu erkennen, wenn die Dateien nicht durch das Schlüsselwort `$schema` eindeutig als JSON-Schema-Dateien gekennzeichnet sind und die Dateien einander referenzieren. Sie können diese Eigenschaft auf *JSON-Instanz, JSON-Schema* oder *Automatisch ermitteln* setzen. Mit der Standardeinstellung *Automatisch ermitteln* würde XMLSpy den Typ der Datei anhand der Struktur und des Inhalts von JSON-Dateien ermitteln.

Anmerkungen zu Projekteigenschaften

Wenn in einer Hierarchie verschiedene Dateien für Verfahren wie die Validierung oder Transformation definiert wurden, beachten Sie hinsichtlich der Vorrangigkeit die folgenden Punkte:

- Wenn die Validierung oder XSLT/XQuery-Transformation über Kontextmenüs des Projektordners ausgeführt wird, so haben die in diesem Dialogfeld definierten Validierungs- bzw. Transformationsdateien Vorrang vor in der XML-Datei getätigten Zuweisungen.
- Außerdem haben für den jeweiligen Projektordner getroffene Einstellungen Vorrang vor Einstellungen, die für übergeordnete Ordner getroffen wurden.
- Wenn eine Datei in mehreren Ordnern des Projekts vorhanden ist und ihr in den verschiedenen Ordnern unterschiedliche Validierungs- und Transformationsdateien zugewiesen wurden, so können Sie auswählen, welche Zuweisung verwendet werden soll, wenn die Datei außerhalb des Projekts verarbeitet wird. Gehen Sie dazu folgendermaßen vor: Navigieren Sie in dem Projektordner, dessen Zuweisung(en) Sie verwenden möchten, zur gewünschten Datei. Klicken Sie mit der rechten Maustaste auf die Datei in diesem Projektordner und wählen Sie den Befehl **Eigenschaften.** Aktivieren Sie im daraufhin angezeigten Dialogfeld (*Abbildung unten*) die Option *Einstellungen des aktuellen Ordners als Standard verwenden*. (Der aktuelle Ordner ist der Projektordner, in dem sich die Datei befindet.) Wenn die Option deaktiviert ist, bedeutet dies, dass die Einstellungen des aktuellen Ordners bereits als Standardeinstellungen ausgewählt sind. Wenn Sie eine Dateiinstanz auswählen, die sich in einem Projektordner befindet, der nicht der Standardordner ist, so steht die Option zur Verfügung und Sie können die Standardeinstellungen in die Einstellungen dieses Ordners ändern. Beachten Sie, dass die lokale Zuweisung (d.h. eine Zuweisung innerhalb der Datei selbst) verwendet wird, falls die Datei eine lokale Zuweisung hat. In diesem Fall werden die Einstellungen des Standardordners ignoriert.



13.3.17 Zuletzt verwendete Projekte

Mit dieser Funktion werden die neun zuletzt verwendeten Projekte angezeigt um einen schnelleren Zugriff auf diese zu erhalten.

Beachten Sie bitte, dass Authentic Desktop automatisch das zuletzt verwendete Projekt [268] beim Starten von Authentic Desktop öffnen kann. (**Extras | Optionen|** Register **Datei**, Projekt | Letztes Projekt beim Programmstart öffnen).

 © 2016-2022 Altova GmbH

13.4 Menü "XML"

Das XML-Menü enthält alle nötigen Befehle um mit XML-Dokumenten zu arbeiten. Zu den am häufigsten verwendeten XML-Funktionen zählen die Überprüfung von Dokumenten auf ihre Wohlgeformtheit 219 und Gültigkeit 219. Sie finden diese Befehle in diesem Menü.

13.4.1 Wohlgeformtheit prüfen

F7

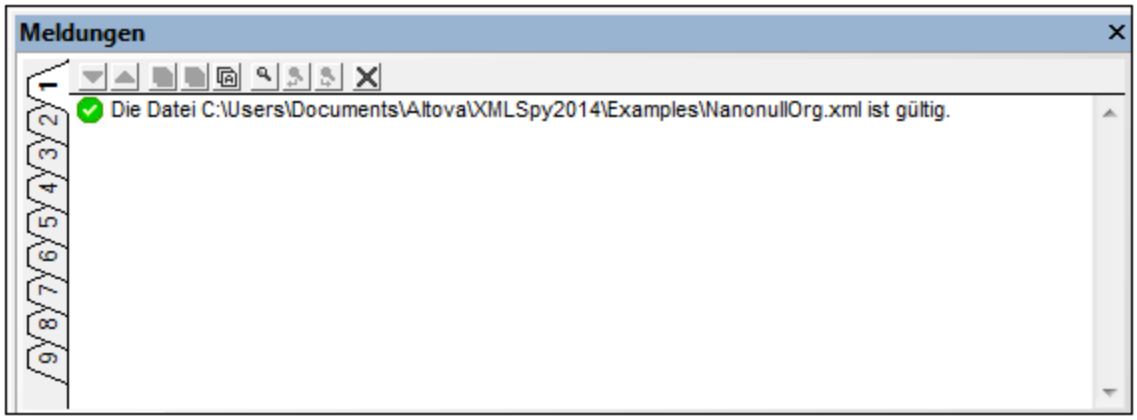
Mit dem Befehl **XML** | **Wohlgeformtheit prüfen (F7)** überprüfen Sie das aktive Dokument auf Wohlgeformtheit gemäß der Definition der XML 1.0 Spezifikation. Jedes XML-Dokument **muss** wohlgeformt sein. Authentic Desktop überprüft ein Dokument automatisch auf Wohlgeformtheit, sobald es geöffnet, neu geladen oder gespeichert wird. Wenn ein XML-Dokument nicht wohlgeformt ist, wird dies automatisch im Fenster "Meldungen" ausgegeben und das XML-Dokument wird in der Textansicht angezeigt. Sie können den Fehler in der Textansicht beheben und anschließend zurück in die Authentic-Ansicht wechseln.

13.4.2 XML validieren

F8

Mit dem Befehl **XML** | **Validieren (F8)** können Sie ein XML-Dokument anhand von DTDs, XML-Schemas und anderen Schemas validieren. Sie können einstellen, dass ein Dokument beim Öffnen oder Speichern automatisch validiert wird (**Extras | Optionen | Datei**). Anmerkung: Um Ihre Daten während der Bearbeitung zu validieren, aktivieren Sie den Befehl **Bei Bearbeitung validieren** 220.

Wenn das Dokument gültig ist, wird eine entsprechende Meldung im Fenster "Meldungen" angezeigt (*siehe Abbildung unten*). Andernfalls wird eine Meldung, die den Fehler beschreibt, angezeigt. Wenn Sie auf die Links in der Fehlermeldung klicken, können Sie zu der Stelle im XML-Dokument springen, an der der Fehler gefunden wurde. Nachdem Sie einen Fehler korrigiert haben, sollten Sie den Befehl **Validieren (F8)** erneut ausführen, um zu überprüfen, ob der Fehler wirklich behoben wurde.



Anmerkung: Das Fenster "Meldungen" enthält neun Register. Das Ergebnis der Validierung wird immer auf dem aktuellen Register angezeigt. Sie können ein XML-Dokuments auf Register 1 validieren und das Ergebnis auf diesem Register aufbewahren. Um ein zweites Dokument zu validieren, wechseln Sie zu Register 2 (oder Register 3, wenn Sie möchten), bevor Sie die Validierung durchführen. Wenn Sie die Register nicht wechseln, wird Register 1 (bzw. das aktive Register) durch das Ergebnis der neuesten Validierung überschrieben.

Validierung über das Projektfenster

Der Befehl **Validieren** kann auch auf eine Datei, einen Ordner oder eine Gruppe von Dateien im aktiven Projekt angewendet werden. Wählen Sie die gewünschte Datei bzw. den gewünschten Ordner im Projekt-Fenster (durch Anklicken) aus und klicken Sie auf **XML | Validieren** oder **F8**. Ungültige Dateien werden im Projekt als aktive Datei im Hauptfenster geöffnet und die entsprechende Fehlermeldung wird angezeigt.

Automatisches Validieren mit RaptorXML 2023

Altova RaptorXML ist eine eigenständige Altova-Applikation zur XML-Validierung, XSLT-Transformation und XQuery-Ausführung. Sie kann über die Befehlszeile, über eine COM-Schnittstelle, in Java-Programmen und in .NET-Applikationen aufgerufen werden. So können mit RaptorXML Validierungsaufgaben automatisiert werden. Sie können z.B. eine Batch-Datei erstellen, die RaptorXML aufruft, um damit eine Reihe von Dokumenten zu validieren und die Ausgabe an eine Textdatei zu senden. Nähere Informationen dazu finden Sie in der Dokumentation zu RaptorXML.

13.4.3 Bei Bearbeitung validieren

Mit dem Befehl **Bei Bearbeitung validieren** wird der *Validierung bei Bearbeitung*-Modus, in dem das Dokument während der Eingabe in die Authentic-Ansicht validiert wird, aktiviert bzw. deaktiviert. Dieser Modus kann auch über die Symbolleisten-Schaltfläche des Befehls oder die Option *Validierung* > *Bei Bearbeitung* des Abschnitts "Datei" des Dialogfelds "Optionen" 268 aktiviert bzw. deaktiviert werden.

 © 2016-2022 Altova GmbH

13.5 Menü "XSL/XQuery"

Mit Hilfe der XSL Transformationssprache können Sie festlegen, wie ein XML-Dokument in andere XML-Dokumente oder Textdateien konvertiert werden soll. Eine Art von XML-Dokument, die mittels eines XSLT-Dokuments generiert wird, ist ein FO-Dokument, das anschließend weiter in ein PDF-Ausgabedokument konvertiert wird. Authentic Desktop enthält integrierte XSLT-Prozessoren (für XSLT 1.0, XSLT 2.0 und XSLT 3.0) und kann mit einem FO-Prozessor auf Ihrem System verknüpft werden, um XML-Dateien zu transformieren und verschiedene Arten von Ausgabedokumenten zu generieren. Der Pfad zum FO-Prozessor muss im Dialogfeld "Optionen" im Abschnitt XSL (Extras | Optionen 276) angegeben werden, um direkt innerhalb der Authentic Desktop Benutzeroberfläche verwendet werden zu können.

Im Menü "XSL/XQuery" stehen die folgenden Befehle zur Verfügung:

- XSL-Transformation 221
- XSL-FO-Transformation 222
- XSL-Parameter / XQuery-Variablen 223

13.5.1 XSL-Transformation

F10

Mit diesem Befehl führen Sie mittels eines zugewiesenen XML-Dokuments eine **XSL-Transformation** durch. Die Transformationen kann entweder mit Hilfe des entsprechenden Altova XSLT-Prozessors (Altova XSLT 1.0-Prozessor für XSLT 1.0 Stylesheets; Altova XSLT 2.0-Prozessor für XSLT 2.0 Stylesheets; Altova XSLT 3.0-Prozessor für XSLT 3.0 Stylesheets), mit Hilfe des MSXML-Moduls von Microsoft oder eines externen XSLT-Prozessors durchgeführt werden. Der in Verbindung mit diesem Befehl verwendete Prozessor, wird im Dialogfeld "Optionen" (**Extras | Optionen**) auf dem Register XSL 276 festgelegt.

Sollte Ihr XML-Dokument eine Referenz zu einem XSLT Stylesheet enthalten, wird dieses Stylesheet automatisch für die XSLT Transformation verwendet., Wenn das XML-Dokument Teil eines Projekts ist, können Sie im Dialogfeld "Projekteigenschaften 215" ein XSLT Stylesheet auch für einzelne Ordner definieren. Klicken Sie mit der rechten Maustaste auf den/die Projektordner oder die zu transformierenden Datei(en) und wählen Sie XSL-Transformation. Wurde einer XML-Datei kein XSLT Stylesheet zugewiesen, werden Sie aufgefordert das XSLT Stylesheet anzugeben. Sie können eine Datei auch über eine globale Ressource, eine URL (klicken Sie dazu auf die Schaltfläche **Durchsuchen)** 169 oder eines der offenen Fenster in XMLSpy auswählen (Klicken Sie dazu auf die Schaltfläche **Fenster**).

Automatisieren von Transformationen mit RaptorXML

Altova RaptorXML ist eine Applikation zur XML-Validierung, XSLT-Transformation und XQuery-Ausführung. Sie kann über die Befehlszeile, über eine COM-Schnittstelle, in Java-Programmen und in .NET-Applikationen aufgerufen werden. So können mit RaptorXML z.B. Aufgaben wie die XSLT-Transformation automatisiert werden. Sie können z.B. eine Batch-Datei erstellen, die RaptorXML aufruft, um damit eine Reihe von Dokumenten zu transformieren. Nähere Informationen dazu finden Sie in der Dokumentation zu RaptorXML.

Transformationen in ZIP-Dateien

Damit die Ausgabe in Form einer ZIP-Datei, z.B. einer Open Office XML-Datei (OOXML) wie `.docx` erfolgt, muss im Dateipfad der Ausgabedatei das ZIP-Protokoll definiert werden. Zum Beispiel:

```
dateiname.zip|zip/dateiname.xxx

dateiname.docx|zip/dateiname.xxx
```

Anmerkung: Vor Ausführung der Transformation muss eventuell die Verzeichnisstruktur erstellt werden. Wenn Sie Dateien für ein Open Office XML-Archiv generieren, müssen Sie die Archivdateien zippen, um OOXML-Datei der obersten Ebene zu erstellen (z.B. `.docx`).

13.5.2 XSL-FO-Transformation

Strg+F10

FO ist ein XML-Format zur Beschreibung von seitenweise formatierten Dokumenten. Ein FO-Prozessor wie z.B. der FOP-Prozessor von Apache XML Projects verarbeitet eine FO-Datei und generiert anhand dieser Datei ein PDF-Dokument. Die Generierung eines PDF-Dokuments erfolgt also in zwei Schritten.

1. Das XML-Dokument wird mittels eines XSLT Stylesheet in ein FO-Dokument transformiert.
2. Aus dem FO-Dokument wird von einem FO-Prozessor ein PDF-Dokument (oder ein anderes Ausgabeformat) generiert.

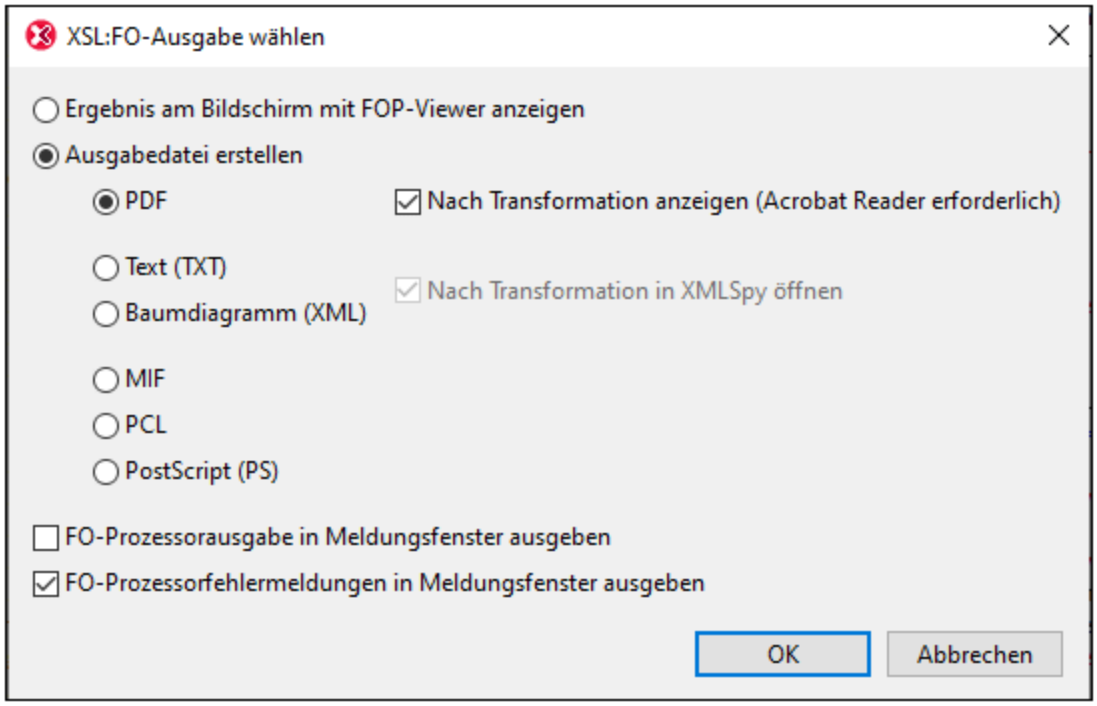
Mit dem Befehl **XSL:FO-Transformation** wird ein XML-Dokument oder ein FO-Dokument ins PDF-Format transformiert.

- Wird die **XSL:FO-Transformation** an einem XML-Quelldokument durchgeführt, werden beide oben beschriebenen Schritte nacheinander durchgeführt. Wenn das für die Transformation in FO benötigte XSLT Stylesheet im XML-Dokument nicht referenziert ist, werden Sie aufgefordert, eines für die Transformation zuzuweisen. Sie können eine Datei auch über eine globale Ressource, eine URL (klicken Sie dazu auf die Schaltfläche **Durchsuchen)** 169 oder eines der offenen Fenster in XMLSpy auswählen (Klicken Sie dazu auf die Schaltfläche **Fenster**). Die Transformation von XML in XSL-FO wird durch den XSLT-Prozessor durchgeführt, der im Dialogfeld "Optionen" (**Extras | Optionen**) auf dem Register XSL 276 definiert wurde. Standardmäßig ist der integrierte XSLT-Prozessor von XMLSpy als XSLT-Prozessor ausgewählt. Das erzeugte FO-Dokument wird von dem FO-Prozessor, der im Dialogfeld "Optionen" (**Extras | Optionen**), Register XSL 276 definiert wurde, direkt verarbeitet.
  Wird der Befehl **XSL:FO-Transformation** an einem FO-Dokument ausgeführt, wird das Dokument mit dem FO-Prozessor, der im Dialogfeld "Optionen" (**Extras | Optionen**), Register XSL 276 definiert wurde, verarbeitet.

XSL:FO-Transformationsausgabe

Wenn Sie den Befehl **XSL:FO-Transformation** aufrufen, wird das Dialogfeld "XSL:FO-Ausgabe wählen" angezeigt (*siehe Abbildung unten*). (Wenn dem aktiven XML-Dokument kein XSLT Stylesheet zugewiesen wurde, werden Sie zuerst nach der XSLT-Datei gefragt).

 © 2016-2022 Altova GmbH



Sie können das vom FO-Prozessor erzeugte Dokument mit Hilfe des FOP-Viewers direkt am Bildschirm anzeigen oder Sie können eine Ausgabedatei in einem der folgenden Formate erzeugen: PDF, Text, als XML-Baumdiagramm, MIF PCL oder PostScript. Sie können auch Meldungen von FO-Prozessor aktivieren, damit (i) die Standard-Ausgabemeldung des Prozessors im Fenster "Meldungen" angezeigt wird und (ii) die Fehlermeldungen des Prozessors im Fenster "Meldungen" angezeigt werden. Um eine der beiden Optionen zu aktivieren, klicken Sie auf das entsprechende Kontrollkästchen um unteren Rand des Dialogfelds.

Hinweis:

- Falls Sie die Option zur Installation des FOP-Prozessors des Apache XML-Projekts nicht deaktiviert haben, wurde dieser im Ordner `C:\ProgramData\Altova\SharedBetweenVersions` installiert. In diesem Fall wurde der Pfad zum Prozessor automatisch auf dem XSL-Register 276 des Dialogfelds "Optionen" (**Extras | Optionen**) als zu verwendender FO-Prozessor eingegeben. Sie können den Pfad zu jedem beliebigen FO-Prozessor, den Sie verwenden möchten, definieren.
- Sie können den Befehl XSL:FO-Transformation nicht nur an der aktiven Datei im Hauptfenster anwenden, sondern auch an jeder Datei/jedem beliebigen Ordner die/den Sie im Projektordner auswählen. Klicken Sie dazu mit der rechten Maustaste und wählen Sie den Befehl **XSL:FO-Transformation**. Es wird das XSLT Stylesheet verwendet, das dem ausgewählten Projektordner zugewiesen wurde.

13.5.3 XSL-Parameter / XQuery-Variablen

Mit dem Befehl **XSL-Parameter/XQuery-Variablen** öffnen Sie das Dialogfeld "XSLT-Eingabeparameter/externe XQuey-Variablen" (*siehe Abbildung unten*). Sie können den Namen und die

dazugehörigen Werte von einem oder mehreren Parametern eingeben, die Sie an das XSLT Stylesheet übergeben möchten, oder den Namen und die dazugehörigen Werte von einer oder mehreren XQuery-Variablen, die an das XQuery-Dokument übergeben werden sollen. Diese Parameter werden in XMLSpy folgendermaßen verwendet:

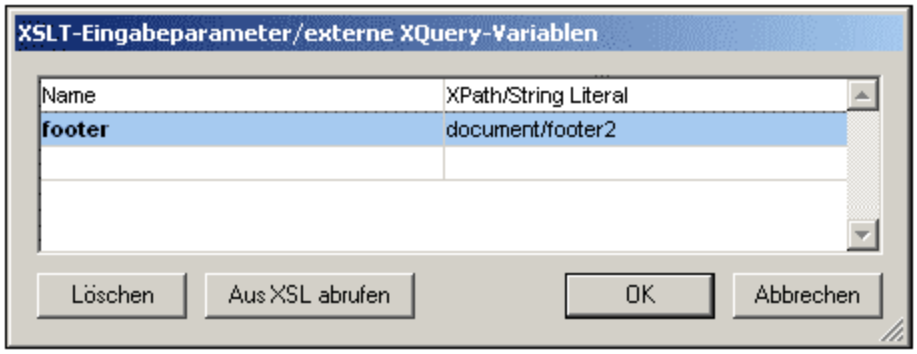
- Wenn ein XML-Dokument über den Befehl **XSL-Transformation** aus dem Menü "XSL/XQuery" transformiert wird, werden die derzeit im Dialogfeld gespeicherten Parameterwerte an das ausgewählte XSLT-Dokument übergeben und für die Transformation verwendet.
- Wenn ein XQuery-Dokument über den Befehl **XQuery-Ausführung** aus dem Menü "XSL/XQuery" verarbeitet wird, werden die aktuell im Dialogfeld gespeicherten externen XQuery-Variablen zur Ausführung an das XQuery-Dokument übergeben.

Anmerkung: Parameter oder Variablen, die Sie im Dialogfeld "XSLT-Eingabeparameter/externe XQuery-Variablen" eingeben, werden nur an den integrierten Altova XSLT-Prozessor übergeben. Wenn Sie daher MSXML oder einen anderen externen Prozessor verwenden, den Sie konfiguriert haben, werden diese Parameter nicht an diesen Prozessor übergeben.

Anmerkung: Es handelt sich nicht um einen Fehler, wenn ein XSLT-Parameter oder eine externe XQuery-Variable im Dialogfeld "XSLT-Eingabeparameter/externe XQuery-Variablen" definiert, aber im XSLT/XQuery-Dokument oder der Transformation nicht verwendet wird.

Verwendung von XSLT-Parametern

Bei dem Wert, den Sie für den Parameter eingeben, handelt es sich um einen XPath-Ausdruck. Beachten Sie, dass Strings in XPath in Anführungszeichen gesetzt werden.



Sobald Sie im Dialogfeld die Parameterwerte eingegeben haben, werden diese für alle zukünftigen Transformationen verwendet, bis sie ausdrücklich gelöscht werden oder die Applikation neu gestartet wird. Parameter, die im Dialogfeld eingegeben wurden, werden für diese Sitzung auf Applikationsebene definiert und werden von diesem Zeitpunkt an bei jeder Transformation, die über die IDE durchgeführt wird, an das jeweilige XSLT-Dokument übergeben. Dies bedeutet:

- Parameter sind nicht mit einem bestimmten Dokument verknüpft und
- Alle Parameter, die im Dialogfeld eingegeben wurden, werden gelöscht, sobald die Applikation (Authentic Desktop) geschlossen wird.

Anmerkung: Nur wenn das aktive Dokument ein XSLT-Dokument ist, ist in der Authentic-Ansicht die Schaltfläche **Aus XSL abrufen** aktiv. Damit werden die im aktiven XSLT-Dokument deklarierten Parameter zusammen mit ihren Standardwerten in das Dialogfeld eingefügt.

 © 2016-2022 Altova GmbH

Beispiel für die Verwendung von XSLT-Parametern

Im folgenden Beispiel haben wir ein XML-Dokument, das die Namen von Ländern und deren jeweiliger Hauptstadt enthält:

```
<document>
   <countries>
      <country name="USA" capital="Washington DC"/>
      <country name="UK" capital="London"/>
      <country name="France" capital="Paris"/>
      <country name="Russia" capital="Moscow"/>
      <country name="China" capital="Beijing"/>
   </countries>
</document>
```

Das folgende XSLT-Dokument generiert ein XML-Dokument, in dem ein Land aus der XML-Datei zusammen mit der Hauptstadt des jeweiligen Landes angezeigt wird. Das Land wird durch Auswahl seines Names als Wert des Parameters namens **`country`** (*unten gelb markiert*) ausgewählt.

```
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
   <xsl:param name="country" select="'USA'"/>
   <xsl:template match="countries">
      <xsl:for-each select="country[@name=$country]">
         <country>
            <name><xsl:value-of select="$country"/></name>
            <capital><xsl:value-of select="@capital"/></capital>
         </country>
      </xsl:for-each>
   </xsl:template>
</xsl:stylesheet>
```

Wenn dieses XSLT-Dokument an oben aufgelisteten XML-Dokument ausgeführt wird, erhalten Sie folgendes Ergebnis:

```
<country><name>USA</name><capital>Washington DC</capital></country>
```

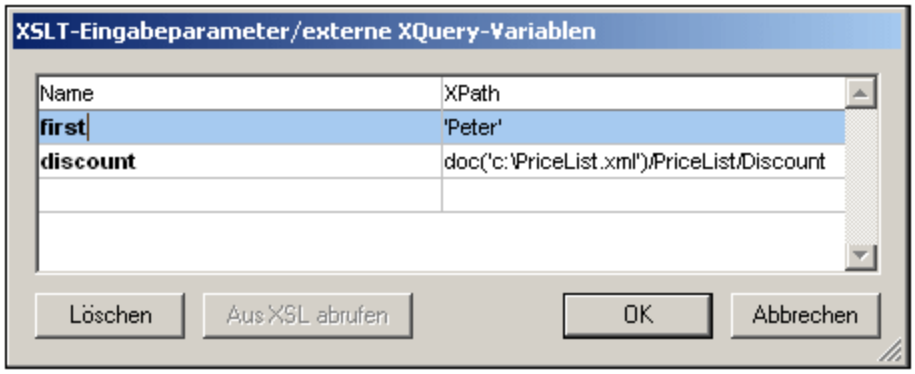
Wenn Sie nun im Dialogfeld "XSLT-Eingabeparameter/Externe XQuery-Variablen" einen Parameter namens **`country`** erstellen und ihm einen Wert geben (*siehe Abbildung oben*), wird dieser Wert an den Parameter **`country`** im XSLT-Stylesheet für die Transformation übergeben. Auf diese Art können Sie zur Laufzeit verschiedene Werte an verschiedene Parameter übergeben.

Anmerkung:

- Wenn Sie den Befehl **XSL:FO-Transformation** (**XSL/XQuery | XSL:FO-Transformation**) verwenden, werden Parameter, die im Dialogfeld "XSLT-Eingabeparameter/externe XQuery-Variablen" eingegeben werden, **nicht** an das Stylesheet übergeben. Damit diese Parameter in der PDF-Ausgabe verwendet werden, führen Sie mittels des Befehls XSLT-Transformation **(XSL/XQuery | XSL-Transformation)** zuerst eine XSL-FO-Transformation durch und anschließend mit Hilfe des Befehls **XSL:FO-Transformation** (**XSL/XQuery | XSL:FO-Transformation**) eine FO-PDF-Transformation.
- Falls Sie einen anderen als den integrierten Altova XSLT-Prozessor verwenden, werden Parameter, die Sie über das Dialogfeld "XSLT-Eingabeparameter" eingeben, nicht an den externen Prozessor übergeben.

Verwendung von externen XQuery-Variablen

Als Wert, den Sie für eine externe XQuery-Variable eingeben, können Sie entweder einen XPath-Ausdruck ohne Anführungszeichen oder einen Textstring innerhalb von Anführungszeichen eingeben. Der Datentyp der externen Variable wird in der Variablendeklaration im XQuery-Dokument angegeben.



Anmerkung: Sobald Sie im Dialogfeld eine Reihe von externen XQuery-Variablen eingegeben haben, werden diese für alle zukünftigen Ausführungen verwendet, bis sie ausdrücklich gelöscht werden oder die Applikation neu gestartet wird. Variablen, die im Dialogfeld eingegeben wurden, werden auf Applikationsebene definiert und werden von diesem Zeitpunkt an bei jeder Transformation, die über die IDE durchgeführt wird, an das jeweilige XSLT-Dokument übergeben. Dies bedeutet:

- Variablen sind nicht mit einem bestimmten Dokument verknüpft und
- Alle Variablen, die im Dialogfeld eingegeben wurden, werden gelöscht, sobald die Applikation (Authentic Desktop) geschlossen wird.

Anwendungsbeispiel für externe XQuery-Variablen

Im folgenden Beispiel wird im XQuery-Dokument eine Variable `$first` deklariert und anschließend in der Return Clause des FLWOR-Ausdrucks verwendet:

```
xquery version "1.0";
declare variable $first as xs:string external;
let $last := "Jones"
return concat($first, " ", $last )
```

Diese XQuery gibt `Peter Jones` zurück, wenn der Wert der externen Variable (der im Dialogfeld "XSLT-Eingabeparameter/externe XQuery-Variablen" eingegeben wurde) `Peter` ist. Beachten Sie Folgendes:

- Das externe Schlüsselwort in der Variablendeklaration im XQuery-Dokument gibt an, dass es sich bei dieser Variable um eine externe Variable handelt.
- Die Definition des statischen Typs der Variable ist optional. Wenn der Datentyp für die Variable nicht in der Variablendeklaration angegeben wurde, wird dem Variablenwert der Typ `xs:untypedAtomic` zugewiesen.
- Wenn im XQuery-Dokument eine externe Variable deklariert wurde, aber keine externe Variable dieses Namens an das XQuery-Dokument übergeben wird, wird eine Fehlermeldung ausgegeben.
- Wenn eine externe Variable deklariert und im Dialogfeld "XSLT-Eingabeparameter/externe XQuery-Variablen" eingegeben wurde, wird diese als Gültigkeitsbereich für das auszuführende XQuery-

 © 2016-2022 Altova GmbH

Dokument verwendet. Wird eine neue Variable dieses Namens im XQuery-Dokument deklariert, setzt diese die externe Variable für den Gültigkeitsbereich vorübergehend außer Kraft. Das unten gezeigte XQuery-Dokument gibt z.B. `Paul Jones` zurück, obwohl die externe in-scope Variable `$first` den Wert `Peter` hat.

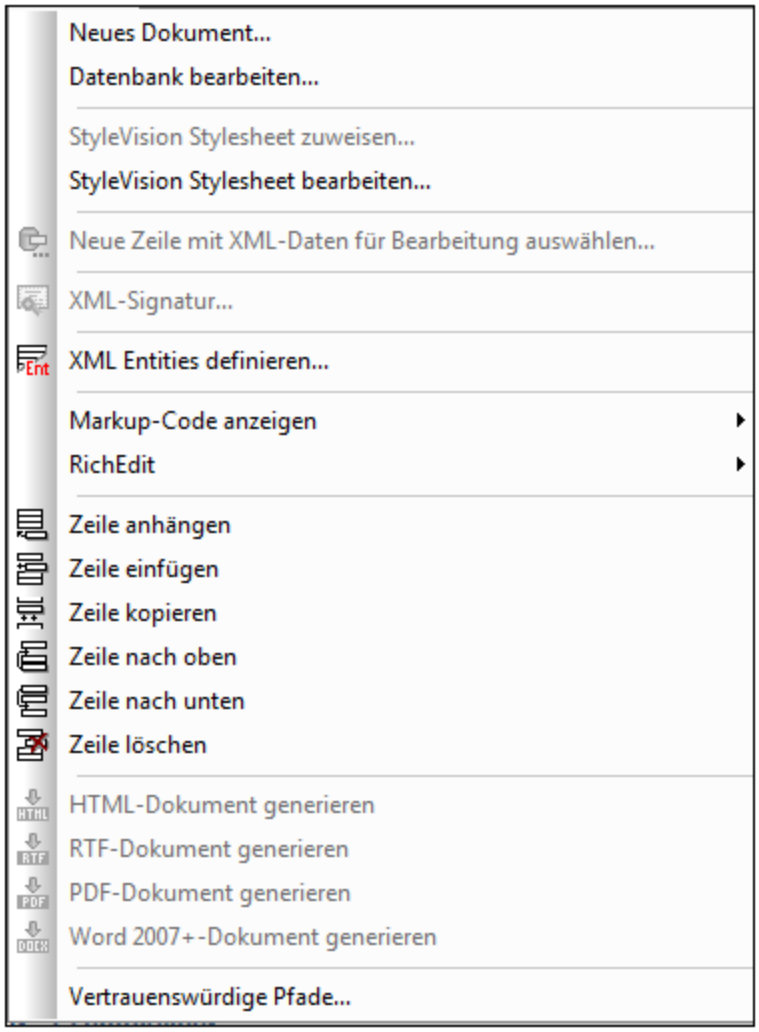
```
xquery version "1.0";
declare variable $first as xs:string external;
let $first := "Paul"
let $last := "Jones"
return concat($first, " ", $last )
```

13.6 Menü "Authentic"

Über die Authentic View-Ansicht können Sie XML-Dokumente editieren, **die auf StyleVision Power Stylesheets basieren, die mit StyleVision von Altova erstellt worden sind!** Diese Stylesheets enthalten alle Informationen, die benötigt werden, damit eine XML-Datei grafisch in der Authentic View-Ansicht dargestellt werden kann. Zusätzlich zu diesen Anzeigeinformationen ermöglichen Ihnen StyleVision Power Stylesheets außerdem, Daten in eine XML-Datei zu schreiben. Diese Daten werden mit Hilfe aller XSLT-Stylesheet-Funktionen dynamisch verarbeitet und sofort in Form einer Authentic View-Ansicht ausgegeben.

Des Weiteren können Sie StyleVision Power Stylesheets erstellen, um eine editierbare XML-Ansicht einer Datenbank anzuzeigen. Das StyleVision Power Stylesheet enthält Informationen, um eine Verbindung zu einer Datenbank herzustellen, die Datenbankdaten in der Authentic View-Ansicht anzuzeigen und wieder zurück in die Datenbank zu schreiben.

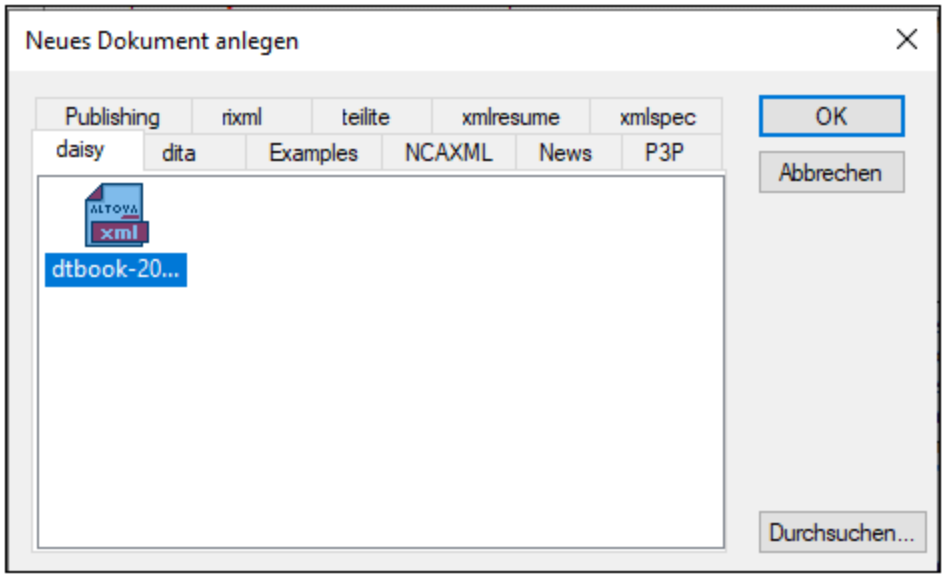
Das Menü "Authentic" enthält Befehle zum Editieren von XML-Dokumenten in der Authentic View-Ansicht. Ein Tutorial zur Authentic View-Ansicht finden Sie im Abschnitt Tutorials Authentic-Ansicht 24.



 © 2016-2022 Altova GmbH

13.6.1 Neues Dokument

Mit diesem Befehl öffnen Sie eine neue XML-Dokumentvorlage in der Authentic View-Ansicht. Die XML-Dokumentvorlage basiert auf einem StyleVision Power Stylesheet (.sps-Datei) und wird durch Auswahl des StyleVision Power Stylesheet im Dialogfeld "Neues Dokument anlegen" (*Abbildung unten*) geöffnet. Nachdem Sie ein SPS ausgewählt und mit OK bestätigt haben, wird die für diese SPS-Datei definierte XML-Dokumentvorlage in der Authentic View-Ansicht geöffnet.



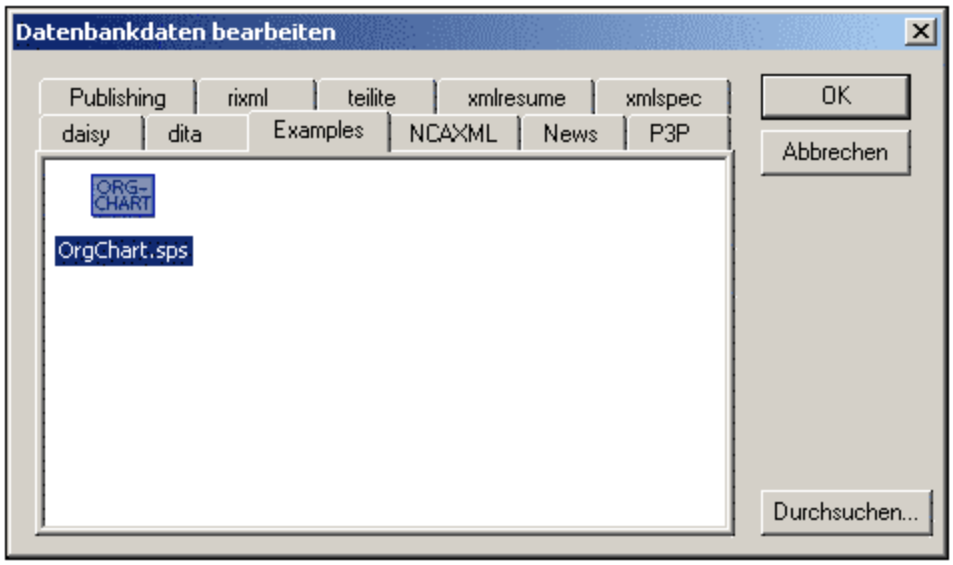
Im Dialogfeld "Neues Dokument anlegen" steht eine Reihe von XML-Dokumentvorlagen, die auf beliebten DTDs oder Schemas basieren, zur Auswahl zur Verfügung. Alternativ dazu können Sie eine benutzerdefinierte SPS-Datei, der eine XML-Vorlagendatei zugewiesen wurde, aussuchen. SPS-Dateien werden mit Altova StyleVision, einer Applikation zum Erstellen von XML-Dokumentvorlagen anhand einer DTD oder eines XML-Schemas, erstellt. Nachdem Sie die erforderliche SPS-Datei in StyleVision erstellt haben, wird der SPS-Datei (in StyleVision) eine XML-Datei als XML-Vorlagendatei zugewiesen. Die Daten in dieser XML-Datei liefern die Ausgangsdaten der neuen, in der Authentic View-Ansicht von Authentic Desktop geöffneten Dokumentvorlage.

Die neue XML-Dokumentvorlage enthält daher die im SPS definierten Layout-Eigenschaften und die Daten der als XML-Vorlagendatei ausgewählten XML-Datei. Der Benutzer der Authentic View -Ansicht kann die XML-Dokumentvorlage nun über eine grafische WYSIWYG-Oberfläche bearbeiten und als XML-Dokument speichern.

13.6.2 Datenbankdaten bearbeiten

Mit dem Befehl **Authentic | Datenbankdaten bearbeiten...** öffnen Sie eine editierbare Ansicht einer Datenbank (DB) in der Authentic View-Ansicht. Alle Informationen über die Herstellung einer Verbindung zur Datenbank, die Anzeige der DB und die Übernahme von Änderungen an der Datenbank über die Authentic View-Ansicht befinden sich in einem StyleVision Power Stylesheet. Mit dem Befehl **Datenbankdaten bearbeiten...** öffnen Sie ein solches DB-basiertes StyleVision Power Stylesheet. Damit stellen Sie eine Verbindung zur DB her und zeigen die DB-Daten (durch eine XML-Linse) in der Authentic View-Ansicht an.

Bei Aufruf des Befehls **Datenbankdaten bearbeiten** erscheint das Dialogfeld "Datenbankdaten bearbeiten" (*Abbildung unten*). Suchen Sie die gewünschte SPS-Datei und wählen Sie diese aus. Dadurch wird eine Verbindung zur DB hergestellt und eine editierbare Ansicht der Datenbank in der Authentic View-Ansicht geöffnet. Das Design der in der Authentic View-Ansicht angezeigten DB-Ansicht ist im StyleVision Power Stylesheet enthalten.



Anmerkung: Wenn Sie versuchen, ein StyleVision Power Stylesheet mit dem Befehl **Datenbankdaten bearbeiten** zu öffnen, das nicht auf einer DB basiert, oder wenn Sie versuchen, ein DB-basiertes StyleVision Power Stylesheet zu öffnen, das in einer StyleVision-Version vor der Version StyleVision 2005 erstellt wurde, erhalten Sie eine Fehlermeldung.

Anmerkung: StyleVision Power Stylesheets werden mit Altova StyleVision erstellt.

13.6.3 StyleVision Stylesheet bearbeiten

Der Befehl **Authentic | StyleVision Stylesheet bearbeiten** steht nur in der Authentic-Ansicht, d.h. nur, wenn dem XML-Dokument ein StyleVision Power Stylesheet zugewiesen wurde, zur Verfügung. Er startet StyleVision, wo Sie das StyleVision Power Stylesheet sofort bearbeiten können.

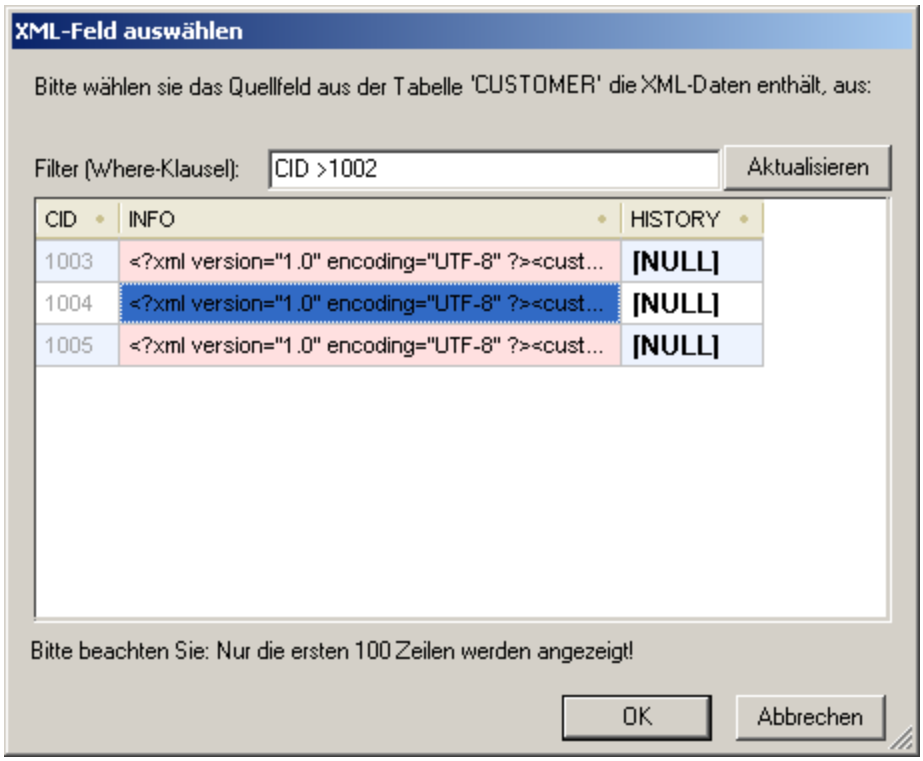
13.6.4 Neue Zeile mit XML-Daten für Bearbeitung auswählen

Mit diesem Befehl können Sie eine neue Zeile in der entsprechenden Tabelle einer XML-Datenbank wie z.B. IBM DB2 auswählen. Diese Zeile wird nun in der Authentic View-Ansicht angezeigt, kann hier bearbeitet und wieder in der Datenbank gespeichert werden.

Bei Verwendung einer XML-Datenbank als XML-Datenquelle handelt es sich bei den in der Authentic View-Ansicht angezeigten XML-Daten um das XML-Dokument, das in einer der Zellen der XML-Datenspalte enthalten ist. Mit dem Befehl **Neue Zeile mit XML-Daten für Bearbeitung auswählen** können Sie ein XML-Dokument

 © 2016-2022 Altova GmbH

aus einer anderen Zelle (oder Zeile) dieser XML-Spalte auswählen. Bei Auswahl des Befehls **Neue Zeile...auswählen** wird das Dialogfeld "XML-Feld auswählen" aufgerufen (*Abbildung unten*), in dem die Tabelle, die die XML-Spalte enthält angezeigt wird.



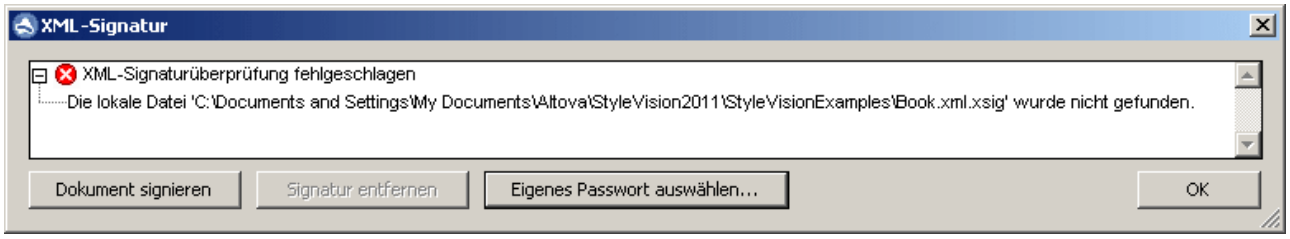
Sie können für diese Tabelle einen Filter eingeben. Dabei sollte es sich um eine SQL `WHERE`-Klausel handeln (nur die Bedingung ohne das `WHERE`-Schlüsselwort wie z.B.: `CID>1002`). Klicken Sie auf **Aktualisieren**, um das Dialogfeld zu aktualisieren. In der Abbildung oben sehen Sie das Ergebnis einer gefilterten Ansicht. Wählen Sie als nächstes die Zelle, die das gewünschte XML-Dokument enthält aus, und klicken Sie auf **OK**. Das XML-Dokument in der ausgewählten Zelle (Zeile) wird in die Authentic View-Ansicht geladen.

13.6.5 XML-Signatur

Der Befehl **XML-Signatur** steht in der Authentic-Ansicht zur Verfügung, wenn XML-Signaturen im dazugehörigen SPS aktiviert sind. Außerdem steht der Befehl **XML-Signatur** in Form der Schaltfläche "XML-Signatur" in der Authentic-Symbolleiste zur Verfügung.

Überprüfung und eigenes Zertifikat/Passwort

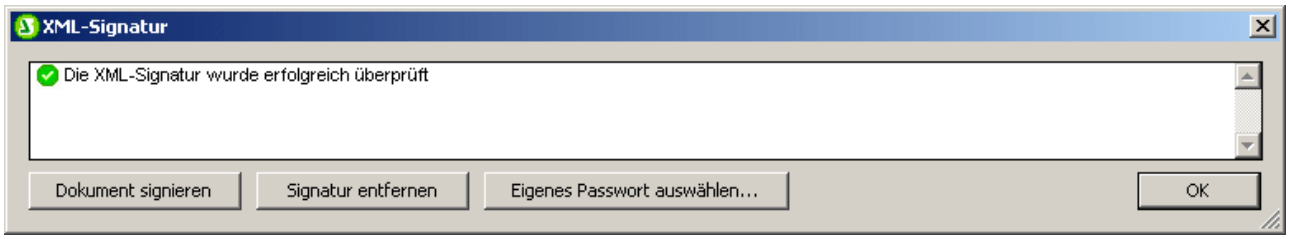
Bei Auswahl des Befehls **XML-Signatur** wird mit der Überprüfung der Signatur begonnen. Wenn keine Signatur im Dokument vorhanden ist, erscheint im Dialogfeld "XML-Signatur" eine entsprechende Meldung (*siehe Abbildung unten*) und das Dialogfeld weist eine Schaltfläche auf, mit der der Benutzer der Authentic View-Ansicht das Dokument signieren kann.



Wenn in diesem Dialogfeld die Schaltfläche **Eigenes Zertifikat auswählen** oder **Eigenes Passwort auswählen** vorhanden ist, so bedeutet das, dass der Benutzer der Authentic View-Ansicht die Option hat, ein eigenes Zertifikat/Passwort auszuwählen. (Die Entscheidung, ob ein Zertifikat oder Passwort gewählt werden soll, hat der SPS-Designer zum Zeitpunkt der Konfiguration der Signatur getroffen. Die Signatur ist entweder zertifikatbasiert oder passwortbasiert.) Wenn eine dieser Schaltflächen im Dialogfeld vorhanden ist und der Benutzer der Authentic View-Ansicht darauf klickt, kann er ein Zertifikat auswählen oder ein Passwort eingeben. Das vom Benutzer der Authentic View-Ansicht gewählte Zertifikat/Passwort wird im Arbeitsspeicher gespeichert und gilt nur für die aktuelle Sitzung. Wenn das Dokument oder die Applikation geschlossen wird, nachdem ein Zertifikat oder Passwort ausgewählt wurde, wird die Zertifikat-/Passworteinstellung wieder auf die mit dem SPS gespeicherte Originaleinstellung zurückgesetzt.

Informationen für die Überprüfung und Authentifizierung

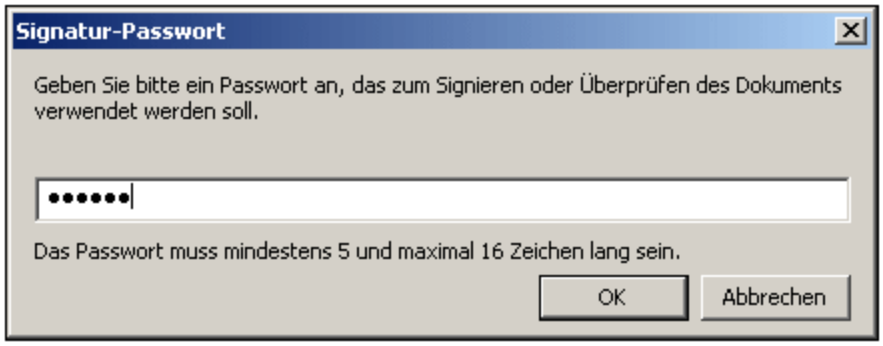
Wenn die Überprüfung an einem signierten Dokument durchgeführt wird, sind zwei allgemeine Situationen möglich. Erstens: Wenn die Authentifizierungsinformationen (in der Signatur oder dem SPS) verfügbar sind, so wird die Überprüfung direkt durchgeführt und das Ergebnis wird angezeigt (*Abbildung unten*).



Die Authentifizierungsinformationen sind entweder die Schlüsselinformationen des Zertifikats oder das zum Signieren verwendete Passwort. Der SPS-Designer hat definiert, ob die Schlüsselinformationen des Zertifikats bei der Signierung des XML-Dokuments in der Signatur gespeichert werden sollen oder ob das Passwort im Fall einer passwortbasierten Signatur im SPS gespeichert werden soll. In beiden Fällen stehen die Authentifizierungsinformationen zur Verfügung. Infolgedessen wird die Überprüfung direkt durchgeführt, ohne dass Eingaben durch den Benutzer der Authentic View-Ansicht erforderlich sind.

Die zweite mögliche Situation tritt ein, wenn die Authentifizierungsinformationen in der Signatur (Schlüsselinformationen des Zertifikats) oder im SPS (Passwort) nicht verfügbar sind. In dieser Situation wird der Benutzer der Authentic View-Ansicht aufgefordert, die Authentifizierungsinformationen anzugeben, nämlich ein Passwort (*siehe Abbildung unten*) oder den Pfad des Zertifikats.

 © 2016-2022 Altova GmbH



13.6.6 XML Entities definieren

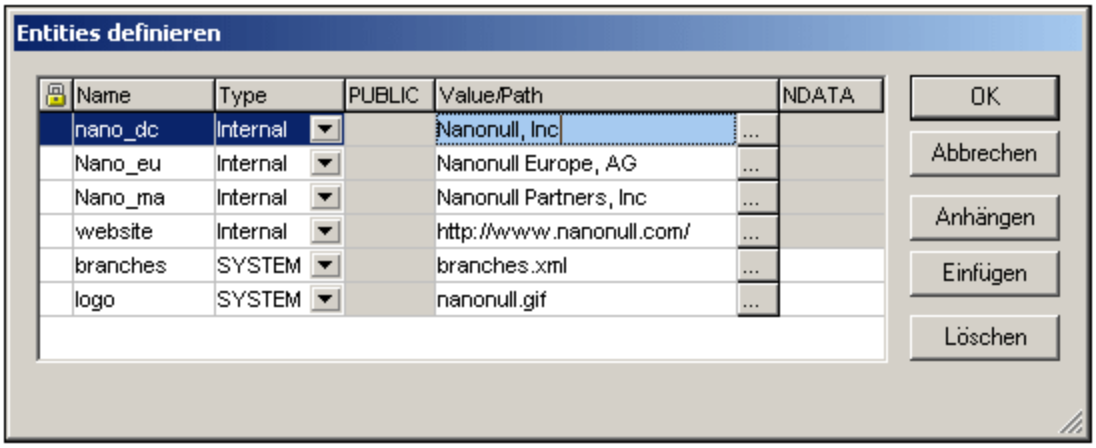
Unabhängig davon, ob Ihr Dokument auf einer DTD oder einem XML-Schema basiert, können Sie für die <% AUTH-VIEW%>-Ansicht Entities definieren, die dann in der Eingabehilfe "Entities" und im Kontext-Untermenü **Entity einfügen** angezeigt werden. Wenn Sie in der Entities-Eingabehilfe auf eine Entity doppelklicken, wird diese an der Cursorposition eingefügt.

Eine Entity eignet sich v.a. dann, wenn Sie einen Textstring, ein XML-Fragment oder einige andere externe Ressourcen an mehreren Stellen in Ihrem Dokument verwenden möchten. Die Entity, bei der es sich im Grunde um eine Kurzbezeichnung für die benötigten Daten handelt, wird im Dialogfeld "Entities definieren" definiert. Nachdem Sie die Entity definiert haben, können Sie sie an mehreren Stellen in Ihrem Dokument einfügen. Dies spart Zeit und ist effizient, wenn Sie Änderungen am Dokument vornehmen möchten.

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

So definieren Sie eine Entity:

1. Wählen Sie **Authentic | XML Entities definieren**. Dadurch wird das Dialogfeld "Entities definieren" geöffnet.



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

   - Wenn die Entity "Internal" ist, geben Sie den Textstring ein, der als Wert der Entity verwendet werden soll. Setzen Sie den Textstring nicht in Anführungszeichen. Alle Anführungszeichen werden als Teil des Textstrings behandelt.
   - Wenn der Entity-Typ SYSTEM ist, geben Sie die URI der Ressource ein oder wählen Sie mit Hilfe der **Durchsuchen**-Schaltfläche eine Ressource auf Ihrem lokalen Netzwerk aus. Wenn die Ressource geparste Daten enthält, muss es eine **.xml**-Datei sein (d.h. sie muss eine XML-Erweiterung haben). Als Alternative dazu kann die Ressource eine Binärdatei sein, z.B. eine GIF-Datei.
   - Wenn der Entity-Typ PUBLIC ist, müssen Sie zusätzlich einen System Identifier in dieses Feld eingeben.

6. Der Eintrag NDATA sagt dem Prozessor, dass diese Entity nicht geparst, sondern an den entsprechenden Prozessor gesendet werden soll. Das Feld NDATA sollte daher nur bei nicht geparsten Entities verwendet werden.

Dialogfeldfunktionen

Sie können Entities anhängen, einfügen und löschen, indem Sie auf die entsprechende Schaltfläche klicken. Des Weiteren können Sie Entities spaltenweise alphabetisch sortieren, indem Sie auf die Kopfzeile der jeweiligen Spalte klicken; bei einmaligem Klicken werden die Entities aufsteigend sortiert, bei zweimaligem Klicken absteigend. Die Größe des Dialogfelds und die Spaltenbreite lassen sich anpassen.

 © 2016-2022 Altova GmbH

Sobald eine Entity im XML-Dokument verwendet wird, ist sie gesperrt und kann im Dialogfeld "Entities definieren" nicht mehr bearbeitet werden. Gesperrte Entities werden durch das Schlosssymbol in der ersten Spalte gekennzeichnet. Durch das Sperren einer Entity wird sichergestellt, dass das Dokument in Bezug auf Entities gültig ist. (Das Dokument wäre ungültig, wenn eine Entity referenziert aber nicht definiert ist.)

Doppelt vorhandene Entities werden markiert.

Einschränkungen

- Eine in einer anderen Entity enthaltene Entity wird weder im Dialogfeld, noch der Authentic View-Ansicht noch der XSLT-Ausgabe aufgelöst. Das kaufmännische Und-Zeichen einer solchen Entity wird in seiner Escape-Form also als `&amp` angezeigt.
- Externe Entities werden in der Authentic View-Ansicht nicht aufgelöst, außer es handelt sich bei der Entity um eine Bilddatei, die als Wert eines Attributs eingegeben wurde, dessen Typ im Schema als `ENTITY` oder `ENTITIES` definiert wurde. Solche Entities werden aufgelöst, wenn das Dokument mit einem in der SPS generierten XSLT verarbeitet wird.

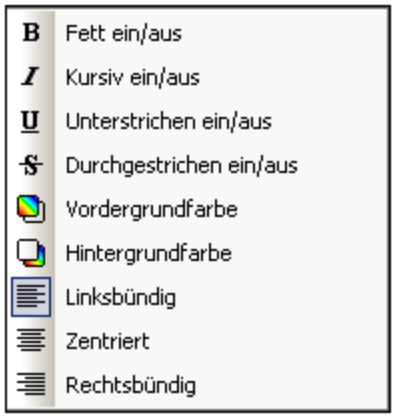
13.6.7 Markup-Code anzeigen

Der Befehl **Markup-Code anzeigen** hat ein Untermenü mit Optionen, wie der Markup-Code in der Authentic-Ansicht angezeigt werden soll. Diese Optionen sind weiter unten beschrieben.

	Markup-Symbole ausblenden blendet alle Markup-Symbole aus.
	Kleine Markup-Symbole einblenden zeigt kleine Markup-Symbole an.
	Große Markup-Symbole einblenden zeigt große Markup-Symbole an.
	Gemischte Markup-Symbole einblenden: Der Autor des StyleVision Power Stylesheet kann für einzelne Elemente/Attribute im Dokument große, kleine oder gar keine Markup-Symbole einstellen. Im gemischten Markup-Modus wird diese benutzerdefinierte Markup-Ansicht angezeigt.

13.6.8 RichEdit

Wenn Sie den Mauszeiger über einen RichEdit-Befehl platzieren, wird ein Untermenü mit den RichEdit-Markup-Befehlen (*Abbildung unten*) angezeigt. Die Menübefehle in diesem Untermenü sind nur in der Authentic View-Ansicht und nur, wenn der Cursor sich innerhalb eines als RichEdit-Komponente erstellten Elements befindet, aktiv.



Wenn Sie auf einen RichEdit-Befehl klicken, werden die Textstileigenschaften des RichEdit-Menüs auf den ausgewählten Text angewendet. Der Benutzer der Authentic View-Ansicht kann neben der in der Authentic-Symbolleiste definierbaren Schriftart und -größe die Schriftbreite, den Schriftstil, die Schriftverzierung, die Farbe, Hintergrundfarbe und Ausrichtung des ausgewählten Texts definieren.

13.6.9 Zeile anhängen/einfügen/kopieren/löschen

Mit Hilfe der unten aufgelisteten **Tabellenzeilenbefehle** können Sie Tabellen in der Authentic-Ansicht gliedern.

	Zeile anhängen hängt eine Zeile an die aktuelle Tabelle an.
	Zeile einfügen fügt eine Zeile in die aktuelle Tabelle ein.
	Zeile kopieren fügt eine Kopie der aktuellen Tabellenzeile unterhalb der aktuellen Zeile ein.
	Zeile löschen löscht die aktuelle Tabellenzeile.

13.6.10 Zeile nach unten/Zeile nach oben

Mit Hilfe der unten aufgelisteten **Tabellenzeilenbefehle** können Sie Zeilen innerhalb von Tabellen in der Authentic-Ansicht verschieben.

- Mit dem Befehl **Zeile nach oben** wird in der Authentic View-Ansicht die markierte Zeile um eine Zeile nach oben verschoben.
- Mit dem Befehl **Zeile nach unten** wird in der Authentic View-Ansicht die markierte Zeile um eine Zeile nach unten verschoben.

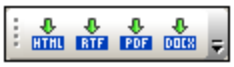
 © 2016-2022 Altova GmbH

13.6.11 HTML-, RTF-, PDF- Word 2007+-, Text-Dokument generieren

Diese fünf Befehle dienen zum Generieren von Ausgabedokumenten anhand eines in einer PXF-Datei gespeicherten XML-Dokuments in der Authentic View-Ansicht:

- **HTML-Dokument generieren**
- **RTF-Dokument generieren**
- **PDF-Dokument generieren**
- **Word 2007+-Dokument generieren**
- **Textdokument generieren**

Diese Befehle stehen auch über die Symbolleiste "Portable XML Form (PXF)" (*Abbildung unten*) zur Verfügung.

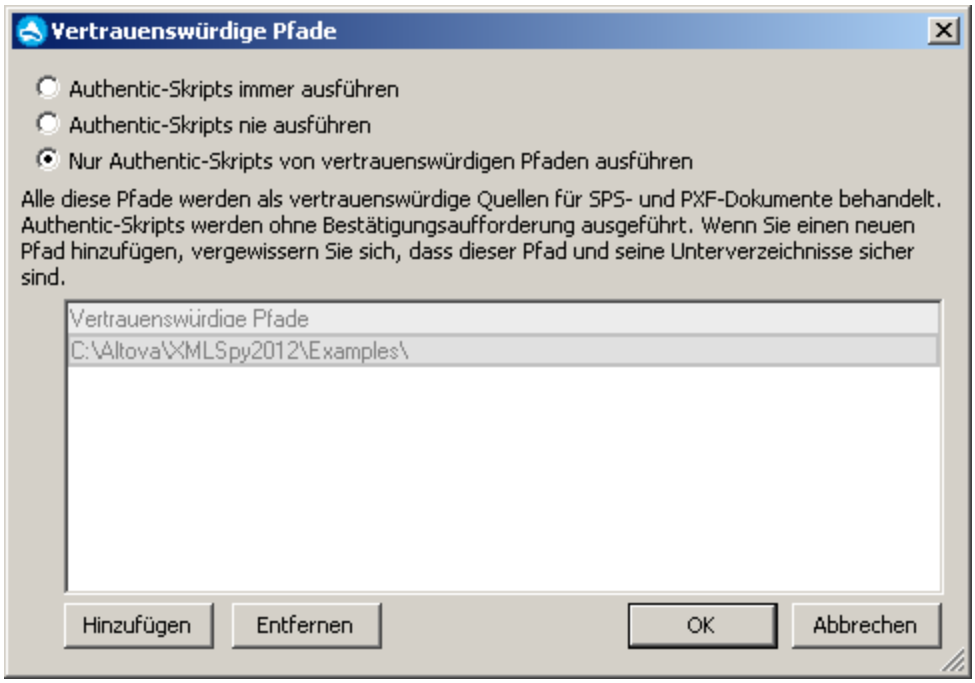


Durch Klicken auf den jeweiligen Befehl wird ein HTML-, RTF-, PDF- bzw. DocX-Ausgabedokument generiert.

Diese Schaltflächen sind aktiv, wenn eine PXF-Datei in der Authentic View-Ansicht geöffnet ist. Die einzelnen Befehle und Schaltflächen sind dann aktiviert, wenn die PXF-Datei auch das XSLT Stylesheet für das jeweilige Ausgabeformat enthält. Wenn die PXF-Datei z.B. die XSLT Stylesheets für HTML und RTF enthalten, sind nur die Befehle und Symbolleistenschaltflächen für HTML und RTF aktiviert, während diejenigen für die Text-, PDF- und DocX (Word 2007+)-Ausgabe deaktiviert sind.

13.6.12 Vertrauenswürdige Pfade

Mit dem Befehl "Vertrauenswürdige Pfade" wird das gleichnamige Dialogfeld (*Abbildung unten)* aufgerufen, in dem Sie die Sicherheitseinstellungen für die Verwendung von Skripts in einem SPS vornehmen können. In diesem Dialogfeld können Sie festlegen, ob ein Skript in einem SPS, auf dem eine XML-Datei basiert, ausgeführt werden darf oder nicht, wenn Sie bei Anzeige dieser XML-Datei in die Authentic-Ansicht wechseln.



Es stehen die folgenden drei Optionen zur Auswahl:

- Authentic-Skripts werden immer ausgeführt, wenn eine Datei in der Authentic-Ansicht geöffnet wird.
- Authentic-Skripts werden nie ausgeführt, wenn eine Datei in der Authentic-Ansicht geöffnet wird.
- Nur Authentic-Skripts aus vertrauenswürdigen Pfaden werden ausgeführt. Die Liste der vertrauenswürdigen Pfade wird im unteren Bereich angezeigt. Über die Schaltfläche **Hinzufügen** können Sie zu einem Ordner navigieren und diesen zur Liste hinzufügen. Um einen Eintrag aus der Liste zu entfernen, wählen Sie ihn in der Liste der vertrauenswürdigen Pfade aus und klicken Sie auf **Entfernen.**

 © 2016-2022 Altova GmbH

13.7 Menü "Ansicht"

Im Menü **Ansicht** (*Abbildung unten*) kontrollieren Sie die Anzeige für das aktive Hauptfenster 15 und können die Art, wie ein Dokument angezeigt wird, ändern. In diesem Abschnitt finden Sie eine Beschreibung aller Befehle im Menü "**Ansicht**".

13.7.1 Authentic-Ansicht

Mit diesem Befehl wechseln Sie mit Ihrem Dokument in die Authentic View 41-Ansicht.

In der Authentic View-Ansicht können Sie XML-Dokumente bearbeiten, die auf in Altova StyleVision erstellten StyleVision Power Stylesheet-Vorlagen basieren. Mit Hilfe dieser Vorlagen (StyleVision Stylesheets oder SPS-Dateien) werden XML-Dokumente in einem grafischen Format angezeigt, in dem das Dokument leichter als in einem mit Markup-Code versehenen Textformat bearbeitet werden kann.

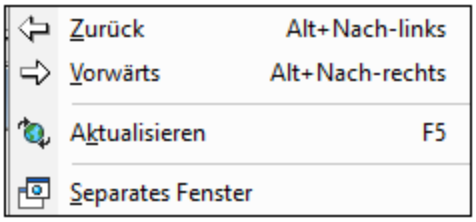
13.7.2 Browser-Ansicht

Mit diesem Befehl wechseln Sie mit Ihrem aktuellen Dokument in die Browser-Ansicht 94. Ein XML-fähiger Browser zeigt das XML-Dokument anhand der verfügbaren CSS- und/oder XSL-Stylesheets an.

Wenn Sie in die Browser Ansicht wechseln, wird das Dokument zuerst auf Gültigkeit geprüft, wenn Sie im Dialogfeld Optionen im Abschnitt "Datei" 268 (**Extras | Optionen**) die Option *Validieren beim Speichern der Datei* aktivieren. Nähere Informationen dazu erhalten Sie im Abschnitt Browser-Ansicht 94 dieser Dokumentation.

13.8 Menü "Browser"

Die Befehle im Menü **Browser** sind nur in der Browser Ansicht 94 aktiv.



Zurück, Vorwärts

Der Befehl **Zurück** (*Tastenkürzel*: **Alt + Pfeil nach links**) zeigt die zuvor angezeigte Seite an. Mit der **Zurück**-Taste auf Ihrer Tastatur erreichen Sie denselben Effekt. Dieser Befehl ist hilfreich, wenn Sie auf einen Link in Ihrem XML-Dokument klicken und danach zu Ihrem XML-Dokument zurückkehren möchten.

Mit dem Befehl **Vorwärts** (*Tastenkürzel*: **Alt + Pfeil nach rechts**) gelangen Sie vorwärts in zuvor in der Browser-Ansicht angezeigten Seiten.

Aktualisieren

Der Befehl **Aktualisieren** (**F5**) ist in der Browser-Ansicht aktiv. Mit diesem Befehl aktualisieren Sie die Browser Ansicht durch erneutes Laden des aktuellen Dokuments sowie damit verknüpfter Dokumente (wie CSS und XSL Stylesheets und DTDs).

Separates Fenster

Dieser Befehl ist in der Browser-Ansicht aktiv. Mit diesem Befehl entkoppeln Sie die Browser-Ansicht vom Applikationsfenster. Dadurch kann die Browser-Ansicht als separates Fenster Seite an Seite mit der Bearbeitungsansicht des Dokuments angezeigt werden.

Um die separate Browser-Ansicht zu aktualisieren, nachdem Sie in der Bearbeitungsansicht eine Änderung vorgenommen haben, drücken Sie in der Bearbeitungsansicht die Taste **F5.** Um ein separates Browser-Fenster wieder im Applikationsfenster anzudocken, machen Sie die Browser-Ansicht zur aktiven Ansicht und wählen Sie den Befehl **Separates Fenster**.

 © 2016-2022 Altova GmbH

13.9 Menü "Extras"

Im Menü "Extras" können Sie:

- die Rechtschreibung 241 Ihrer XML-Dokumente überprüfen
- die Skripting-Umgebung 297 von Authentic Desktop aufrufen. Sie können Ihre eigenen Formulare, Makros und Event Handler erstellen, bearbeiten und speichern.
- derzeit zugewiesene Makros ansehen 248
- benutzerdefinierte Befehle aufrufen, die auf externe Applikationen zugreifen. Diese Befehle können im Dialogfeld "Anpassen" im Abschnitt "Extras" 254 erstellt werden.
- Globalen Ressourcen definieren und verwenden 248
- Ihre Version von Authentic Desktop anpassen 250: eigene Symbolleisten, Tastaturkürzel, Menüs und Makros erstellen
- Globale Authentic Desktop-Einstellungen 267 definieren

13.9.1 Rechtschreibung

Die Rechtschreibprüfung von Authentic Desktop mit integrierten Wörterbüchern (*siehe Anmerkung unten*) kann in der Authentic View-Ansicht verwendet werden.

Anmerkung: Die in die Altova-Software integrierten Wörterbücher sind nicht Ausdruck einer Präferenz für bestimmte Sprachen durch Altova. Diese Auswahl basiert lediglich auf der Verfügbarkeit von Wörterbüchern, die mit kommerzieller Software im Rahmen von Lizenzen wie z.B. MPL-,LGPL- oderBSD-Lizenzen weitergegeben werden dürfen. Es gibt noch viele weitere Open Source-Wörterbücher, doch werden diese mit größeren Lizenzeinschränkungen wie z.B. im Rahmen von GPL Lizenzen weitergegeben. Viele dieser Wörterbücher stehen als Teil eines separaten Installationsprogramms unter http://www.altova.com/dictionaries zur Verfügung. Sie sollten die gewünschten Wörterbücher auf Basis der dazugehörigen Lizenz und ihrer Eignung für Ihre Zwecke auswählen.

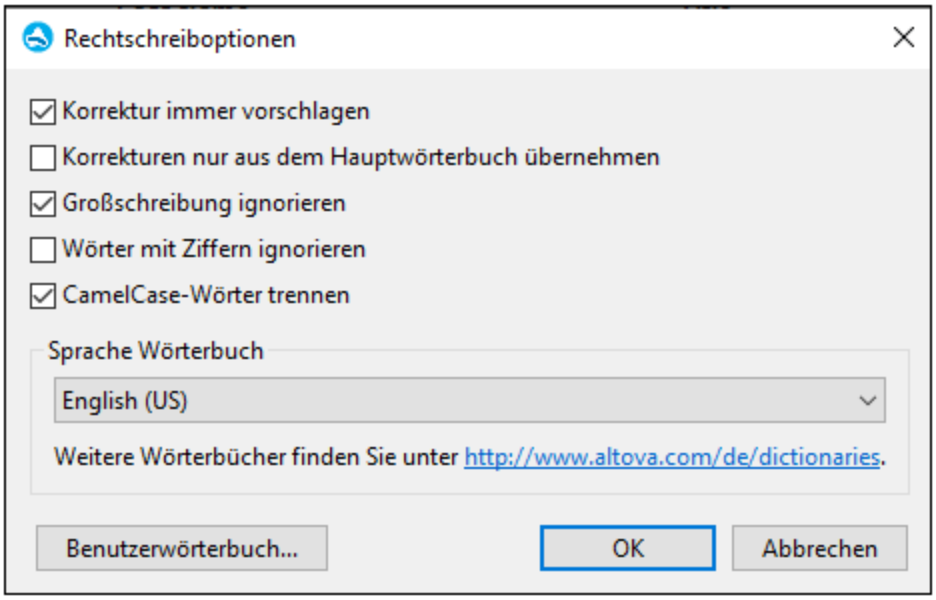
In diesem Abschnitt finden Sie eine Anleitung zur Verwendung der Rechtschreibprüfung. Der Abschnitt ist in die folgenden Unterabschnitte gegliedert:

- Auswahl der Sprache für die Rechtschreibprüfung 241
- Ausführen der Rechtschreibprüfung 242

Auswahl der Sprache für die Rechtschreibprüfung

Die Sprache für die Rechtschreibprüfung wird folgendermaßen festgelegt:

1. Wählen Sie den Menübefehl **Extras | Rechtschreiboptionen**.
2. Wählen Sie im nun angezeigten Dialogfeld "Rechtschreiboptionen" (*Abbildung unten*) eines der installierten Wörterbücher aus der Dropdown-Liste des Auswahlfelds "Sprache Wörterbuch" aus.



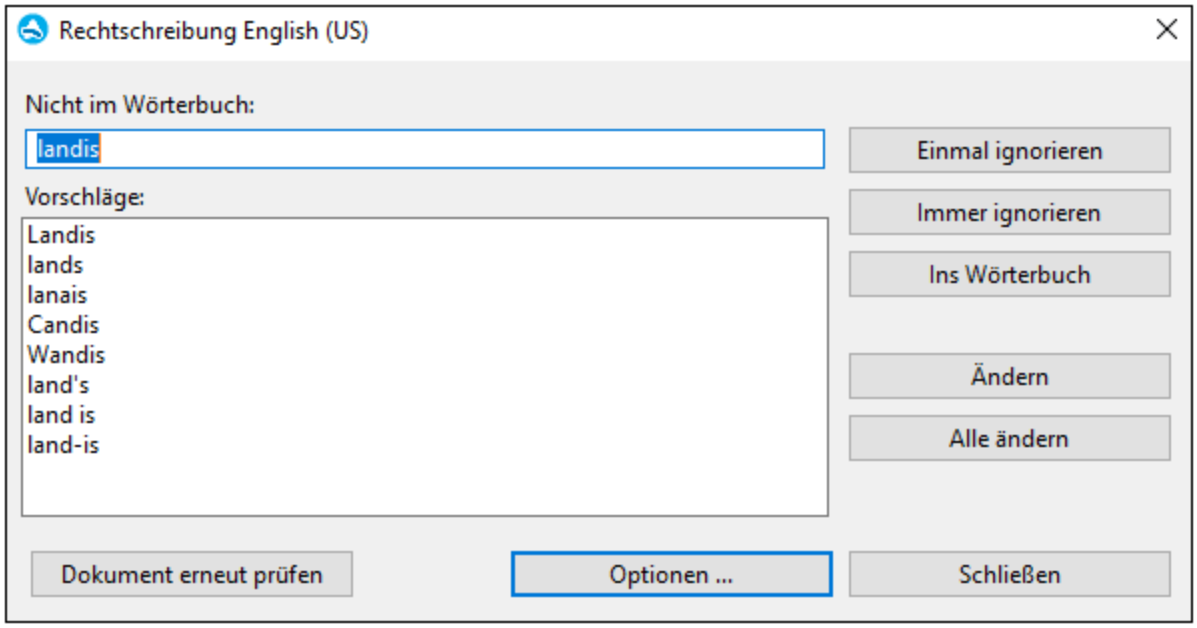
3. Klicken Sie abschließend auf **OK**.

Die ausgewählte Wörterbuchsprache wird von der Rechtschreibprüfung verwendet. Wenn die gewünschte Sprache noch nicht installiert ist, können Sie zusätzliche Wörterbücher herunterladen. Eine Anleitung dazu finden Sie im Abschnitt Hinzufügen von Wörterbüchern zur Rechtschreibprüfung 245.

Ausführen der Rechtschreibprüfung

Bei Klicken auf den Befehl **Extras | Rechtschreibung (Umschalt+F7)** wird die Rechtschreibung des gerade aktiven XML-Dokuments automatisch überprüft. Wenn ein unbekanntes Wort gefunden wird, erscheint das Dialogfeld *Rechtschreibung: Nicht im Wörterbuch* (*Abbildung unten*). Andernfalls wird das Dokument weiter bis zum Ende überprüft.

 © 2016-2022 Altova GmbH



Die verschiedenen Bereiche des Dialogfelds *Rechtschreibung: Nicht im Wörterbuch* und die verfügbaren Optionen werden weiter unten beschrieben:

Nicht im Wörterbuch
Dieses Textfeld enthält das Wort, das weder im ausgewählten Wörterbuch noch im Benutzerwörterbuch gefunden wurde. Es stehen die folgenden Optionen zur Verfügung:

- Sie können das Wort im Textfeld manuell bearbeiten oder einen Vorschlag aus dem Bereich *Vorschläge* auswählen. Klicken Sie anschließend auf **Ändern**, um das Wort im XML-Dokument durch das bearbeitete Wort zu ersetzen. (Durch Doppelklick auf ein vorgeschlagenes Wort wird dieses direkt ins XML-Dokument eingefügt). Ein Wort, das im Textfeld *Nicht im Wörterbuch* angezeigt wird, wird im XML-Dokument markiert anzeigt, sodass Sie das Wort auch direkt im Dokument bearbeiten können. Bei Auswahl der Schaltfläche **Alle ändern** werden alle Instanzen des Worts im XML-Dokument durch das geänderte Wort ersetzt.
- Sie können auch keine Änderung vornehmen und die Warnung der Rechtschreibprüfung ignorieren - und zwar entweder nur für die aktuelle Instanz des Worts oder für alle Instanzen.
- Sie können das Wort zum Benutzerwörterbuch hinzufügen, sodass das Wort in Zukunft bei jeder Rechtschreibprüfung als korrekt erkannt wird.

Vorschläge
Dieses Listenfeld zeigt Wörter an, die Ähnlichkeiten mit dem unbekannten Wort haben (die Wörter stammen aus dem Wörterbuch und Benutzerwörterbüchern). Durch Doppelklicken auf einen Eintrag wird dieser anstelle des unbekannten Wortes eingefügt und die Rechtschreibprüfung wird fortgesetzt.

Einmal ignorieren
Mit diesem Befehl können Sie das unbekannte Wort bei seinem ersten Auftreten ignorieren und die Rechtschreibprüfung fortsetzen. Sollte das Wort ein weiteres Mal im Dokument vorkommen, wird die Rechtschreibprüfung es dann wieder als unbekannt markieren.

Immer ignorieren
Mit diesem Befehl werden alle Instanzen des unbekannten Worts im gesamten Dokument ignoriert.

Ins Wörterbuch
Mit diesem Befehl können Sie das unbekannte Wort in das **Benutzerwörterbuch** einfügen. Das Benutzerwörterbuch kann über das Dialogfeld Rechtschreiboptionen 246 aufgerufen und dort bearbeitet werden..

Ändern
Mit diesem Befehl ersetzen Sie das im XML-Dokument markierte Wort durch das bearbeitete Wort aus dem Textfeld *Nicht im Wörterbuch*.

Alle ändern
Mit diesem Befehl ersetzen Sie das Wort an allen Stellen im XML-Dokument, an denen es vorkommt, durch das bearbeitete Wort aus dem Textfeld *Nicht im Wörterbuch*.

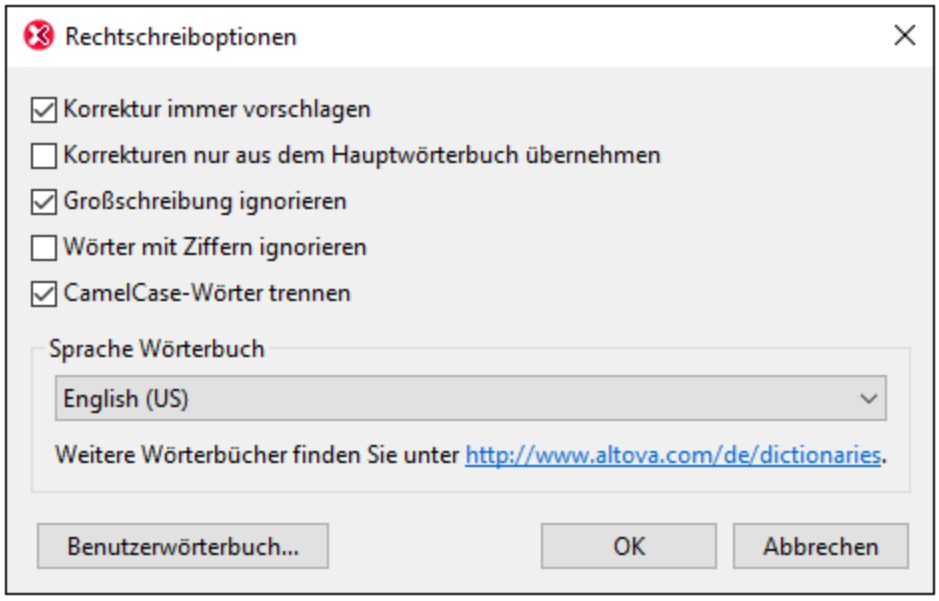
Dokument erneut prüfen
Damit wird das Dokument von Anfang an neu überprüft.

Optionen
Wenn Sie auf die Schaltfläche **Optionen** klicken, wird das Dialogfeld Rechtschreiboptionen 244 geöffnet.

Schließen
Mit diesem Befehl schließen Sie das Dialogfeld "Rechtschreibung".

13.9.2 Rechtschreiboptionen

Mit dem Befehl **Extras | Rechtschreiboptionen** wird das Dialogfeld **Rechtschreiboptionen** (*Abbildung unten*) in dem Sie globale Rechtschreiboptionen definieren können, geöffnet.



Korrektur immer vorschlagen:
Wenn Sie dieses Kontrollkästchen aktivieren, werden die Vorschläge (sowohl aus dem Wörterbuch als auch

aus dem Benutzerwörterbuch) in der Vorschlagsliste angezeigt. Wenn Sie diese Option deaktivieren, werden keinen Vorschläge angezeigt.

Korrekturen nur aus dem Hauptwörterbuch übernehmen:
Bei Aktivierung dieser Option werden nur Vorschläge angezeigt, die im Hauptwörterbuch zur Verfügung stehen. Das Benutzerwörterbuch wird nicht durchsucht. Die Schaltfläche **Benutzerwörterbuch** ist deaktiviert, sodass das Benutzerwörterbuch nicht bearbeitet werden kann.

Großschreibung ignorieren:
Großgeschriebene Wörter werden in der Rechtschreibprüfung ignoriert.

Wörter mit Ziffern ignorieren:
Wörter, die Ziffern enthalten, werden in der Rechtschreibprüfung ignoriert.

CamelCase-Wörter trennen
CamelCase-Wörter sind Wörter, die einen Großbuchstaben innerhalb des Worts haben. So ist z.B. im Wort "CamelCase" das "C" von "Case" groß geschrieben, d.h. es handelt sich hierbei um ein CamelCase-Wort. Da man derartige Wörte selten in einem Wörterbuch findet, würden Sie von der Rechtschreibprüfung als Fehler markiert. Um dies zu vermeiden, werden CamelCase-Wörter mit dieser Option in ihre beiden Komponenten aufgeteilt, sodass die Komponenten einzeln überprüft werden können. Diese Option ist standardmäßig aktiviert.

Sprache Wörterbuch
Wählen Sie im Dropdown-Listenfeld eine Sprache, in der die Rechtschreibprüfung erfolgen soll. Die Standardeinstellung ist US-Englisch. Wörterbücher in weiteren Sprachen können kostenlos von der Altova Website heruntergeladen werden.

Hinzufügen von Wörterbüchern für die Rechtschreibprüfung

Für jede Wörterbuchsprache stehen zwei Hunspell-Wörterbuchdateien zur Verfügung, die miteinander verwendet werden: eine `.aff` Datei und eine `.dic` Datei. Alle Wörterbücher werden unter dem folgenden Pfad im Ordner `Lexicons` installiert: `C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons`.

Im Ordner `Lexicons` werden unterschiedliche Sprachwörterbücher in jeweils anderen Ordnern gespeichert: `<Sprachenname>\<Wörterbuchdateien>`. So werden z.B. Dateien für die beiden englischsprachigen Wörterbücher (`English (British)` und `English (US)`) folgendermaßen gespeichert:

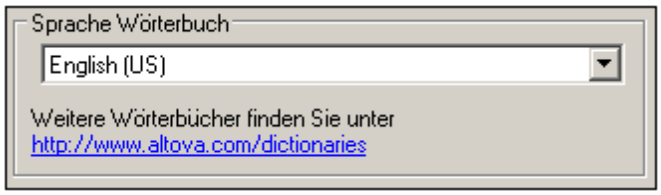
```
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (British)
\en_GB.aff
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (British)
\en_GB.dic
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (US)\en_US.aff
C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (US)\en_US.dic
```

Die Sprachwörterbücher werden im Dialogfeld "Rechtschreiboptionen" in der Dropdown-Liste der Auswahlliste *Sprache Wörterbuch* aufgelistet. Diese Wörterbücher sind diejenigen, die im Ordner `Lexicons` zur Verfügung stehen, und haben dieselben Namen wie die Unterordner für die jeweilige Sprache im Ordner `Lexicons`. So würden z.B. im Fall der oben angegebenen englischsprachigen Wörterbücher die Wörterbücher in der Auswahlliste "Sprache Wörterbuch" als *English (British)* und *English (US)* angezeigt werden.

Alle installierten Wörterbücher stehen für alle Benutzer auf dem Rechner und alle Altova-Produktversionen (ob 32-Bit oder 64-Bit) zur Verfügung

Sie können Wörterbücher für die Rechtschreibprüfung auf zwei Arten hinzufügen, wobei die Dateien in keinem Fall im System registriert werden müssen:

- Durch Hinzufügen von Hunspell-Wörterbüchern in einem neuen Unterordner des Ordners `Lexicons`. Hunspell-Wörterbücher können z.B. von https://wiki.services.openoffice.org/wiki/Dictionaries oder http://extensions.services.openoffice.org/en/dictionaries heruntergeladen werden. (Beachten Sie bitte, dass in OpenOffice das Zip-Format `OXT` verwendet wird. Ändern Sie also die Erweiterung in `.zip` und entpacken Sie die `.aff`- und `.dic`-Datei in die Sprachordner im Ordner `Lexicons`. Beachten Sie außerdem, dass Hunspell-Wörterbücher auf Myspell-Wörterbüchern basieren. Daher können auch Myspell-Wörterbücher verwendet werden.)
- Durch Verwendung des Altova-Wörterbuchinstallationsprogramms, das ein Paket mit mehreren Sprachwörterbüchern standardmäßig im richtigen Ordner auf Ihrem Rechner installiert. Das Installationsprogramm kann durch Klicken auf den Link im Bereich "Sprache Wörterbuch" des Dialogfelds "Rechtschreibung Optionen" (*siehe Abbildung unten*) heruntergeladen werden. Für die Installation benötigen Sie Administratorrechte, da die Installation sonst nicht durchgeführt werden kann.

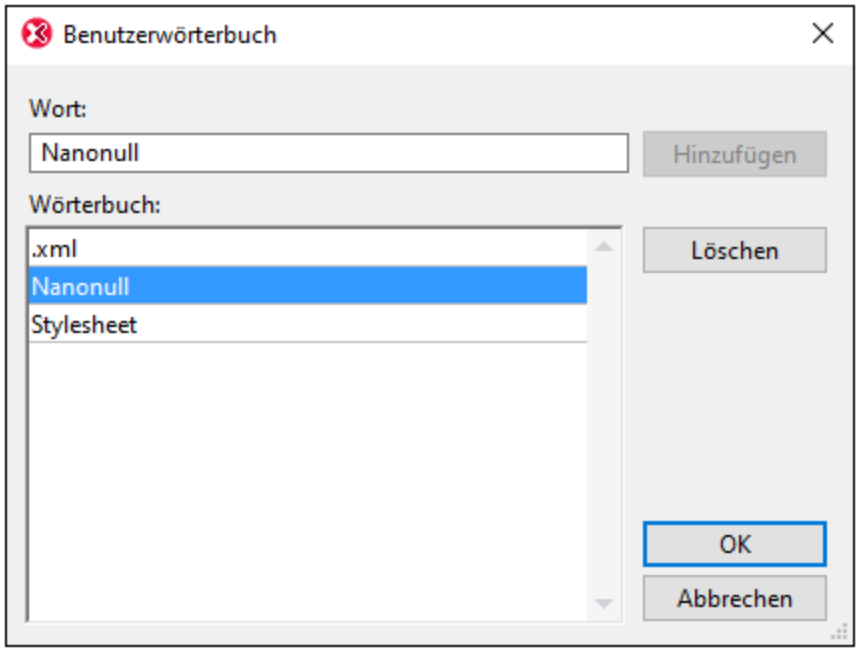


Anmerkung: Es bleibt Ihnen überlassen, ob Sie mit den für das jeweilige Wörterbuch geltenden Lizenzbedingungen einverstanden sind und ob das Wörterbuch sich für die Verwendung mit der auf Ihrem Rechner installierten Software eignet.

Arbeiten mit dem Benutzerwörterbuch

Jeder Benutzer hat ein Benutzerwörterbuch, in dem vom Benutzer genehmigte Wörter gespeichert werden können. Bei der Rechtschreibprüfung wird die Rechtschreibung anhand einer Wörterliste bestehend aus den Wörtern im Sprachwörterbuch und denen im Benutzerwörterbuch durchgeführt. Sie können Wörter über das Dialogfeld "Benutzerwörterbuch" (*Abbildung unten*) zum Benutzerwörterbuch hinzufügen bzw. diese daraus löschen. Dieses Dialogfeld wird durch Klick auf die Schaltfläche "Benutzerwörterbuch" im Dialogfeld "Rechtschreiboptionen" aufgerufen (*siehe zweite Abbildung in diesem Abschnitt*).

 © 2016-2022 Altova GmbH



Um ein Wort zum Benutzerwörterbuch hinzuzufügen, geben Sie das Wort in das Textfeld "Wort" ein und klicken Sie auf **Hinzufügen**. Daraufhin wird das Wort zur alphabetischen Liste im Fenster "Wörterbuch" hinzugefügt. Um ein Wort aus dem Wörterbuch zu löschen, wählen Sie das Wort im Fenster "Wörterbuch" aus und klicken Sie auf **Löschen**. Daraufhin wird das Wort aus dem Fenster "Wörterbuch" gelöscht. Wenn Sie mit der Bearbeitung des Benutzerwörterbuchs fertig sind, klicken Sie auf **OK**, damit die Änderungen im Benutzerwörterbuch gespeichert werden.

Wörter können auch während einer Rechtschreibprüfung zum Benutzerwörterbuch hinzugefügt werden. Wenn bei der Rechtschreibprüfung ein unbekanntes Wort gefunden wird, wird das Dialogfeld Rechtschreibprüfung 241 aufgerufen und Sie werden aufgefordert zwischen verschiedenen Aktionen zu wählen. Wenn Sie auf die Schaltfläche **Ins Wörterbuch** klicken, wird das unbekannte Wort zum Benutzerwörterbuch hinzugefügt.

Das Benutzerwörterbuch befindet sich im folgenden Ordner: `C:\Benutzer\<Benutzer>\Dokumente\Altova\SpellChecker\Lexicons\user.dic`

13.9.3 Skript-Editor

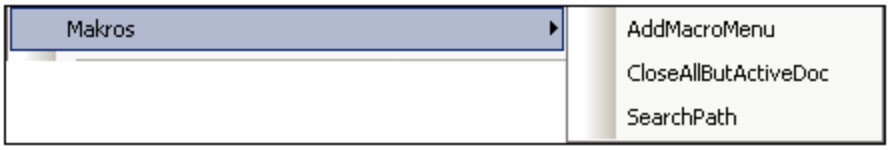
Mit dem Befehl "Skript-Editor" wird das Fenster "Skript-Editor" geöffnet. Eine Beschreibung zur Verwendung des Skript-Editors finden Sie in dieser Dokumentation im Abschnitt Skript-Editor 297.

Anmerkung: Zur Ausführung des Skript-Editors muss die .NET Framework Version 2.0 oder höher auf Ihrem Rechner installiert sein.

13.9.4 Makros

Makros werden folgendermaßen ausgeführt:

1. Platzieren Sie Ihren Cursor über den Befehl **Makros**.
2. Das daraufhin angezeigte Untermenü enthält eine Liste von Makros im Skripting-Projekt, das gerade in Authentic Desktop aktiv ist (*Abbildung unten*). Das aktive Skripting-Projekt wird im Dialogfeld "Optionen" auf dem Register Skripting-Umgebung 280 definiert.



3. Klicken Sie auf ein Makro, um es auszuführen.

13.9.5 Benutzerdefinierte Tools

Wenn Sie den Mauszeiger über den Befehl **Benutzerdefinierte Tools** platzieren, wird ein Untermenü mit benutzerdefinierten Befehlen angezeigt, die auf externe Applikationen zugreifen.Diese Befehle können im Dialogfeld "Anpassen" auf dem Register "Extras" 254 erstellt werden. Wenn Sie auf einen dieser benutzerdefinierten Befehle klicken, wird die mit diesem Befehl verknüpfte Aktion ausgeführt.

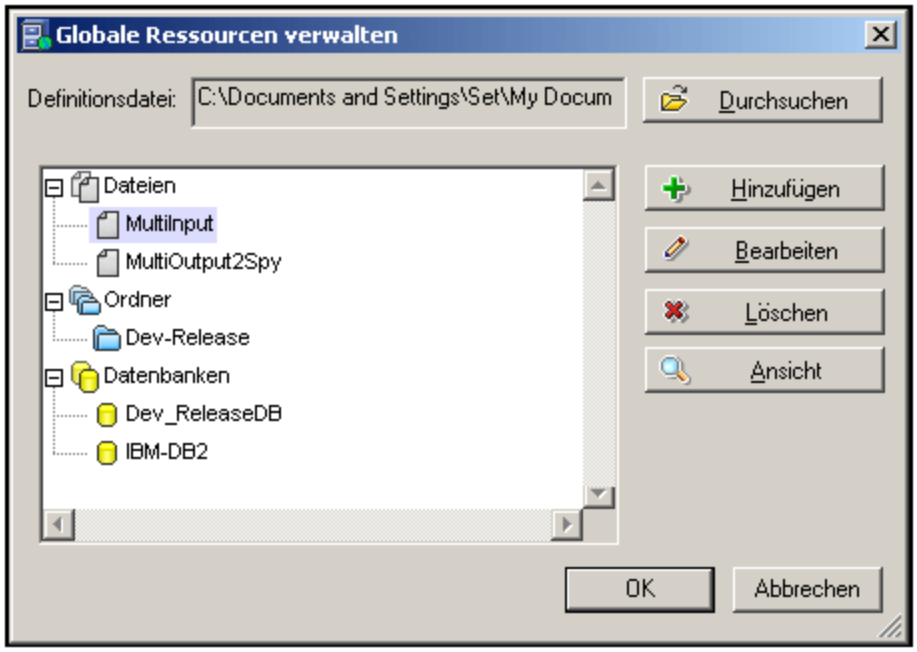
Mit dem Befehl **Benutzerdefinierte Tools | Anpassen** wird das Register "Extras" im Dialogfeld "Anpassen" 254 geöffnet (in dem Sie den benutzerdefinierten Befehl, der im Menü des Befehls **Benutzerdefinierte Tools** angezeigt wird, erstellen können.)

13.9.6 Globale Ressoucen

MIt dem Befehl **Globale Ressourcen** wird das Dialogfeld "Globale Ressourcen" aufgerufen (*Abbildung unten*), in dem Sie Folgendes tun können:

- Angabe der XML-Datei für globale Ressourcen.
- Hinzufügen von globalen Ressourcen (oder Aliassen) vom Typ Datei, Ordner und Datenbank
- Angabe verschiedener Konfigurationen für die einzelnen globalen Ressourcen (Aliasse). Jede Konfiguration wird einer bestimmten Ressource zugeordnet.

 © 2016-2022 Altova GmbH

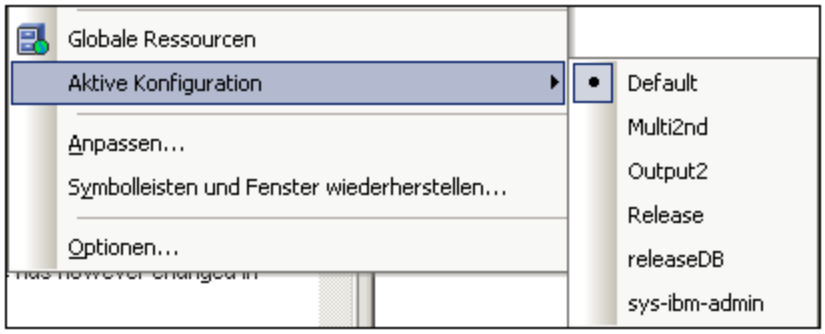


Eine Beschreibung, wie man globale Ressoucen definiert, finden Sie im Abschnitt Definieren globaler Ressourcen 96.

Anmerkung: Das Dialogfeld "Globale Ressourcen" kann auch über die Symbolleiste "Globale Ressourcen" 252 aufgerufen werden (**Ansicht | Anpassen |Symbolleisten | Globale Ressourcen**)

13.9.7 Aktive Konfiguration

Wenn Sie die Maus über den Menübefehl **Aktive Konfiguration** platzieren, wird ein Untermenü ausgerollt, das alle in der aktiven XML-Datei für globale Ressourcen 248 definierten Konfigurationen enthält. (*Abbildung unten*).



Die gerade aktive Konfiguration ist durch einen Punkt gekennzeichnet. In der Abbildung oben ist die gerade aktive Konfiguration `Default`. Um die aktive Konfiguration zu wechseln, wählen Sie die Konfiguration aus, die aktiv gemacht werden soll.

Anmerkung: Die aktive Konfiguration kann auch über die Symbolleiste "Globale Ressourcen" 252 ausgewählt werden (**Ansicht | Anpassen | Symbolleisten | Globale Ressourcen**).

13.9.8 Anpassen

Mit diesem Befehl können Sie die Applikationsmenüs und Symbolleisten an Ihre eigenen Bedürfnisse anpassen. Bei Aufruf des Befehls, wird das Dialogfeld "Anpassen" angezeigt, das die folgenden Register enthält:

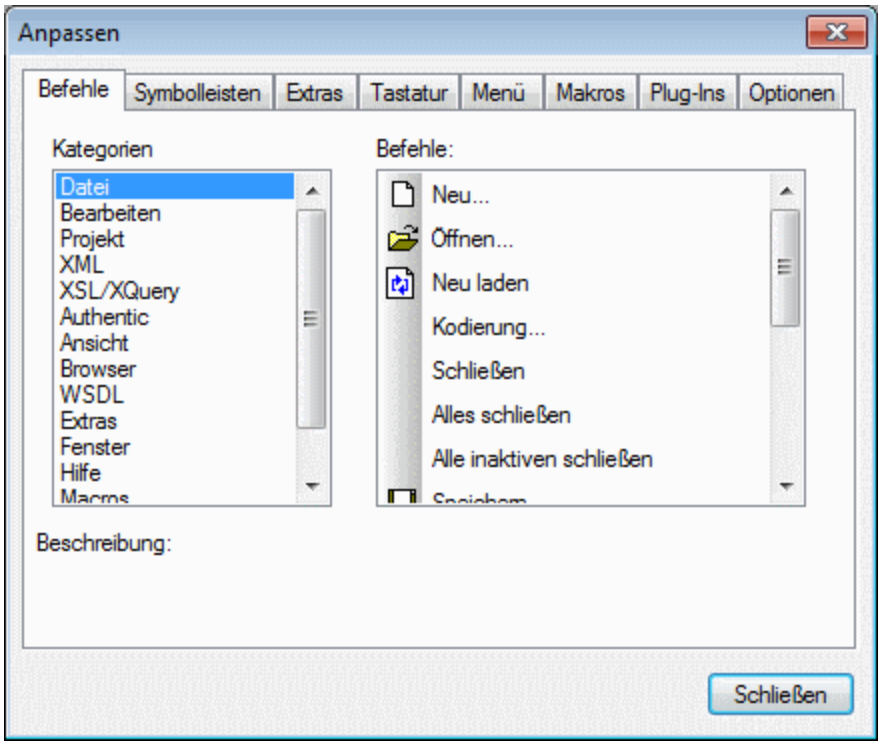
- Befehle 250: Alle Applikations- und Makrobefehle können von diesem Register in Menüleisten, Menüs und Symbolleisten gezogen werden.
- Symbolleisten 252: Die Symbolleisten können einzeln aktiviert, deaktiviert und zurückgesetzt werden.
- Extras 241: Befehle, mit denen externe Programme von der Benutzeroberfläche aus geöffnet werden, können zur Benutzeroberfläche hinzugefügt werden.
- Tastatur 255: Für einzelne Applikationen und Makrobefehle können Tastaturkürzel erstellt werden.
- Menü 258: Auf diesem Register können Menüleisten und Kontextmenüs, die angepasst werden sollen, ausgewählt und aktiviert werden. Dieses Register wird gemeinsam mit dem Register "Befehle" verwendet.
- Makros 259: Makros können hier mit neuen Befehlen verknüpft werden.
- Plug-Ins 262: Plug-Ins können aktiviert und in die Benutzeroberfläche integriert werden.
- Optionen 267: Auf diesem Register werden Anzeigeoptionen für Symbolleisten eingestellt.

In diesem Abschnitt wird auch das Kontextmenü 263 beschrieben, das bei geöffneten Dialogfeld "Anpassen" bei Rechtsklick auf Elemente der Menüleiste, des Menüs oder der Symbolleiste angezeigt wird.

13.9.8.1 Befehle

Auf dem Register Befehle können Sie Ihre Menüs oder Symbolleisten anpassen. Sie können ganz nach Wunsch Applikationsbefehle zu Ihren Menüs und Symbolleisten hinzufügen. Beachten Sie, dass Sie selbst keine neuen Applikationsbefehle oder Menüs erstellen können.

© 2016-2022 Altova GmbH



So fügen Sie einen Befehl zu einer Symbolleiste oder einem Menü hinzu:

1. Öffnen Sie das Dialogfeld über die Menübefehle **Extras | Anpassen.** Daraufhin wird das Dialogfeld "Anpassen" angezeigt.
2. Wählen Sie die Befehlskategorie **Alle Befehle** im Kategorie-Listenfeld. Die verfügbaren Befehle werden im entsprechenden Listenfeld angezeigt.
3. Markieren Sie einen Befehl aus dem Listenfeld, und ziehen Sie ihn in ein existierendes Menü oder eine Symbolleiste.
4. Ein **I**-Zeichen wird angezeigt, wenn Sie den Cursor an eine Stelle bewegen, an die der ausgewählte Befehl gestellt werden kann.
5. Lassen Sie die Maustaste los, um den Befehl an der gewünschten Stelle einzufügen.

Beachten Sie die folgenden Punkte.

- An der Spitze des Mauszeigers wird beim Ziehen eines Befehls eine kleine Schaltfläche angezeigt. Damit wird angezeigt, dass der Befehl gerade mit der Maus gezogen wird
- Ein "**x**" unterhalb des Mauszeigers bedeutet, dass der Befehl nicht an die aktuelle Cursorposition gezogen werden kann.
- Sobald der Cursor an eine Stelle verschoben wird, an die der Befehl gezogen werden kann (eine Symbolleiste oder ein Menü), verschwindet das **x**. Anstelle des **x**-Zeichens erscheint ein **I**-Zeichen.
- Befehle können in Menüs oder Symbolleisten eingefügt werden. Wenn Sie eine eigene Symbolleiste erstellt 252 haben, können Sie nun die gewünschten Befehle einfügen.
- Wenn Sie den Cursor über ein geschlossenes Menü platzieren, wird dieses Menü geöffnet, sodass Sie den Befehl an einer beliebigen Stelle in diesem Menü einfügen können.

Hinzufügen von Befehlen zu Kontextmenüs

Sie können Befehle auch zu Kontextmenüs hinzufügen, indem Sie die Befehle aus dem Listenfeld *Befehle* in das Kontextmenü ziehen. Gehen Sie dazu folgendermaßen vor:

1. Klicken Sie im Dialogfeld "Anpassen" auf das Register 258 **"Menü"** 258.
2. Wählen Sie im Bereich "Kontextmenü" ein Kontextmenü aus der Auswahlliste aus. Daraufhin wird das ausgewählte Kontextmenü angezeigt.
3. Wechseln Sie im Dialogfeld "Anpassen" zurück zum Register "Befehle".
4. Ziehen Sie den gewünschten Befehle aus der Liste *Befehle* an die gewünschte Stelle im Kontextmenü.

Löschen eines Befehls oder Menüs

Um einen Befehl aus einem Menü, einem Kontextmenü (nähere Informationen zum Aufrufen von Kontextmenüs siehe oben) oder einer Symbolleiste zu löschen oder um ein ganzes Menü zu löschen, gehen Sie folgendermaßen vor:

1. Öffnen Sie das Dialogfeld "Anpassen" (**Extras | Anpassen).**
2. Klicken Sie, während das Dialogfeld "Anpassen" geöffnet (und ein beliebiges Register ausgewählt) ist, mit der rechten Maustaste auf ein Menü oder einen Menübefehl und wählen Sie anschließend im Kontextmenü den Befehl **Löschen**. Alternativ dazu können Sie das Menü oder den Menübefehl auch mit der Maus ziehen, bis ein "x"-Symbol unterhalb des Mauszeigers erscheint. Wenn Sie die Maustaste nun loslassen, wird das Menü bzw. der Menübefehl gelöscht.

Um gelöschte Menübefehle wiederherzustellen, gehen Sie vor, wie in diesem Abschnitt beschrieben. Um ein gelöschtes Menü wiederherzustellen, gehen Sie zu **Extras | Anpassen | Menü** und klicken Sie im Bereich *Applikations-Menüs* auf die Schaltfläche **Zurücksetzen**. Alternativ dazu können Sie das Menü auch über **Extras | Anpassen | Symbolleisten**, durch Auswahl von "Menüleiste" und klicken auf die Schaltfläche "Zurücksetzen" zurücksetzen.

13.9.8.2 Symbolleisten

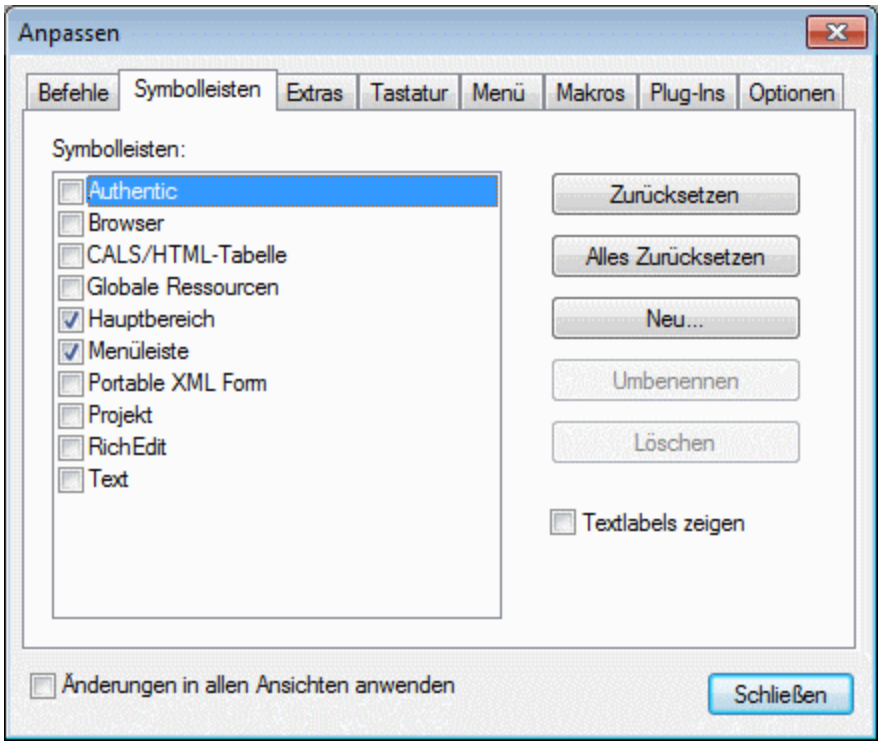
Auf dem Register **Symbolleisten** können Sie (i) spezifische Symbolleisten aktivieren oder deaktivieren, d.h. festlegen, welche davon auf der Benutzeroberfläche angezeigt werden sollen; (ii) festlegen, welche Schaltflächen in jeder Symbolleiste angezeigt werden sollen und (iii) Ihre eigenen Symbolleisten erstellen.

Die Symbolleisten enthalten Schaltflächen für die meistverwendeten Befehle. Zu jeder Schaltfläche werden in einem Tooltip und in der Statusleiste Informationen angezeigt, wenn Sie den Cursor über die Schaltfläche platzieren. Sie können die Symbolleisten aus Ihrer Standardposition an eine andere beliebige Stelle am Bildschirm verschieben, wo diese als frei schwebendes Fenster angezeigt werden.

Anmerkung: Um einen Befehl zu einer Symbolleiste hinzuzufügen, ziehen Sie den gewünschten Befehl aus dem Listenfeld *Befehle* des Registers **Befehle** 250 in die Symbolleiste. Um einen Befehl aus einer Symbolleiste zu löschen, öffnen Sie das Dialogfeld "Anpassen" und ziehen Sie den Befehl mit der Maus aus der Symbolleiste hinaus (näherer Informationen siehe **Befehle** 250).

Anmerkung: Symbolleisteneinstellungen, die in einer bestimmten Ansicht definiert wurden, stehen standardmäßig nur in der jeweiligen Ansicht zur Verfügung. Damit die Einstellungen auf alle Ansichten angewendet werden, aktivieren Sie das Kontrollkästchen am unteren Rand des Dialogfelds.

 © 2016-2022 Altova GmbH



Es stehen die folgenden Funktionalitäten zur Verfügung:

- *Um eine Symbolleiste zu aktivieren oder deaktivieren*: Klicken Sie im Listenfeld *Symbolleisten* auf das jeweilige Kontrollkästchen.

- *Um Änderungen auf alle Ansichten anzuwenden*: Aktivieren Sie das Kontrollkästchen am unteren Rand des Dialogfelds. Andernfalls werden die Änderungen nur auf die aktuelle Ansicht angewendet. Beachten Sie, dass nur Änderungen, die **nach** Aktivierung des Kontrollkästchens vorgenommen wurden, auf alle Ansichten angewendet werden.

- *Um eine neue Symbolleiste hinzuzufügen*: Klicken Sie auf die Schaltfläche **Neu** und geben Sie der neuen Symbolleiste einen Namen (Dialogfeld Symbolleistenname). Ziehen Sie Befehle aus dem Register **Befehle** 250 in die neue Symbolleiste.

- *Um den Namen einer hinzugefügten Symbolleiste zu ändern:* Wählen Sie die hinzugefügte Symbolleiste aus, klicken Sie auf die Schaltfläche **Umbenennen** und bearbeiten Sie den Namen im Dialogfeld "Symbolleistenname".

- *Um die Menüleiste zurückzusetzen:* Klicken Sie im Bereich *Symbolleisten* auf den Menüleisteneintrag und klicken Sie auf die Schaltfläche **Zurücksetzen.** Daraufhin wird die Menüleiste auf die Standardeinstellung zurückgesetzt.

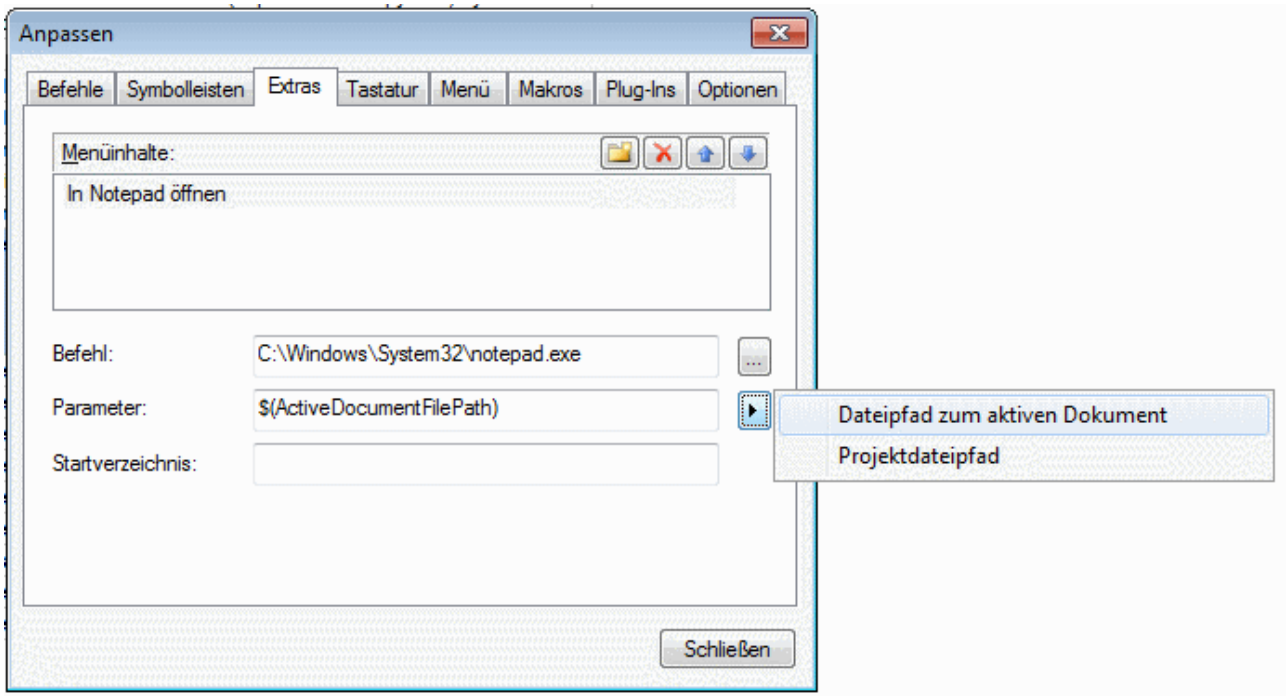
- *Um alle Symbolleisten- und Menübefehle zurückzusetzen:* Klicken Sie auf die Schaltfläche **Alles Zurücksetzen.** Daraufhin werden alle Symbolleisten und Menüs auf die Standardinstallationseinstellung zurückgesetzt.

- *Um eine Symbolleiste zu löschen:* Wählen Sie die gewünschte Symbolleiste im Bereich *Symbolleisten* aus und klicken Sie auf **Löschen**.

- *Um Textlabels von Befehlen in einer bestimmten Symbolleiste anzuzeigen:* Wählen Sie die gewünschte Symbolleiste aus und aktivieren Sie das Kontrollkästchen *Textlabels anzeigen*. Beachten Sie, dass die Textlabels für die einzelnen Symbolleisten separat aktiviert werden müssen.

13.9.8.3 Extras

Auf dem Register **Extras** können Sie Befehle definieren, mit denen Sie von Authentic Desktop aus externe Applikationen verwenden können. Diese Befehle werden zum Menü **Extras | Benutzerdefinierte Tools** hinzugefügt. So kann z.B. die aktive Datei im Hauptfenster von Authentic Desktop durch Auswahl eines Befehls, den Sie im Menü **Extras | Benutzerdefinierte Tools** erstellt haben, in einer externen Applikation wie z.B. Notepad geöffnet werden.



Um einen Befehl zur Verwendung einer externen Applikation zu definieren, gehen Sie folgendermaßen vor:

1. Klicken Sie im Bereich *Menüinhalte* (*siehe Abbildung oben*) in der Titelleiste des Bereichs auf die Schaltfläche "Neu" und geben Sie in der Zeile, die für den Eintrag erstellt wurde, den Namen des gewünschten Menübefehls ein. In der Abbildung oben wurde ein einziger Menübefehl eingegeben, nämlich **In Notepad öffnen**. Mit diesem Befehl soll das aktive Dokument in der externen Applikation Notepad geöffnet werden. Durch Klicken auf die Schaltfläche "Neu" können weitere Befehle zur Befehlsliste hinzugefügt werden. Mit Hilfe der Schaltflächen **Element hinauf** und **Element hinab** können Sie die Befehle in der Liste nach oben oder unten verschieben. Um einen Befehl zu löschen, wählen Sie ihn aus und klicken Sie auf die Schaltfläche **Löschen.**
2. Um eine externe Applikation mit einem Befehl zu verknüpfen, wählen Sie den Befehl im Bereich *Menüinhalte* aus. Geben Sie anschließend im Feld *Befehl* den Pfad zur ausführbaren Datei der externen Applikation ein oder wählen Sie ihn aus. In der Abbildung oben wurde im Feld *Befehl* der Pfad zur Applikation Notepad eingegeben.

 © 2016-2022 Altova GmbH

3. Wenn Sie auf die Flyout-Schaltfläche des Felds *Parameter* klicken (*siehe Abbildung oben)*, werden die Aktionen angezeigt, die für die externe Applikation verfügbar sind. Diese Aktionen sind in der Liste unterhalb davon beschrieben. Wenn Sie eine Aktion auswählen, wird ein Code-String für die Aktion in das Feld *Parameter* eingetragen.
4. Wenn Sie ein aktuelles Arbeitsverzeichnis definieren möchten, geben Sie dieses in das Feld *Startverzeichnis* ein.
5. Klicken Sie auf **Schließen,** um den Vorgang abzuschließen.

Die von Ihnen erstellten Befehle werden im Menü **Extras | Benutzerdefinierte Tools** und im Kontextmenü von Dateien und Ordnern im Projektfenster im Untermenü **Benutzerdefinierte Tools** angezeigt.

Wenn Sie (im Menü **Extras | Benutzerdefinierte Tools**) auf den von Ihnen erstellten Befehl klicken, wird die mit dem Befehl verknüpfte Aktion ausgeführt. In dem in der Abbildung oben gezeigten Beispiel geschieht Folgendes: Das im Hauptfenster von Authentic Desktop aktive Dokument wird in Notepad geöffnet. Der Befehl für die externe Applikation steht auch im Projektfenster im Kontextmenü für die Dateien zur Verfügung (Klicken Sie mit der rechten Maustaste auf eine Datei im Projektfenster, um das Kontextmenü für die Datei anzuzeigen). Sie können über das Projektfenster auch mehrere Dateie öffnen (wenn dies die Applikation erlaubt). Wählen Sie dazu mehrere Dateien aus und klicken Sie anschließend im Kontextmenü auf den entsprechenden Befehl.

Parameter

Im Feld *Parameter* wird die Aktion definiert, die vom Befehl der externen Applikation ausgeführt werden soll. Die folgenden Parameter stehen zur Verfügung.

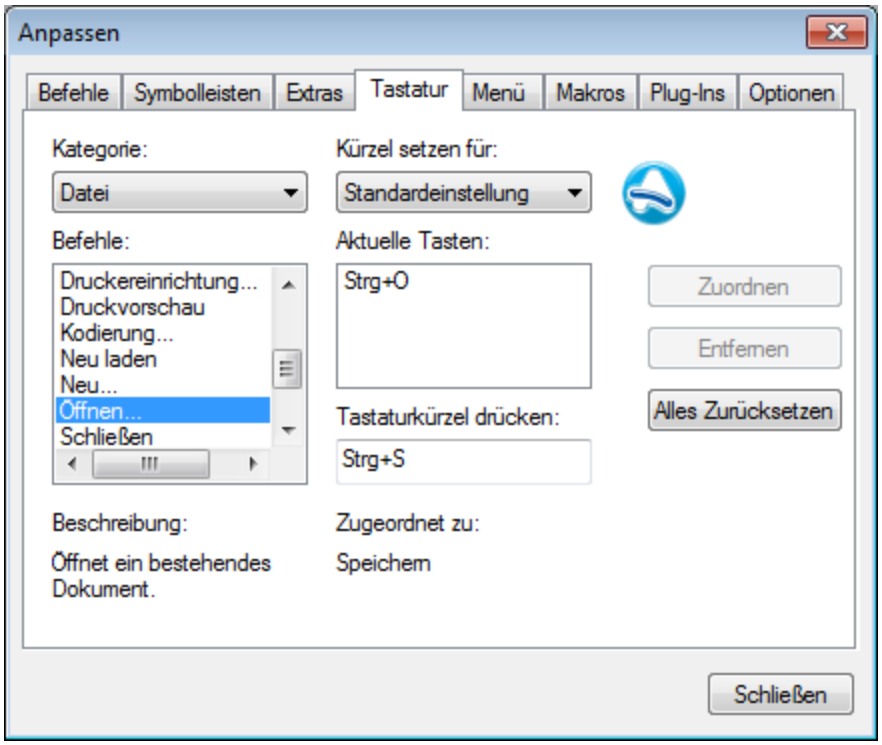
- *Dateipfad zum aktiven Dokument:* Mit dem Befehl im Menü **Extras | Benutzerdefinierte Tools** wird das in Authentic Desktop aktive Dokument in der externen Applikation geöffnet. Der Befehl im Kontextmenü einer Datei im Projektfenster öffnet die ausgewählte Datei in der externen Applikation.
- *Projektdateipfad:* Öffnet die Authentic Desktop Projektdatei (die `.spp` Datei) in der externen Applikation.

Startverzeichnis

Der Eintrag unter *Startverzeichnis* ist optional und ist ein Pfad, der als das aktuelle Verzeichnis verwendet wird.

13.9.8.4 Tastatur

Auf dem Register Tastatur können Sie neue Tastaturkürzel für jeden Applikationsbefehl definieren oder bestehende ändern.



Um einem Befehl ein neues Kürzel zuzuweisen oder ein bestehendes Tastaturkürzel zu ändern, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Auswahlliste die Kategorie *Alle Befehle* aus. Beachten Sie: Wenn ein Makro als verknüpfter Befehl ausgewählt wurde 259, können über die Auswahlliste auch Makros ausgewählt werden und es kann für das Makro ein Tastenkürzel definiert werden.
2. Wählen Sie im Listenfeld *Befehle* den **Befehl** aus, welchem Sie ein neues Kürzel zuweisen möchten, bzw. dessen Kürzel Sie ändern möchten.
3. Klicken Sie in das Textfeld *Tastaturkürzel drücken* und drücken Sie das Tastaturkürzel, das Sie diesem Befehl zuweisen möchten. Das Kürzel wird im Textfeld *Tastaturkürzel drücken* angezeigt. Wenn das Kürzel noch mit keinem Befehl verknüpft ist, ist die Schaltfläche **Zuweisen** aktiv, Wenn das Kürzel schon einem anderen Befehl zugeordnet war, wird dieser Befehl unterhalb des Textfelds angezeigt und die Schaltfläche **Zuweisen** ist deaktiviert. (Um den Eintrag im Feld *Tastaturkürzel drücken* zu löschen, drücken Sie eine der Tasten **Strg, Alt** oder **Umschalt**).
4. Klicken Sie auf die Schaltfläche **Zuweisen,** um das Tastaturkürzel zuzuweisen. Das Kürzel wird nun im Listenfeld *Aktuelle Tasten* angezeigt. Sie können einem einzigen Befehl mehrere Tastenkürzel zuweisen.
5. Klicken Sie auf die Schaltfläche **Schließen**, um die Einstellung zu bestätigen.

Löschen eines Tastaturkürzels

Ein Tastaturkürzel kann nicht mehreren Befehlen zugewiesen werden. Um ein Kürzel zu löschen, klicken Sie in das Listenfeld *Aktuelle Tasten* und anschließend auf die Schaltfläche **Entfernen.**

Kürzel setzen für:

Derzeit können Tastaturkürzel nur als Standard definiert werden. Es steht kein anderer Modus zur Verfügung.

 © 2016-2022 Altova GmbH

Standard-Tastaturkürzel

In der unten stehenden Tabelle sind die Standard-Tastenkürzel häufig verwendeter Befehle aufgelistet. Eine Übersicht über alle Applikationsmenübefehle finden Sie in der Tastenzuordnung (**Hilfe | Tastaturzuordnungen** 288).

Funktionstasten-Tastaturkürzel (einschl. Validierung und Transformation)

F1	Menü "Hilfe"
F1 + Alt	Letzte Datei öffnen
F3	Weitersuchen
F4 + STRG	Aktives Fenster schließen
F4 + Alt	Authentic Desktop schließen
F5	Aktualisieren
F6 + STRG	Offene Fenster der Reihe nach aufrufen
F7	Wohlgeformtheit prüfen
F8	Validieren
F10	XSL-Transformation
F10 + STRG	XSL:FO-Transformation

Datei- und Applikationsbefehle

Alt + F1	Letzte Datei öffnen
STRG + O	Datei öffnen
STRG + N	Datei neu
STRG + P	Datei drucken
STRG + S	Datei speichern
STRG + F4	Aktives Fenster schließen
STRG + F6	Offene Fenster der Reihe nach aufrufen
STRG + TAB	Zwischen offenen Dokumenten wechseln
Alt + F4	Authentic Desktop schließen

Diverse Tastaturkürzel

Pfeil nach oben/nach unten	Cursor oder Auswahlleiste verschieben
Esc	Bearbeitungen verwerfen oder Dialogfeld schließen
Eingabetaste	Auswahl bestätigen
Entf	Zeichen oder Auswahl löschen
Umschalt + Entf	Ausschneiden

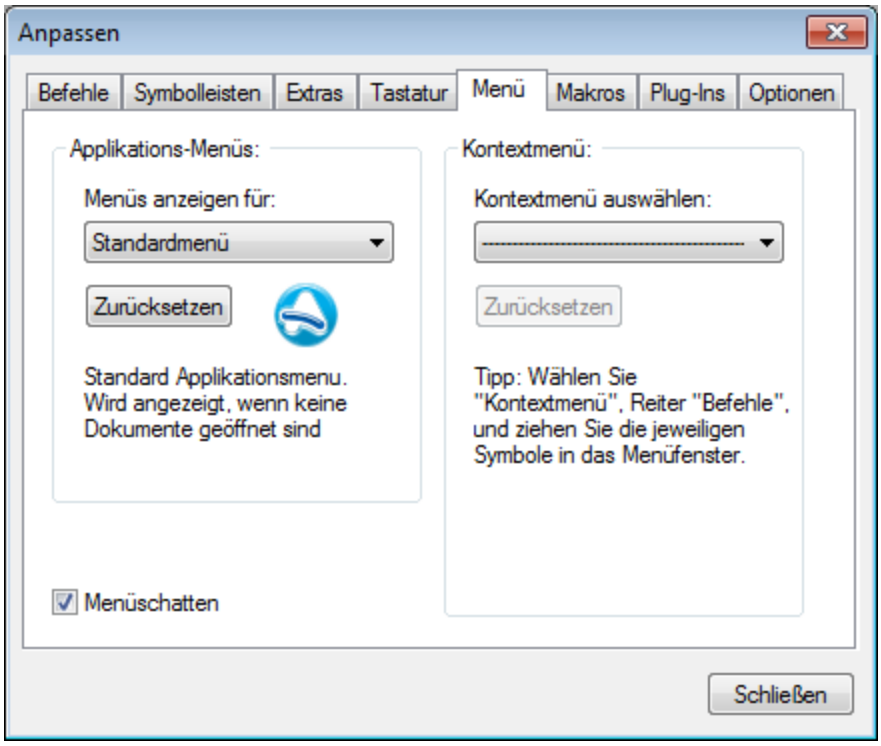
Bearbeitungsbefehle

STRG + A	Alles markieren

STRG + F	Suchen
STRG + G	Gehe zu Zeile/Zeichen
STRG + H	Ersetzen
STRG + V	Einfügen
STRG + X	Ausschneiden
STRG + Y	Wiederherstellen
STRG + Z	Rückgängig machen

13.9.8.5 Menü

Auf dem Register **Menü** können Sie die beiden Hauptmenüleisten (Standardmenüleiste und Applikationsmenüleiste) und die Kontextmenüs der Applikation anpassen.



Anpassen der Standardmenüleiste und der Applikationsmenüleiste

Die Standardmenüleiste ist die Menüleiste, die angezeigt wird, wenn kein Dokument im Hauptfenster geöffnet ist. Die Applikationsmenüleiste ist die Menüleiste, die angezeigt wird, wenn ein oder mehrere Dokumente im Hauptfenster geöffnet sind. Jede Menüleiste kann separat angepasst werden. Änderungen, die an einer der Leisten vorgenommen wurden, haben keine Auswirkung auf andere Menüleisten.

 © 2016-2022 Altova GmbH

Um eine Menüleiste anzupassen, wählen Sie diese in der Auswahlliste *Mernüs anzeigen für* (*siehe Abbildung oben*) aus. Wechseln Sie anschließend zum Register "Befehle" des Dialogfelds "Anpassen" 250 und ziehen Sie die Befehle aus dem Listenfeld "Befehle" in die Menüleiste oder in eines der Menüs.

Löschen von Befehlen aus Menüs und Zurücksetzen der Menüleisten

So **löschen** Sie ein ganzes Menü oder einen Befehl in einem Menü:

1. Wählen Sie im Bereich "Applikations-Menüs" entweder *Standardmenü* (damit werden die Menüs angezeigt, die verfügbar sind, wenn kein Dokument offen ist) oder *Authentic* (damit werden die Menüs angezeigt, die verfügbar sind, wenn ein oder mehrere Dokumente offen sind) aus.
2. Während das Dialogfeld "Anpassen" geöffnet ist, (i) wählen Sie das Menü, das Sie aus der Menüleiste der Applikation löschen möchten, aus oder (ii) wählen Sie den Befehl aus, den Sie aus einem dieser Menüs löschen möchten.
3. Ziehen Sie entweder das Menü aus der Menüleiste oder den Menübefehl aus dem Menü oder (ii) klicken Sie mit der rechten Maustaste auf das Menü oder den Menübefehl und wählen Sie den Befehl **Löschen**.

Sie können jede der beiden Menüleisten (die Standardmenüleiste und die Applikationsmenüleiste) in ihren Originalzustand (Zustand bei der Installation) **zurücksetzen**. Wählen Sie dazu das Menü in der Auswahlliste *Menüs anzeigen für* aus und klicken Sie anschließend unterhalb der Auswahlliste auf den Befehl **Zurücksetzen**.

Anpassen der Kontextmenüs der Applikation

Die Kontextmenüs sind die Menüs, die angezeigt werden, wenn Sie mit der rechten Maustaste auf bestimmte Objekte auf der Benutzeroberfläche der Applikation klicken. Jedes dieser Kontextmenüs kann folgendermaßen angepasst werden:

1. Wählen Sie das gewünschte Kontextmenü in der Auswahlliste *Kontextmenü auswählen* aus. Daraufhin wird das Kontextmenü angezeigt.
2. Wechseln Sie auf das Register Befehle des Dialogfelds "Anpassen" 250.
3. Ziehen Sie den gewünschten Befehl aus dem Listenfeld *Befehle* in das Kontextmenü.
4. Wenn Sie einen Befehl aus dem Kontextmenü löschen möchten, klicken Sie mit der rechten Maustaste auf diesen Befehl im Kontextmenü und wählen Sie den Befehl **Löschen**. Alternativ dazu können Sie den zu löschenden Befehl auch mit der Maus aus dem Kontextmenü herausziehen.

Sie können jedes Kontextmenü in den Originalzustand zurücksetzen. Wählen Sie dazu das Kontextmenü in der Auswahlliste *Kontextmenüs anzeigen* aus und klicken Sie anschließend unterhalb der Auswahlliste auf den Befehl **Zurücksetzen**.

Menüschatten

Klicken Sie auf das Kontrollkästchen *Menüschatten*, wenn Menüs mit Schatten dargestellt werden sollen.

13.9.8.6 Makros

Über das Register **Makros** können Sie Applikationsbefehle für mit dem Skript-Editor von Authentic Desktop erstellte Makros erstellen. Diese Applikationsbefehle (mit denen die damit verknüpften Makros ausgeführt werden), können anschließend in Menüs und Symbolleisten zur Verfügung gestellt werden - entweder direkt

über das Register "Makros" oder über den Mechanismus auf dem Register Befehle des Dialogfelds "Anpassen"[250]. Als Applikationsbefehlen können diesen Makros über das Register "Tastatur" des Dialogfelds "Anpassen"[255] auch Tastaturkürzel zugewiesen werden.

Funktionsweise von Makros in Authentic Desktop

Makros funktionieren in Authentic Desktop folgendermaßen:

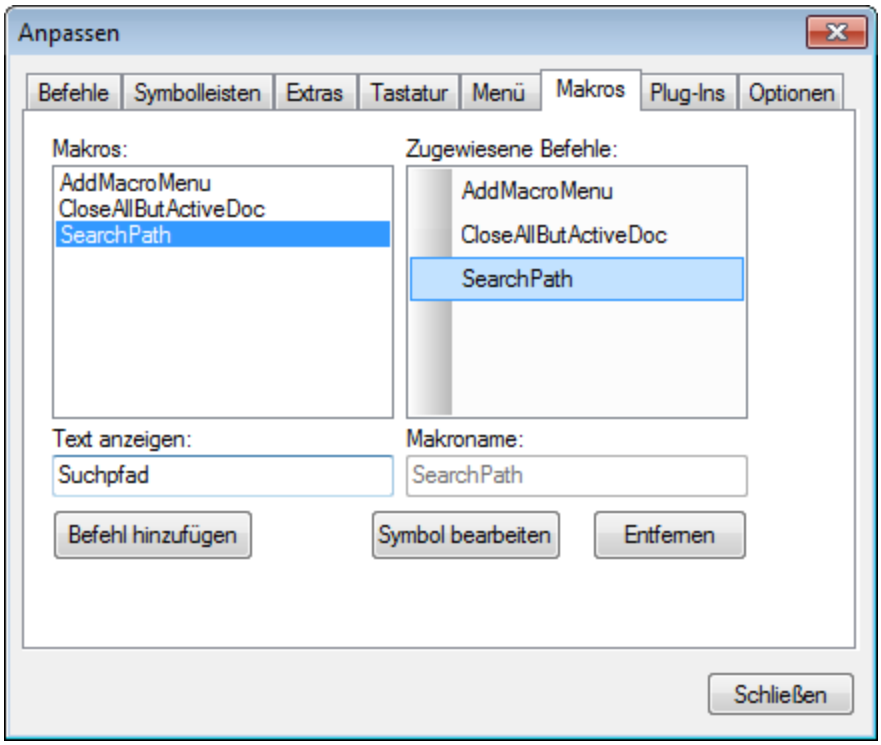
- Altova Skripting-Projekte (`.asprj` Dateien) werden im Skript-Editor[297] von Authentic Desktop erstellt. In diesen Skripting-Projekten sind die in Authentic Desktop verwendeten Makros enthalten.
- In Authentic Desktop können gleichzeitig zwei Skripting-Projekte aktiv sein: (i) ein Applikations-Skripting-Projekt, das im Dialogfeld "Optionen" im Abschnitt "Skripting"[280] definiert wird, und (ii) das Skripting-Projekt des aktiven Authentic Desktop-Projekts[17], das im Dialogfeld "Skript-Einstellungen"[215] (**Projekt | Skript-Einstellungen**[215]) definiert wird.
- Die Makros in diesen beiden Skripting-Projekten stehen in der Applikation im Menü **Extras | Makros** (über das die Makros ausgeführt werden können) und im Dialogfeld "Anpassen" auf dem Register "Makros" (*Abbildung unten*), über das sie als Applikationsbefehle definiert werden können, zur Verfügung. Nachdem ein Makro als Applikationsbefehl definiert wurde, kann der Befehl in ein Menü und/oder eine Symbolleiste platziert werden.

Erstellen eines Applikationsbefehls für ein Makro

Erstellen Sie zuerst im Skript-Editor[297] (**Extras | Skript-Editor**[297]) das gewünschte Makro und speichern Sie es in einem Skripting-Projekt. Definieren Sie diese Datei entweder als das Applikations-Skripting-Projekt (über das Dialogfeld "Optionen", Register "Skripting")[280] oder als Skripting-Projekt des aktiven Applikationsprojekts (über das Dialogfeld "Skript-Einstellungen"[215] (**Projekt | Skript-Einstellungen**[215])). Die Makros im Skripting-Projekt werden nun auf dem Register "Makros" im Bereich *Makros* angezeigt (*siehe Abbildung unten)*

Um einen Applikationsbefehl für ein Makro zu erstellen, wählen Sie das Makro im Bereich *Makros* aus, definieren Sie im Textfeld *Text anzeigen* den Text und klicken Sie auf **Befehl hinzufügen** *(siehe Abbildung unten*). Daraufhin wird ein mit dem ausgewählten Makro verknüpfter Befehl zum Listenfeld *Zugewiesene Befehle* hinzugefügt.

 © 2016-2022 Altova GmbH



Um das Symbol eines zuwiesenen Befehls zu bearbeiten, wählen Sie den Befehl aus und klicken Sie auf die Schaltfläche **Symbol bearbeiten**. Um einen zugewiesenen Befehl zu löschen, klicken Sie auf die Schaltfläche **Entfernen.**

Platzieren eines mit einem Marko verknüpften Befehls in ein Menü oder eine Symbolleiste

Es gibt zwei Möglichkeiten, um einen mit einem Makro verknüpften Befehl in ein Menü oder eine Symbolleiste zu platzieren:

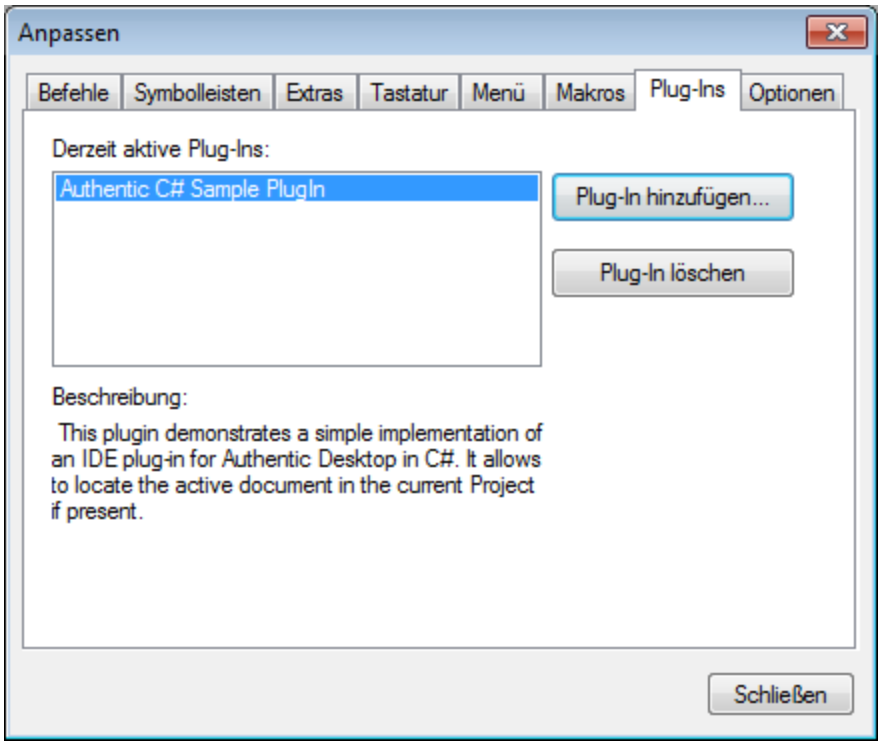
- Ziehen Sie den Befehl aus dem Listenfeld "Zugewiesene Befehle" an die gewünschte Stelle im Menü oder in der Symbolleiste.
- Verwenden Sie die Mechanismen, die im Dialogfeld "Anpassen" auf dem Register "Befehle" 250 zur Verfügung stehen.

In beiden Fällen wird der Befehl an der gewünschten Stelle erstellt. Wenn Sie im Menü doer in der Symbolleiste auf den Befehl klicken, wird das Makro ausgeführt.

Anmerkung: Wenn ein Makro als "zugewiesener Befehl" ausgewählt wurde, können Sie dafür ein Tastaturkürzel definieren 255. Wählen Sie auf dem Register "Tastatur" des Dialogfelds "Anpassen" 255 in der Auswahlliste *Kategorie* die Kategorie *Makros* aus, klicken Sie auf das gewünschte Makro und definieren Sie das Tastenkürzel. Sie müssen ein Makro zuerst als "zugewiesenen Befehl" definieren, damit für Makros Tastenkürzel definiert werden können.

13.9.8.7 Plug-Ins

Auf dem Register **Plug-Ins** können Sie Plug-Inszu integrieren und entsprechend programmierte Befehle in ein Applikationsmenü und/oder eine Symbolleiste einzufügen. Klicken Sie auf dem Register "Plug-In" (*Abbildung unten*) auf **Plug-In hinzufügen** und navigieren Sie zur DLL-Datei des Plug-In (*siehe "Erstellen von Plug-ins" weiter unten*). Klicken Sie auf **OK**, um das Plug-In hinzuzufügen. Es können auch mehrere Plug-Ins hinzugefügt werden.



Nachdem ein Plug-In erfolgreich hinzugefügt worden ist, erscheint eine Beschreibung des Plug-In im Dialogfeld und die Schaltfläche **Plug-In löschen** wird aktiv. Wenn mit dem Plug-In Code Symbolleisten und Menüs erstellt werden, sind diese sofort auf der Benutzeroberfläche zu sehen. Um ein Plug-In zu entfernen, klicken Sie auf **Plug-In löschen**.

Erstellen von Plug-Ins

Im Ordner (Eigene) Dateien [11]: `Examples\IDEPlugin` der Applikation finden Sie Quellcode für Beispiel-Plug-Ins. Um ein Plug-in anhand von solchem Quellcode zu erstellen, gehen Sie folgendermaßen vor:

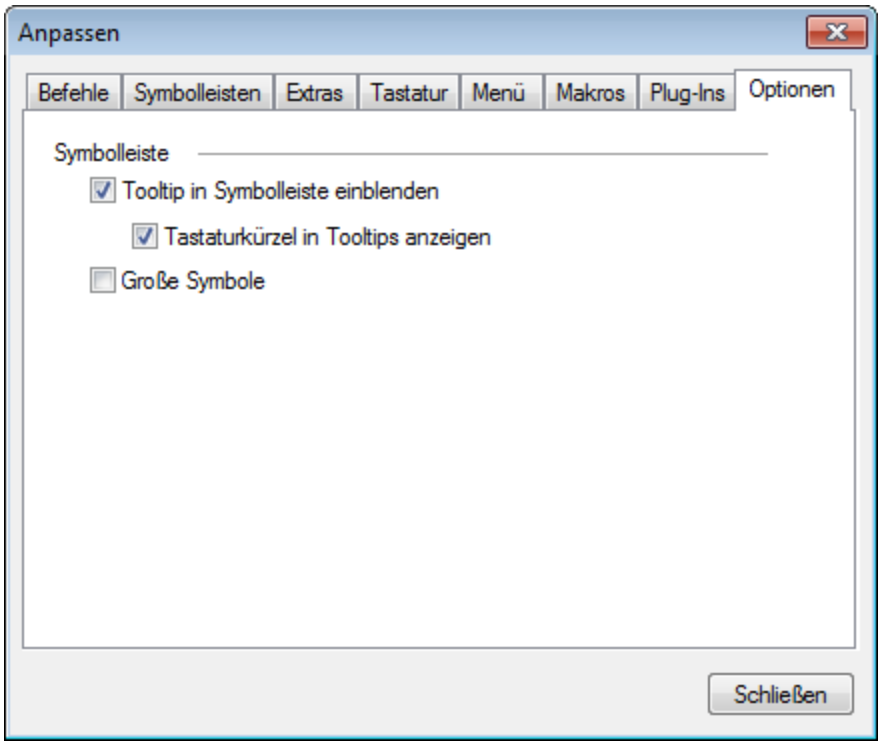
1. Öffnen Sie die Lösung, die Sie als Plug-in in Visual Studio erstellen möchten.
2. Erstellen Sie das Plug-in mit dme Befehl im Menü "Build".
3. Die DLL-Datei des Plug-in wird im Ordner `Bin` oder `Debug` erstellt. Diese DLL ist die Datei, die als Plug-in hinzugefügt werden muss (*siehe oben*).

Nähere Informationen zum Erstellen von Plug-Ins finden Sie im Abschnitt IDE Plug-Ins [325].

 © 2016-2022 Altova GmbH

13.9.8.8 Optionen

Auf dem Register Optionen können Sie allgemeine Einstellungen für die Umgebung vornehmen.

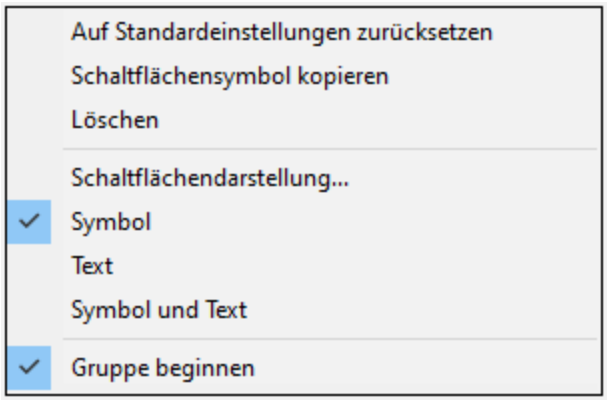


Klicken Sie auf die Kontrollkästchen, um die folgenden Optionen zu aktivieren bzw. zu deaktivieren.

- *Tooltip in Symbolleiste einblenden:* Wenn diese Option aktiviert ist, wird ein Popup beim Ziehen des Mauszeigers über das entsprechende Symbol in der Symbolleiste angezeigt. Das Popup enthält eine Kurzbeschreibung der Funktion, sowie, falls vorhanden und falls die Option *Tastaturkürzel in Tooltips anzeigen* aktiviert ist, ein Tastaturkürzel.
- *Tastaturkürzel in Tooltips anzeigen:* Gibt an, ob Tastaturkürzel in Tooltips angezeigt werden sollen.
- *Große Symbole:* Damit können Sie bei der Größe der angezeigten Symbole zwischen "Standard" und "Groß" wechseln.

13.9.8.9 Kontextmenü 'Anpassen'

Das **Kontextmenü "Anpassen"** (*Abbildung unten*) wird angezeigt, wenn Sie bei geöffnetem Dialogfeld "Anpassen" mit der rechten Maustaste auf ein Applikationsmenü, einen Menübefehl oder eine Symbolleisten-Schaltfläche klicken.



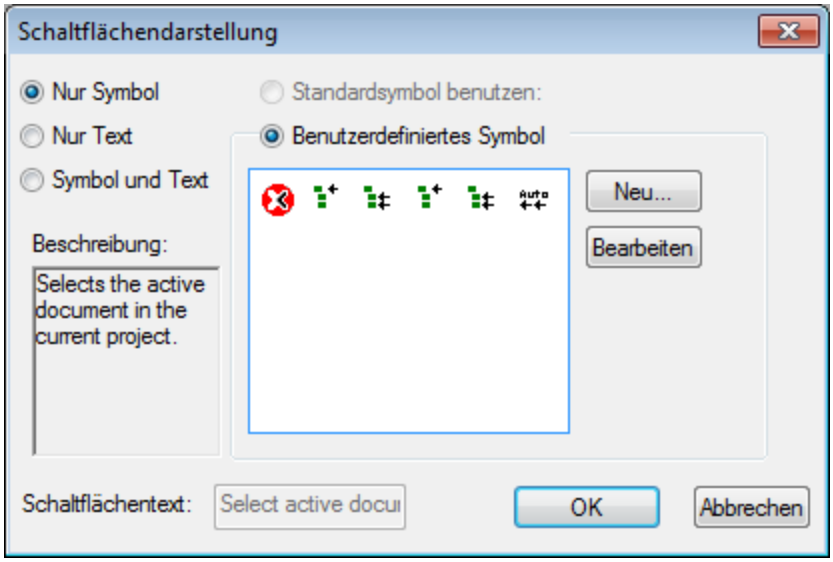
Es stehen die folgenden Funktionalitäten zur Verfügung:

- *Auf Standardeinstellungen zurücksetzen*: Derzeit keine Funktion.
- *Schaltflächensymbol kopieren:* Kopiert das Symbol, auf das Sie mit der rechten Maustaste klicken, in die Zwischenablage.
- *Löschen*: Löscht das ausgewählte Menü, den ausgewählten Menübefehl oder die ausgewählte Symbolleisten-Schaltfläche. Informationen, wie man gelöschte Elemente wiederherstellt, finden Sie weiter unten.
- *Schaltflächendarstellung*: Damit wird das Dialogfeld "Schaltflächendarstellung" (*siehe Abbildung unten*) aufgerufen, in dem Sie die Eigenschaften definieren können, mit denen das Aussehen der ausgewählten Symbolleisten-Schaltfläche definiert wird. Nähere Informationen dazu finden Sie in der Beschreibung unten.
- *Nur Symbol, Nur Text, Symbol und Text:* Einander gegenseitig ausschließende Optionen, mit denen festgelegt wird, ob die ausgewählte Symbolleisten-Schaltfläche nur als Symbol, nur als Text oder als Symbol und Text angezeigt werden soll. Wählen Sie eine dieser Opionen aus, um die Änderung vorzunehmen. Alternativ dazu können Sie diese Änderung auch im Dialogfeld "Schaltflächendarstellung" vornehmen.
- *Gruppe beginnen:* Fügt links von der ausgewählten Symbolleisten-Schaltfläche eine vertikale Trennlinie für die Gruppe ein. Dadurch wird die ausgewählte Symbolleisten-Schaltfläche zur ersten in einer Gruppe von Schaltflächen.

Das Dialogfeld "Schaltflächendarstellung"

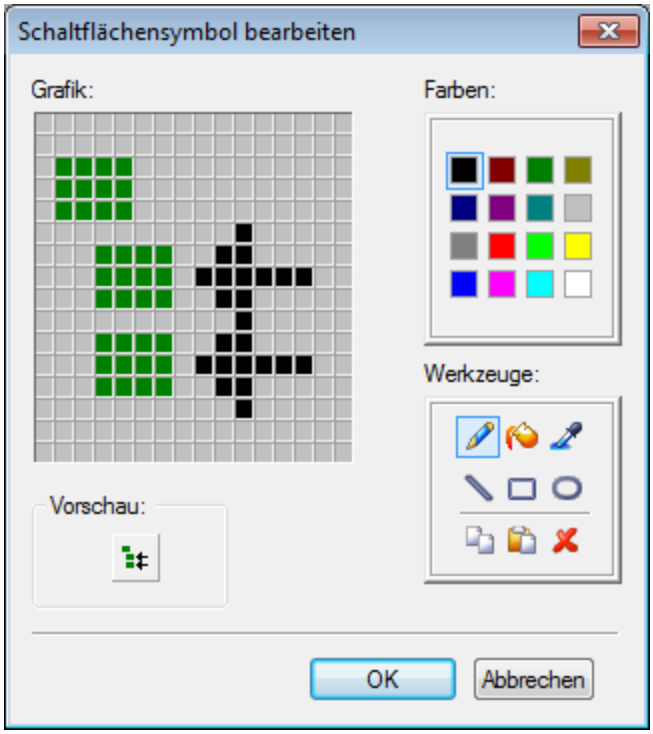
Klicken Sie mit der rechten Maustaste auf eine Symbolleisten-Schaltfläche und wählen Sie den Befehl **Schaltflächendarstellung** um das gleichnamige Dialogfeld aufzurufen (*Abbildung unten*). Über dieses Dialogfeld können Sie das Symbol der Symbolleisten-Schaltfläche sowie seinen Text bearbeiten. Zur Zeit können nur Symbolleisten-Schaltflächen für Makros und solche von Plug-Ins über dieses Dialogfeld bearbeitet werden.

 © 2016-2022 Altova GmbH



Für die ausgewählte Symbolleisten-Schaltfläche (für die durch Rechtsklick das Kontextmenü aufgerufen wurde) stehen die folgenden Bearbeitungsmöglichkeiten zur Verfügung:

- *Nur Symbol, Nur Text, Symbol und Text:* Aktivieren Sie das gewünschte Optionsfeld, um festzulegen, wie die Symbolleisten-Schaltfläche aussehen soll.
- *Bearbeitung des Symbols:* Bei Auswahl der Option *Nur Symbol* oder *Symbol und Text* stehen die Optionen zur Bildbearbeitung zur Verfügung. Klicken Sie auf **Neu,** um ein neues Symbol zu erstellen, das zu den benutzerdefinierten Symbolen im Bereich "Benutzerdefinierte Symbole" hinzugefügt wird. Wählen Sie ein Symbol aus und klicken Sie auf **Bearbeiten** um es zu bearbeiten.



- *Auswahl des Symbols:* Wählen Sie im Bereich der Symbole ein Symbol aus und klicken Sie auf OK, um als ausgewählte Symbol als Symbolleisten-Schaltfläche zu verwenden.
- *Textbearbeitung und Auswahl:* Bei Auswahl der Option *Nur Text* oder *Symbol und Text* wird das Textfeld *Schaltflächentext* aktiviert. Geben Sie den Text ein oder bearbeiten Sie ihn und klicken Sie auf **OK,** um diesen Text als Beschriftung der Symbolleisten-Schaltfläche anzuzeigen.

Anmerkung: Über das Dialogfeld "Schaltflächendarstellung" kann auch der Text von Menübefehlen bearbeitet werden. Klicken Sie (während das Dialogfeld "Anpassen" geöffnet ist) mit der rechten Maustaste auf den Menübefehl, klicken Sie auf **Schaltflächendarstellung** und bearbeiten Sie anschließend den Text des Menübefehls im Textfeld *Schaltflächentext.*

Wiederherstellen von gelöschten Menüs, Menübefehlen und Symbolleisten-Schaltflächen

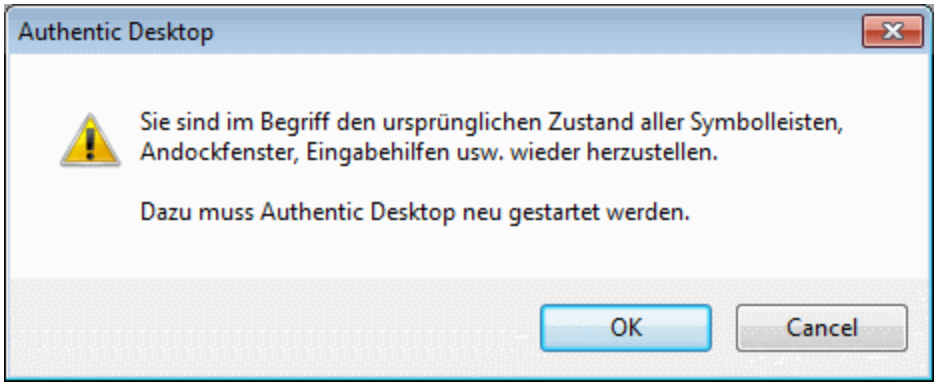
Wenn ein Menü, ein Menübefehl oder eine Symbolleisten-Schaltfläche mit dem Befehl **Löschen** (Kontextmenü "Anpassen") gelöscht wurde, kann der Befehl folgendermaßen wiederhergestellt werden:

- *Menüs:* Gehen Sie zu **Extras | Anpassen | Menü** 258 und klicken Sie im Bereich *Applikationsmenüs* auf die Schaltfläche **Zurücksetzen.** Gehen Sie alternativ dazu zu **Extras | Anpassen | Symbolleisten** 252, wählen Sie die Menüleiste aus und klicken Sie auf die Schaltfläche **Zurücksetzen.-**
- *Menübefehle:* Gehen Sie zu **Extras | Anpassen | Befehle** 250 und ziehen Sie den Befehl aus dem Listenfeld "Befehle" in das Menü.
- *Symbolleisten-Schaltflächen:* Gehen Sie zu **Extras | Anpassen | Befehle** 250 und ziehen Sie den Befehl aus dem Listenfeld "Befehle" in die Symbolleiste.

 © 2016-2022 Altova GmbH

13.9.9 Symbolleisten und Fenster wiederherstellen

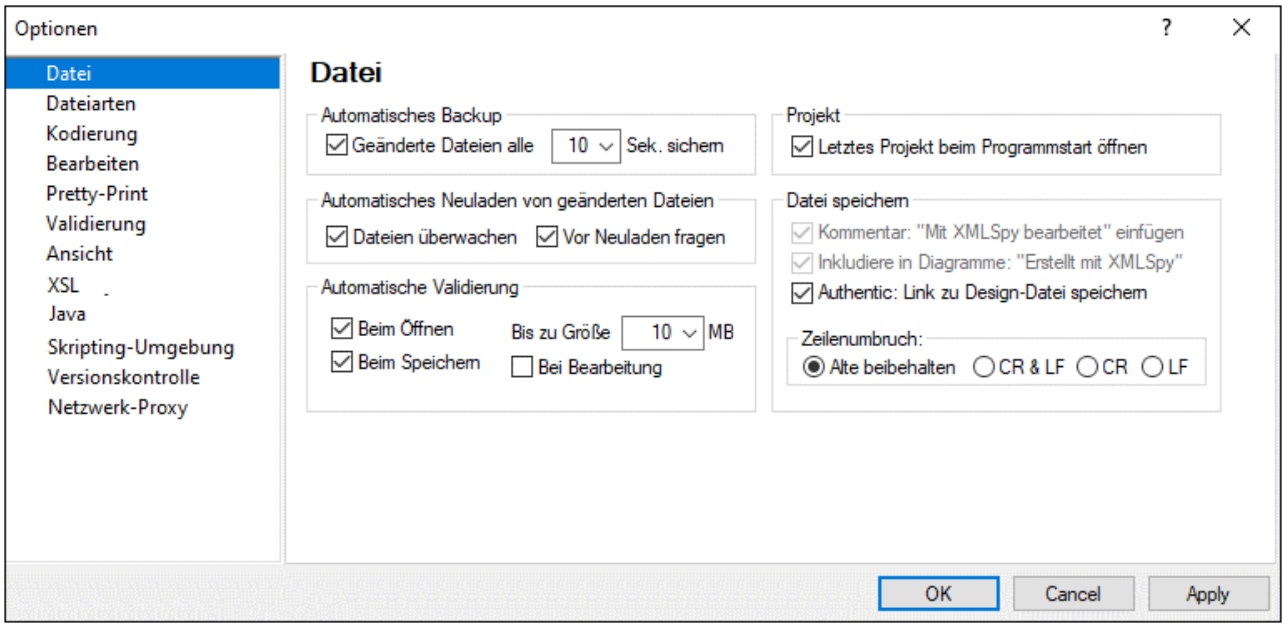
Mit diesem Befehl wird Authentic Desktop geschlossen und mit den Standardeinstellungen neu gestartet. Vor dem Schließen wird ein Dialogfeld angezeigt, in dem Sie gebeten werden, den Befehl zu bestätigen (*Abbildung unten*).



Dieser Befehl ist hilfreich, wenn Sie Symbolleisten oder Fenster verschoben oder ausgeblendet oder deren Größe angepasst haben und nun alle Originaleinstellungen wiederherstellen möchten.

13.9.10 Optionen

Mit dem Befehl **Extras | Optionen** können Sie globale Programmeinstellungen vornehmen. Diese Einstellungen sind in Abschnitte gegliedert (*siehe linker Bereich in der Abbildung unten*). So enthält etwa der Abschnitt "Datei" 268 (*siehe Abb. unten)* Optionen, mit denen Sie festlegen, wie Authentic Desktop Dateien öffnen und speichern soll. Um die Optionen eines bestimmten Abschnitts zu definieren, wählen Sie diesen Abschnitt im linken Bereich aus und definieren Sie die gewünschten Eigenschaftswerte. Durch Klicken auf die Schaltfläche **OK** speichern Sie die Änderungen in der Registry und schließen das Dialogfeld. Durch Klicken auf die Schaltfläche **Übernehmen** werden die Änderungen in den derzeit offenen Dokumenten angezeigt.

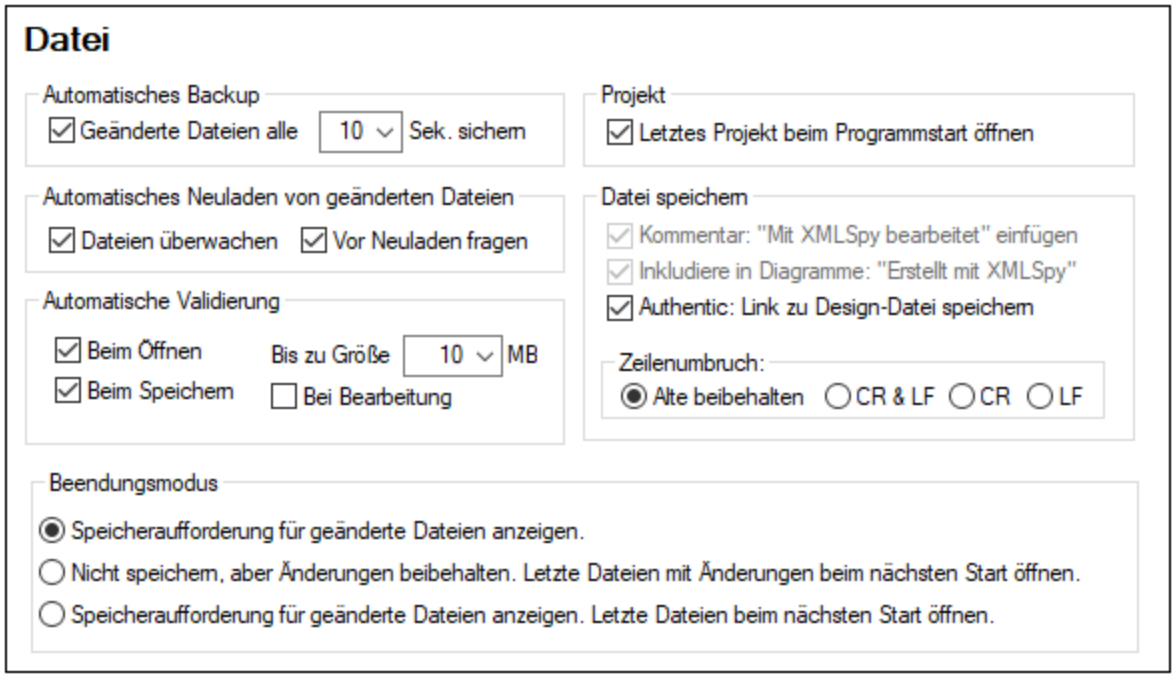


Die einzelnen Abschnitte des Dialogfelds "Optionen" werden in den Unterabschnitten dieses Abschnitts näher beschrieben.

13.9.10.1 Datei

Im Abschnitt "**Datei**" wird die Art definiert, wie Authentic Desktop Ihre XML-Dokumente öffnet und speichert. Weitere Einstellungen hierzu finden Sie im Abschnitt Kodierung 272.

 © 2016-2022 Altova GmbH



Automatisches Backup

Gerade in Bearbeitung befindliche Dateien werden automatisch gesichert, wenn diese Option aktiviert ist. Sie können aus der Auswahlliste ein Backup-Intervall zwischen 5 und 60 Sekunden auswählen oder einen benutzerdefinierten Wert bis zu 300 Sekunden eingeben. Nähere Informationen dazu finden Sie im Abschnitt *Automatisches Backup von Dateien* 59.

Automatisches Neuladen von geänderten Dateien

Wenn Sie in einer Multi-User Umgebung, oder mit Dateien arbeiten, die von Servern dynamisch erstellt werden, können Sie die Änderungen während des Bearbeitens direkt verfolgen. Jedes Mal, wenn Authentic Desktop eine Änderung in einem offenen Dokument bemerkt, werden Sie gefragt, ob die geänderte Datei neu geladen werden soll.

Automatische Validierung

Wenn Sie DTDs oder XML-Schemas zur Definition der Struktur Ihrer XML-Dokumente verwenden, können Sie Ihre Instanzdokumente in den folgenden Situationen automatisch validieren:

- beim Öffnen der Datei, wenn die Datei unter einer von Ihnen in MB definierten Größe ist
- beim Speichern der Datei
- beim Bearbeiten der Datei

Wenn das Dokument nicht gültig ist, wird eine Fehlermeldung angezeigt. Wenn es gültig ist, wird keine Meldung angezeigt und die Operation wird ohne weitere Benachrichtigung fortgesetzt.

Projekt

Beim Starten von Authentic Desktop können Sie das zuletzt verwendete Projekt automatisch anzeigen lassen.

Datei speichern

Beim Speichern eines XML-Dokuments fügt Authentic Desktop im oberen Bereich des Dokuments den kurzen Kommentar `<!-- Mit Authentic Desktop bearbeitet http://www.altova.com -->` ein. Dieser Kommentar kann nur von lizenzierten Benutzern deaktiviert werden und wird beim Bearbeiten oder Speichern von Dateien in der Grid-Ansicht oder der Schema-Ansicht verwendet.

Wenn ein StyleVision Power Stylesheet mit einer XML-Datei verknüpft ist, wird der Link zum StyleVision Power Stylesheet bei Auswahl der Option 'Authentic: Link in Design-Datei speichern' mit der XML-Datei gespeichert.

Zeilenumbrüche

Wenn Sie eine Datei öffnen, wird die Zeichenkodierung für Zeilenumbrüche beibehalten, wenn die Option **Alte beibehalten** aktiviert ist. Alternativ dazu können Sie festlegen, dass Zeilenumbrüche in einer dieser drei Kodierungsarten kodiert werden: **CR&LF** (für PC), **CR** (für MacOS) oder **LF** (für Unix).

Beendungsmodus

Mit diesen Optionen legen Sie fest, wie Dateien, die bei der Beendung von Authentic Desktop geöffnet sind, behandelt werden sollen. Es stehen die folgenden Optionen zur Verfügung:

- *Speicheraufforderung für geänderte Dateien anzeigen*: Wenn eine offene Datei nicht gespeicherte Änderungen enthält, wird eine Meldung angezeigt, in der Sie gefragt werden, ob Sie die Dateiänderungen speichern möchten. Je nach Antwort wird die Datei entweder gespeichert oder nicht gespeichert, bevor das Programm anschließend beendet wird.
- *Speicheraufforderung für geänderte Dateien anzeigen. Letzte Dateien beim nächsten Start öffnen*: Für offene Dateien, die nicht gespeicherte Änderungen enthalten, wird das Speicherdialogfeld angezeigt. Der Benutzer kann eine oder mehrere geänderte Dateien speichern oder diese nicht speichern. Beim Neustart des Programms werden alle bei der Beendung des Programms geöffneten Dateien erneut geöffnet. (Wenn Änderungen nicht gespeichert wurden, gehen sie verloren.)
- *Nicht speichern, aber Änderungen beibehalten. Letzte Dateien mit Änderungen beim nächsten Start öffnen*: Das Programm wird sofort beendet, ohne dass nicht gespeicherte Änderungen gespeichert werden. Beim Neustart des Programms werden alle bei der Beendung des Programms geöffneten Dateien geöffnet und enthalten die nicht gespeicherten Änderungen und Sie können dort weiterarbeiten, wo sie zuletzt aufgehört haben.

Wenn Sie das Programm zum ersten Mal beenden, werden Ihnen die Beendungsmodusoptionen angezeigt, sodass Sie das gewünschte Verhalten auswählen können. Diese Option stehen im Dialogfeld "Optionen" im Abschnitt "Datei" zur Verfügung.

Speichern und Beenden

Klicken Sie zum Abschluss auf **OK**.

© 2016-2022 Altova GmbH

13.9.10.2 Dateiarten

Im Abschnitt **Dateiarten** (*Abbildung unten*) können Sie für Authentic Desktop verschiedene Einstellungen für die einzelnen Dateiarten festlegen. Wählen Sie eine Dateiart im Listenfeld "Dateiarten" aus und passen Sie anschließend die Funktionen für die ausgewählte Dateiart, wie unten beschrieben, an. Beachten Sie, dass die Liste "Dateiarten" zwei Sondereinträge enthält:

- <*Standard*>: Damit können Sie festlegen, wie Dateien mit einer Erweiterung, die nicht in der Liste der Dateiarten enthalten ist, behandelt werden sollen.
- <*keine*>: Damit können Sie festlegen, wie Dateien, die keine Erweiterung haben, behandelt werden sollen.

Windows Explorer-Einstellungen

Sie können die Beschreibung der Dateiart und den MIME-konformen Inhaltstyp, der von Windows Explorer verwendet wird, ändern. Weiters können Sie definieren, ob Authentic Desktop als der standardmäßige Editor für diese Dateiart verwendet werden soll.

Konformität

Authentic Desktop bietet für verschiedene Dateiarten spezifische intelligente Editierfunktionen und andere Funktionen. Die Funktionalitäten für eine bestimmte Dateiart werden von Authentic Desktop auf Basis der in dieser Option von Ihnen gewählten Konformität konfiguriert. Vielen Dateiarten ist bereits eine für den jeweiligen Typ entsprechende Standardkonformität zugewiesen. Wir empfehlen Ihnen, diese Einstellungen nicht zu ändern, außer Sie fügen einen neuen Dateityp hinzu oder Sie möchten die Art der Konformität bei einem bestimmten Dateityp absichtlich ändern.

Standardansicht

In dieser Gruppe definieren Sie, welche Standardansicht für welche Dateiart verwendet werden soll.

Textansicht

Über das Kontrollkästchen *Syntaxfärbung* können Sie die Syntaxfärbung für bestimmte Dateiarten bestimmen.

Automatische Validierung deaktivieren

Mit dieser Option können Sie für bestimmte Dateiarten die automatische Validierung deaktivieren. Die automatische Validierung wird normalerweise beim Öffnen oder Speichern einer Datei oder beim Wechseln der Ansicht durchgeführt.

Neue Dateierweiterung hinzufügen

Fügt eine neue Dateiart zur Liste der Dateiarten hinzu. Sie müssen anschließend die Einstellungen für diesen neuen Dateityp über die Optionen auf diesem Register definieren.

Markierte Dateierweiterung löschen

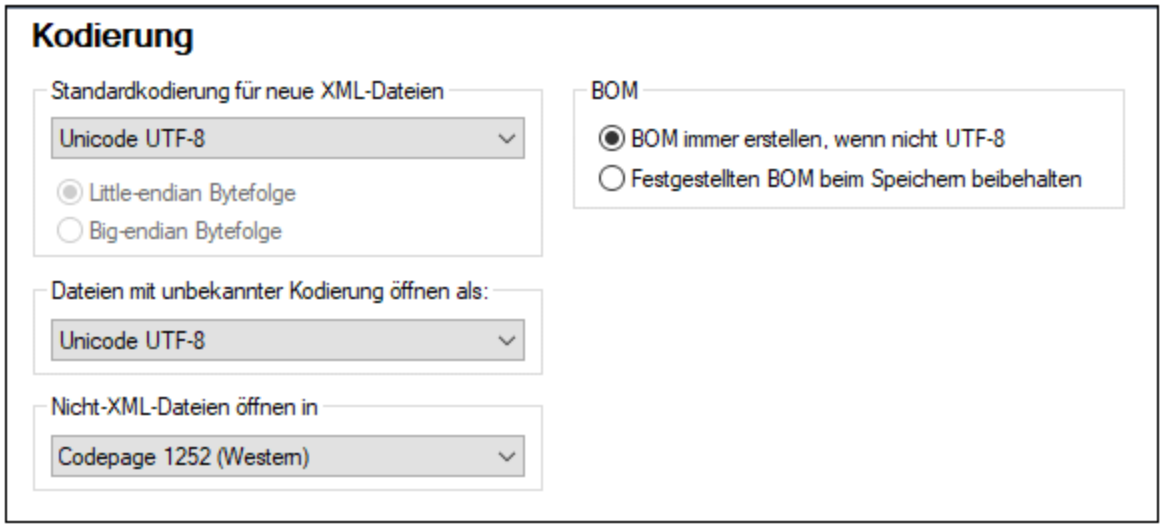
Löscht die aktuell ausgewählte Dateiart sowie alle damit in Zusammenhang stehenden Einstellungen.

Speichern und Beenden

Klicken Sie zum Abschluss auf **OK**.

13.9.10.3 Kodierung

Im Abschnitt **Kodierung** definieren Sie die Einstellungen für die Dateikodierung.



Standardkodierung für neue XML-Dateien

Die Standardkodierung für neue Dateien kann im Dialogfeld "Optionen" durch Auswahl einer Option aus der Dropdown-Liste definiert werden, sodass jedes neue Dokument automatisch mit der entsprechenden XML-Deklaration und der hier gewählten Kodierungsspezifikation erstellt wird. Wird eine 2- oder 4-Byte-Kodierung als standardmäßige Kodierung ausgewählt (z.B. UTF-16, UCS-2 oder UCS-4) können Sie weiters wählen, ob die Byteordnung für XML-Dokumente Little Endian oder Big Endian sein soll.

Die Kodierung für existierende Dateien wird immer beibehalten und kann nur mit dem Befehl **Datei | Kodierung** 174 geändert werden.

Dateien mit unbekannter Kodierung öffnen als

Sie können in dieser Auswahlliste auswählen, mit welcher Kodierung eine Datei geöffnet werden soll, die keine Kodierungsspezifikation enthält oder deren Kodierung nicht automatisch bestimmt werden konnte.

Nicht-XML-Dateien öffnen in

In dieser Auswahlliste können Sie auswählen, welche Kodierung für nicht XML-Dateien verwendet werden soll. Die können die Kodierung des Dokuments mit dem Befehl **Datei | Kodierung** 174 ändern.

 © 2016-2022 Altova GmbH

BOM (Byte Order Mark)

Bei Speicherung eines Dokuments mit einer 2-Byte- oder einer 4-Byte-Zeichenkodierung kann es entweder mit (i) der Bytefolge Little Endian und einer Little Endian BOM gespeichert werden (*BOM immer erstellen, wenn nicht UTF-8*); oder (ii) mit der ermittelten Bytefolge und der ermittelten BOM (*Festgestellten BOM beim Speichern beibehalten*).

Speichern und Beenden

Klicken Sie zum Abschluss auf **OK**.

13.9.10.4 Pretty-Print

Im Abschnitt **Pretty-Print** *(siehe Abbildungen unten)* können Sie festlegen, wie Text in dem in der Authentic-Ansicht angezeigten XML-Dokument dargestellt werden soll. Zwar gibt es in Authentic Desktop keine Textansicht des XML-Dokuments, doch werden die hier getroffenen Einstellungen auf den reinen Text der in der Authentic View-Ansicht bearbeiteten XML-Datei angewendet.

Es stehen die folgenden Einstellungen zur Verfügung:

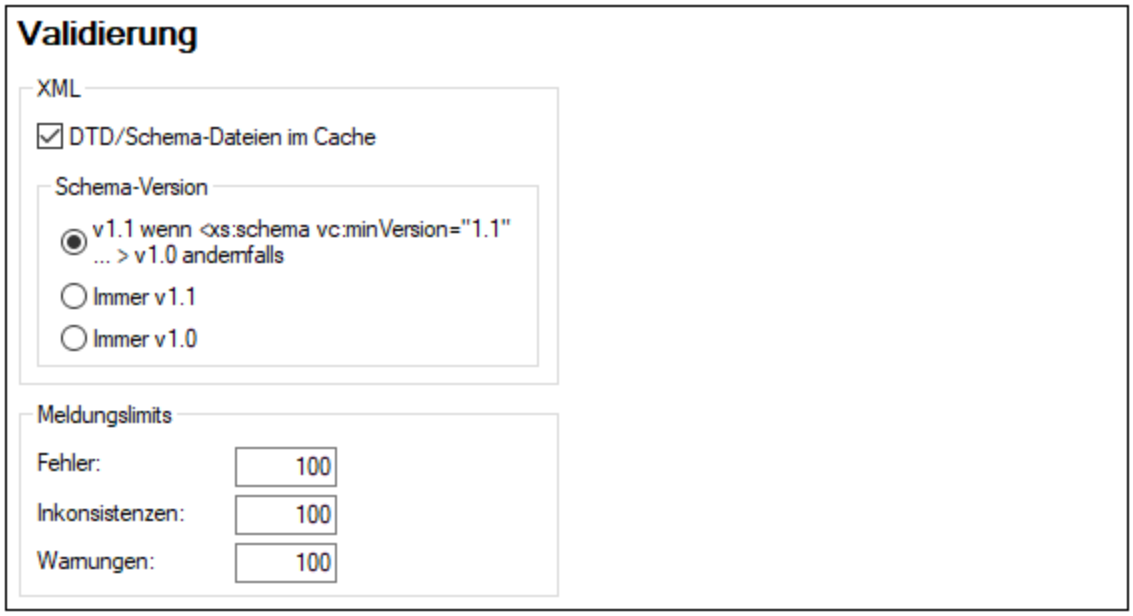
- Ob Tabulatoren oder Leerzeichen für die Pretty-Print-Darstellung verwendet werden sollen.
- Wie viele leere Elemente geschrieben werden sollen. Ein selbstschließendes Element ist ein Element, in dem der öffnende und schließende Tag in einem einzigen Element kombiniert werden, wie z.B.: **`<element/>`** oder **`<element />`**.

Speichern und Beenden

Klicken Sie zum Abschluss auf **OK**.

13.9.10.5 Validierung

Im Abschnitt **Validierung** können Sie die Optionen für die Validierung von XML-Dokumenten festlegen.

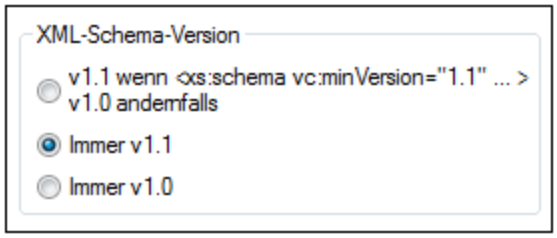


XML

Authentic Desktop kann DTD- und XML-Schema-Dateien im Arbeitsspeicher aufbewahren, um unnötiges Neuladen zu vermeiden (z.B. wenn ein Schema nicht lokal, sondern über eine URL aufgerufen wird). Beachten Sie jedoch, dass Änderungen bei Verwendung von zwischengespeicherten Schemas bei der Validierung nicht sofort berücksichtigt werden; Sie müssten die XML-Datei in diesem Fall neu laden oder Authentic Desktop neu starten.

Schema-Version

Welcher XSD-Modus in der Schema-Ansicht aktiviert wird, hängt sowohl (i) vom Vorhandensein bzw. Fehlen - und, falls vorhanden, vom Wert - des `/xs:schema/@vc:minVersion` Attributs des XSD-Dokuments als auch (ii) von der XML-Schema-Version ab, die im Dialogfeld "Optionen" (**Extras | Optionen**, *siehe Abbildung unten*) im Abschnitt "Datei" ausgewählt wurde.



 © 2016-2022 Altova GmbH

Es sind die folgenden Situationen möglich. *XML-Schema-Version* in der Tabelle unten bezieht sich auf die Auswahl im oben gezeigten Bereich XML-Schema-Version. Die `vc:minVersion` Werte in der unten stehenden Tabelle beziehen sich auf den Wert des `xs:schema/@vc:minVersion` Attributs im XML-Schema-Dokument.

XML-Schema-Version	`vc:minVersion` Attribut	XSD-Modus
Immer v 1.0	Fehlt oder ist mit einem beliebigen Wert vorhanden	1.0
Immer v 1.1	Fehlt oder ist mit einem beliebigen Wert vorhanden	1.1
Wert von @vc:minVersion	Das Attribut hat den Wert `1.1`	1.1
Wert von @vc:minVersion	Das Attribut fehlt oder ist vorhanden und hat einen anderen Wert als `1.1`	1.0

Meldungslimits

Mit Hilfe dieser Optionen können Sie für die Anzahl der angezeigten Fehler, XBRL-Inkonsistenzen und Warnungen separate Limits definieren. Der Standardwert für die einzelnen Kategorien ist 100. Ändern Sie ihn in den gewünschten Wert.

Speichern und Beenden

Klicken Sie zum Abschluss auf **OK**.

13.9.10.6 Ansicht

Im Abschnitt **Ansicht** können Sie die Darstellung der XML-Dokumente in Authentic Desktop anpassen.

Programmlogo

Das Programmlogo beim Starten kann deaktiviert werden, um die Applikation schneller starten zu können. Wenn Sie eine Lizenz erworben haben (und nicht mit einer Testversion arbeiten) haben Sie außerdem die Möglichkeit, das Programmlogo, die Copyright-Informationen und die Registrierungsinformationen beim Drucken eines Dokuments von XMLSpy aus zu deaktivieren.

Titelleiste-Inhalt

Im Fenstertitel jedes Dokumentfensters kann entweder nur der Dateiname oder der komplette Pfad angezeigt werden.

Browser-Ansicht

Derzeit wird in der Authentic-und der Browser-Ansicht Internet Explorer (IE) verwendet. IE ist daher der Standard-Browser für diese beiden Ansichten. Alternativ dazu kann Microsoft Edge Web View 2 für die Browser-Ansicht verwendet werden. Wenn Edge auf Ihrem Rechner nicht installiert ist, gehen Sie zur WebView2-Download-Seite, um den Evergreen Bootstrapper zu installieren. Dadurch kann Microsoft Edge WebView 2 für die Browser-Ansicht verwendet werden.

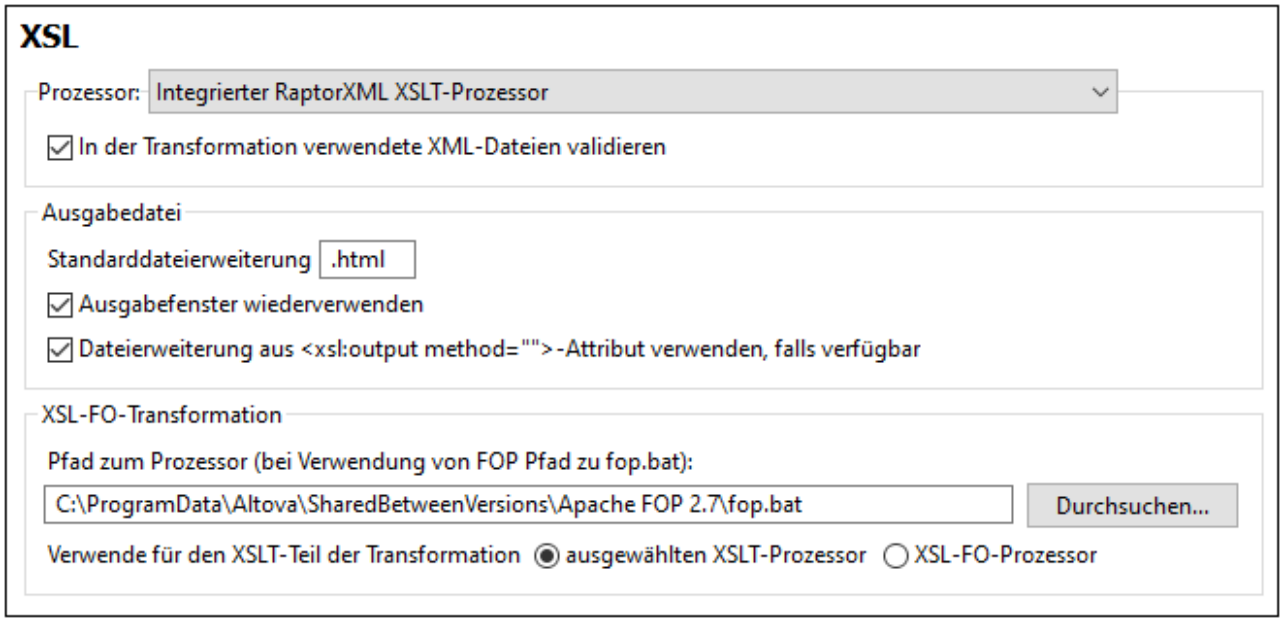
Nähere Informationen dazu finden Sie im Abschnitt Browser-Ansicht 94.

Speichern und Beenden

Klicken Sie zum Abschluss auf **OK**.

13.9.10.7 XSL

Im Abschnitt **XSL** (*Abbildung unten*) werden die Optionen für XSLT-Transformationen 276 und über die Applikation ausgeführte XSL-FO-Transformationen 278 definiert.



Prozessoreinstellungen

Sie können einen XSLT-Prozessor definieren, der bei Aufruf des Befehls **XSLT-Transformation** 221 XSLT-Transformationen durchführt.

Zur Auswahl stehen die folgenden XSLT-Prozessor-Optionen:

- Integrierter RaptorXML XSLT-Prozessor
- Microsoft XML Parser (MSXML)
- Externes XSL-Transformationsprogramm

Anmerkung: Für das XSLT-Debugging wird in Authentic Desktop immer der integrierte RaptorXML XSLT-Prozessor verwendet, selbst wenn für die Transformation hier ein anderer XSLT-Prozessor ausgewählt wurde.

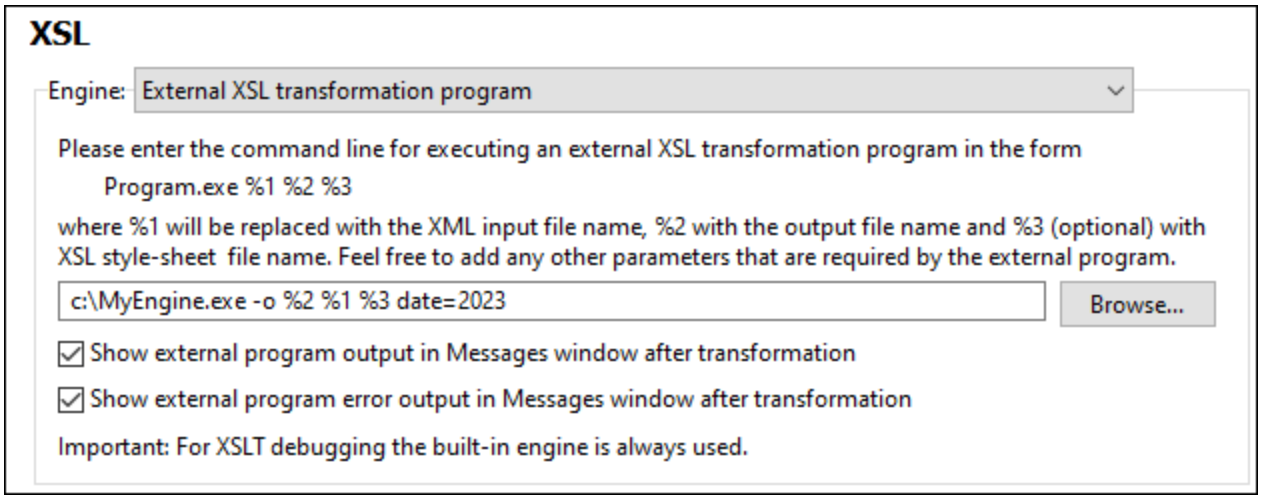
 © 2016-2022 Altova GmbH

Altova RaptorXML XSLT-Prozessor
Authentic Desktop enthält den Altova RaptorXML XSLT 1.0-, XSLT 2.0- und XSLT 3.0-Prozessor, den Sie für Ihre XSLT-Transformationen verwenden können. Je nach dem Wert des `version`-Attributs im Element `xsl:stylesheet` oder `xsl:transform` wird immer der jeweilige XSLT-Prozessor für Transformationen (1.0, 2.0 oder 3.0) verwendet. Dies gilt sowohl für XSLT-Transformationen als auch für das XSLT Debugging mit dem XSLT/XQuery-Debugger von XMLSpy.

Wenn Sie die in der Transformation verwendeten XML-Dateien validieren möchten, wählen Sie die Option *Validieren* (*siehe Abbildung oben*) aus.

Microsoft XML Parser (MSXML)
Auf Ihrem Rechner sind ein oder mehrere MSXML 3.0-, 4.0- oder 6.0-Parser vorinstalliert. Wenn Sie wissen, welche installierte Version Sie verwenden möchten, können Sie diese auswählen. Andernfalls wird die Version automatisch von Authentic Desktop ausgewählt. (Die Option *Version automatisch auswählen* ist standardmäßig aktiv.) In diesem Fall versucht Authentic Desktop, die neueste verfügbare Version auszuwählen.

Externes XSL-Transformationsprogramm
Wählen Sie einen externen XSLT-Prozessor Ihrer Wahl aus, indem Sie den Pfad zur ausführbaren Datei eingeben.



Sie müssen den Befehlszeilenstring, den der externe XSLT-Prozessor zur Ausführung einer Transformation verwendet, definieren. Sie können den Befehlszeilenstring mit den folgenden Komponenten erstellen:

`%1` = zu verarbeitendes XML-Dokument
`%2` = zu generierende Ausgabedatei
`%3` = zu verwendendes XSLT Stylesheet (wenn das XML-Dokument keine Referenz auf ein Stylesheet enthält)

Angenommen, Sie haben einen Prozessor, der zur Ausführung einer XSLT-Transformation den folgenden Befehl verwendet:

```
myxsltengine.exe -o <output.xml> <input.xml> <stylesheet.xslt> <param-name>=<param-value>?
```

Dann können Sie in Authentic Desktop die entsprechenden Variablen an der entsprechenden Stelle zur Generierung dieses Befehls verwenden. Beispiel:

```
c:\MyEngine.exe -o %2 %1 %3 date=2023
```

Authentic Desktop sendet die korrekten Input-Dateien zur Verarbeitung an den externen Prozessor und retourniert die Ausgabedatei(en) an einen Ausgabepfad, falls einer definiert wurde, und/oder an ein Applikationsfenster.

Aktivieren Sie die entsprechenden Kontrollkästchen, um die Ausgabe- und Fehlermeldungen des externen Programms im Fenster "Meldungen" in Authentic Desktop anzuzeigen.

Anmerkung: Die Parameter, die im Dialogfeld "XSLT-Eingabeparameter" 223 (Aufruf über das Menü **XSL**) definiert sind, werden nur an die internen Altova-XSLT-Prozessoren übergeben und nicht an andere XSLT-Prozessoren, die als XSLT-Standardprozessor konfiguriert wurden.

Ausgabedateieinstellungen

Es stehen die folgenden Optionen zur Verfügung:

- *Standarddateierweiterung:* Definiert eine Standarddateierweiterung für Ausgabedateien, die durch die im XSLT-Element `xsl:output` definierte Dateierweiterung (*siehe letzter Eintrag in der Liste*) außer Kraft gesetzt werden kann.
- *Ausgabefenster wiederverwenden:* Dadurch wird das Ergebnisdokument bei nachfolgenden Transformationen im selben Ausgabefenster angezeigt. Wenn die XML-Input-Datei zu einem Projekt gehört und die Option *Ausgabefenster wiederverwenden* deaktiviert ist, wird diese Einstellung nur dann aktiv, wenn auch der unter dem Feld *In Ordner speichern* definierte Ausgabedateipfad (*Abbildung unten*) in den entsprechenden Projekteigenschaften 215 **ebenfalls** deaktiviert ist.

- *Dateierweiterung aus xsl:output-Element verwenden:* Gibt an, ob die im Element `xsl:output` des XSLT-Stylesheet definierte Dateierweiterung die in der ersten Option dieser Liste definierte Standarddateierweiterung außer Kraft setzen soll.

XSL-FO-Transformationen

FO-Dokumente werden mit Hilfe des FO-Prozessors verarbeitet. Der Pfad zur ausführbaren Datei des FO-Prozessors muss im Textfeld für den XSL-FO-Transformationsprozessor definiert werden. Die Transformation wird mit Hilfe des Menübefehls **XSL/XQuery | XSL-FO-Transformation** 222 ausgeführt. Wenn es sich bei der Quelldatei (dem Dokument, das bei Ausführung des Befehls in der IDE aktiv ist) um ein XSL-FO-Dokument handelt, wird für die Transformation der FO-Prozessor aufgerufen. Wenn es sich beim Quelldokument um ein XML-Dokument handelt, muss das XML-Dokument zuerst mittels einer XSLT-Transformation in ein XSL-FO-Dokument konvertiert werden. Diese XSLT-Transformation kann entweder von dem für die Applikation als Standardprozessor definierten XSLT-Prozessor ausgeführt werden (*siehe oben* 276) oder vom XSLT-Prozessor, der möglicherweise bereits in den FO-Prozessor integriert ist, den Sie als Standard-FO-Prozessor für die Applikation definiert haben. Kicken Sie zur Auswahl zwischen den beiden Optionen auf das entsprechende Optionsfeld.

 © 2016-2022 Altova GmbH

Klicken Sie zum Abschluss auf **OK**.

Anmerkung: Falls Sie die Option zur Installation des FOP-Prozessors des Apache XML-Projekts nicht deaktiviert haben, wurde dieser im Ordner `C:\ProgramData\Altova\SharedBetweenVersions` installiert. In diesem Fall wurde der Pfad zum Prozessor automatisch in das Feld "XSL-FO-Prozessor" eingegeben. Sie können den Pfad zu jedem gewünschten FO-Prozessor definieren. Beachten Sie, dass derselbe Pfad auch von anderen Altova-Produkten, die FO-Prozessoren verwenden (StyleVision und Authentic Desktop) verwendet wird.

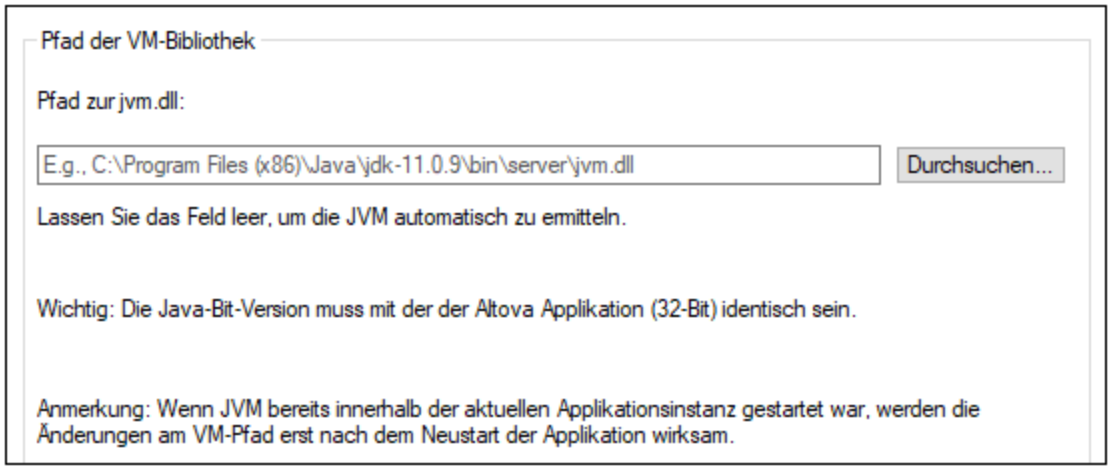
Speichern und Beenden

Klicken Sie zum Abschluss auf **OK**.

13.9.10.8 Java

Im Abschnitt *Java* (*siehe Abbildung unten*) haben Sie die Möglichkeit, den Pfad zu einer Java VM (Virtual Machine) auf Ihrem Dateisystem einzugeben. Beachten Sie, dass dies nicht immer notwendig ist. Authentic Desktop versucht standardmäßig den Java VM-Pfad automatisch zu ermitteln. Dazu wird zuerst die Windows Registry und anschließend die JAVA_HOME-Umgebungsvariable gelesen. Ein in dieses Dialogfeld eingegebener benutzerdefinierter Pfad hat Vorrang vor allen automatisch ermittelten Java VM-Pfaden.

Wenn Sie eine Java Virtual Machine verwenden, die keinen Installer hat und keine Registry-Einträge erstellt (z.B. OpenJDK von Oracle), müssen Sie eventuell einen benutzerdefinierten Java VM-Pfad angeben. Auch wenn Sie automatisch von Authentic Desktop ermittelte Java VM-Pfade aus irgendeinem Grund außer Kraft setzen müssen, müssen Sie diesen Pfad eventuell definieren.



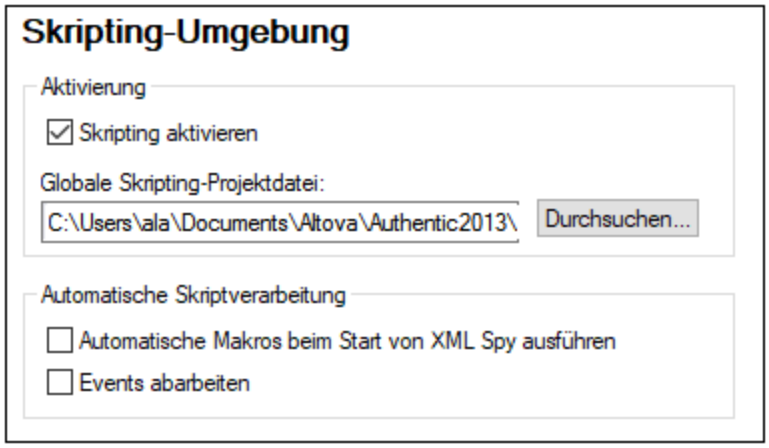
Beachten Sie dazu Folgendes:

- Der Java VM-Pfad wird gemeinsam von allen Altova Desktop-Applikationen (nicht aber den Server-Applikationen) verwendet. Wenn Sie den Pfad daher in einer Applikation ändern, gilt dies automatisch auch für alle anderen Altova-Applikationen.
- Der Pfad muss auf die Datei `jvm.dll` im Verzeichnis `\bin\server` oder `\bin\client` (relativ zum Verzeichnis, in dem JDK installiert ist) verweisen.
- Die Authentic Desktop-Plattform (32-Bit, 64-Bit) muss mit der des JDK identisch sein.

- Nachdem Sie den Java VM-Pfad geändert haben, müssen Sie Authentic Desktop eventuell neu starten, damit die neuen Einstellungen wirksam werden.

13.9.10.9 Skripting-Umgebung

Im Abschnitt **Skripting-Umgebung** (*Abbildung unten*) können Sie die Skripting-Umgebung 247 beim Start der Applikation aktivieren. Aktivieren Sie dazu das Kontrollkästchen *Skripting aktivieren*. Anschließend können Sie die globale Skripting-Projektdatei definieren. (*siehe Abbildung unten*).



Um ein globales Skripting-Projekt für Authentic Desktop zu definieren, aktivieren Sie das Kontrollkästchen *Skripting aktivieren* und suchen Sie die gewünschte Altova Skripting-Projektdatei (`.asprj`-Datei). Außerdem können Sie folgende Einstellungen definieren: (i) ob automatische Makros im Skripting-Projekt beim Start von Authentic Desktop automatisch ausgeführt werden sollen und (ii) ob Application Event Handler-Skripts im Projekt automatisch ausgeführt werden sollen oder nicht; aktivieren bzw. deaktivieren Sie das entsprechende Kontrollkästchen.

Klicken Sie anschließend auf **OK**. Die Makros im globalen Skripting-Projekt werden nun im Untermenü des Befehls **Makros** angezeigt.

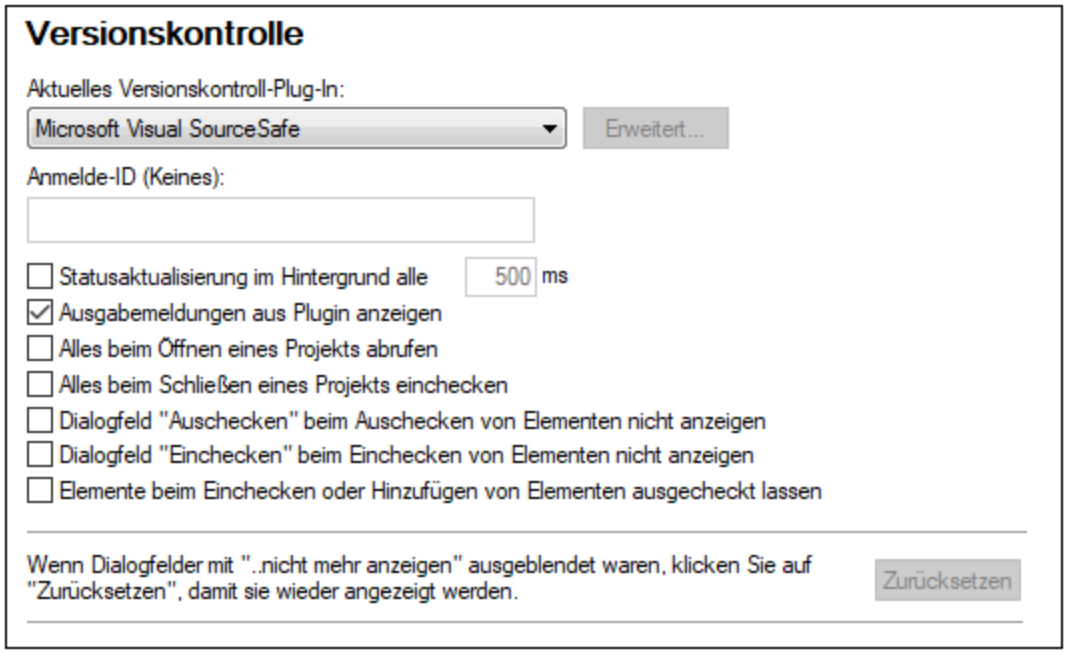
Speichern und Beenden

Klicken Sie anschließend auf **OK**. Die Makros im globalen Skripting-Projekt werden nun im Untermenü des Befehls **Makros** angezeigt.

13.9.10.10 Versionskontrolle

Im Abschnitt **Versionskontrolle** (*Abbildung unten*) können Sie den Versionskontrollprovider, die Einstellungen und die Standard-Anmelde-IDs für die einzelnen Versionskontrollsysteme definieren.

 © 2016-2022 Altova GmbH



Versionskontroll-Plugin

Das aktuelle Versionskontroll-Plugin kann aus den derzeit installierten Versionskontrollsystemen ausgewählt werden. Diese werden in der Dropdown-Auswahlliste aufgelistet. Definieren Sie nach Auswahl der gewünschten Versionskontrolle im nächsten Textfeld die Anmelde-ID dafür. Über die Schaltfläche **Erweitert** rufen Sie ein Dialogfeld zum jeweiligen Versionskontroll-Plugin auf, in dem Sie die Einstellungen für dieses Plugin definieren können. Diese Einstellungen unterscheiden sich je nach Plugin.

Benutzereinstellungen

Es steht eine Reihe von Benutzereinstellungen zur Verfügung, darunter die folgenden:

- Statusaktualisierungen können im Hintergrund in benutzerdefinierten Zeitabständen durchgeführt oder komplett deaktiviert werden. Sehr große Versionskontrolldatenbanken könnten beträchtlich viel Arbeitsspeicher beanspruchen. Indem Sie die Statusaktualisierungen im Hintergrund deaktivieren oder das Zeitintervall vergrößern, können Sie Ressourcen frei machen, sodass Ihr System schneller arbeitet.
- Dateien können beim Öffnen und Schließen von Projekten automatisch aus- bzw. eingecheckt werden.
- Die Anzeige der Dialogfelder "Einchecken" und "Auschecken" kann unterdrückt werden.
- Die Schaltfläche "**Zurücksetzen**" steht zur Verfügung, wenn Sie in einem der Dialogfelder die Option "*Wenn Dialogfelder mit…"*aktiviert haben. Wenn Sie auf die Schaltfläche **Zurücksetzen** klicken, wird diese Option wieder aktiviert.

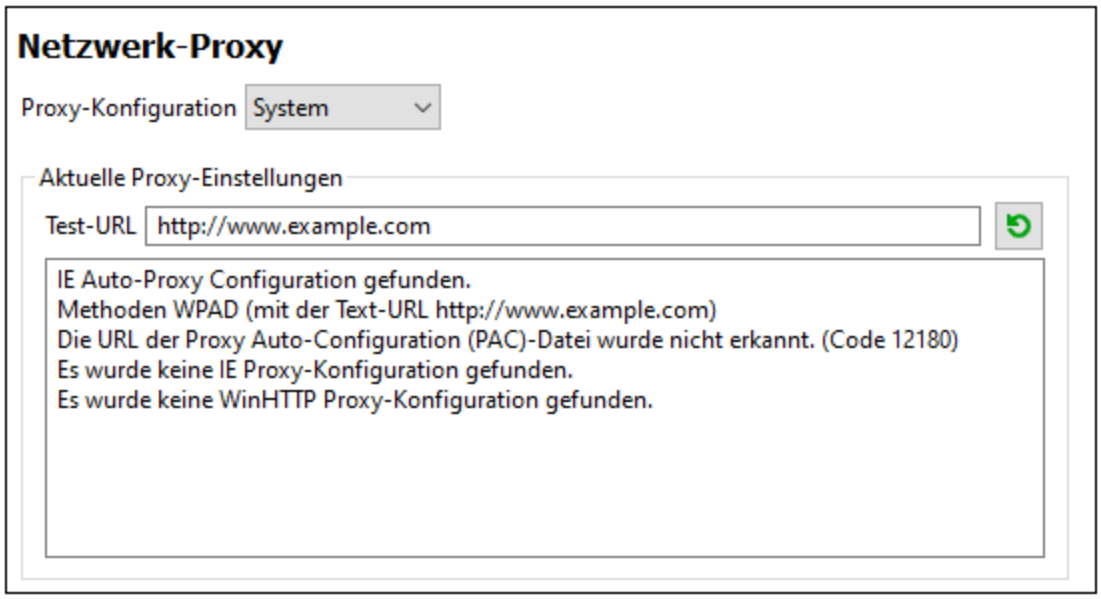
Speichern und Beenden

Klicken Sie zum Abschluss auf **OK**.

13.9.10.11 Netzwerk-Proxy

Im Abschnitt *Netzwerk-Proxy* können Sie die benutzerdefinierten Proxy-Einstellungen konfigurieren. Diese Einstellungen beeinflussen, wie die Applikation eine Verbindung mit dem Internet herstellt (z.B. zur XML-Validierung). Standardmäßig werden die Proxy-Einstellungen des Systems verwendet, d.h. die Einstellungen funktionieren, ohne dass der Benutzer etwas daran ändern muss. Falls nötig, können Sie jedoch einen anderen Netzwerk-Proxy-Server definieren. Wählen Sie dazu in der Auswahlliste *Proxy-Konfiguration* entweder die Option *Automatisch* oder *Manuell*, um die Einstellungen entsprechend zu konfigurieren.

Anmerkung: Die Nethwerk-Proxy-Einstellungen werden von allen Altova MissionKit-Applikationen gemeinsam verwendet. Wenn Sie daher die Einstellungen in einer Applikation ändern, wirkt sich dies automatisch auf alle anderen Applikationen aus.



System-Proxy-Einstellungen verwenden
Dadurch werden die über die System-Proxy-Einstellungen konfigurierbaren Internet Explorer (IE)-Einstellungen verwendet. Führt außerdem eine Abfrage der mit **`netsh.exe winhttp`** konfigurierten Einstellungen durch.

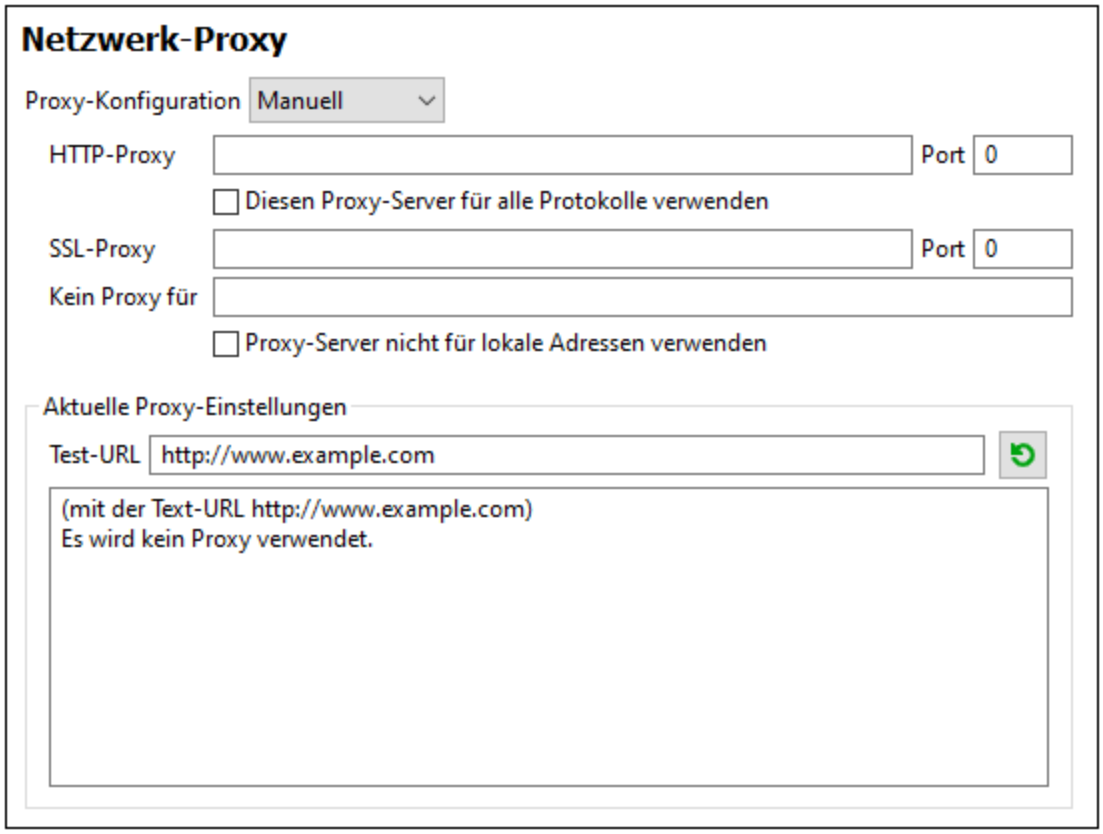
Automatische Proxy-Konfiguration
Es stehen die folgenden Optionen zur Verfügung:

- *Einstellungen automatisch ermitteln:* **`Verwendet ein WPAD-Skript`** (http://wpad.LOCALDOMAIN/wpad.dat) über DHCP oder DNS, um die Einrichtung des Proxy-Servers zu konfigurieren.
- *Skript-URL:* Definieren Sie eine HTTP URL zu einem automatischen Proxy-Konfigurationsskript (**`.pac`**), mit dem der Proxy-Server eingerichtet wird.
- *Neu laden:* Setzt die aktuelle automatische Proxy-Konfiguration zurück und lädt sie neu. Dafür ist Windows 8 oder neuer erforderlich. Die Rücksetzung kann bis zu 30 Sekunden dauern.

Manuelle Proxy-Konfiguration
Definieren Sie den vollständig qualifizierten Host-Namen und Port für die Proxy-Server der jeweiligen Protokolle manuell. Im Host-Namen kann ein unterstütztes Schema inkludiert werden (z.B.: **`http://hostname`**). Das

 © 2016-2022 Altova GmbH

Schema muss nicht mit dem entsprechenden Protokoll übereinstimmen, wenn der Proxy-Server das Schema unterstützt.



Es stehen die folgenden Optionen zur Verfügung:

- *HTTP-Proxy:* Verwendet den angegebenen Host-Namen und Port für das HTTP-Protokoll. Wenn *Diesen Proxy-Server für alle Protokolle verwenden* aktiviert ist, wird der angegebene HTTP-Proxy-Server für alle Protokolle verwendet.
- *SSL-Proxy:* Verwendet den angegebenen Host-Namen und Port für das SSL-Protokoll.
- *Kein Proxy für:* eine durch Semikola (`;`) getrennte Liste von voll qualifizierten Host-Namen, Domain-Namen oder IP-Adressen für Hosts, die ohne einen Proxy-Server verwendet werden sollen. IP-Adressen dürfen nicht abgeschnitten werden und IPv6-Adressen müssen innerhalb von eckige Klammern gesetzt werden (z.B.: `[2606:2800:220:1:248:1893:25c8:1946]`). Domain-Namen muss ein Punkt vorangestellt werden (z.B: `.example.com`).
- *Proxy-Server nicht für lokale Adressen verwenden:* Falls dieses Kontrollkästchen aktiviert ist, wird `<local>` zur *Kein Proxy für*-Liste hinzugefügt. Falls diese Option ausgewählt ist, wird für die folgenden Adressen kein Proxy-Server verwendet: (i) `127.0.0.1`, (ii) [::1], (iii) alle Host-Namen, die kein Punktzeichen (`.`) enthalten.

Aktuelle Proxy-Einstellungen
Stellt ein ausführliches Protokoll der Proxy-Ermittlung bereit. Es kann über die Schaltfläche **Aktualisieren** rechts vom Feld *Test-URL* aktualisiert werden (z.B. bei Wechsel zu einer anderen Test-URL oder wenn die Proxy-Einstellungen geändert wurden).

- *Test-URL:* Anhand einer Test-URL kann ermittelt werden, welcher Proxy-Server für diese bestimmte

URL verwendet wird. Mit dieser URL erfolgt kein I/O. Dieses Feld darf nicht leer sein, wenn die automatische Proxy-Konfiguration verwendet wird (entweder über *System-Proxy-Einstellungen verwenden* oder *Automatische Proxy-Konfiguration*).

© 2016-2022 Altova GmbH

13.10 Menü "Fenster"

Das Menü **Fenster** enthält Befehle, mit denen Sie einzelne Applikations- und Dokumentfenster der Benutzeroberfläche verwalten können. Sie können die offenen Dokumente überlappend oder nebeneinander anzeigen sowie Eingabehilfenfenster und Ausgabefenster anordnen und ein- oder ausblenden.

Überlappend, horizontal/vertikal anordnen

Mit dem Befehl **Überlappend** ordnen Sie Dokumentfenster gestaffelt übereinander von hinten nach vorne an.

Mit den Befehlen **Horizontal anordnen** und **Vertikal anordnen** arrangieren Sie alle offenen, nicht minimierten Dokumente horizontal nebeneinander bzw. vertikal übereinander, sodass alle Dokumentfenster gleichzeitig im Applikationsfenster sichtbar sind.

Projekt-Fenster, Info-Fenster, Eingabehilfen, Ausgabefenster

Mit diesen Befehlen blenden Sie das Projekt-Fenster 17, bzw. das Info-Fenster 19, die Eingabehilfen 19 bzw. Ausgabefenster 19 ein oder aus.

Jedes dieser Fenster kann an- und abgedockt werden. Wenn Sie es an der Titelleiste ziehen, können Sie es von der aktuellen Position abdocken und es frei schwebend anzeigen. Wenn Sie mit der rechten Maustaste auf die Titelleiste klicken, können Sie das Fenster andocken oder ausblenden.

Projekt und Eingabehilfen

Mit diesem Befehl können Sie das Projekt-Fenster 17 und die Eingabehilfen 19 gleichzeitig ein- und ausblenden. Sie müssen diese Fenster dadurch nicht separat ein- und ausblenden.

Alle ein/aus

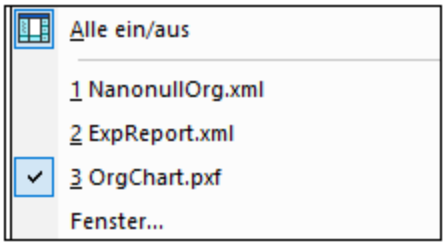
Mit diesem Befehl blenden Sie alle an- und abdockbaren Fenster (*siehe Liste unten*) ein oder aus.

- Projekt-Fenster 17
- Info-Fenster 19
- Eingabehilfen 19
- Ausgabefenster 19

Diese Funktion ist besonders hilfreich, wenn Sie schnell alle anderen Fenster ausblenden möchten, um das/die Dokument(e), an dem/denen Sie gerade arbeiten, in maximaler Größe anzeigen zu können.

Liste der derzeit geöffneten Fenster

In der Liste sehen Sie alle offenen Fenster und können jederzeit zwischen diesen Fenstern wechseln.

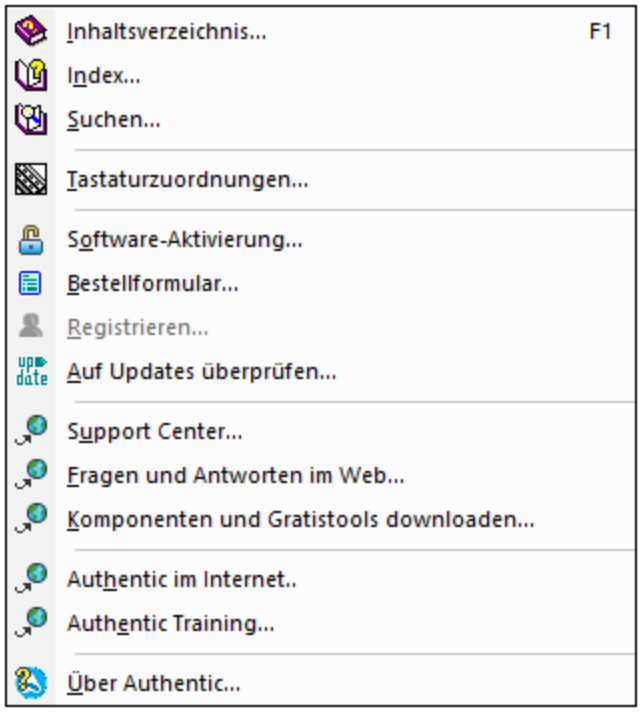


Sie können auch mit Hilfe der Tastenkürzel **STRG+F6** zwischen den Fenstern wechseln.

© 2016-2022 Altova GmbH

13.11 Menü "Hilfe"

Im Hilfemenü finden Sie Befehle, die Sie benötigen, um weitere Informationen zu spezifischen Themen von Authentic Desktop zu erhalten, sowie Links zu Informationen und Supportseiten auf dem Altova Webserver.



Im Hilfemenü finden Sie weiters den Registrierungsdialog 288, über den Sie Ihren Keycode eingeben können, sobald Sie das Produkt erworben haben.

13.11.1 Inhaltsverzeichnis, Index, Suche

- Inhaltsverzeichnis

  Mit dem Befehl "Inhaltsverzeichnis" wird die Bildschirmhilfe für Authentic Desktop geöffnet, wobei das Inhaltsverzeichnis auf der linken Seite des Hilfefensters angezeigt wird. Das Inhaltsverzeichnis bietet eine Übersicht über das gesamte Hilfedokument. Wenn Sie auf einen Eintrag im Inhaltsverzeichnis klicken, gelangen Sie zu diesem Thema.

- Index

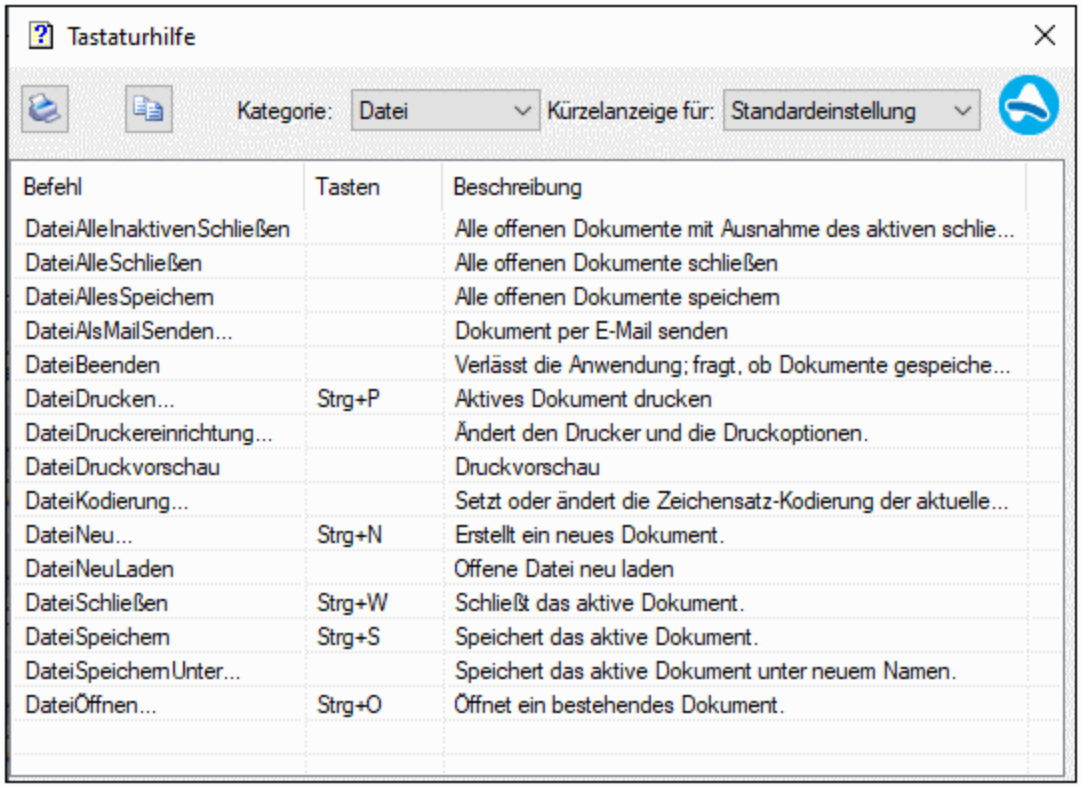
  Mit dem Befehl Index wird die Bildschirmhilfe für Authentic Desktop geöffnet, wobei auf der linken Seite des Hilfefensters der Stichwortindex angezeigt wird. Im Index werden Stichwörter aufgelistet und Sie können durch Doppelklick auf das Stichwort zum jeweiligen Thema navigieren. Wenn ein Stichwort mit mehr als einem Thema verknüpft ist, können Sie das Thema aus einer Liste von Themen auswählen.

- Suchen

Mit dem Befehl Suchen wird die Bildschirmhilfe für Authentic Desktop geöffnet, wobei auf der linken Seite des Hilfefensters das Suchdialogfeld angezeigt wird. Um nach einem Begriff zu suchen, geben Sie ihn in das Eingabefeld ein und drücken Sie die **Eingabetaste** oder klicken Sie auf **Kapitel auflisten**. Das Hilfesystem führt eine Volltextsuche am gesamten Hilfedokument durch und gibt eine Trefferliste zurück. Doppelklicken Sie auf einen Eintrag, um das Thema anzuzeigen.

13.11.2 Tastaturzuordnungen

Bei Auswahl des Befehls **Tastenzuordnungen** wird eine Infobox angezeigt, in der alle Befehle in Authentic Desktop nach Menüs geordnet angezeigt werden. Die Menübefehle werden mit einer Beschreibung und dem entsprechenden Tastaturkürzel für den jeweiligen Befehl angezeigt.



Um die Befehle eines bestimmten Menüs anzuzeigen, wählen Sie den Menünamen in der Dropdown-Liste "Kategorie" aus. Durch Klicken auf das Druckersymbol können Sie den Befehl ausdrucken.

13.11.3 Aktivierung, Bestellformular, Registrieren, Updates

- Software-Aktivierung

  Lizenzieren Ihres Produkts
  Nachdem Sie Ihre Altova-Software heruntergeladen haben, können Sie sie entweder mit Hilfe eines

 © 2016-2022 Altova GmbH

kostenlosen Evaluierungs-Keycode oder eines käuflich erworbenen permanenten Lizenzkeycode lizenzieren oder aktivieren.

- ***Kostenlose Evaluierungs-Lizenz.*** Wenn Sie die Software zum ersten Mal starten, wird das Dialogfeld **Software-Aktivierung** angezeigt. Es enthält eine Schaltfläche, über die Sie eine kostenlose Evaluierungs-Lizenz anfordern können. Geben Sie in das Dialogfeld Ihren Namen, den Namen Ihrer Firma und Ihre E-Mail-Adresse ein und klicken Sie auf **Anfordern**. Daraufhin wird an die von Ihnen angegebene E-Mail-Adresse eine Lizenz gesendet und sollte in wenigen Minuten eintreffen. Speichern Sie die Lizenzdatei in einem geeigneten Ordner.

  Wenn Sie auf **Anfordern** klicken, wird am unteren Rand des Dialogfelds ein Eingabefeld angezeigt. Über dieses Feld gelangen Sie zum Pfad der Lizenzdatei. Navigieren Sie zur Lizenzdatei oder geben Sie den Pfad dazu ein und klicken Sie auf **OK**. (Sie können im Dialogfeld **Software-Aktivierung** auch auf **Neue Lizenz hochladen** klicken, um das Dialogfeld, in das der Pfad zur Lizenzdatei eingegeben wird, aufzurufen). Die Software wird für einen Zeitraum von 30 Tagen aktiviert.

- ***Permanenter Lizenz-Keycode.*** Über das Dialogfeld **Software-Aktivierung** können Sie einen permanenten Lizenz-Keycode erwerben. Wenn Sie auf diese Schaltfläche klicken, gelangen Sie zum Altova Online Shop, in dem Sie einen permanenten Lizenzschlüssel für Ihr Produkt erwerben können. Ihre Lizenz wird Ihnen in Form einer Lizenzdatei, die Ihre Lizenzdaten enthält, per E-Mail zugesendet.

  Es gibt drei Arten von permanenten Lizenzen: *Einzelplatzlizenzen*, *Parallellizenzen* und *Named User-Lizenzen* (benutzerdefinierte Nutzung). Mit einer Einzelplatzlizenz wird die Software auf einem einzigen Rechner freigeschaltet. Wenn Sie eine Einzelplatzlizenz für *N* Rechner erwerben, gestattet Ihnen die Lizenz, die Software auf bis zu *N* Rechnern zu verwenden. Mit einer Parallellizenz für *N* Parallelbenutzer dürfen *N* Benutzer die Software gleichzeitig ausführen. (Die Software darf auf 10N Rechnern installiert sein.) Mit einer Named User-Lizenz darf ein bestimmter Benutzer die Software auf bis zu 5 verschiedenen Rechnern verwenden. Um Ihre Software zu aktivieren, klicken Sie auf **Neue Lizenz hochladen** und geben Sie im daraufhin angezeigten Dialogfeld den Pfad zur Lizenzdatei ein und klicken Sie auf **OK**.

Anmerkung: Bei Mehrplatzlizenzen wird jeder Benutzer aufgefordert, seinen eigenen Namen einzugeben.

Ihre Lizenz-E-Mail und die verschiedenen Methoden, Ihr Altova-Produkt zu lizenzieren
Die Lizenz-E-Mail, die Sie von Altova erhalten, enthält Ihre Lizenzdatei im Anhang. Die Lizenzdatei hat die Dateierweiterung `.altova_licenses`.

Sie haben folgende Möglichkeiten, Ihr Altova-Produkt zu aktivieren:

- Speichern Sie die Lizenzdatei (`.altova_licenses`) in einem geeigneten Ordner, doppelklicken Sie auf die Lizenzdatei, geben Sie etwaige erforderliche Informationen in das Dialogfeld ein, das daraufhin angezeigt wird und beenden Sie den Vorgang durch Klicken auf **Lizenzschlüssel anwenden**.
- Speichern Sie die Lizenzdatei (`.altova_licenses`) in einem geeigneten Ordner. Wählen Sie in Ihrem Altova-Produkt den Menübefehl **Hilfe | Software-Aktivierung** und klicken Sie anschließend auf **Neue Lizenz hochladen.** Navigieren Sie zur Lizenzdatei oder geben Sie den Pfad dazu ein und klicken Sie auf **OK**.

- Speichern Sie die Lizenzdatei (`.altova_licenses`) in einem geeigneten Ordner und laden Sie diese von dort aus in den Lizenz-Pool Ihres Altova LicenseServer hoch. Sie können die Lizenz anschließend (i) entweder von Ihrem Altova-Produkt über das Dialogfeld "Software-Aktivierung" abrufen (*siehe unten*) oder (ii) dem Produkt die Lizenz von Altova LicenseServer aus zuweisen. *Nähere Informationen zur Lizenzierung über LicenseServer finden Sie weiter unten in diesem Kapitel.*

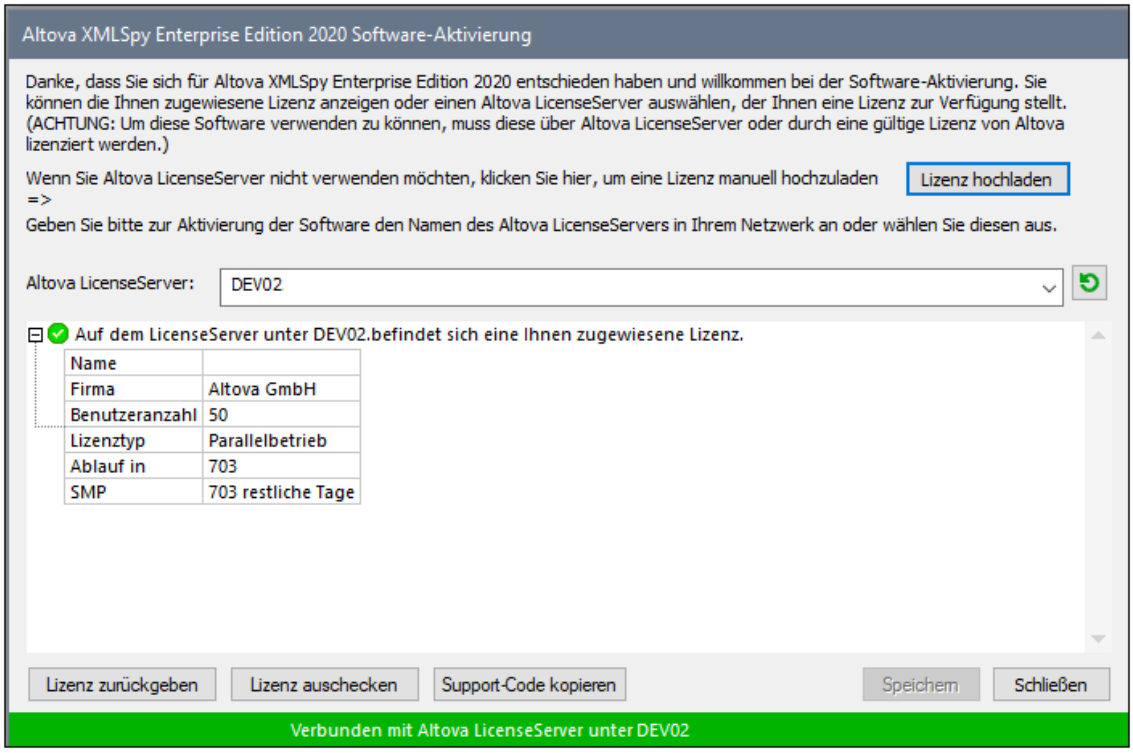
Das Dialogfeld **Software-Aktivierung** (*Abbildung unten*) kann über den Befehl **Hilfe | Software-Aktivierung** aufgerufen werden.

Aktivieren Ihrer Software
Sie können die Software durch Registrieren der Lizenz im Dialogfeld "Software-Aktivierung" oder durch Lizenzierung über Altova LicenseServer (*nähere Informationen siehe unten*) aktivieren.

- *Registrierung der Lizenz im Dialogfeld "Software-Aktivierung".* Klicken Sie im Dialogfeld auf **Neue Lizenz hochladen** und navigieren Sie zur Lizenzdatei. Klicken Sie auf **OK,** um den Pfad zur Lizenzdatei und alle eingegebenen Daten (im Fall einer Mehrplatzlizenz Ihren Namen) zu bestätigen und abschließend auf **Speichern**.

- *Lizenzierung über einen Altova LicenseServer in Ihrem Netzwerk:* Um eine Lizenz über einen Altova LicenseServer in Ihrem Netzwerk abzurufen, (klicken Sie am unteren Rand des Dialogfelds **Software-Aktivierung** auf **Altova LicenseServer verwenden**). Wählen Sie den Rechner aus, auf dem der gewünschte LicenseServer installiert wurde. Beachten Sie, dass die automatische Ermittlung von License Servern durch die Aussendung eines Signals ins LAN erfolgt. Da diese Aussendung auf ein Subnetz beschränkt ist, muss sich der LicenseServer im selben Subnetz wie der Client-Rechner befinden, damit die Ermittlung von License Servern funktioniert. Falls die automatische Ermittlung nicht funktioniert, geben Sie den Namen des Servers ein. Der Altova LicenseServer muss in seinem Lizenzpool eine Lizenz für Ihre Altova-Produkt haben. Wenn im LicenseServer-Pool eine Lizenz verfügbar ist, wird dies im Dialogfeld **Software-Aktivierung** angezeigt (*siehe Abbildung unten, in der Sie das Dialogfeld in Altova XMLSpy sehen*) und Sie können auf **Speichern** klicken, um die Lizenz abzurufen.

 © 2016-2022 Altova GmbH



Eine rechnerspezifische Lizenz (Einzelplatzlizenz) kann erst nach Ablauf von sieben Tagen wieder an LicenseServer zurückgegeben werden. Danach können Sie die rechnerspezifische Lizenz durch Klick auf **Lizenz zurückgeben** an den Server zurückgeben, sodass sie von einem anderen Client vom LicenseServer abgerufen werden kann. Ein LicenseServer-Administrator kann die Zuweisung einer abgerufenen Lizenz jedoch über die Web-Benutzeroberfläche von LicenseServer jederzeit aufheben. Beachten Sie, dass eine Rückgabe von Lizenzen nur bei rechnerspezifischen Lizenzen, nicht aber bei Parallellizenzen möglich ist.

Lizenz-Check-Out
Über den Lizenzpool können Sie eine Lizenz für einen Zeitraum von bis zu 30 Tagen auschecken, sodass die Lizenz auf dem lokalen Rechner gespeichert wird. Dadurch können Sie offline arbeiten, was nützlich ist, wenn Sie z.B. in einer Umgebung arbeiten möchten, in der Sie keinen Zugriff auf Ihren Altova LicenseServer haben (z.B. wenn Ihr Altova-Produkt auf einem Laptop installiert ist und Sie gerade unterwegs sind). Solange die Lizenz ausgecheckt ist, zeigt LicenseServer die Lizenz als in Verwendung an. Diese Lizenz kann dann von keinem anderen Rechner verwendet werden. Die Lizenz wird nach Ablauf des Check-Out-Zeitraums automatisch wieder eingecheckt. Alternativ dazu kann eine ausgecheckte Lizenz jederzeit über die Schaltfläche **Einchecken** des Dialogfelds **Software-Aktivierung** wieder eingecheckt werden.

Um eine Lizenz auszuchecken, gehen Sie folgendermaßen vor: (i) Klicken Sie im Dialogfeld **Software-Aktivierung** auf **Lizenz auschecken** (*siehe Abbildung oben*); (ii) Wählen Sie im daraufhin angezeigten Dialogfeld **Lizenz-Check-Out** den gewünschten Check-Out-Zeitraum aus und klicken Sie auf **Auschecken**. Daraufhin wird die Lizenz ausgecheckt. Nachdem Sie eine Lizenz ausgecheckt haben, geschehen zwei Dinge: (i) Die Check-Out-Informationen und das Ende des Check-Out-Zeitraums werden im Dialogfeld **Software-Aktivierung** angezeigt; (ii) Die Schaltfläche **Lizenz auschecken** im Dialogfeld ändert sich nun in **Einchecken**. Sie können die Lizenz jederzeit durch Klicken auf **Einchecken** einchecken. Da die Lizenz nach Ablauf des

Check-Out-Zeitraums automatisch wieder in den Zustand "Eingecheckt" zurück wechselt, sollte der von Ihnen ausgewählte Zeitraum für das Check-Out den gewünschten Zeitraum, in dem Sie offline arbeiten möchten, entsprechend abdecken.

Wenn eine Lizenz wieder eingecheckt wird, muss diese Lizenz für dieselbe Hauptversion eines Altova-Produkts ausgestellt sein, wie die Lizenz, die ausgecheckt wurde. Stellen Sie daher sicher, dass die Lizenz eingecheckt ist, bevor Sie für Ihr Altova-Produkt ein Upgrade auf die nächste Hauptversion installieren.

Anmerkung: Damit Lizenzen ausgecheckt werden können, muss die Check-Out-Funktion auf dem LicenseServer aktiviert werden. Wenn diese Funktion nicht aktiviert wurde, erhalten Sie eine entsprechende Fehlermeldung, wenn Sie versuchen die Lizenz auszuchecken. Wenden Sie sich in diesem Fall an Ihren LicenseServer-Administrator.

Support-Code kopieren
Klicken Sie auf **Support-Code kopieren,** um Lizenzinformationen in die Zwischenablage zu kopieren. Dies sind die Daten, die Sie bei einer Support-Anfrage über das Online Support-Formular benötigen.

Altova LicenseServer bietet IT-Administratoren einen Echtzeitüberblick über alle Altova-Lizenzen in einem Netzwerk. Dazu werden die Einzelheiten zu jeder Lizenz sowie Client-Zuweisungen und die Verwendung von Lizenzen durch Clients angezeigt. Der Vorteil der Verwendung von LicenseServer liegt in seinen Funktionen zur Verwaltung großer Altova-Lizenzpools. Altova LicenseServer steht kostenlos auf der Altova Website zur Verfügung. Nähere Informationen zu Altova LicenseServer und der Lizenzierung mittels Altova LicenseServer finden Sie in der Dokumentation zu Altova LicenseServer.

- Bestellformular

Sobald Sie eine lizenzierte Version des Software-Produkts bestellen möchten, klicken Sie im Dialogfeld **Software-Aktivierung** (*siehe oben*) auf die Schaltfläche **Permanenten Key-Code erwerben...** oder wählen Sie den Befehl **Bestellformular**, um zum sicheren Online-Shop von Altova weitergeleitet zu werden.

- Registrierung

Bei Aufruf dieses Befehls wird die Altova-Produktregistrierungsseite auf einem Register Ihres Browsers geöffnet. Durch Registrierung Ihrer Altova-Software stellen Sie sicher, dass Sie immer die neuesten Produktinformationen erhalten.

- Auf Updates überprüfen

Überprüft, ob am Altova Server eine neuere Version Ihres Produkts vorhanden ist und zeigt eine entsprechende Meldung an.

13.11.4 Weitere Befehle

- Support Center

Der Befehl "Support Center" ist ein Link zum Altova Support Center im Internet. Im Support Center finden Sie Antworten auf häufig gestellt Fragen, Diskussionsforen, in denen Sie Software-Probleme besprechen

 © 2016-2022 Altova GmbH

können und ein Formular, um unsere Mitarbeiter vom technischen Support zu kontaktieren.

- Fragen und Antworten im Web

  Der Befehl "Fragen und Antworten im Web" ist ein Link zur Altova-Fragen- und Antworten-Datenbank im Internet. Diese Datenbank wird ständig aktualisiert, wenn die Mitarbeiter von Altova von unseren Kunden mit neuen Fragen konfrontiert werden.

- Komponenten und Gratistools downloaden

  Dieser Befehl ist ein Link zum Komponenten Download Center von Altova im Internet. Von hier können Sie Software-Komponenten verschiedener anderer Anbieter herunterladen, die Sie mit Altova Produkten verwenden können. Dabei handelt es sich um XSLT- und XSL-FO-Prozessoren, Applikationsserverplattformen usw. Die im Komponenten Download Center verfügbare Software ist normalerweise kostenlos.

- Authentic Desktop im Internet

  Der Befehl Authentic Desktop im Internet ist ein Link zur Altova Website im Internet. Hier erfahren Sie mehr über Authentic Desktop und verwandte Technologien und Produkte auf der Altova Website.

- Über Authentic Desktop

  Mit dem Befehl Über Authentic Desktop wird das Willkommensfenster und die Versionsnummer Ihres Produkts angezeigt. Wenn Sie die 64-Bit-Verson von Authentic Desktop verwenden, wird dies durch das Suffix (x64) nach dem Applikationsnamen angezeigt. Die 32-Bit-Version hat kein Suffix.

13.12 Befehlszeile

Bestimmte Authentic Desktop-Aktionen können über die Befehlszeile ausgeführt werden. Diese Befehle sind im Folgenden aufgelistet:

Datei öffnen

authentic.exe `file.xml`

Öffnet die Datei `file.xml`, in Authentic Desktop

Mehrere Dateien öffnen

authentic.exe `file1.xml file2.xml`

Öffnet die Dateien `file1.xml` und `file2.xml` in Authentic Desktop.

Einer XML-Datei eine SPS-Datei für die Bearbeitung in der Authentic-Ansicht zuweisen

authentic.exe `myxml.xml /sps mysps.sps`

Öffnet die Datei `myxml.xml` in der Authentic-Ansicht mit `mysps.sps` als ihrer SPS-Datei. Das Flag `/sps` gibt an, dass (für die Bearbeitung in der Authentic-Ansicht) die nachfolgende SPS-Datei mit der XML-Datei zu verwenden ist, die vor dem `/sps` Flag steht.

Neue XML-Vorlagendatei über eine SPS-Datei öffnen

authentic.exe `mysps.sps`

Öffnet eine neue XML-Datei in der Authentic-Ansicht. Die Anzeige basiert auf dem SPS und die neue XML-Datei hat eine Skelettstruktur, die auf dem SPS-Schema basiert. Der Name der neu erstellten XML-Datei muss der XML-Datei beim Speichern zugewiesen werden.

© 2016-2022 Altova GmbH

14 Referenz für Programmierer

Authentic Desktop ist ein Automation Server, d.h. eine Applikation, die anderen Applikationen, so genannten Automation Clients, programmierbare Objekte zur Verfügung stellt. Ein Automation Client hat direkten Zugriff auf die vom Automation Server bereitgestellten Objekte und Funktionalitäten, sodass sich der Automation Client die Funktionalitäten von Authentic Desktop zunutze machen kann. Dadurch können Sie von anderen Applikationen aus die fertigen Funktionen von Authentic Desktop nutzen.

Die programmierbaren Objekte von Authentic Desktop stehen den Automation Clients über die Applikations-API von Authentic Desktop, eine COM API, zur Verfügung. Die Applikations-API Authentic Desktop wird in dieser Dokumentation in der Folge auch als Applikations-API bezeichnet. Das Objektmodell der API und eine vollständige Beschreibung aller verfügbaren Objekte finden Sie in dieser Dokumentation (siehe Abschnitt Applikations-API 341).

Die Applikations-API kann von den folgenden Umgebungen aus aufgerufen werden:

- Skript-Editor 297
- IDE Plug-ins 325
- Externe Programme 341
- ActiveX Integration 619

Im Folgenden finden Sie eine Beschreibung der einzelnen Umgebungen.

Skript-Editor: Anpassen von Authentic Desktop und Ändern der Funktionalität

Sie können Ihre Installation von Authentic Desktop anpassen, indem Sie sie ändern und Funktionalitäten dazu hinzufügen. Sie können auch Formulare für die Benutzereingabe erstellen und neue Menübefehle und Symbolleisten-Schaltflächen zur Benutzeroberfläche hinzufügen. Zu diesem Zweck werden Skripts geschrieben, die mit Objekten der Applikations-API interagieren. Zur effizienten Ausführung dieser Aufgaben steht in Authentic Desktop ein integrierter Skript-Editor zur Verfügung. Eine ausführliche Beschreibung der im Skript-Editor verfügbaren Funktionalitäten und eine Anleitung zur Verwendung der Scripting-Umgebung finden Sie im Abschnitt Skript-Editor 297 dieser Dokumentation. Unterstützt werden die Programmiersprachen **JScript** und **VBScript**.

IDE Plug-ins: Erstellen von Plug-ins für Authentic Desktop

Sie haben in Authentic Desktop die Möglichkeit Ihre eigenen Plug-Ins zu schreiben und in Authentic Desktop zu integrieren. Dazu können Sie die spezielle Authentic Desktop Schnittstelle für Plug-ins verwenden. Eine Beschreibung zum Erstellen von Plug-ins finden Sie im Abschnitt Authentic Desktop IDE Plug-ins 325.

An die meisten Methoden, die von einem IDE Plug-in implementiert werden müssen und das von der Applikation aufgerufen wird, wird ein Applikationsobjekt übergeben. Häufig werden zur Implementierung eines IDE Plug-in die Programmiersprachen **C#** und **C++** verwendet. Nähere Informationen dazu finden Sie im Abschnitt Authentic Desktop IDE Plug-ins 325.

Externe Programme

Außerdem können Sie Authentic Desktop mittels externer Skripts bedienen. So könnten Sie z.B. ein Skript schreiben, um Authentic Desktop zu einem bestimmten Zeitpunkt zu öffnen, dann eine XML-Datei in Authentic Desktop zu öffnen, die Datei zu validieren und auszudrucken. Externe Skripts würden sich zur Ausführung dieser Aufgaben wiederum der Applikations-API bedienen. Eine Beschreibung der Applikations-API finden Sie im Abschnitt Applikations-API[341].

Um die Applikations-API außerhalb von Authentic Desktop verwenden zu können, muss zuerst eine Instanz von Authentic Desktop gestartet werden. Wie dies erfolgt, hängt von der verwendeten Programmiersprache ab. Nähere Informationen zu den einzelnen Sprachen finden Sie unter Programmiersprachen[343].

Im Wesentlichen wird Authentic Desktop über seine COM Registrierung gestartet. Anschließend wird das mit der Authentic Desktop-Instanz verknüpfte`Application` Objekt zurückgegeben. Je nach COM-Einstellungen kann ein mit einer bereits laufenden Authentic Desktop-Instanz verknüpftes Objekt zurückgegeben werden. Es kann jede Programmiersprache verwendet werden, die die Erstellung und den Aufruf von COM-Objekten unterstützt. Die am häufigsten dazu verwendeten Sprachen sind unten aufgelistet.

- JScript[364] und VBScript[349] Skript-Dateien haben eine einfache Syntax und dienen dazu COM-Objekte aufzurufen. Sie können direkt über eine DOS-Befehlszeile oder durch Doppelklick im Windows Explorer ausgeführt werden. Am besten eignen sich diese Sprachen für einfache Automationsaufgaben.
- C#[352] ist eine umfangreiche Programmiersprache mit einer breiten Palette an Funktionalitäten. Der Zugriff auf COM-Objekte kann in C# automatisch über Wrapper erfolgen.
- C++ ermöglicht die direkte Steuerung des COM-Zugriffs. Allerdings muss dafür im Vergleich zu anderen Sprachen relativ viel Code geschrieben werden.
- Java[364]: Im Lieferumfang von Altova-Produkten sind native Java-Klassen inkludiert, die als Wrapper für die Applikations-API verwendet werden und dadurch eine Java-Umgebung ermöglichen.
- Weitere nützliche Alternativen sind Visual Basic für Applikationen, Perl und Python.

ActiveX Integration

Ein Sonderfall ist der Aufruf der Applikations-API über das Authentic Desktop ActiveX Control. Diese Funktion steht nur zur Verfügung, wenn das Authentic Desktop Integration Package[619] installiert ist. Jedes ActiveX Control hat eine Eigenschaft, die das entsprechende COM-Objekt für die zugrunde liegende Funktion zurückgibt. Das Manager Control liefert ein `Application` Object, das Document Control ein `Document` Object und das Platzhalterobjekt gibt, wenn es die Projektstruktur enthält, das `Project` Objekt zurück. Die von diesen Objekten unterstützten Methoden sind genau diegleichen, wie im Abschnitt "Schnittstellen" des Abschnitts[372] Applikations-API[341] beschrieben. Achten Sie darauf, keine Methoden zu verwenden, die keinen Sinn im Zusammenhang mit der ActiveX Control Integration ergeben. Nähere Informationen dazu finden Sie unter ActiveX Integration[619].

Informationen über die Referenz für Programmierer

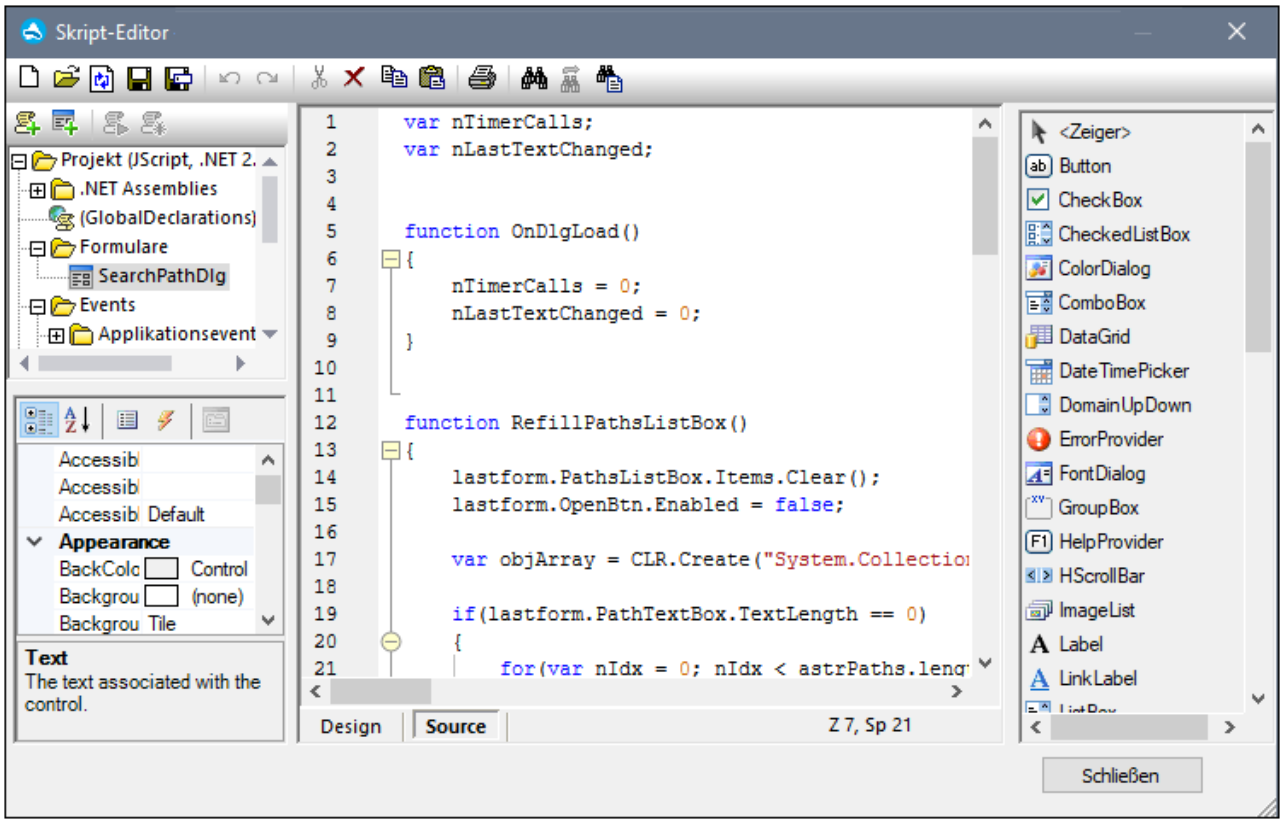
Die Authentic Desktop Referenz für Programmierer besteht aus den folgenden Abschnitten:

- Skript-Editor[297]: eine Benutzerreferenz für die Skripting-Umgebung in Authentic Desktop
- IDE Plug-ins[325]: eine Anleitung zum Erstellen von Plug-ins für Authentic Desktop
- Applikations-API[341]: eine Referenz für die Applikations-API
- ActiveX Integration[619]: Anleitung und Referenz zur Integration der Benutzeroberfläche und Funktionalitäten von XMLSpy mit Hilfe eines ActiveX Control.

© 2016-2022 Altova GmbH

14.1 Skript-Editor

Der Skript-Editor ist eine in Authentic Desktop integrierte Entwicklungsumgebung, über die Sie die Funktionalitäten von Authentic Desktop mit Hilfe von JScript- oder VBScript-Skripts anpassen können. So können Sie z.B. einen neuen Menüeintrag zur Durchführung einer benutzerdefinierten Projektaufgabe hinzufügen oder Sie können festlegen, dass jedes Mal, wenn ein Dokument in Authentic Desktop geöffnet oder geschlossen wird, ein bestimmtes Verhalten ausgelöst wird. Zu diesem Zweck werden Skripting-Projekte - Dateien mit der Erweiterung .asprj (Altova Skripting-Projekt) - erstellt.



Skript-Editor

Skripting-Projekte enthalten normalerweise ein oder mehrere Makros, d.h. Programme, die bei Aufruf verschiedene benutzerdefinierte Aufgaben ausführen. Makros können entweder explizit über einen Menübefehl (oder ggf. eine Symbolleisten-Schaltfläche) gestartet werden oder so konfiguriert werden, dass sie beim Start von Authentic Desktop automatisch ausgeführt werden. Die Skripting-Umgebung ist auch mit der Authentic Desktop COM API integriert. So können Ihre VBScript- oder JScript-Skripts etwa Applikations- oder Dokument-Ereignisse wie das Starten oder Beenden von Authentic Desktop, das Öffnen oder Schließen eines Projekts, usw. behandeln. Skripting-Projekte können Windows-Formulare enthalten, die Sie ähnlich wie in Visual Studio visuell entwerfen können. Zusätzlich dazu steht eine Reihe von vordefinierten Befehlen, die Ihnen dabei helfen, .NET-Klassen aus VBScript oder JScript-Code zu instantiieren und zu verwenden zur Verfügung.

Sobald Ihr Scripting-Projekt fertig gestellt wurde, können Sie es in Authentic Desktop entweder global oder nur für bestimmte Projekte aktivieren.

Für den Skript-Editor muss vor der Installation von Authentic Desktop .NET Framework 2.0 oder höher installiert sein.

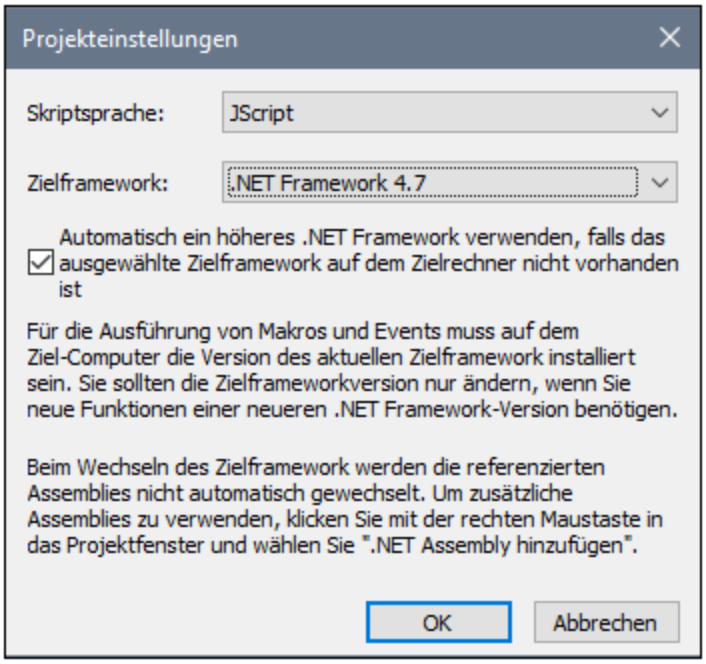
14.1.1 Erstellen eines Skripting-Projekts

Alle im Skript-Editor erstellten Skripting-Informationen werden in Altova Skripting-Projekten (.asprj-Dateien) gespeichert. Ein Skripting-Projekt kann Makros, Applikation Event Handler und Formulare (die wiederum ihre eigenen Event Handler haben können) enthalten. Zusätzlich dazu können Sie globale Variablen und Funktionen zu einem "Global Declarations"-Skript hinzufügen, sodass diese Variablen und Funktionen im gesamten Projekt aufgerufen werden können.

Um ein neues Projekt zu beginnen, klicken Sie auf den Menübefehl **Extras | Skript-Editor**.

Für die Verwendung in einem Skripting-Projekt werden die Sprachen JScript und VBScript unterstützt (nicht zu verwechseln mit Visual Basic, welches nicht unterstützt wird). Diese Skripting-Prozessoren stehen standardmäßig in Windows zur Verfügung. Es müssen keine speziellen Voraussetzungen erfüllt werden, um diese Sprachen auszuführen. Folgendermaßen können Sie eine Skripting-Sprache auswählen:

1. Klicken Sie mit der rechten Maustaste im linken oberen Bereich auf den Eintrag **Projekt** und wählen Sie im Kontextmenü den Befehl **Projekteinstellungen** aus.
2. Wählen Sie eine Sprache (JScript oder VBScript) aus und klicken Sie auf **OK**.



Über das oben gezeigte Dialogfeld "Projekteinstellungen" können Sie auch die .NET-Zielframework-Version ändern. Normalerweise ist dies dann notwendig, wenn für Ihr Skripting-Projekt Funktionen benötigt werden, die in einer neueren .NET Framework-Version zur Verfügung stehen. Beachten Sie, dass dieselbe .NET Framework-Version (oder eine höhere kompatible Version) auch auf allen Clients, auf denen Ihr Skripting-Projekt verwendet wird, installiert sein muss.

 © 2016-2022 Altova GmbH

Ein Skripting-Projekt referenziert standardmäßig mehrere .NET Assemblys wie `System`, `System.Data`, `System.Windows.Forms` und andere. Falls nötig, können Sie weitere .NET Assemblys, darunter Assemblys aus .NET Global Assembly Cache (GAC) oder benutzerdefinierte .dll-Dateien importieren. Assemblys können folgendermaßen importiert werden:

1. Statisch durch manuelles Hinzufügen zum Projekt. Klicken Sie mit der rechten Maustaste im linken oberen Bereich auf **Projekt** und wählen Sie im Kontextmenü den Befehl **.NET Assembly hinzufügen**.
2. Dynamisch zur Laufzeit durch Aufruf des Befehls `CLR.LoadAssembly` 315 über den Code.

Gegebenenfalls können Sie mehrere Skripting-Projekte erstellen. Sie können ein Skripting-Projekt auf der Festplatte speichern und es später wieder über die Windows-Standard-Symbolleisten-Schaltflächen **Neu, Öffnen, Speichern, Speichern unter** in den Skript-Editor laden. Nachdem Sie das Skripting-Projekt getestet haben und es bereitgestellt werden kann, können Sie es in Authentic Desktop laden und jedes beliebige seiner Makros und Event Handler ausführen. Nähere Informationen dazu finden Sie unter Aktivieren von Skripts und Makros 322.

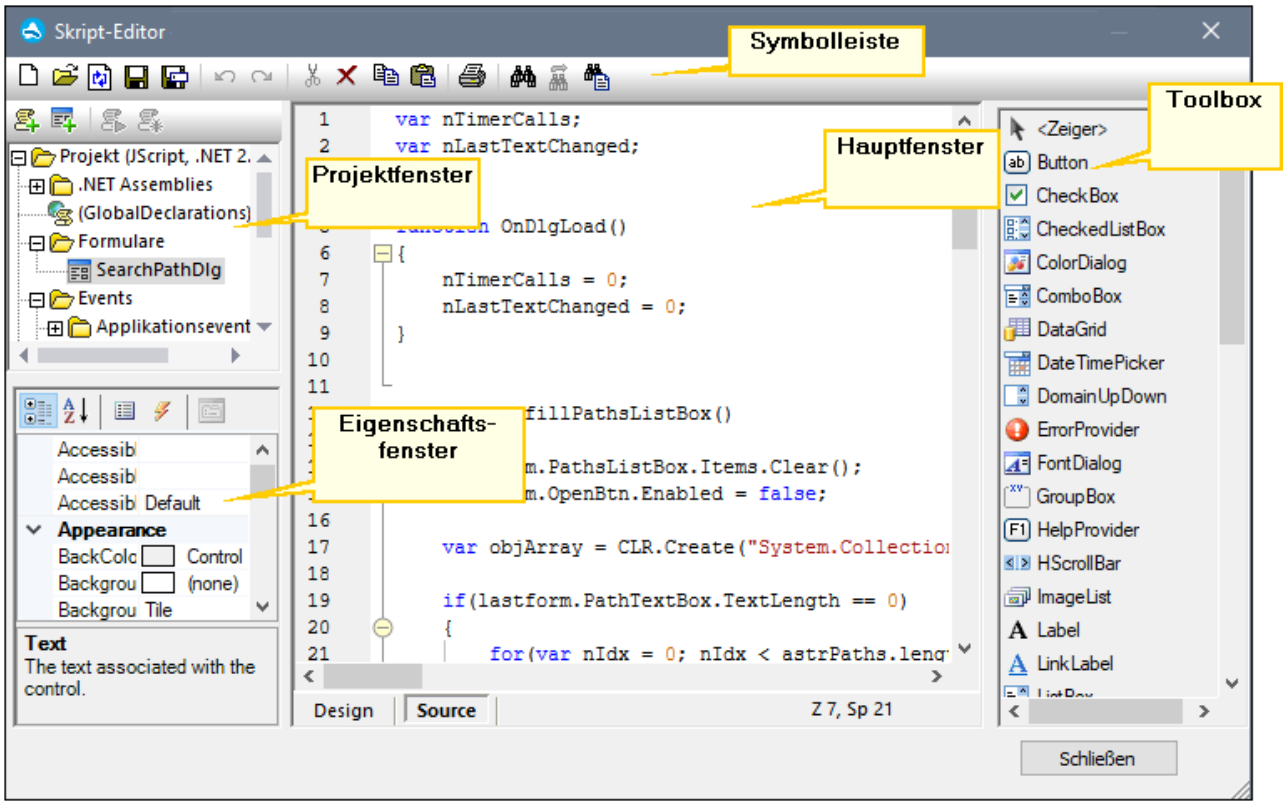
Ein Beispiel für ein Skripting-Projekt finden Sie unter dem folgenden Pfad: **C:\Benutzer\<Benutzer>\Dokumente\Altova\Authentic2023\AuthenticExamples\SampleScripts.asprj**.

In den nächsten Abschnitten wird beschrieben, welche Teile in Ihrem Skripting-Projekt eventuell benötigt werden: Globale Deklarationen, Makros, Formulare und Events.

14.1.1.1 Übersicht über die Umgebung

Der Skript-Editor besteht aus den folgenden Teilen:

- Symbolleiste
- Projektfenster
- Eigenschaftsfenster
- Hauptfenster
- Toolbox



Symbolleiste

Die Symbolleiste enthält Windows-Standardbefehle für die Dateiverwaltung (**Neu**, **Öffnen**, **Speichern**, **Speichern unter**) sowie Bearbeitungsbefehle (**Kopieren, Ausschneiden, Löschen, Einfügen**). Bei der Bearbeitung von Quellcode stehen zusätzlich die Befehle **Suchen** und **Ersetzen** sowie **Drucken** zur Verfügung.

Projektfenster

Im Projektfenster können Sie die Struktur des Projekts anzeigen und verwalten. Ein Skripting-Projekt besteht aus mehreren Komponenten, die gemeinsam verwendet und in jeder beliebigen Reihenfolge erstellt werden können.

- Ein Skript *"Global Declarations"*. Wie der Name schon sagt, sind in diesem Skript Informationen gespeichert, die global im gesamten Projekt zur Verfügung stehen. Sie können in diesem Skript alle Variablen oder Funktionen, die in alle Formularen, Event Handlern und Makros zur Verfügung stehen sollen, deklarieren.
- *Formulare.* Formulare dienen normalerweise zum Erfassen von Benutzer-Eingaben oder stellen Informationsdialogfelder zur Verfügung. Ein Formular wird entweder durch einen Aufruf innerhalb einer Funktion (im Globale Deklaration-Skript) oder durch einen Aufruf direkt in einem Makro aufgerufen.
- *Events.* In Ordner "Events" werden Authentic Desktop Applikations-Events aus der COM API angezeigt. Um ein Skript zu schreiben, das bei Auftreten eines Events ausgeführt werden soll, doppelklicken Sie auf ein beliebiges Event und geben Sie anschließend den behandelnden Code in den Editor ein. Die Applikations-Events sind nicht mit Formular-Events zu verwechseln; letztere werden auf Formularebene behandelt, wie weiter unten beschrieben.

 © 2016-2022 Altova GmbH

- *Makros*. Ein Makro ist ein Skript, das entweder bei Bedarf über ein Kontextmenü aufgerufen werden kann oder automatisch beim Start von Authentic Desktop ausgeführt wird. Makros haben keine Parameter oder Rückgabewerte. Ein Makro kann alle im Global Declarations-Skript deklarierten Variablen und Funktionen aufrufen und auch Formulare anzeigen.

Klicken Sie mit der rechten Maustaste auf eine der Komponenten, um die verfügbaren Kontextmenübefehle und deren Tastenkürzel zu sehen. Doppelklicken Sie auf eine beliebige Datei (wie z.B. ein Formular oder ein Skript), um diese(s) im Hauptfenster zu öffnen.

Über die Symbolleisten-Schaltflächen stehen die folgenden Schnellbefehle zur Verfügung:

	Neues Makro	Fügt im Verzeichnis "Makros" ein neues Makro zum Projekt hinzu.
	Neues Formular	Fügt im Verzeichnis "Formulare" ein neues Formular zum Projekt hinzu.
	Makro ausführen	Führt das ausgewählte Makro aus.
	Makro debuggen	Führt das ausgewählte Makro im Debug-Modus aus.

Eigenschaftsfenster

Das Eigenschaftsfenster ist dem Eigenschaftsfenster in Visual Studio sehr ähnlich. Darin werden die folgenden Elemente angezeigt:

- Wenn ein Formular ausgewählt ist, Formulareigenschaften
- Wenn ein Objekt in einem Formular ausgewählt ist, Objekteigenschaften
- Wenn ein Formular ausgewählt ist, Formular-Events
- Wenn ein Objekt in einem Formular ausgewählt ist, Objekt-Events

Um zwischen den Eigenschaften und Events der ausgewählten Komponente hin- und herzuwechseln, klicken Sie auf die Schaltfläche **Properties** bzw. auf die Schaltfläche **Events** .

Mit den Schaltflächen **Categorized** und **Alphabetical** werden die Eigenschaften oder Events, entweder nach Kategorie oder in aufsteigender alphabetischer Reihenfolge geordnet.

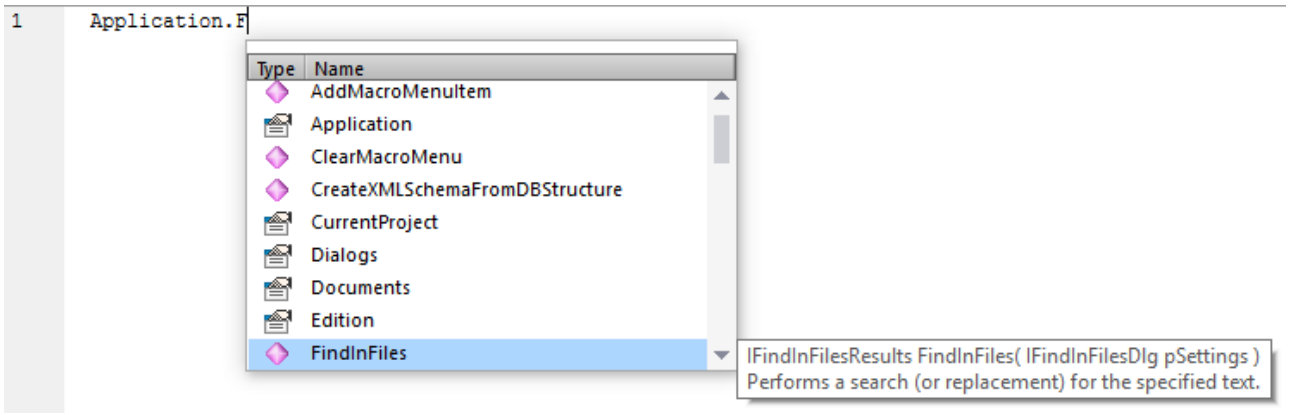
Wird eine Eigenschaft oder ein Event ausgewählt, wird am unteren Rand des Eigenschaftsfensters eine kurze Beschreibung angezeigt.

Hauptfenster

Das Hauptfenster ist der Arbeitsbereich, in den Sie Quellcode eingeben oder das Design des Formulars bearbeiten können. Bei der Bearbeitung von Formularen können Sie auf zwei Registern arbeiten: dem Register **Design** und dem Register **Source**. Auf dem Register **Design** sehen Sie das Layout des Formulars, während das Register **Source** den Quellcode wie z.B. Handler-Methoden für die Formular-Events enthält.

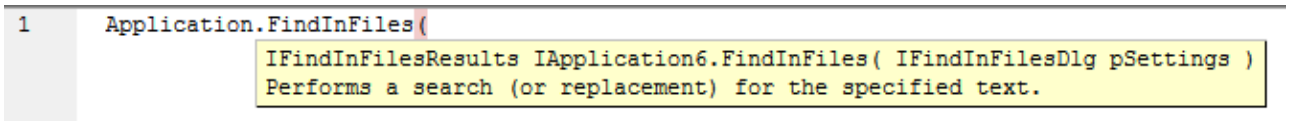
Der Quellcode-Editor bietet Codebearbeitungshillfen wie Syntaxfärbung, Quellcode-Klappleiste, Markierung von öffnenden und schließenden geschwungenen Klammern, Zoomen, Autokomplettierungsvorschläge, Lesezeichen.

Autokomplettierungsvorschläge
JScript und VBScript sind typenlose Sprachen, daher ist die Autokomplettierung auf COM API-Namen und vordefinierte Authentic Desktop-Befehle 312 beschränkt. Die vollständige Methode oder Eigenschaftssignatur wird neben der Autokomplettierungseingabehilfe angezeigt.



Wenn Namen mit `objDocument`, `objProject`, `objXMLData`, oder `objAuthenticRange` beginnen, werden Mitglieder der entsprechenden Schnittstelle angezeigt.

Wenn Sie den Mauszeiger über eine bekannte Methode oder Eigenschaft platzieren, wird deren Signatur (und, falls vorhanden, Dokumentation) angezeigt, z.B:



Die Autokomplettierungseingabehilfe wird normalerweise automatisch bei der Bearbeitung angezeigt, kann aber auch jederzeit durch Drücken von **Strg+Leerzeichen** angezeigt werden.

Lesezeichen
- Um ein Lesezeichen zu setzen oder zu entfernen, klicken Sie in eine Zeile und drücken Sie **Strg+F2.**
- Um zum nächsten Lesezeichen zu gelangen, drücken Sie **F2.**
- Um zum vorhergehenden Lesezeichen zu gelangen, drücken Sie **Umschalt+F2.**
- Um alle Lesezeichen zu löschen, drücken Sie **Strg+Umschalt+F2.**

Vergrößern/Verkleinern
- Um etwas zu vergrößern oder zu verkleinern, halten Sie die Strg-Taste gedrückt und drücken Sie anschließend die "+" oder "-" Taste oder drehen Sie das Mausrad.

Einstellungen für Textansicht
Um die Texteinstellungen aufzurufen, klicken Sie mit der rechten Maustaste in den Editor und wählen Sie im Kontextmenü den Befehl **Einstellungen für Textansicht.**

 © 2016-2022 Altova GmbH

Schriftarteinstellungen
Um die Schriftart zu ändern, klicken Sie mit der rechten Maustaste in den Editor und wählen Sie im Kontextmenü den Befehl **Schriftart Textansicht**.

Toolbox

Die Toolbox enthält alle Objekte, die für das Formulardesign zur Verfügung stehen, wie z.B. Schaltflächen, Textfelder, Auswahllisten, usw.

So fügen Sie ein Element aus der Toolbox zu einem Formular hinzu:

1. Erstellen oder öffnen Sie ein Formular und stellen Sie sicher, dass das Register **Design** ausgewählt ist.
2. Klicken Sie auf das Toolbox-Objekt (z.B.**Button**) und anschließend auf die Stelle im Formular, an der Sie es einfügen möchten. Alternativ dazu können Sie das Objekt auch mit der Maus direkt in das Formular ziehen.

Einige Objekte wie z.B. `Timer` werden nicht zum Formular hinzugefügt, sondern in der Ablageunteren Bereich des Hauptfensters angezeigt. Sie können das Objekt dort auswählen und über das Eigenschaftsfenster dafür Eigenschaften und Event Handler definieren. Ein Beispiel für die Behandlung solcher Komponenten über den Code finden Sie unter Behandeln von Formular-Events 305.

Sie können auch registrierte ActiveX Controls zum Formular hinzufügen. Klicken Sie dazu mit der rechten Maustaste in den Toolbox-Bereich und wählen Sie im Kontextmenü den Befehl **ActiveX Control hinzufügen.**

14.1.1.2 Globale Deklarationen

Das Skript "Global Declarations" ist standardmäßig in jedem Skripting-Projekt vorhanden und muss nicht explizit erstellt werden. Alle Variablen oder Funktionen, die Sie zu diesem Skript hinzufügen, werden als global für das gesamte Projekt behandelt, sodass Sie solche Variablen und Funktionen von jedem beliebigen Makro oder Event des Projekts aus referenzieren können. Im Folgenden sehen Sie ein Beispiel für ein Global Declarations-Skript, das den Namespace `System.Windows.Forms` in das Projekt importiert. Der unten stehende Code ruft zu diesem Zweck den im Skript-Editor vordefinierten Befehl `CLR.Import` auf.

```
// import System.Windows.Forms namespace for all macros, forms and events:
CLR.Import( "System.Windows.Forms" );
```

Anmerkung: Die globalen Deklarationen werden jedes Mal, wenn ein Makro ausgeführt oder ein Event Handler aufgerufen wird, erneut initialisiert.

14.1.1.3 Makros

Makros sind Skripts, die JScript-Anweisungen (oder, je nach Projektsprache, VBScript-Anweisungen) wie z.B. Variablendeklarationen und Funktionen enthalten.

Wenn in Ihren Projekten Makros verwendet werden sollen, können Sie diese folgendermaßen hinzufügen: Klicken Sie mit der rechten Maustaste in das Projektfenster, wählen Sie im Kontextmenü den Befehl Makro

hinzufügen aus und geben Sie den Code des Makros in das Hauptformular ein. Beim Makrocode kann es sich z.B. um eine einfache Warnmeldung handeln:

```
alert("Hello, I'm a macro!");
```

Komplexere Makros können Variablen und lokale Funktionen enthalten. Makros können auch Code, der Formulare aus dem Projekt aufruft, enthalten. Im Codefragment unten sehen Sie ein Beispiel für ein Makro, das ein Formular aufruft. Es wird davon ausgegangen, dass dieses Formular im Ordner "Formulare" bereits erstellt wurde und den Namen "SampleForm" hat, siehe auch Formulare 304.

```
// display a form
ShowForm( "SampleForm" );
```

`ShowForm` im obigen Codefragment ist ein im Skript-Editor vordefinierter Befehl. Eine Liste anderer ähnlicher Befehle, die für die Arbeit mit Formularen und .NET-Objekten zur Verfügung stehen, finden Sie unter Vordefinierte Befehle 312.

Sie können mehrere Makros zum selben Projekt hinzufügen und jedes beliebige Makro als "automatisches Makro" definieren. Wenn ein Makro als automatisches Makro definiert ist, wird es beim Start von Authentic Desktop automatisch ausgeführt. Um ein Makro als automatisches Makro zu definieren, klicken Sie mit der rechten Maustaste darauf und wählen Sie im Kontextmenü den Befehl **Als automatisches Makro definieren**.

Es kann immer nur ein Makro gleichzeitig ausgeführt werden. Nachdem ein Makro (oder Event) ausgeführt wurde, wird das Skript geschlossen und die globalen Variablen verlieren ihre Werte.

Um ein Makro direkt im Skript-Editor auszuführen, klicken Sie auf **Makro ausführen**. Um ein Makro mit dem Visual Studio Debugger zu debuggen, klicken Sie auf **Makro debuggen**. Informationen darüber, wie Sie in Authentic Desktop Makros aktivieren und ausführen, finden Sie unter Aktivieren von Skripts und Makros 322.

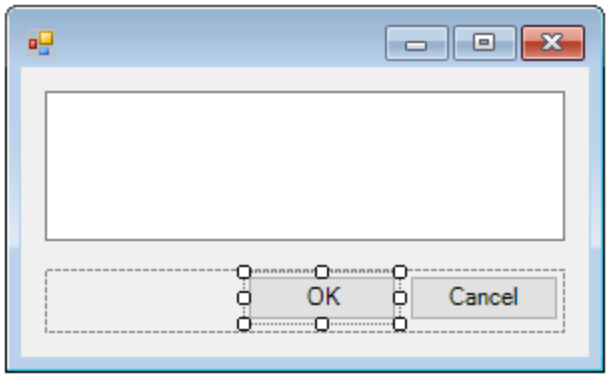
14.1.1.4 Formulare

Formulare eigenen sich vor allem, um Eingabedaten von Benutzern zu erfassen oder Benutzern Daten anzuzeigen. Ein Formular kann diverse Steuerelemente wie z.B. Schaltflächen, Kontrollkästchen, Auswahllisten, usw. enthalten.

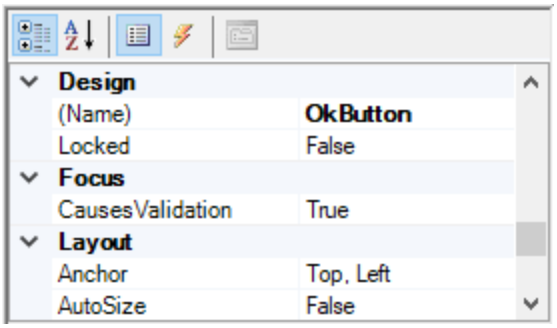
Um ein Formular hinzuzufügen, klicken Sie im Projektfenster mit der rechten Maustaste auf das Formular und wählen Sie im Kontextmenü den Befehl **Formular hinzufügen**. Um ein Steuerelement zu einem Formular hinzuzufügen, ziehen Sie es aus der Toolbox auf der rechten Seite des Skript-Editors in das Formular.

Sie können die Position und Größe der Steuerelemente direkt im Formular über die Ziehpunkte, die angezeigt werden, wenn Sie auf das Steuerelement klicken, ändern, z.B:

 © 2016-2022 Altova GmbH



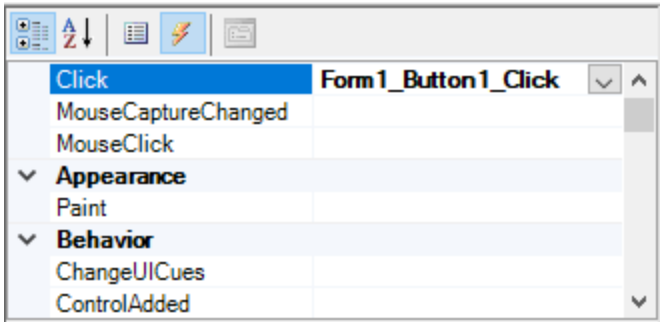
Alle Formularsteuerelemente haben Eigenschaften, die Sie im Eigenschaftsfenster mühelos anpassen können. Wählen Sie dazu das Steuerelement zuerst im Formular aus und bearbeiten Sie anschließend die gewünschten Eigenschaften im Eigenschaftsfenster.



Behandeln von Formular-Events

Jedes Formularsteuerelement stellt außerdem verschiedene Events bereit, an die Ihr Skripting-Projekt gebunden werden kann. So können Sie z.B. jedes Mal, wenn ein Benutzer auf eine Schaltfläche klickt, eine bestimmte Authentic Desktop COM API-Methode aufrufen lassen. Um eine Funktion zu erstellen, die an ein Formular-Event gebunden ist, gehen Sie folgendermaßen vor:

1. Klicken Sie im Eigenschaftsfenster auf die Schaltfläche **Events** .
2. Doppelklicken Sie in der Spalte **Action** auf das Event, für das Sie die Methode benötigen (in der Abbildung unten ist dies "Click").



Sie können Handler-Methoden auch durch Doppelklick auf ein Steuerelement im Formular hinzufügen. Wenn Sie z.B. im Formulardesign auf eine Schaltfläche doppelklicken, wird eine Handler-Methode für das "Click"-Event dieser Schaltfläche erstellt.

Nachdem der Rumpf der Handler-Methode generiert wurde, können Sie Code eingeben, der dieses Event behandelt, z.B:

```
//Occurs when the component is clicked.
function MyForm_ButtonClick( objSender, e_EventArgs )
{
   alert("A button was clicked");
}
```

Um ein Formular, an dem Sie gerade arbeiten, getrennt vom Skript-Editor anzuzeigen, klicken Sie mit der rechten Maustaste im Projektfenster auf das Formular und wählen Sie im Kontextmenü den Befehl **Formular testen**. Beachten Sie, dass mit dem Befehl **Formular testen** nur das Formular angezeigt wird; die Events des Formulars (wie z.B. Schaltflächenklicks) sind weiterhin deaktiviert. Damit das Formular auf Events reagiert, rufen Sie es von einem Makro aus auf, z.B:

```
// Instantiate and display a form
ShowForm( "SampleForm" );
```

Aufrufen von Formularsteuerelementen

Sie können alle Komponenten eines Formulars von Ihrem Code aus mit Hilfe von Feldaufrufsyntax aufrufen. Angenommen, Sie haben das folgenden Formular:

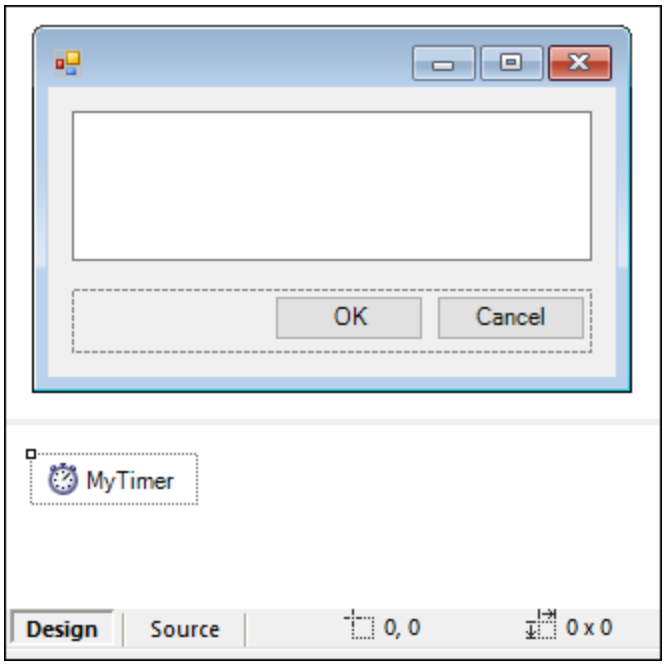
```
// MyForm
//   ButtonPanel
//      OkButton
//      CancelButton
//   TextEditor
//      AxMediaPlayer1
// TrayComponents
//   MyTimer
```

Im unten stehenden Code sehen Sie, wie das Formular instantiiert wird, einige seiner Steuerelemente mittels Feldaufrufsyntax aufgerufen und das Formular anschließend angezeigt wird.

 © 2016-2022 Altova GmbH

```
// Instantiate the form
var objForm = CreateForm("MyForm");
// Disable the OK button
objForm.ButtonPanel.OkButton.Enabled = false;
// Change the text of TextEditor
objForm.TextEditor.Text = "Hello";
// Show the form
objForm.ShowDialog();
```

Wenn Sie bestimmte Steuerelemente wie Timer zum Formular hinzufügen, werden diese nicht im Formular, sondern als Komponenten der Ablage am unteren Rand des Formulardesigns angezeigt, z.B:



Mit Hilfe der Methode `GetTrayComponent` können Sie Steuerelemente aus der Ablage aufrufen und den Namen des Steuerelements als Argument bereitstellen. Um in diesem Beispiel eine Referenz zu `MyTimer` zu erhalten und zu aktivieren, verwenden Sie den folgenden Code:

```
var objTimer = objForm.GetTrayComponent("MyTimer");
objTimer.Enabled = true;
```

Bei ActiveX Controls können Sie das zugrunde liegende COM-Objekt über die OCX-Eigenschaft aufrufen:

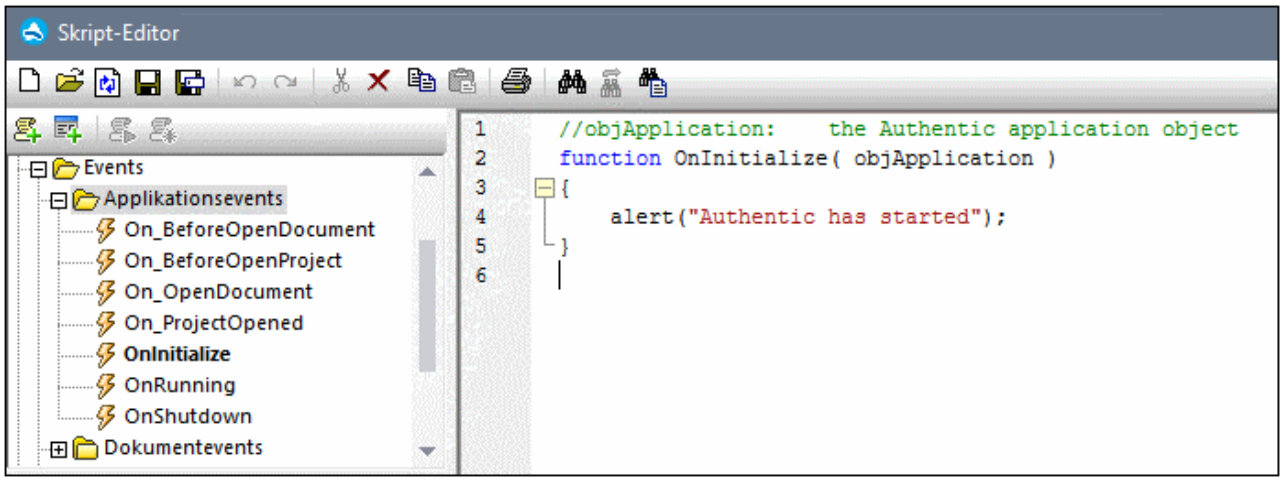
```
var ocx = lastform.AxMediaPlayer1.OCX; // get underlying COM object
ocx.enableContextMenu = true;
ocx.URL = "mms://apasf.apa.at/fm4_live_worldwide";
```

14.1.1.5 Events

Ihr Skripting-Projekt kann optional Skripts enthalten, die Authentic Desktop Events wie das Öffnen, Schließen oder Speichern eines Dokuments, das Starten oder Schließen von Authentic Desktop, das Hinzufügen eines Elements zu einem Diagramm und andere behandeln. Diese Events werden von der Authentic Desktop COM API bereitgestellt. Sie finden diese Events im Ordner "Ereignisse" Ihres Skripting-Projekts. Beachten Sie, dass diese Events im Gegensatz zu Formular-Events Authentic Desktop-spezifisch sind. Die Events befinden Sie in den folgenden Ordnern:

- Applikationsevents
- Dokumentevents
- Events der Authentic-Ansicht
- Events der Grid-Ansicht
- Events der Textansicht

Um ein Event Handler-Skript zu erstellen, klicken Sie mit der rechten Maustaste auf ein Event und wählen Sie im Kontextmenü den Befehl **Öffnen** (oder doppelklicken Sie auf das Event). Daraufhin wird das Event Handler-Skript im Hauptfenster, wo Sie es nun bearbeiten können, angezeigt. So zeigt z.B. der unten gezeigte Event Handler bei jedem Start von Authentic Desktop eine entsprechende Benachrichtigung an:



Beachten Sie die folgenden Punkte:

- Der Befehl `alert` ist auf JScript anwendbar. Das VBScript-Äquivalent dazu ist `MsgBox`. Siehe auch alert 313.
- Der Name der Event Handler-Funktion darf nicht geändert werden, da das Event Handler-Skript sonst nicht aufgerufen wird.
- Damit Events verarbeitet werden, muss das Kontrollkästchen **Ereignisse abarbeiten** ausgewählt sein, wenn Sie das Skripting-Projekt in Authentic Desktop aktivieren. Nähere Informationen dazu finden Sie unter Aktivieren von Skripts und Makros 322.

Sie können in Event Handler-Skripts optional lokale Variablen und Hilfsfunktionen definieren, z.B:

```
var local;
```

 © 2016-2022 Altova GmbH

```
function OnInitialize( objApplication )
{
   local = "OnInitialize";
   Helper();
}

function Helper()
{
   alert("I'm a helper function for " + local);
}
```

14.1.1.6 Tipps zur Programmierung mit JScript

Im Folgenden finden Sie einige Tipps zum Programmieren in JScript, die sich bei der Erstellung eines Skripting-Projekts im Authentic Desktop Skript-Editor als nützlich erweisen könnten.

Out-Parameter

Für Out-Parameter von Methoden des .NET Framework werden in JScript spezielle Variablen benötigt, z.B:

```
var dictionary =
CLR.Create("System.Collections.Generic.Dictionary<System.String,System.String>");
dictionary.Add("1", "A");
dictionary.Add("2", "B");

// use JScript method to access out-parameters
var strOut = new Array(1);
if ( dictionary.TryGetValue("1", strOut) ) // TryGetValue will set the out parameter
  alert( strOut[0] ); // use out parameter
```

Ganzzahl-Argumente

.NET-Methoden, für die Ganzzahl-Argumente benötigt werden, sollten nicht direkt mit JScript-Zahlenobjekten, die Gleitkommawerte enthalten, aufgerufen werden. Verwenden Sie z.B. anstelle von:

```
var objCustomColor = CLR.Static("System.Drawing.Color").FromArgb(128,128,128);
```

Folgendes:

```
var objCustomColor =
CLR.Static("System.Drawing.Color").FromArgb(Math.floor(128),Math.floor(128),Math.floor(12
8));
```

Iterieren über .NET Collections

Für die Iteration über .NET Collections können sowohl der JScript Enumerator als auch .NET-Iterierungstechnologien verwendet werden, z.B:

```
// iterate using the JScript iterator
var itr = new Enumerator( coll );
for ( ; !itr.atEnd(); itr.moveNext() )
  alert( itr.item() );

// iterate using the .NET iterator
var itrNET = coll.GetEnumerator();
while( itrNET.MoveNext() )
  alert( itrNET.Current );
```

.NET Templates

.NET Templates können, wie unten gezeigt, instantiiert werden:

```
var coll = CLR.Create( "System.Collections.Generic.List<System.String>" );
```

oder

```
CLR.Import( "System" );
CLR.Import( "System.Collections.Generic" );
var dictionary = CLR.Create( "Dictionary<String,Dictionary<String,String>>" );
```

.NET-Enumerationswerte

.NET-Enumerationswerte werden, wie unten gezeigt, aufgerufen:

```
var enumValStretch = CLR.Static( "System.Windows.Forms.ImageLayout" ).Stretch;
```

Enumerationsliterale

Die Enumerationsliterale aus der Authentic Desktop API können, wie unten gezeigt, aufgerufen werden (ihr numerischer Wert muss nicht bekannt sein).

```
objExportXMIFileDlg.XMIType = eXMI21ForUML23;
```

14.1.1.7 Skripting-Beispielprojekt

Unter dem folgenden Pfad steht ein Demo-Projekt, in dem die Skripterstellung mit Authentic Desktop gezeigt wird, zur Verfügung: **C:\Benutzer\<Benutzer>\Dokumente\Altova\Authentic2023\AuthenticExamples\SampleScripts.asprj**. Dieses Skripting-Projekt besteht aus einem Makro und einem Windows-Formular.

So laden Sie das Skripting-Projekt in den Skript-Editor:

1. Klicken Sie im Menü **Extras** auf **Skript-Editor**.
2. Klicken Sie auf **Öffnen** und navigieren Sie zur Datei **SampleScripts.asprj** unter dem obigen Pfad.

 © 2016-2022 Altova GmbH

Das Projekt enthält im Verzeichnis "Makros" eine Reihe von Makros.

Makro	Beschreibung
AddMacroMenu	Dieses Makro fügt durch Aufruf der Methode `Application.AddMacroMenuItem` der COM API einen neuen Menüeintrag zu Authentic Desktop hinzu. Das erste Argument der Methode `AddMacroMenuItem` ist der Name des hinzuzufügenden Makros (in diesem Beispiel "CloseAllButActiveDoc") und das zweite Argument ist der Anzeigetext für den Menüeintrag. Immer, wenn dieses Makro ausgeführt wird, wird ein neuer Menübefehl namens "CloseAllButActiveDoc") zum Menü **Extras** hinzugefügt. Um zuvor erstellte Menüeinträge zu löschen, starten Sie Authentic Desktop entweder neu oder erstellen Sie ein Makro, das die API-Methode `Application.ClearMacroMenu` aufruft.
CloseAllButActiveDocument	Wenn dieses Makro ausgeführt wird, iteriert es durch die offenen Dokumente in Authentic Desktop und schließt alle bis auf das aktive Dokument.
SearchPath	Dieses Makro zeigt ein Formular an, über das Benutzer im aktuellen Projekt nach Dateien suchen können. Das Formular steht im Verzeichnis "Formulare", wo Sie sein Design und die damit verknüpften Event Handler anzeigen können, zur Verfügung. Die Methode `GetAllPathsFromProject()` gibt alle Dateipfade, die zum aktuell geöffneten Projekt gehören als Array zurück. Die Definition dieser Methode befindet sich im **GlobalDeclarations**-Skript dieses Projekts. Die `InsertStringInArrayUnique`-Methode stellt sicher, dass nur eindeutige Pfade zum Array hinzugefügt werden. Als nächstes wird das Formular mit CreateForm 318 initialisiert. Schließlich wird das Array mit Hilfe der CLR.Create 314-Methode in einen .NET-Typ konvertiert und das Formular wird mit der erzeugten `ArrayList` Collection befüllt. Die Schaltfläche **Open** des Formulars hat einen Handler, der die API-Methode `Application.Documents.OpenFile` aufruft, um die aktuell ausgewählte Datei zu öffnen.

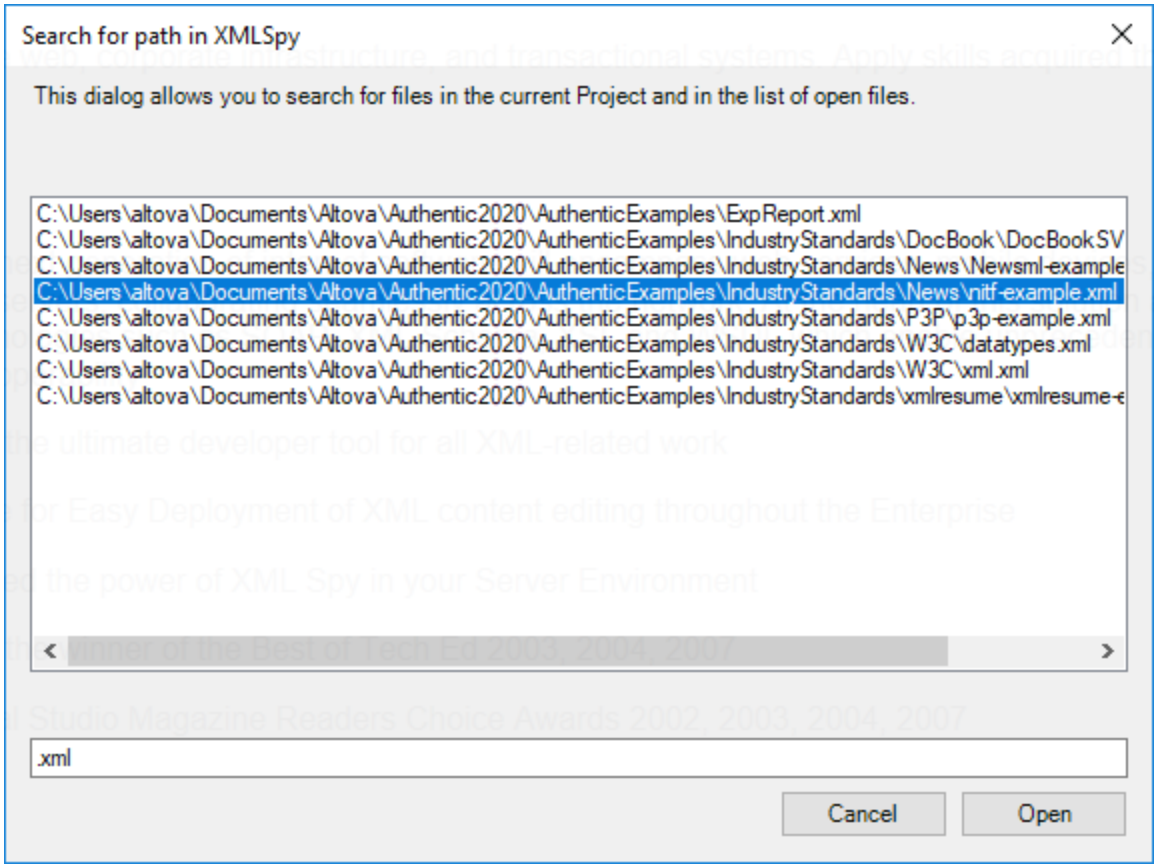
So aktivieren Sie das Skripting-Projekt als globales Authentic Desktop-Skripting-Projekt:

1. Klicken Sie im Menü **Extras** auf **Optionen**.
2. Klicken Sie auf das Register **Skripting-Umgebung**.
3. Klicken Sie unter "Globale Skripting-Projektdatei" auf **Durchsuchen** und wählen Sie die Datei **SampleScripts.asprj** aus dem obigen Pfad aus.
4. Dieses Skripting-Projekt hat keine automatischen Makros und Applikations-Event-Handler, daher müssen die Kontrollkästchen **Automatische Makros...ausführen** und **Ereignisse abarbeiten** nicht aktiviert werden.
5. Klicken Sie auf **Anwenden**.

Daraufhin stehen unter dem Menü **Extras | Makros** einige neue Menüeinträge (für jedes Makro einer) zur Verfügung.

So führen Sie das Makro "SearchPath" aus:

1. Öffnen Sie ein Authentic Desktop-Projekt, das mehrere Dateien enthält (in diesem Beispiel **C:\Benutzer\<Benutzer>\Dokumente\Altova\Authentic2023\AuthenticExamples\Examples.spp).**
2. Klicken Sie im Menü **Extras** auf **Makros** und anschließend auf **Search Path**.
3. Geben Sie den Suchbegriff ein (in diesem Beispiel ".xml").



Wie oben gezeigt, werden daraufhin alle Dateinamen, die den Suchbegriff enthalten, aufgelistet. Sie können auf jedes beliebige Element in der Liste und anschließend auf **Open** klicken, um es im Haupt-Editor anzuzeigen.

14.1.2 Vordefinierte Befehle

Dieser Abschnitt enthält eine Referenz aller Befehle, die Sie im Authentic Desktop Skript-Editor verwenden können.

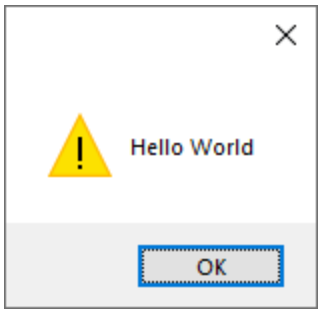
- alert 313
- confirm 313
- CLR.Create 314
- CLR.Import 315
- CLR.LoadAssembly 315

 © 2016-2022 Altova GmbH

- CLR.ShowImports 316
- CLR.ShowLoadedAssemblies 317
- CLR.Static 318
- CreateForm 318
- doevents 319
- lastform 320
- prompt 320
- ShowForm 321
- watchdog 322

14.1.2.1 alert

Zeigt ein Meldungsfeld mit einer angegebenen Meldung und der Schaltfläche "OK" an. Um fortzufahren, muss der Benutzer auf "OK" klicken.



Signatur

Für JScript lautet die Signatur:

```
alert(strMessage : String) -> void
```

Für VBScript lautet die Signatur:

```
MsgBox(strMessage : String) -> void
```

Beispiel

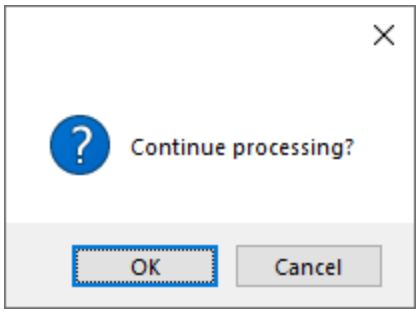
Mit dem folgenden JScript-Code wird ein Meldungsfeld mit dem Text "Hello World" angezeigt.

```
alert("Hello World");
```

14.1.2.2 confirm

Öffnet ein Dialogfeld, in dem eine angegebene Meldung, eine Bestätigungsschaltfläche und eine Abbrechen-Schaltfläche angezeigt werden. Der Benutzer muss entweder auf "OK" oder auf "Cancel" klicken, um fortfahren zu können. Der Rückgabewert ist ein Boolescher Wert, der für die Antwort des Benutzers steht. Wenn der

Benutzer auf "OK" klickt, gibt die Funktion **true** zurück, wenn der Benutzer auf "Cancel" klickt, gibt das Dialogfeld **false** zurück.



Signatur

```
confirm(strMessage : String) -> result : Boolean
```

Beispiel (JScript)

```
if ( confirm( "Continue processing?" ) == false )
  alert("You have cancelled this action");
```

Beispiel (VBScript)

```
If ( confirm( "Continue processing?" ) = false ) Then
  MsgBox("You have cancelled this action")
End If
```

14.1.2.3 CLR.Create

Erstellt eine neue .NET-Objektinstanz für den als Argument angegebenen Typnamen. Wenn mehr als ein Argument übergeben wird, werden die nachfolgenden Argumente als Argumente für den Konstruktor des .NET-Objekts interpretiert. Der Rückgabewert ist eine Referenz auf das erstellte .NET-Objekt.

Signatur

```
CLR.Create(strTypeNameCLR : String, constructor arguments ... ) -> object
```

Beispiel

Im folgenden JScript-Code wird gezeigt, wie Sie Instanzen von verschiedenen .NET-Klassen erstellen.

```
// Create an ArrayList
var objArray = CLR.Create("System.Collections.ArrayList");
// Create a ListViewItem
```

 © 2016-2022 Altova GmbH

```
var newItem = CLR.Create( "System.Windows.Forms.ListViewItem", "NewItemText" );
// Create a List<string>
var coll = CLR.Create( "System.Collections.Generic.List<System.String>" );
// Import required namespaces and create a Dictionary object
CLR.Import( "System" );
CLR.Import( "System.Collections.Generic" );
var dictionary = CLR.Create( "Dictionary< String, Dictionary< String, String > >" );
```

14.1.2.4 CLR.Import

Importiert einen Namespace. Dies ist das Skripting-Äquivalent zu den C# und VB.Net-Schlüsselwörtern `using` und `imports`. Damit können Sie den Namespace-Teil in aufeinander folgenden Aufrufen wie z.B. CLR.`Create()` und `CLR.Static()` auslassen.

Anmerkung: Durch den Import eines Namespace wird die entsprechende Assembly nicht zum Skripting-Projekt hinzugefügt bzw. geladen. Assemblys können mittels CLR.LoadAssembly 315 dynamisch (zur Laufzeit) im Quellcode eingefügt werden.

Signatur

```
CLR.Import(strNamespaceCLR : String) -> void
```

Beispiel

Anstatt vollständig qualifizierte Namespaces wie die folgenden verwenden zu müssen:

```
if ( ShowForm( "FormName" ) == CLR.Static( "System.Windows.Forms.DialogResult" ).OK )
{
    var sName = lastform.textboxFirstName.Text + " " + lastform.textboxLastName.Text;
    CLR.Static( "System.Windows.Forms.MessageBox" ).Show( "Hello " + sName );
}
```

können Sie Namespaces zuerst importieren und danach die Kurzform verwenden:

```
CLR.Import( "System.Windows.Forms" );

if ( ShowForm( "FormName" ) == CLR.Static( "DialogResult" ).OK )
{
    var sName = lastform.textboxFirstName.Text + " " + lastform.textboxLastName.Text;
    CLR.Static( "MessageBox" ).Show( "Hello " + sName );
}
```

14.1.2.5 CLR.LoadAssembly

Lädt die .NET Assembly mit dem angegebenen Assembly-Namen oder Dateipfad. Gibt den Booleschen Wert **true** zurück, wenn die Assembly geladen werden konnte, **false**, wenn nicht.

Signatur

```
CLR.LoadAssembly(strAssemblyNameCLR : String, showLoadErrors : Boolean) -> result :
Boolean
```

Beispiel

Mit dem folgenden JScript-Code wird versucht, den Text in der Zwischenablage durch dynamisches Laden der erforderlichen Assembly zu definieren.

```
// set clipboard text (if possible)
// System.Windows.Clipboard is part of the PresentationCore assembly, so load this
assembly first:
if ( CLR.LoadAssembly( "PresentationCore, Version=3.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35", true ) )
{
    var clipboard = CLR.Static( "System.Windows.Clipboard" );
    if ( clipboard != null )
        clipboard.SetText( "HelloClipboard" );
}
```

14.1.2.6 CLR.ShowImports

Öffnet ein Meldungsfeld, in dem die aktuell importierten Namespaces angezeigt werden. Der Benutzer muss zum Fortfahren auf OK klicken.

Signatur

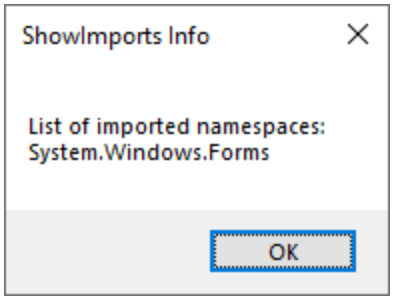
```
CLR.ShowImports() -> void
```

Beispiel

Mit dem folgenden JScript-Code wird zuerst ein Namespace importiert und anschließend die Liste der importierten Namespaces angezeigt:

```
CLR.Import( "System.Windows.Forms");
CLR.ShowImports();
```


© 2016-2022 Altova GmbH



14.1.2.7 CLR.ShowLoadedAssemblies

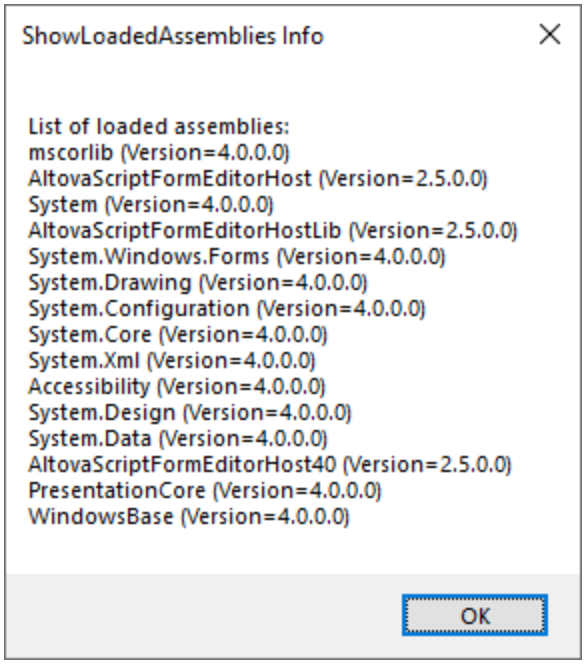
Öffnet ein Meldungsfeld, in dem die aktuell geladenen Assemblys angezeigt werden. Der Benutzer muss zum Fortfahren auf OK klicken.

Signatur

```
CLR.ShowLoadedAssemblies() -> void
```

Beispiel

```
CLR.ShowLoadedAssemblies();
```



14.1.2.8 CLR.Static

Gibt eine Referenz auf ein statisches .NET-Objekt zurück. Verschafft Zugriff auf .NET-Typen, die keine Instanzen haben und nur statische Mitglieder enthalten.

Signatur

```
CLR.Static(strTypeNameCLR : String) -> object
```

Beispiel (JScript)

```
// Get the value of a .NET Enum into a variable
var enumValStretch = CLR.Static( "System.Windows.Forms.ImageLayout" ).Stretch

// Set the value of the Windows clipboard
var clipboard = CLR.Static( "System.Windows.Clipboard" );
clipboard.SetText( "HelloClipboard" );

// Check the buttons pressed by the user on a dialog box
if ( ShowForm( "FormName" ) == CLR.Static( "System.Windows.Forms.DialogResult" ).OK )
    alert( "ok" );
else
    alert( "cancel" );
```

14.1.2.9 CreateForm

Instantiiert das `Form`-Objekt, das durch den als Argument bereitgestellten Namen identifiziert wird. Das Formular muss im Ordner "Formulare" des Skripting-Projekts vorhanden sein. Gibt das Formularobjekt (`System.Windows.Forms.Form`) des angegebenen Namens oder `null` zurück, wenn kein Formular dieses Namens vorhanden ist.

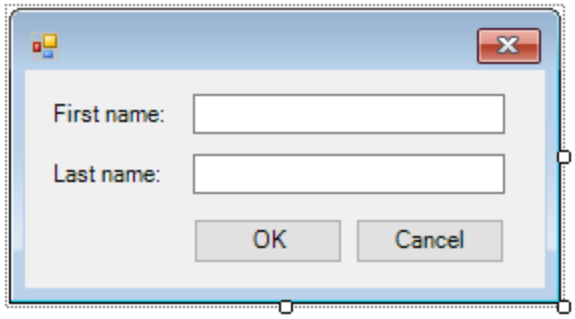
Signatur

```
CreateForm (strFormName : String) -> System.Windows.Forms.Form | null
```

Beispiel

Angenommen, es gibt im Skripting-Projekt ein Formular namens "FormName".

© 2016-2022 Altova GmbH



Der folgende JScript-Code instantiiert das Formular mit einige Standardwerten und zeigt es dem Benutzer an.

```
var myForm = CreateForm( "FormName" );
if ( myForm != null )
{
   myForm.textboxFirstName.Text = "Daniela";
   myForm.textboxLastName.Text = "Heidegger";
   var dialogResult = myForm.ShowDialog();
}
```

Das `dialogResult` kann anschließend folgendermaßen ausgewertet werden:

```
if ( dialogResult == CLR.Static( "System.Windows.Forms.DialogResult" ).OK )
    alert( "ok" );
else
    alert( "cancel" );
```

Anmerkung: Der oben gezeigte Code funktioniert nur, wenn die Eigenschaft **DialogResult** der Schaltflächen "OK" und "Cancel" über das Eigenschaftsfenster korrekt definiert wurde (für die Schaltfläche "OK" muss er z.B. **OK** sein).

14.1.2.10 doevents

Verarbeitet alle Windows-Meldungen, die sich derzeit in der Meldungswarteschlange befinden.

Signatur

```
doevents() -> void
```

Beispiel (JScript)

```
for ( i=0; i < nLongLastingProcess; ++i )
{
    // do long lasting process
```

```
    doevents(); // process Windows messages; give UI a chance to update
}
```

14.1.2.11 lastform

Dies ist ein globales Feld, das eine Referenz auf das zuletzt mit `CreateForm()` oder `ShowForm()` erstellte Formularobjekt zurückgibt.

Signatur

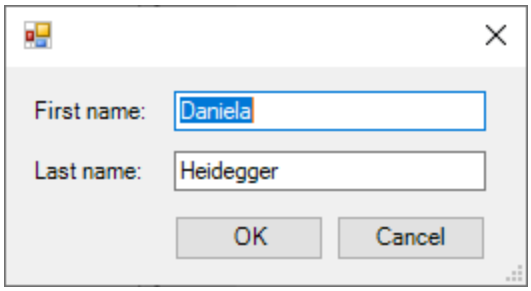
```
lastform -> formObj : System.Windows.Forms.Form
```

Beispiel

Mit dem folgenden JScript-Code wird das Formular "FormName" als Dialogfeld angezeigt.

```
CreateForm( "FormName" );
if ( lastform != null )
{
    lastform.textboxFirstName.Text = "Daniela";
    lastform.textboxLastName.Text = "Heidegger";
    var dialogResult = lastform.ShowDialog();
}
```

Die Werte beider Textfeld-Steuerelemente werden mit Hilfe von `lastform` initialisiert.



14.1.2.12 prompt

Öffnet ein Dialogfeld, in dem einem Meldung sowie ein Textfeld-Steuerelement mit einer Standardantwort angezeigt wird. Damit kann der Benutzer einen einfachen String-Wert eingeben. Der Rückgabewert ist ein String, der den Textfeldwert enthält oder Null, wenn der Benutzer auf "Cancel" geklickt hat.

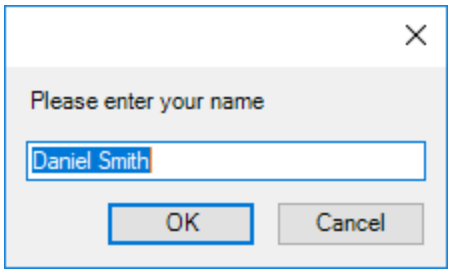
Signatur

```
prompt(strMessage : String, strDefault : String) -> val : String
```

 © 2016-2022 Altova GmbH

Beispiel

```
var name = prompt( "Please enter your name", "Daniel Smith" );
if ( name != null )
    alert( "Hello " + name + "!" );
```



14.1.2.13 ShowForm

Instantiiert ein neues Formularobjekt anhand des angegebenen Formularnamens und zeigt es sofort als Dialogfeld an. Der Rückgabewert ist eine Ganzzahl, die für das generierte Dialogfeldergebnis `DialogResult` (`System.Windows.Forms.DialogResult`) steht. Eine Liste der möglichen Werte finden Sie in der Dokumentation zur `DialogResult` Enum (https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.dialogresult?view=netframework-4.8).

Signatur

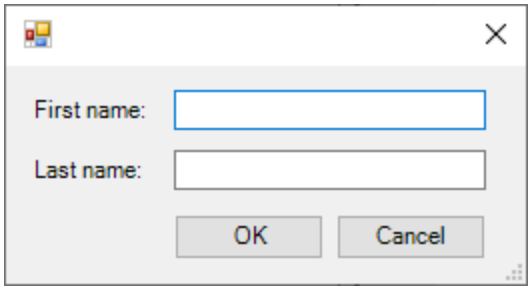
```
ShowForm(strFormName : String) -> result : Integer
```

Beispiel

Der folgende JSCript-Code

```
var dialogResult = ShowForm( "FormName" );
```

zeigt das Formular "FormName" als Dialogfeld an:



Das Ergebnis `DialogResult` kann anschließend ausgewertet werden, z.B:

```
if ( dialogResult == CLR.Static( "System.Windows.Forms.DialogResult" ).OK )
    alert( "ok" );
else
    alert( "cancel" );
```

Anmerkung: Der oben gezeigte Code funktioniert nur, wenn die Eigenschaft **DialogResult** der Schaltflächen "OK" und "Cancel" über das Eigenschaftsfenster korrekt definiert wurde (für die Schaltfläche "OK" muss er z.B. **OK** sein).

14.1.2.14 watchdog

Bei CPU-intensiven Skripts, deren Verarbeitung lange dauert, fragt der Watchdog den Benutzer, ob das Skript abgebrochen werden soll. Mit Hilfe der `watchdog()` Methode können Sie dieses Verhalten aktivieren bzw. deaktivieren. Standardmäßig ist der Watchdog aktiviert.

Durch Aufruf von `watchdog(true)` können Sie den Watchdog auch zurücksetzen. Dies empfiehlt sich manchmal vor der Ausführung langer speicherintensiver Aufgaben, um sicherzustellen, dass diesen der maximale vom Skript her zulässige Verarbeitungsspeicher zur Verfügung steht.

Signatur

```
watchdog(bEnable : boolean) -> void
```

Beispiel

```
watchdog( false );  // disable watchdog - we know the next statement is CPU intensive but
it will terminate for sure
doCPUIntensiveScript();
watchdog( true );   // re-enable watchdog
```

14.1.3 Aktivieren von Skripts und Makros

Nachdem ein Skripting-Projekt fertig gestellt und getestet wurde, können Sie es auf folgende Arten verwenden:

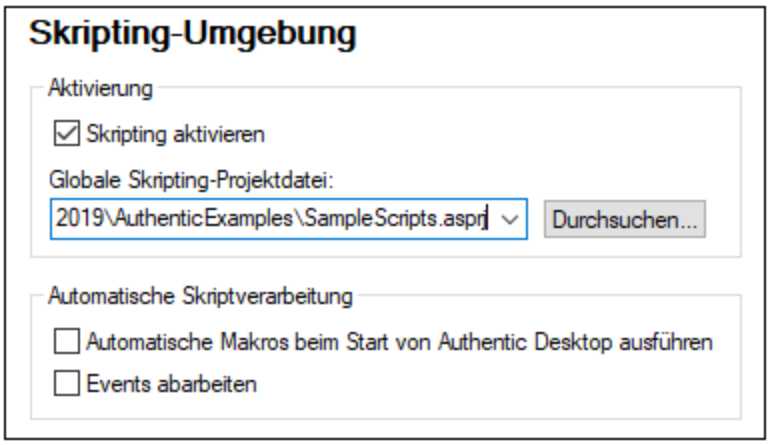
1. Als das globale Skripting-Projekt für Authentic Desktop. Das bedeutet, dass alle Skripts und Makros aus dem Skripting-Projekt Authentic Desktop zur Verfügung stehen.
2. Auf Authentic Desktop-Projektebene. Das bedeutet, es wird eine Referenz auf die .asprj-Datei zusammen mit dem Authentic Desktop-Projekt gespeichert. Wenn das Authentic Desktop-Projekt geöffnet wird, können die damit verknüpften Skripts und Makros aufgerufen werden.

So definieren Sie ein Skripting-Projekt als globales Skripting-Projekt:

1. Klicken Sie im Menü **Extras** auf **Optionen**.
2. Klicken Sie auf das Register **Skripting-Umgebung**.

 © 2016-2022 Altova GmbH

3. Aktivieren Sie das Kontrollkästchen **Skripting aktivieren** und navigieren Sie zu der .asprj-Datei, die als globales Skripting-Projekt verwendet werden soll.



Sie können optional die folgenden zusätzlichen Skriptverarbeitungsoptionen aktivieren:

Automatische Makros beim Start von Authentic Desktop ausführen	Wenn Sie dieses Kontrollkästchen aktivieren, werden alle im Projekt als automatische Makros definierten Makros beim Start von Authentic Desktop automatisch ausgelöst.
Ereignisse abarbeiten	Aktivieren Sie dieses Kontrollkästchen, wenn Ihre Skripts an Applikations-Events gebunden sind. Deaktivieren Sie dieses Kontrollkästchen, damit die Skripts nicht auf Events reagieren.

So aktivieren Sie ein Skripting-Projekt auf Projektebene:

1. Öffnen Sie das Projekt.
2. Klicken Sie im Menü **Projekt** auf **Skript-Einstellungen**.
3. Aktivieren Sie das Kontrollkästchen **Projektskripts aktivieren** und navigieren Sie zur .asprj-Datei.

Das Kontrollkästchen **Beim Laden des... automatische Makros ausführen** hat dieselbe Bedeutung wie oben beschrieben.

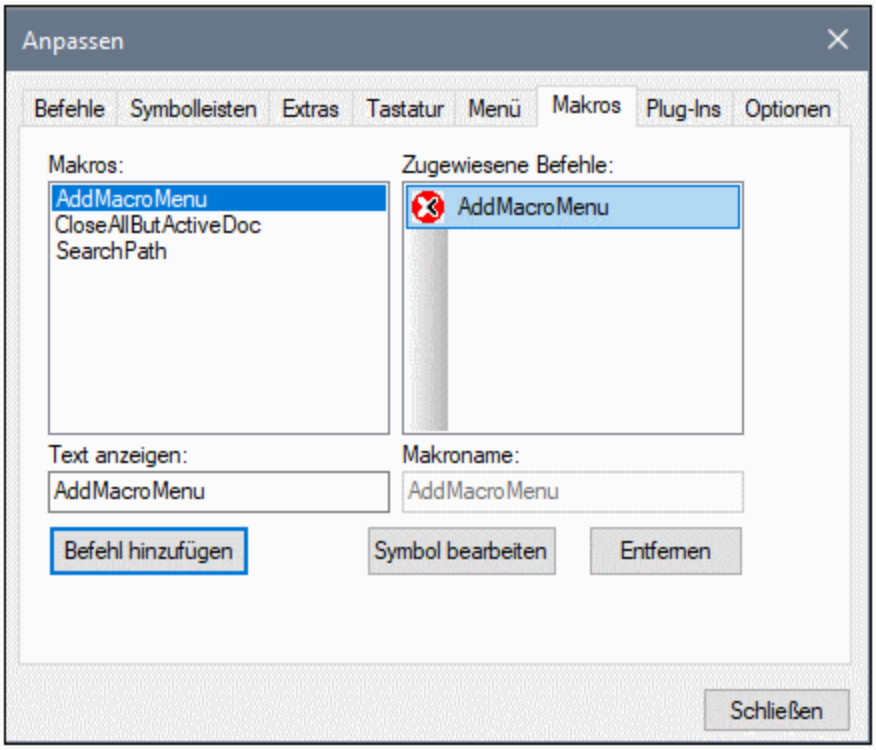
14.1.3.1 Ausführen von Makros

Wenn ein Skripting-Projekt in Authentic Desktop aktiv ist, werden alle in diesem Projekt verfügbaren Makros im Menü **Extras | Makros** angezeigt. Sie können ein Makro daher durch Ausführen des entsprechenden Menübefehls wie z.B. **Extras | Makros | <Makroname>** jederzeit ausführen.

Makros die als automatische Makros konfiguriert wurden, werden bei jedem Start von Authentic Desktop automatisch ausgeführt, vorausgesetzt dieses Verhalten wurde in den Optionen, wie unter Aktivieren von Skripts und Makros 322 beschrieben, aktiviert.

Sie können auch Symbolleisten-Schaltflächen für Makros erstellen. Gehen Sie dazu folgendermaßen vor:

1. Klicken Sie im Menü **Extras** auf **Anpassen**.
2. Klicken Sie auf das Register **Makros**. Alle auf Applikationsebene (im *globalen* Skripting-Projekt) verfügbaren Makros werden aufgelistet.
3. Klicken sie auf **Befehl hinzufügen**.



4. Klicken Sie optional auf **Symbol bearbeiten** und erstellen Sie ein neues Symbol für das neue Makro. Sie können dem Makro über das Register **Tastatur** auch ein Tastaturkürzel zuweisen.
5. Ziehen Sie das Makro aus dem Bereich **Zugewiesene Befehle** an die gewünschte Stelle in der Symbolleiste.

So entfernen Sie ein Makro aus einer Symbolleiste:

1. Klicken Sie im Menü **Extras** auf **Anpassen**.
2. Klicken Sie auf das Register **Makros**.
3. Ziehen Sie das Makro aus der Symbolleiste, in der es angezeigt wird, zurück in den Bereich **Zugewiesene Befehle.**

© 2016-2022 Altova GmbH

14.2 IDE Plugins

Sie können in Authentic Desktop Ihre eigenen IDE-Plug-Ins erstellen und in Authentic Desktop integrieren.

Plug-Ins können zu folgenden Zwecken verwendet werden:

- zum Konfigurieren Ihrer Version von Authentic Desktop, zum Hinzufügen von Befehlen in Form von Menübefehlen, Symbolleisten-Schaltflächen, Schaltflächen usw.
- zum Reagieren auf Events aus Authentic Desktop.
- zum Ausführen Ihres eigenen Codes in Authentic Desktop, wobei Sie Zugriff auf die komplette Authentic Desktop API haben.

Authentic Desktop erwartet von Ihrem Plug-In, dass es die IXMLSpyPlugIn 335 Schnittstelle implementiert. VB.NET, C#- und C++ sind die derzeit unterstützten Sprachen und Ihr Installationspaket enthält im Ordner `Authentic2023\AuthenticExamples\IDEPlugin` Ihrer Authentic Desktop Beispiele zu diesen Sprachen.

Windows 7, 8, 10, 11	`C:/Benutzer/<Benutzername>/Dokumente`

Ein Beispiel zur Verwendung von C++ finden Sie in den ATL-Beispieldateien 330.

14.2.1 Registrierung von IDE Plug-Ins

Authentic Desktop enthält einen speziellen Schlüssel in der Registrierungsdatei, wo alle registrierten IDE-Plug-Ins gespeichert sind:

```
HKEY_CURRENT_USER\Software\Altova\XML Spy\PlugIns
```

Alle Werte dieses Schlüssels werden als Referenzen auf registrierte Plug-Ins behandelt und müssen dem folgenden Format entsprechen:

Wertname:	ProgID des Plug-in
Werttyp:	muss REG_SZ sein
Wertdaten:	CLSID der Komponente

Bei jedem Start der Applikation werden die Werte des **`PlugIns`**-Schlüssels gelesen und die registrierten Plug-ins werden geladen.

Manuelle Registrierung von Plug-Ins

Um ein Plug-In manuell zu registrieren, verwenden Sie in Authentic Desktop im Menü **Extras** das Dialogfeld "Anpassen". Klicken Sie auf die Schaltfläche **Plug-in hinzufügen...** um die DLL anzugeben, die Ihr Plug-In implementiert. Authentic Desktop registriert die DLL als einen COM Server und fügt den entsprechenden Eintrag in seinem **`PlugIns`** Schlüssel hinzu.

Wenn es bei der manuellen Registrierung zu Problemen kommt, überprüfen Sie, ob die CLSID Ihres Plug-in korrekt im **`PlugIns`** Schlüssel registriert wurde. Wenn dies nicht der Fall ist, war der Name Ihrer Plug-In DLL

wahrscheinlich nicht eindeutig genug. Verwenden Sie einen anderen Namen oder führen Sie eine direkte Registrierung durch.

Direkte Registrierung von Plug-Ins

Ein Plug-In kann direkt als IDE Plug-In registriert werden. Dazu muss zuerst die DLL registriert und anschließend müssen z.B. beim Einrichten des Plug-In die entsprechenden Werte zum **`PlugIns`** Schlüssel von Authentic Desktop hinzugefügt werden. Das neue Plug-In wird beim nächsten Start von Authentic Desktop aktiviert.

Erstellen von Plug-Ins

Im Ordner (Eigene) Dateien [11]: `Examples\IDEPlugin` der Applikation finden Sie Quellcode für Beispiel-Plug-Ins. Um ein Plug-in anhand von solchem Quellcode zu erstellen, gehen Sie folgendermaßen vor:

1. Öffnen Sie die Lösung, die Sie als Plug-in in Visual Studio erstellen möchten.
2. Erstellen Sie das Plug-in mit dem Befehl im Menü "Build".
3. Die DLL-Datei des Plug-in wird im Ordner `Bin` oder `Debug` erstellt. Diese DLL ist die Datei, die als Plug-in hinzugefügt werden muss (*siehe oben*).

Anmerkung: VB.NET, C# und C++ sind die derzeit unterstützten Sprachen.

14.2.2 ActiveX Controls

ActiveX Controls werden unterstützt. Jedes IDE PlugIn, das auch ein ActiveX Control ist, wird in einer Dialogfeldsteuerleiste angezeigt. Im Ordner `Examples` Ihres Ordners `XMLSpyPlugInActiveX` des Applikationsordners finden Sie auch ein Beispiel-Plug-In, das auch ein ActiveX Control ist.

14.2.3 Konfigurations-XML

Mit Hilfe des IDE-Plug-In können Sie die Benutzeroberfläche von Authentic Desktop ändern. Dazu wird jede einzelne Änderung mit Hilfe eines XML-Datenstroms beschrieben. Die XML-Konfiguration wird über die Methode GetUIModifications [339] der IXMLSpyPlugIn-Schnittstelle an Authentic Desktop übergeben.

Die XML-Datei für das IDE Plug-In, die die Änderungen an der Benutzeroberfläche enthält, muss die folgende Struktur haben:

```
<ConfigurationData>
   <ImageFile>Path to image file</ImageFile>
   <Modifications>
      <Modification>
         ...
      </Modification>
         ...
    </Modifications>
</ConfigurationData>
```

 © 2016-2022 Altova GmbH

Sie können Symbole oder Symbolleisten-Schaltflächen für die neuen Menüeinträge, die über das Plug-In zur Benutzeroberfläche von Authentic Desktop hinzugefügt werden, definieren. Der Pfad zur Datei, die das Bild enthält, wird über das Element `ImageFile` definiert. Jedes Bild muss 16x16 Pixel haben und darf maximal 256 Farben enthalten. Die Bildreferenzen müssen in einem einzigen **`<ImageFile>`**-Element von links nach rechts angeordnet werden. Der am weitesten rechts gelegene Bild-Indexwert ist Null.

Das Element **`Modification`** kann beliebig viele **`Modification`** Child-Elemente enthalten. Jedes **`Modification`**-Element definiert eine bestimmte Änderung an der Standardbenutzeroberfläche von Authentic Desktop. Welche Änderungen Sie vornehmen können, ist im nächsten Abschnitt weiter unten beschrieben.

Struktur von Modification-Elementen

Ein `Modification`-Element besteht aus den folgenden beiden Child-Elementen:

```
<Modification>
   <Action>Type of action</Action>
   <UIElement Type="Type of UI element" />
</Modification>
```

Gültige Werte für das Element **`Action`** sind:

`Add:` um das darauf folgende Benutzeroberflächenelement zu Authentic Desktop hinzuzufügen
`Hide:` um das darauf folgende Benutzeroberflächenelement in Authentic Desktop auszublenden
`Remove:` um das Benutzeroberflächenelement im Dialogfeld "Anpassen" aus dem Listenfeld "Befehle" zu entfernen.

Mehrere Änderungen können folgendermaßen in einem **`Action`**-Element kombiniert werden: **`"Add Hide"`**

Das Element **`UIElement`** definiert alle neuen oder bestehenden Elemente der Benutzeroberfläche und kann eines der folgenden Typen sein: Symbolleisten, Schaltflächen, Menüs oder Menübefehle. Das Attribut **`type`** definiert, zu welchem dieser Typen das Benutzeroberflächen-Element gehört. Die Struktur von **`UIElement`** wird im folgenden Abschnitt beschrieben.

Gemeinsame Child-Elemente von UIElement

Die Elemente `ID` und `Name` sind für alle verschiedenen Typen von UI-Elementen definiert. Bei einigen Typen wird jedoch eines dieser Elemente ignoriert. So wird z.B. **`Name`** für ein Trennzeichen ignoriert.

```
<ID></ID>
<Name></Name>
```

Wenn **`UIElement`** ein bestehendes Element der Benutzeroberfläche beschreibt, wird der Wert des Elements **`ID`** von Authentic Desktop vorgegeben. Normalerweise sind diese ID-Werte dem User nicht bekannt. Wenn das XML-Fragment einen neuen Teil der Benutzeroberfläche beschreibt, kann die ID beliebig gewählt werden und der Wert sollte kleiner als **`1000`** sein. Das Element **`Name`** definiert den Textwert. Bestehende UI-Elemente können nur nach ihrem Namen identifiziert werden, z.B. Menüs und Menübefehle, die Untermenüs haben. Bei neuen Benutzeroberflächenelementen definiert das Element **`Name`** die Überschrfit, z.B. den Titel einer Symbolleiste oder den Text eines Menübefehls.

Symbolleisten und Menüs

Um eine Symbolleiste zu definieren, muss die ID und/oder der Name der Symbolleiste definiert werden. Eine bestehende Symbolleiste kann nur durch den Namen oder die ID, unter der sie bekannt ist, definiert werden. Um eine **neue** Symbolleiste zu erstellen, müssen beide Werte definiert werden. Das Attribut **type** muss **ToolBar** sein.

```
<UIElement Type="ToolBar">
   <ID>1</ID>
   <Name>TestPlugIn</Name>
</UIElement>
```

Um ein Authentic Desktop Menü zu definieren, benötigen Sie zwei Parameter:

- die ID der Menüleiste, die das Menü enthält. Wenn im Hauptfenster keine XML-Dokumente geöffnet sind, ist die ID der Menüleiste `128`. Wenn ein oder mehrere Dokumente geöffnet sind, ist die ID der Menüleiste `129`.
- den Menünamen. Menüs haben keinen dazugehörigen ID-Wert. Im folgenden Beispiel wird das Menü "Edit" (Bearbeiten) der Menüleiste definiert, die aktiv ist, wenn mindestens ein XML-Dokument geöffnet ist:

  ```
  <UIElement Type="Menu">
     <ID>129</ID>
     <Name>Edit</Name>
  </UIElement>
  ```

Wenn Sie ein neues Menü erstellen möchten, wird ein zusätzliches Element verwendet. Das Element **Place** definiert die Position des neuen Menüs in der Menüleiste:

```
<UIElement Type="Menu">
   <ID>129</ID>
   <Name>PlugIn Menu</Name>
   <Place>12</Place>
</UIElement>
```

Ein Wert von `-1` für das Element `Place` definiert die neue Schaltfläche oder den neuen Menübefehl am Ende des Menüs oder der Symbolleiste.

Befehle

Wenn Sie einen neuen Befehl in Form einer Symbolleisten-Schaltfläche oder eines Menübefehls hinzufügen, kann das **UIElement**-Fragment jedes beliebige dieser Subelemente enthalten:

```
<MacroName></MacroName>
<Info></Info>
<ImageID></ImageID>
```

Wenn **MacroName** definiert ist, sucht Authentic Desktop in der Skripting-Umgebung nach einem Makro dieses Namens und führt es jedes Mal, wenn dieser Befehl aufgerufen wird, aus. Das Element **Info** enthält einen Beschreibungsstring, der in der Statusleiste angezeigt wird, wenn sich der Mauszeiger über dem entsprechenden Befehl (Schaltfläche oder Menübefehl) befindet. **ImageID** definiert den Index des Symbols in der Bilddatei. Bitte beachten Sie, dass alle Symbole in einer Bilddatei gespeichert sind.

 © 2016-2022 Altova GmbH

Um einen Symbolleistenbefehl zu definieren, erstellen Sie ein `UIElement` mit der folgenden Struktur:

```
<UIElement Type="ToolBarItem">
   <!--don't reuse local IDs even the commands do the same-->
   <ID>5</ID>
   <Name>Open file from repository...</Name>
   <!--Set Place To -1 If this is the first button To be inserted-->
   <Place>-1</Place>
   <ImageID>0</ImageID>
   <ToolBarID>1</ToolBarID>
   <!--instead of the toolbar ID the toolbar name could be used-->
   <ToolBarName>TestPlugIn</ToolBarName>
</UIElement>
```

Zusätzliche Elemente zum Deklarieren einer Symbolleisten-Schaltfläche sind **`Place`**, **`ToolBarID`** und **`ToolBarName`**. `ToolBarID` und `ToolBarName` dienen dazu, die Symbolleiste zu identifizieren, die die neue oder bestehende Schaltfläche enthält. Die Groß- und Kleinschreibung des Textwerts von **`ToolBarName`** spielt eine Rolle. Das Attribut **`type`** von UIElement muss **`ToolBarItem`** sein.

Um einen Menübefehl zu definieren, stehen zusätzlich zu den Standardelementen, die zum Deklarieren eines Befehls verwendet werden, die Elemente **`MenuID`**, **`Place`** und **`Parent`** zur Verfügung. Der Inhalt des **`MenuID`**-Elements kann **`128`** oder **`129`** sein. *Nähere Informationen zu diesen Werten finden Sie unter "Symbolleisten und Menüs".*

Das Element **`Parent`** dient dazu, das **Menü** anzugeben, in das der neue Menübefehl eingefügt werden soll. Da die Befehle von Untermenüs keine eindeutige Windows ID haben, benötigen wir eine andere Methode, um das übergeordnete Element des Menübefehls zu definieren. Wir tun dies, indem wir als Inhalt des **`Parent`**-Elements den Pfad zu Menübefehl definieren. Die Schritte im Pfad werden durch einen Doppelpunkt gekennzeichnet. Das Muster wäre **`ParentMenü:Submenü`**. Wenn das Menü kein übergeordnetes Menü hat (da es kein Untermenü ist), stellen Sie dem Namen am Anfang einen Doppelpunkt voran (siehe Beispiel unten). Das Attribut **`type`** muss den Wert **`MenuItem`** haben.

Im Beispiel unten wird ein Menübefehl definiert, dessen übergeordnetes Menü kein Untermenü ist:

```
<UIElement Type="MenuItem">
   <!--the following element is a Local command ID-->
   <ID>3</ID>
   <Name>Open file from repository...</Name>
   <Place>-1</Place>
   <MenuID>129</MenuID>
   <Parent>:PlugIn Menu</Parent>
   <ImageID>0</ImageID>
</UIElement>
```

Sie können Symbolleistentrennzeichen und Menüs hinzufügen, wenn der Wert des **`ID`** Elements auf **`0`** gesetzt wird.

14.2.4 ATL-Beispieldateien

In diesem Abschnitt wird beschrieben, wie Sie mit Hilfe von ATL eine einfache Authentic Desktop IDE Plug-In DLL erstellen. Sie müssen mit der Arbeit mit MSVisualStudio, ATL und den Assistenten zur Generierung von ATL-Objekten vertraut sein. Zum Aufrufen der API importiert die Implementierung die Typbibliothek von Authentic Desktop. Der Code liest verschiedene Eigenschaften aus und ruft mit Hilfe der über die **`#import`**-Anweisung zur Verfügung gestellten Smart Pointer Methoden auf. Außerdem werden im Beispielcode der MFC Class **`CString`** und die ATL-Konvertierungsmakros wie z.B. W2T verwendet.

Die grundlegenden Schritte zum Erstellen einer ATL DLL lauten wie folgt:

1. Öffnen Sie VisualStudio und wählen Sie **File | New**.
2. Wählen Sie das Register *Projects*.
3. Wählen Sie *ATL COM AppWizard* und geben Sie einen Projektnamen ein.
4. Aktivieren Sie *Support for MFC*, wenn Sie MFC-Klassen verwenden oder ein Projekt für den Beispielcode erstellen möchten.

Nach Erstellung der Projektdateien können Sie ein ATL-Objekt hinzufügen, um die **`IXMLSpyPlugIn`**-Schnittstelle zu implementieren:

1. Wählen Sie **Insert | New ATL Object**.
2. Wählen Sie im Assistenten die Option *Simple Object* und klicken Sie auf **Next**.
3. Geben Sie einen Namen für das Objekt ein.
4. Wählen Sie auf dem Register *Attributes* als Schnittstellentyp *Custom* aus und deaktivieren Sie die *Aggregation*.

Damit wird der grundlegende Code für die Implementierung der IDE Plug-In Schnittstelle erzeugt. Wie man den Code ändert und einige grundlegende Funktionalitäten definiert, wird auf den nächsten Seiten beschrieben.

14.2.4.1 Schnittstellenbeschreibung (IDL)

Die IDL des neu erstellten ATL-Objekts enthält eine Deklaration für eine COM-Schnittstelle.

- Diese Schnittstellenbeschreibung muss, wie unten gezeigt, durch die Deklaration von IXMLSpyPlugIn ersetzt werden.
- Die IDL muss außerdem die Definition der SPYUpdateAction Enumeration enthalten.
- Ersetzen Sie in der coclass-Deklaration den (vom Assistenten) generierten Standard-Schnittstellennamen durch "IXMLSpyPlugIn".

Die IDL sollte anschließend in etwa so aussehen wie im Beispielcode unten: Nach Erstellung des ATL-Objekts müssen Sie nun die IDE Plug-In-Schnittstelle von Authentic Desktop implementieren.

```
import "oaidl.idl";
import "ocidl.idl";

// ----- please insert the following block into your IDL file -----
    typedef enum {
       spyEnable = 1,
       spyDisable = 2,
```

 © 2016-2022 Altova GmbH

```
      spyCheck = 4,
      spyUncheck = 8
   } SPYUpdateAction;

// ----- end insert block ----


// ----- E.g. Interface entry automatically generated by the ATL wizard -----
//     [
//           object,
//           uuid(AB7CD86A-8145-429A-A1F3-270692EO8AFC),

//           helpstring("IXMLSpyPlugIn Interface")
//           pointer_default(unique)
//     ]
//     interface IXMLSpyPlugIn : IUnknown
//     {
//     };
// ----- end automatically generated Interface Entry


// ----- replace the Interface Entry (shown above) generated for you by the ATL wizard,
with the following block -----

       [
          odl,
          uuid(88F2A622-4B7E-42CD-8D04-3C0E5389DD85),
          helpstring("IXMLSpyPlugIn Interface")
       ]
       interface IXMLSpyPlugIn : IUnknown
          {
             HRESULT _stdcall OnCommand([in] long nID, [in] IDispatch* pXMLSpy);
             HRESULT _stdcall OnUpdateCommand([in] long nID, [in] IDispatch* pXMLSpy, [out,
retval] SPYUpdateAction* pAction);
             HRESULT _stdcall OnEvent([in] long nEventID, [in] SAFEARRAY(VARIANT)*
arrayParameters, [in] IDispatch* pXMLSpy, [out, retval] VARIANT* pReturnValue);
             HRESULT _stdcall GetUIModifications([out, retval] BSTR* pModificationsXML);
             HRESULT _stdcall GetDescription([out, retval] BSTR* pDescription);
          };

// ----- end replace block -----


// ----- The code below is automatically generated by the ATL wizard and will look slightly
different in your case -----

       [
          uuid(24FE0D1B-3FC0-494E-B36E-1D4CE412B014),
          version(1.0),
          helpstring("XMLSpyIDEPlugInDLL 1.0 Type Library")
       ]
       library XMLSPYIDEPLUGINDLLLib
          {
       importlib("stdole32.tlb");
       importlib("stdole2.tlb");
```

```
        [
           uuid(3800E791-7F6B-4ACD-9E32-2AC184444501),
           helpstring("XMLSpyIDEPlugIn Class")
        ]
        coclass XMLSpyIDEPlugIn
        {
           [default] interface IXMLSpyPlugIn;   // ----- define IXMLSpyPlugIn as the default
interface -----
        };
};
```

14.2.4.2 Klassendefinition

In der Klassendefinition des ATL-Objekts müssen die folgenden Änderungen vorgenommen werden.

- Die Klasse muss von `IXMLSpyPlugIn` abgeleitet sein.
- Die "Interface Map" benötigt einen Eintrag für `IXMLSpyPlugIn`.
- Die Methoden der IDE Plug-In-Schnittstelle müssen deklariert werden.

Diese Änderungen können, wie unten gezeigt, gemacht werden:

```
#ifndef __XMLSPYIDEPLUGIN_H_
#define __XMLSPYIDEPLUGIN_H_

#include "resource.h"       // main symbols

/////////////////////////////////////////////////////////////////////////////
// CXMLSpyIDEPlugIn
class ATL_NO_VTABLE CXMLSpyIDEPlugIn :
   public CComObjectRootEx<CComSingleThreadModel>,
   public CComCoClass<CXMLSpyIDEPlugIn, &CLSID_XMLSpyIDEPlugIn>,
   public IXMLSpyPlugIn
{
public:
   CXMLSpyIDEPlugIn()
   {
   }

DECLARE_REGISTRY_RESOURCEID(IDR_XMLSPYIDEPLUGIN)
DECLARE_NOT_AGGREGATABLE(CXMLSpyIDEPlugIn)

DECLARE_PROTECT_FINAL_CONSTRUCT()

BEGIN_COM_MAP(CXMLSpyIDEPlugIn)
   COM_INTERFACE_ENTRY(IXMLSpyPlugIn)
END_COM_MAP()

// IXMLSpyIDEPlugIn
public:
   virtual HRESULT _stdcall OnCommand(long nID, IDispatch* pXMLSpy);
```


© 2016-2022 Altova GmbH

```
   virtual HRESULT _stdcall OnUpdateCommand(long nID, IDispatch* pXMLSpy, SPYUpdateAction*
pAction);
   virtual HRESULT _stdcall OnEvent(long nEventID, SAFEARRAY **arrayParameters, IDispatch*
pXMLSpy, VARIANT* pReturnValue);
   virtual HRESULT _stdcall GetUIModifications(BSTR* pModificationsXML);
   virtual HRESULT _stdcall GetDescription(BSTR* pDescription);
};

#endif //__XMLSPYIDEPLUGIN_H_
```

14.2.4.3 Implementierung

Im Code unten sehen Sie eine einfache Implementierung eines Authentic Desktop IDE Plug-In. Damit wird ein Menübefehl und eine Trennlinie (verfügbar in Authentic Desktop) zum Menü "Extras" hinzugefügt. Der neue Befehl innerhalb der OnUpdateCommand() Methode ist nur aktiv, wenn das aktive Dokument über die Grid-Ansicht angezeigt wird. Der Befehl sucht nach dem XML-Element, das sich gerade im Fokus befindet, und öffnet jede URL, die mit "http://" beginnt am Textwert des Elements.

```
/////////////////////////////////////////////////////////////////////////////
// CXMLSpyIDEPlugIn

#import "XMLSpy.tlb"
using namespace XMLSpyLib;

HRESULT CXMLSpyIDEPlugIn::OnCommand(long nID, IDispatch* pXMLSpy)
{
   USES_CONVERSION;

   if(nID == 1)      {
      IApplicationPtr       ipSpyApp;

      if(pXMLSpy)    {
         if(SUCCEEDED(pXMLSpy->QueryInterface(__uuidof(IApplication),(void **)&ipSpyApp)))
{
            IDocumentPtr  ipDocPtr = ipSpyApp->ActiveDocument;

            // we assume that grid view is active
            if(ipDocPtr) {
               IGridViewPtr   ipGridPtr = ipDocPtr->GridView;

               if(ipGridPtr)       {
                  IXMLDataPtr      ipXMLData = ipGridPtr->CurrentFocus;

                  CString   strValue = W2T(ipXMLData->TextValue);

                  if(!strValue.IsEmpty() && (strValue.Left(7) == _T("http://")))
                     ::ShellExecute(NULL,_T("open"),W2T(ipXMLData-
>TextValue),NULL,NULL,SW_SHOWNORMAL);
               }
            }
         }
      }
   }
```

```
   return S_OK;
}


HRESULT CXMLSpyIDEPlugIn::OnUpdateCommand(long nID, IDispatch* pXMLSpy, SPYUpdateAction*
pAction)
{
   *pAction = spyDisable;

   if(nID == 1)      {
      IApplicationPtr       ipSpyApp;

      if(pXMLSpy)    {
         if(SUCCEEDED(pXMLSpy->QueryInterface(__uuidof(IApplication),(void **)&ipSpyApp)))
       {
            IDocumentPtr    ipDocPtr = ipSpyApp->ActiveDocument;

            // only enable if grid view is active
            if((ipDocPtr != NULL) && (ipDocPtr->CurrentViewMode == spyViewGrid))
               *pAction = spyEnable;
         }
      }
   }

   return S_OK;
}


HRESULT CXMLSpyIDEPlugIn::OnEvent(long nEventID, SAFEARRAY **arrayParameters, IDispatch*
pXMLSpy, VARIANT* pReturnValue)
{
   return S_OK;
}

HRESULT CXMLSpyIDEPlugIn::GetUIModifications(BSTR* pModificationsXML)
{
   CComBSTR   bstrMods = _T(" \
            <ConfigurationData>    \
               <Modifications>    ");
   // add "Open URL..." to Tools menu
   bstrMods.Append (_T(" \
                  <Modification> \
                     <Action>Add</Action> \
                     <UIElement type=\"MenuItem\"> \
                        <ID>1</ID> \
                        <Name>Open URL...</Name> \
                        <Place>0</Place> \
                        <MenuID>129</MenuID> \
                        <Parent>:Tools</Parent> \
                     </UIElement> \
                  </Modification> "));
   // add Seperator to Tools menu
   bstrMods.Append (_T(" \
                  <Modification> \
                     <Action>Add</Action> \
```

```
                    <UIElement type=\"MenuItem\"> \
                       <ID>0</ID> \
                       <Place>1</Place> \
                       <MenuID>129</MenuID> \
                       <Parent>:Tools</Parent> \
                    </UIElement> \
                 </Modification> "));
   // finish modification description
   bstrMods.Append (_T(" \
               </Modifications> \
            </ConfigurationData>"));

   return bstrMods.CopyTo(pModificationsXML);
}


HRESULT CXMLSpyIDEPlugIn::GetDescription(BSTR* pDescription)
{
   CComBSTR   bstrDescr = _T("ATL C++ XMLSpy IDE PlugIn;This PlugIn demonstrates the
implementation of a simple ATL DLL as a IDE PlugIn for XMLSpy.");
   return bstrDescr.CopyTo(pDescription);
}
```

14.2.5 IXMLSpyPlugIn

Methoden

OnCommand 335
OnUpdateCommand 336
OnEvent 337
GetUIModifications 339
GetDescription 340

Beschreibung

Wenn eine DLL als IDE Plug-In zu Authentic Desktop hinzugefügt wird, muss sie als COM-Komponente registriert werden, die auf eine `IXMLSpyPlugIn` Schnittstelle mit der reservierten `UUID(88F2A622-4B7E-42CD-8D04-3C0E5389DD85)` antwortet, damit sie als Plug-In erkannt wird.

14.2.5.1 OnCommand

Deklaration

```
OnCommand(nID als long, pXMLSpy als IDispatch)
```

Beschreibung

Die `OnCommand()` Methode der Schnittstellenimplementierung wird jedes Mal aufgerufen, wenn ein zum IDE Plug-In hinzugefügter Befehl (Menübefehl oder Symbolleisten-Schaltfläche) verarbeitet wird. `nID` speichert die durch das `ID`-Element des entsprechenden `UIElement` definierte Befehls-ID. `pXMLSpy` enthält eine Referenz auf das Dispatch Interface des `Application`-Objekts von Authentic Desktop.

Beispiel

```
Public Sub IXMLSpyPlugIn_OnCommand(ByVal nID As Long, ByVal pXMLSpy As Object)
```

```
   If (Not (pXMLSpy Is Nothing)) Then
      Dim objDlg
      Dim objDoc As XMLSpyLib.Document
      Dim objSpy As XMLSpyLib.Application
      Set objSpy = pXMLSpy

      If nID = 3 Or nID = 5 Then
         Set objDlg = CreateObject("MSComDlg.CommonDialog")
         objDlg.Filter = "XML Files (*.xml)|*.xml|All Files (*.*)|*.*||"
         objDlg.FilterIndex = 1
         objDlg.ShowOpen

         If Len(objDlg.FileName) > 0 Then
            Set objDoc = objSpy.Documents.OpenFile(objDlg.FileName, False)
            Set objDoc = Nothing
         End If
      End If

      If nID = 4 Or nID = 6 Then
         Set objDlg = CreateObject("MSComDlg.CommonDialog")
         objDlg.Filter = "All Files (*.*)|*.*||"
         objDlg.Flags = cdlOFNPathMustExist
         objDlg.ShowSave

         If Len(objDlg.FileName) > 0 Then
            Set objDoc = objSpy.ActiveDocument

            If Not (objDoc Is Nothing) Then
               objDoc.SetPathName objDlg.FileName
               objDoc.Save
               Set objDoc = Nothing
            End If
         End If
      End If

      Set objSpy = Nothing
   End If
End Sub
```

14.2.5.2 OnUpdateCommand

Deklaration

`OnUpdateCommand(nID als long, pXMLSpy als IDispatch) als SPYUpdateAction`

Beschreibung

Die `OnUpdateCommand()` Methode wird jedes Mal, wenn der Sichtbarkeitsstatus einer Schaltfläche oder eines Menübefehls definiert werden muss, aufgerufen. `nID` speichert die durch das `ID`-Element des entsprechenden `UIElement` definierte Befehls-ID. `pXMLSpy` enthält eine Referenz auf das Dispatch Interface des `Application`-Objekts.

Mögliche Rückgabewerte zum Definieren des Aktualisierungsstatus sind:

```
spyEnable= 1
spyDisable   = 2
```

 © 2016-2022 Altova GmbH

```
   spyCheck     = 4
   spyUncheck   = 8
```

Beispiel

```
Public Function IXMLSpyPlugIn_OnUpdateCommand(ByVal nID As Long, ByVal pXMLSpy As Object)
As SPYUpdateAction
   IXMLSpyPlugIn_OnUpdateCommand = spyDisable

   If (Not (pXMLSpy Is Nothing)) Then
      Dim objSpy As XMLSpyLib.Application
      Set objSpy = pXMLSpy

      If nID = 3 Or nID = 5 Then
         IXMLSpyPlugIn_OnUpdateCommand = spyEnable
      End If
      If nID = 4 Or nID = 6 Then
         If objSpy.Documents.Count > 0 Then
            IXMLSpyPlugIn_OnUpdateCommand = spyEnable
         Else
            IXMLSpyPlugIn_OnUpdateCommand = spyDisable
         End If
      End If
   End If
End Function
```

14.2.5.3 OnEvent

Deklaration

```
OnEvent(nEventID als long, arrayParameters als SAFEARRAY(VARIANT), pXMLSpy als IDispatch)
als VARIANT
```

Beschreibung

`OnEvent()` wird jedes Mal, wenn ein Event von Authentic Desktop ausgelöst wird, aufgerufen.

Mögliche Werte für `nEventID` sind:

```
On_BeforeStartEditing          = 1
On_EditingFinished             = 2
On_FocusChanged                = 3
On_Beforedrag                  = 4
On_BeforeDrop                  = 5
On_OpenProject                 = 6
On_OpenDocument                = 7
On_CloseDocument               = 8
On_SaveDocument                = 9
On_DocEditDragOver             = 10
On_DocEditDrop                 = 11
```

```
On_DocEditKeyDown                    = 12
On_DocEditKeyUp                      = 13
On_DocEditKeyPressed                 = 14
On_DocEditMouseMove                  = 15
On_DocEditButtonUp                   = 16
On_DocEditButtonDown                 = 17
On_DocEditContextMenu                = 18
On_DocEditPaste                      = 19
On_DocEditCut                        = 20
On_DocEditCopy                       = 21
On_DocEditClear                      = 22
On_DocEditSelectionChanged           = 23
On_DocEditDragOver                   = 10
On_BeforeOpenProject                 = 25
On_BeforeOpenDocument                = 26
On_BeforeSaveDocument                = 27
On_BeforeCloseDocument               = 28
On_ViewActivation                    = 29
On_DocEditKeyboardEvent              = 30
On_DocEditMouseEvent                 = 31
On_BeforeValidate                    = 32
On_BeforeShowSuggestions             = 33
On_ProjectOpened                     = 34
On_Char                              = 35
On_Initialize                        = 36
On_Running                           = 37
On_Shutdown                          = 38
On_AuthenticBeforeSave               = 39
On_AuthenticContextMenuActivated     = 40
On_AuthenticLoad                     = 41
On_AuthenticToolbarButtonClicked     = 42
On_AuthenticToolbarButtonExecuted    = 43
On_AuthenticUserAddedXMLNode         = 44
```

Die Namen der Events sind dieselben wie in der Skripting-Umgebung von Authentic Desktop. Bei IDE Plug-Ins spielt es keine Rolle, welche Namen verwendet werden. Die Events werden über den ID Wert definiert.

`arrayParameters` ist ein Array, das mit den Parametern des aktuell ausgelösten Event gefüllt ist. Die Reihenfolge, Art und Bedeutung der einzelnen Parameter stehen über die Skripting-Umgebung von Authentic Desktop zur Verfügung. Das Event-Modul eines Skripting-Projekts enthält vordefinierte Funktionen für alle

 © 2016-2022 Altova GmbH

Events vor Version 4.4. Die an die vordefinierten Funktionen übergebenen Parameter sind mit den Array-Elementen des `arrayParameters`-Parameters identisch.

Events, die über die Authentic-Ansicht von Authentic Desktop ausgelöst werden, übergeben Parameter nicht direkt. Statt dessen wird ein "event" Objekt verwendet. Das Event-Objekt kann über das `Document`-Objekt des aktiven Dokuments aufgerufen werden.

`pXMLSpy` enthält eine Referenz auf das Dispatch Interface des `Application`-Objekts von Authentic Desktop.

Wenn der Rückgabewert von `OnEvent()` definiert ist, so ruft weder das IDE Plug-In noch ein Event Handler innerhalb der Skripting-Umgebung dieses Event später auf. Bitte beachten Sie, dass alle IDE Plug-Ins das Event abrufen/verarbeiten, bevor dies die Skripting-Umgebung tut.

14.2.5.4 GetUIModifications

Deklaration

```
GetUIModifications() als String
```

Beschreibung

Die `GetUIModifications()` Methode wird während der Initialisierung des Plug-In aufgerufen, um die XML-Konfigurationsdaten abzurufen, die die Änderungen an der Benutzeroberfläche von Authentic Desktop definieren. Die Methode wird beim ersten Laden des Plug-In und bei jedem Start von Authentic Desktop aufgerufen. *Eine ausführliche Anleitung zum Ändern der Benutzeroberfläche finden Sie auch unter* *Konfigurations-XML* 326.

Beispiel

```
Public Function IXMLSpyPlugIn_GetUIModifications() As String
   ' GetUIModifications() gets the XML file with the specified modifications of
   ' the UI from the config.xml file in the plug-in folder
   Dim strPath As String
   strPath = App.Path

   If Len(strPath) > 0 Then
      Dim fso As New FileSystemObject
      Dim file As file
      Set file = fso.GetFile(strPath & "\config.xml")

      If (Not (file Is Nothing)) Then
         Dim stream As TextStream
         Set stream = file.OpenAsTextStream(ForReading)

         ' this replaces the token '**path**' from the XML file with
         ' the actual installation path of the plug-in to get the image file
         Dim strMods As String
         strMods = stream.ReadAll
         strMods = Replace(strMods, "**path**", strPath)

         IXMLSpyPlugIn_GetUIModifications = strMods
      Else
         IXMLSpyPlugIn_GetUIModifications = ""
      End If
   End If
```

```
End Function
```

14.2.5.5 GetDescription

Deklaration

```
GetDescription() as String
```

Beschreibung

`GetDescription()` dient zur Definition des Beschreibungsstring für die Plug-In-Einträge, die im Dialogfeld "Anpassen" angezeigt werden.

Beispiel

```
Public Function IXMLSpyPlugIn_GetDescription() As String
    IXMLSpyPlugIn_GetDescription = "Sample Plug-in for XMLSpy;This Plug-in demonstrates the
implementation of a simple VisualBasic DLL as a Plug-in for XMLSpy."
End Function
```


© 2016-2022 Altova GmbH

14.3 Applikations-API

Mit Hilfe der COM-basierten API von Authentic Desktop können andere Applikationen die Funktionalitäten von Authentic Desktop verwenden. Dadurch können die verschiedensten Aufgaben vom Validieren einer XML-Datei bis zum Ändern von komplexem XML-Inhalt (mit der XMLData 593 Schnittstelle) automatisiert werden.

Authentic Desktop und die Applikations-API entsprechen den von Microsoft vorgegebenen allgemeinen Spezifikationen für Automation Server. Die Methoden und Eigenschaften der Applikations-API können von gängigen Entwicklungsumgebungen wie z.B. solchen, die C#, C++, VisualBasic und Delphi verwenden, und von von Skript-Sprachen wie JScript und VBScript aufgerufen werden.

Ausführungsumgebungen für die Applikations-API

Die Applikations-API kann über die folgenden Ausführungsumgebungen aufgerufen werden:

- externe Programme (Beschreibung siehe unten 341 und in diesem Abschnitt im Kapitel Übersicht 342)
- vom integrierten Skript-Editor von Authentic Desktop aus. Eine Beschreibung zur Skripting-Umgebung finden Sie im Abschnitt Skript-Editor 297.
- Sie haben in Authentic Desktop die Möglichkeit, Ihre eigenen Plug-ins zu erstellen und über eine spezielle Schnittstelle für Plug-ins in die Applikation zu integrieren. Eine Beschreibung zur Erstellung von Plug-ins finden Sie im Abschnitt IDE Plug-ins 325.
- über ein ActiveX Control, das verfügbar ist, wenn das Integration Package 619 installiert wurde. Nähere Informationen dazu finden Sie im Abschnitt ActiveX Integration 619.

Externe Programme

Im Abschnitt Übersicht 342 dieses Abschnitts wird beschrieben, wie die Funktionalitäten von Authentic Desktop von externen Programmen aus aufgerufen und automatisiert werden können.

Um die Applikations-API außerhalb von Authentic Desktop verwenden zu können, muss zuerst eine Instanz von Authentic Desktop gestartet werden. Wie dies erfolgt, hängt von der verwendeten Programmiersprache ab. Nähere Informationen zu den einzelnen Sprachen finden Sie unter Programmiersprachen 343.

Im Wesentlichen wird Authentic Desktop über seine COM Registrierung gestartet. Anschließend wird das mit der Authentic Desktop-Instanz verknüpfte `Application` Objekt zurückgegeben. Je nach COM-Einstellungen kann ein mit einer bereits laufenden Authentic Desktop-Instanz verknüpftes Objekt zurückgegeben werden. Es kann jede Programmiersprache verwendet werden, die die Erstellung und den Aufruf von COM-Objekten unterstützt. Die am häufigsten dazu verwendeten Sprachen sind unten aufgelistet.

- JScript und VBScript 349 Skript-Dateien haben eine einfache Syntax und dienen dazu COM-Objekte aufzurufen. Sie können direkt über eine DOS-Befehlszeile oder durch Doppelklick im Windows Explorer ausgeführt werden. Am besten eignen sich diese Sprachen für einfache Automationsaufgaben.
- C# 352 ist eine umfangreiche Programmiersprache mit einer breiten Palette an Funktionalitäten. Der Zugriff auf COM-Objekte kann in C# automatisch über Wrapper erfolgen.
- C++ ermöglicht die direkte Steuerung des COM-Zugriffs. Allerdings muss dafür im Vergleich zu anderen Sprachen relativ viel Code geschrieben werden.
- Java 364: Im Lieferumfang von Altova-Produkten sind native Java-Klassen inkludiert, die als Wrapper für die Applikations-API verwendet werden und dadurch eine Java-Umgebung ermöglichen.
- Weitere nützliche Alternativen sind Visual Basic für Applikationen, Perl und Python.

Anmerkungen zur Programmierung

Berücksichtigen Sie bitte in Ihrem Client-Code die folgenden Einschränkungen:

- Denken Sie daran, dass bei einem Crash Ihres Client-Code eventuell noch Instanzen von Authentic Desktopim System laufen.
- Behalten Sie Referenzen auf Objekte - vor allem jene von der XMLData-Schnittstelle - nicht länger als nötig im Arbeitsspeicher. Wenn der Benutzer zwischen zwei Aufrufen Ihres Client Eingaben macht, ist nicht gewährleistet, dass diese Referenzen noch gültig sind.
- Vergessen Sie nicht, Dialogfelder zu deaktivieren, wenn die Benutzeroberfläche nicht sichtbar ist.
- Nähere Informationen, wie Sie lästige Fehlermeldungen vermeiden, finden Sie unter Fehlerbehandlung in JScript 348 (und in C# 361 and Java 371).
- Geben Sie Referenzen explizit frei, wenn Sie C oder C++ verwenden.

Diese Dokumentation

Die Dokumentation zur Applikations-API ist in zwei Hauptteile gegliedert.

- Der erste Teile besteht aus einer Übersicht 342, in der das Objektmodell für die API beschrieben wird. Weiters wird beschrieben, wie Sie die API über verschiedene Programmiersprachen 343 aufrufen.
- Der zweite Teil ist ein Referenzteil (Schnittstellen 372 und Enumerationen 604), der Beschreibungen der Schnittstellenobjekte der Applikations-APIenthält

14.3.1 Übersicht

Diese Übersicht über die Applikations-API ist in die folgenden Abschnitte gegliedert:

- Das Objektmodell 342: Hier werden die Beziehungen zwischen den Objekten der Applikations-API beschrieben.
- Programmiersprachen 343: Hier wird beschrieben, wie Sie die Funktionalitäten der Applikations-API über die gebräuchlichsten Programmiersprachen (JScript, VBScript, C# und Java) aufrufen können. Die Codefragmente aus den mitinstallierten Beispieldateien dienen zum Veranschaulichen einiger grundlegender Mechanismen.

14.3.1.1 Objektmodell

Der Ausgangspunkt für alle Applikationen, die die Applikations-API verwenden, ist das `Application` 372 Objekt. Dieses Objekt enthält allgemeine Methoden wie Import/Export-Funktionen und Referenzen auf offene Dokumente und das offene Projekt, falls vorhanden.

Das `Application`-Objekt wird in verschiedenen Programmiersprachen unterschiedlich erstellt. In Skriptsprachen wie JScript oder VBScript wird dabei eine Funktion aufgerufen, die das COM-Objekt der Applikation initialisiert. Beispiele dazu finden Sie im Abschnitt Programmiersprachen 343.

© 2016-2022 Altova GmbH

XMLSpy.Application oder AuthenticDesktop.Application

Authentic Desktop installiert eine Typbibliothek, die die XMLSpyLib enthält. Wenn diese Typbibliothek zur Entwicklungsumgebung (z.B. der VB-Entwicklungsumgebung) hinzugefügt wurde, dann kann ein Objekt des Typs `Application` folgendermaßen erstellt werden.

```
Set objSpy = New XMLSpyLib.Application
```

Wenn nur Authentic Desktop (und nicht XMLSpy) installiert ist, dann funktioniert

```
Set objSpy = GetObject("", "XMLSpy.Application")
```

nicht, da in diesem Fall in der Registry kein Objekt mit einer ProgID `XMLSpy.Application` registriert ist. in diesem Fall ist das registrierte Objekt `AuthenticDesktop.Application`.

Bei dem in dieser Dokumentation aufgelisteten Code wird davon ausgegangen, dass sowohl Authentic Desktop als auch XMLSpy installiert ist. Wenn nur Authentic Desktop installiert ist, ändern Sie bitte die Codefragmente entsprechend.
Das Application-Objekt besteht aus den folgenden Teilen:

1. Dokumentsammlung und Referenz auf das aktive Dokument.
2. Referenz auf das aktuelle Projekt und Methoden zum Erstellen und Öffnen von Projekten.
3. Methoden zur Unterstützung des Exports und Imports aus/in Datenbanken, Textdateien und Word-Dokumente.
4. URL-Verwaltung.
5. Methoden für Makro-Menübefehle.

Nach Erstellung eines Application-Objekts können Sie beginnen, die Funktionalitäten von Authentic Desktop zu nutzen. In den meisten Fällen werden Sie entweder ein Projekt öffnen und die Dokumente von dort aufrufen oder ein Dokument direkt über die Documents 493 Schnittstelle öffnen.

14.3.1.2 Programmiersprachen

Der Zugriff auf die COM-Schnittstelle erfolgt in den verschiedenen Programmiersprachen auf unterschiedliche Weise. Eine Beschreibung dazu finden Sie anhand einiger Beispiele für die gebräuchlichsten Programmiersprachen (*Links siehe unten*). Anhand der Codefragmente in diesem Abschnitt wird gezeigt, wie grundlegende Funktionalitäten aufgerufen werden. Sie können diese Funktionalitäten anhand der Dateien im Unterordner "API Examples" des Ordners `Examples` testen:

Windows 7, Windows 8, Windows 10, Windows 11	`C:\Benutzer\<Benutzername>\Dokumente\Altova\Authentic\2023\%APPNAME%>Examples`

JScript

Im JScript-Code werden die folgenden grundlegenden Funktionen verwendet:

- Starten der Applikation oder Anhängen einer laufenden Instanz 345
- Einfacher Dokumentzugriff 346
- Iteration 347
- Fehlerbehandlung 348
- Events 348

VBScript

VBScript unterscheidet sich nur syntaktisch von JScript und funktioniert ansonsten auf genau die gleiche Art. Der unten aufgelistete Code enthält ein Beispiel für die Verwendung von VBScript. Nähere Informationen finden Sie in den JScript-Beispielen 345.

- Events 350: Zeigt, wie Events mit VBScript verwendet werden.

C#

Sie können mit Hilfe von C# auf die Funktionalitäten der Application API zugreifen. Der unten aufgelistete Code enthält ein Beispiel dafür, wie Sie die API für bestimmte grundlegende Funktionalitäten aufrufen können.

- Start Authentic Desktop 358: Startet Authentic Desktop, das als Automation Server registriert ist, oder aktiviert das Programm, wenn Authentic Desktop bereits ausgeführt wird.
- Open OrgChart.pxf 360: Navigiert zu einem der mit Authentic Desktop installierten Beispieldokumente und öffnet es. Wenn das Dokument bereits offen ist, wird es zum aktiven Dokument.
- OnDocumentOpened Event On/Off 362: Zeigt, wie die Applikation auf Authentic Desktop Events hört. Ist die Funktion aktiv, erscheint ein Meldungsfeld, nachdem ein Dokument geöffnet wurde.
- Open ExpReport.xml 360: Öffnet ein weiteres Beispieldokument.
- Toggle View Mode 361:: Wechselt in der Ansicht aller offenen Dokumente zwischen der Browser- und der Authentic-Ansicht. Im Code wird gezeigt, wie Sie durch offene Dokumente iterieren können.
- Validate 361: Validiert das aktive Dokument und zeigt das Ergebnis in einem Meldungsfeld an. Im Code wird gezeigt, wie Fehler und COM-Ausgabeparameter behandelt werden.
- Shutdown Authentic Desktop 358: Beendet Authentic Desktop.

Java

Sie können mit Hilfe von Java auf die Funktionalitäten der Application API zugreifen. In diesem Abschnitt 364 wird erklärt, wie einige grundlegende XMLSpy-Funktionen über Java-Code aufgerufen werden können. Der Abschnitt ist in die folgenden Unterabschnitte gegliedert:

- Mapping-Regeln für den Java Wrapper 364
- Java-Beispielprojekt 365
- Starten und Beenden der Applikation 358
- Einfacher Dokumentaufruf 370
- Iterationen 370
- Verwendung von Out-Parametern 371
- Event Handler 371

 © 2016-2022 Altova GmbH

14.3.1.2.1 JScript

Dieser Abschnitt enthält eine Liste von JScript-Code, anhand dessen die folgenden grundlegenden Funktionalitäten erläutert werden:

- Starten der Applikation oder Anhängen einer laufenden Instanz 345
- Einfacher Dokumentzugriff 346
  - Iteration 347
- Fehlerbehandlung 348
- Events 348

Beispieldateien

Sie finden den Code in diesem Abschnitt in den Beispieldateien, die Sie je nach Bedarf testen und modifizieren können. Die JScript-Beispieldateien befinden sich im Ordner `JScript` des Ordners API Examples:

Windows 7, Windows 8, Windows 10, Windows 11	`C:\Benutzer\<Benutzername>\Dokumente\Altova\Authentic\2023\%APPNAME%>Examples`

Die Beispieldateien können auf zwei Arten ausgeführt werden:

- *Über die Befehlszeile:* Öffnen Sie ein Eingabeaufforderungsfenster, wechseln Sie zum obigen Verzeichnis und geben Sie den Namen eines der Beispiel-Skripts ein (z.B. `Start.js`).
- *Über den Windows Explorer:* Navigieren Sie im Windows Explorer zur JScript-Datei und doppelklicken Sie darauf.

Der mit dem Windows-Betriebssystem verfügbare Windows Script Host führt das Skript aus. Nähere Informationen zum Windows Script Host finden Sie in der MSDN-Dokumentation (https://msdn.microsoft.com).

14.3.1.2.1.1 Applikation starten

Mit dem unten aufgeführten JScript-Code wird die Applikation gestartet und beendet. Wenn das COM-Objekt der 32-Bit-Version von Authentic Desktop nicht gefunden wird, wird nach dem COM-Objekt der 64-Bit-Applikation gesucht; andernfalls wird ein Fehler ausgegeben. Wenn bereits eine Instanz der Applikation ausgeführt wird, so wird diese Instanz aufgerufen.

Anmerkung: Für die 32-Bit-Version von Authentic Desktop ist der registrierte Name oder der programmatische Identifier (ProgId) des COM-Objekts `AuthenticDesktop.Application`. Für die 64-Bit-Version von Authentic Desktop ist der Name `AuthenticDesktop_x64.Application`. Beachten Sie jedoch, dass das aufrufende Programm die CLASSES Registry-Einträge in seiner eigenen Registry Hive oder -Gruppe (32-Bit oder 64-Bit) aufruft. Wenn Sie daher Skripts über die Standardbefehlszeileneingabe und mit Windows Explorer auf einem 64-Bit-Windows-System ausführen, werden die 64-Bit-Registry-Einträge, welche auf die 64-Bit-Version von Authentic Desktop verweisen, aufgerufen. Wenn daher sowohl Authentic Desktop 32-Bit als auch die 64-Bit-Version installiert ist, ist eine spezielle Behandlung erforderlich, damit die 32-Bit-Version von Authentic Desktop aufgerufen wird. Angenommen, der Windows Skripting Host ist das aufrufende Programm, so

gehen Sie folgendermaßen vor:

1. Wechseln Sie in das Verzeichnis **C:\Windows\SysWOW64**.
2. Geben Sie in der Befehlszeile **wscript.exe** gefolgt vom Pfad zum gewünschten Skript ein, z.B:

```
wscript.exe "C:\Users\...
\Documents\Altova\Authentic2023\AuthenticExamples\API\JScript\start.js"
```

```
// Initialize application's COM object. This will start a new instance of the application
and
// return its main COM object. Depending on COM settings, the main COM object of an
already
// running application might be returned.

try{   objAuthentic = WScript.GetObject("", "AuthenticDesktop.Application");   }
catch(err) {}

if( typeof( objAuthentic ) == "undefined")
{
  try  {   objAuthentic = WScript.GetObject("", "AuthenticDesktop_x64.Application")   }
  catch(err)
  {
     WScript.Echo( "Can't access or create AuthenticDesktop.Application");
     WScript.Quit();
  }
}

// if newly started, the application will start without its UI visible. Set it to
visible.
objAuthentic.Visible = true;

WScript.Echo(objAuthentic.Edition + " has successfully started. ");

objAuthentic.Visible = false; // will shutdown application if it has no more COM
connections
//objAuthentic.Visible = true;   // will keep application running with UI visible
```

Der oben angeführte JScript-Code steht in der Datei `Start.js` zur Verfügung (siehe Beispieldateien 345).

14.3.1.2.1.2 Einfacher Dokumentaufruf

Nachdem Sie die Applikation, wie unter Applikation starten 345 beschrieben, gestartet haben, werden Sie wahrscheinlich ein Dokument programmatisch öffnen wollen, um damit zu arbeiten. Im nachstehenden JScript-Codefragment sehen Sie, wie Sie zwei Dokumente aus dem Authentic Desktop-Beispielordner öffnen und eines davon als aktives Dokument definieren.

```
// Locate examples via USERPROFILE shell variable. The path needs to be adapted to major
```

 © 2016-2022 Altova GmbH

```
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();
```

Sie finden den oben aufgelisteten JScript-Code in der Beispieldatei `DocumentAccess.js` (siehe Beispieldateien 345).

14.3.1.2.1.3 Iteration

Im unten aufgelisteten JScript-Code wird gezeigt, wie man durch offene Dokumente iteriert. Es wird davon ausgegangen, dass Sie die Applikation bereits gestartet und einige Dokumente, wie in den vorherigen Abschnitten beschrieben, geöffnet haben.

```
// go through all open documents using a JScript Enumerator
bRequiresSaving = false;
for(variterDocs = newEnumerator(objAuthentic.Documents); !iterDocs.atEnd();
iterDocs.moveNext())
{
  if(iterDocs.item().IsModified)
     bRequiresSaving = true;

  varstrErrorText = newArray(1);
  varnErrorNumber = newArray(1);
  varerrorData = newArray(1);

  if(!iterDocs.item().IsValid(strErrorText, nErrorNumber, errorData))
  {
      vartext = 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);
```

Sie finden den oben aufgelisteten JScript-Code in der Beispieldatei `DocumentAccess.js` (siehe Beispieldateien 345).

14.3.1.2.1.4 Fehlerbehandlung

Die Applikations-API gibt Fehler auf zwei verschiedene Arten zurück:

- das von jeder API-Methode zurückgegebene `HRESULT`
- die `IErrorInfo`-Schnittstelle der Applikations-API

Jede API-Methode gibt ein `HRESULT` zurück. Diese Rückgabewert enthält Informationen über Fehler, die während der Ausführung der Methode aufgetreten sind. Wenn der Aufruf erfolgreich war, so ist der Rückgabewert `S_OK`. Die Option `HRESULT` wird häufig in C/C++ Programmen verwendet.

Programmiersprachen wie VisualBasic sowie Skripting-Sprachen und andere Entwicklungsumgebungen gestatten dem Programmierer jedoch keinen Zugriff auf das zurückgegebene HRESULT eines COM-Aufrufs. Sie verwenden die IErrorInfo Schnittstelle, die ebenfalls von der Applikations-API unterstützt wird. Bei Auftreten eines Fehlers erstellt die API ein neues Objekt, das die IErrorInfo Schnittstelle implementiert. Die aus der IErrorInfo Schnittstelle stammenden Informationen werden in die Entwicklungsumgebung in ihren eigenen Fehlerbehandlungsmechanismus importiert.

So wird etwa mit dem nachstehenden JScript-Code ein Fehler zurückgegeben, weil ein Array falsch deklariert wurde. Weitere Informationen zum Fehler-Objekt stehen in seinen Eigenschaften `number` und `description` zur Verfügung.

```
try{
  vararr = newArray(-1);
}
catch(err) {
  WScript.Echo("Error : ("+ (err.number & 0xffff) + ")"+ err.description);
}
```

14.3.1.2.1.5 Events

COM definiert einen Client, der sich über den Connection Point-Mechanismus als Server für Callbacks registrieren muss. Das Automation Interface für XMLSpy definiert die dafür erforderlichen Event Interfaces. Wie

 © 2016-2022 Altova GmbH

eine Verbindung zu diesen Events hergestellt wird, hängt von der Programmiersprache ab, die Sie auf Ihrem Client verwenden. Im folgenden Codefragment wird gezeigt, wie dies in JScript erfolgt.

Die Methode `WScript.ConnectObject` dient zum Empfang von Events.

```
// The event-handler function
functionDocEvent_OnBeforeCloseDocument(objDocument)
{
  WScript.Echo("Received event - before closing document");
}

// Create or connect to XMLSpy (or Authentic Desktop)
try
{
  // Create the environment and XMLSpy (or Authentic Desktop)
  objWshShell = WScript.CreateObject("WScript.Shell");
  objFSO = WScript.CreateObject("Scripting.FileSystemObject");
  objSpy = WScript.GetObject("", "XMLSpy.Application");

// If only Authentic Desktop is installed (and XMLSpy is not installed) use:
// objSpy = WScript.GetObject("", "AuthenticDesktop.Application")

}
catch(err)
  { WScript.Echo ("Can't create WScript.Shell object or XMLSpy"); }

// Create document object and connect to its events
objSpy.Visible = true;
majorVersionYear = objSpy.MajorVersion + 1998
docPath = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\\Documents\\Altova\
\XMLSpy"+ majorVersionYear + "\\Examples\\ExpReport.xml";
objDoc = objSpy.Documents.OpenFile (docPath, false);
WScript.ConnectObject(objDoc, "DocEvent_");

// Keep running while waiting for the event
// In the meanwhile close this document in XMLSpy (or Authentic Desktop) manually
WScript.Echo ("Sleeping for 10 seconds ...");
WScript.Sleep (10000);

objDoc = null;
WScript.Echo ("Stopped listening for event");
objSpy.Quit();
```

14.3.1.2.2 VBScript

VBScript unterscheidet sich syntaktisch von JScript, funktioniert ansonsten aber auf genau die gleiche Art. Dieser Abschnitt enthält Beispielcode um zu zeigen, wie Events mit VBScript verwendet werden 350 sowie ein Beispiel 351.

Informationen über andere Funktionalitäten finden Sie in den unten aufgeführten JScript-Beispielen:

- Starten der Applikation oder Anhängen einer laufenden Instanz 345
- Einfacher Dokumentaufruf 346
- Iteration 347
- Fehlerbehandlung 348

14.3.1.2.2.1 Events

COM definiert einen Client, der sich über den Connection Point-Mechanismus als Server für Callbacks registrieren muss. Das Automation Interface für XMLSpy definiert die dafür erforderlichen Event Interfaces. Wie eine Verbindung zu diesen Events hergestellt wird, hängt von der Programmiersprache ab, die Sie auf Ihrem Client verwenden. Im folgenden Codefragment wird gezeigt, wie dies im VBScript erfolgt.

Die Methode `WScript.ConnectObject` dient zum Empfangen von Events.

Um diesen Code auszuführen, fügen Sie ihn in eine Datei mit der Erweiterung .vbs ein und doppelklicken Sie entweder in Windows Explorer darauf oder führen Sie die Datei über die Befehlszeile aus.

```
' the event handler function
FunctionDocEvent_OnBeforeCloseDocument(objDocument)
  CallWScript.Echo("received event - before closing document")
EndFunction

' create or connect to XmlSpy
SetobjWshShell = WScript.CreateObject("WScript.Shell")
SetobjFSO = WScript.CreateObject("Scripting.FileSystemObject")
SetobjSpy = 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
SetobjDoc = objSpy.Documents.OpenFile (docPath, False)
CallWScript.ConnectObject(objDoc, "DocEvent_")

' keep running while waiting on the event
' in the meantime close the document in XMLSPY manually
CallWScript.Echo ("sleeping for 10 seconds ...")
CallWScript.Sleep (10000)

SetobjDoc = Nothing
```

 © 2016-2022 Altova GmbH

```
CallWScript.Echo ("stopped listening for event")
CallobjSpy.Quit
```

Anmerkung: Für die 32-Bit-Version von Authentic Desktop ist der registrierte Name oder der programmatische Identifier (ProgId) des COM-Objekts `AuthenticDesktop.Application`. Für die 64-Bit-Version von Authentic Desktop ist der Name `AuthenticDesktop_x64.Application`. Beachten Sie jedoch, dass das aufrufende Programm die CLASSES Registry-Einträge in seiner eigenen Registry Hive oder -Gruppe (32-Bit oder 64-Bit) aufruft. Wenn Sie daher Skripts über die Standardbefehlszeileneingabe und mit Windows Explorer auf einem 64-Bit-Windows-System ausführen, werden die 64-Bit-Registry-Einträge, welche auf die 64-Bit-Version von Authentic Desktop verweisen, aufgerufen. Wenn daher sowohl Authentic Desktop 32-Bit als auch die 64-Bit-Version installiert ist, ist eine spezielle Behandlung erforderlich, damit die 32-Bit-Version von Authentic Desktop aufgerufen wird. Angenommen, der Windows Skripting Host ist das aufrufende Programm, so gehen Sie folgendermaßen vor:

1. Wechseln Sie in das Verzeichnis **C:\Windows\SysWOW64**.
2. Geben Sie in der Befehlszeile **wscript.exe** gefolgt vom Pfad zum gewünschten Skript ein, z.B:

```
wscript.exe "C:\Users\...
\Documents\Altova\Authentic2023\AuthenticExamples\API\JScript\start.js"
```

14.3.1.2.2.2 Beispiel: Verwendung von Events

Die Authentic-Ansicht unterstützt die Verbindung zu Events pro Objekt. Die Implementierung dieser Funktion basiert auf COM Connection Points und steht in Umgebungen zur Verfügung, die diesen Mechanismus unterstützen.

Das folgende Beispiel ist ein Beispiel für VBScript-Code, in dem gezeigt wird, wie man von einem VBScript-Projekt aus Events verwendet.

```
' ---------------------------------------------------------------------------
' VBScript example that demonstrates how to use events.
' ---------------------------------------------------------------------------

' Event handler for OnSelectionChanged event of AuthenticView
FunctionAuthenticViewEvent_OnSelectionChanged(objAuthenticRange)
   IfobjAuthenticRange.FirstTextPosition <> objAuthenticRange.LastTextPosition Then
       CallWScript.Echo("Selection: " & objAuthenticRange.Text & vbNewLine & vbNewLine &
"Close this dialog.")
   Else
       CallWScript.Echo("Cursor position: " & objAuthenticRange.FirstTextPosition &
vbNewLine & vbNewLine & "Close this dialog.")
   EndIf
EndFunction

' Start/access XMLSpy and connect to its automation interface.
```

```
SetWshShell = WScript.CreateObject("WScript.Shell")
SetobjSpy = 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.
Setfso = CreateObject("Scripting.FileSystemObject")
Iffso.FileExists(docPath) Then
   ' open the document
   CallobjSpy.Documents.OpenFile(docPath, False)
   setobjDoc = 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.
   setobjView = objDoc.AuthenticView
   CallWScript.ConnectObject(objView, "AuthenticViewEvent_")
   CallWScript.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.
   setobjView = Nothing
Else
   CallWScript.Echo("The file " & docPath & " does not exist.")
EndIf

' shut down XMLSpy when this script ends
objSpy.Visible = False
```

14.3.1.2.3 C#

Sie können mit Hilfe von C# auf die Funktionalitäten der Application API zugreifen. Zur Erstellung des C#-Code können Sie Visual Studio 2012/2013/2015/2017/2019/2022 verwenden und den Code in einem Visual Studio-Projekt speichern. Erstellen Sie das Projekt folgendermaßen:

1. Fügen Sie in Microsoft Visual Studio mit dem Befehl **Datei | Neu | Projekt** ein neues Projekt hinzu.
2. Fügen Sie durch Auswahl des Befehls **Projekt | Verweis hinzufügen** hinzufügen eine Referenz zur Authentic Desktop-Typbibliothek hinzu. Daraufhin wird das Dialogfeld "Verweis hinzufügen" angezeigt. Navigieren Sie zur Authentic Desktop-Typbibliothekkomponente, die sich im Authentic Desktop-Applikationsordner befindet und fügen Sie sie hinzu.

 © 2016-2022 Altova GmbH

3. Geben Sie den gewünschten Code ein.
4. Kompilieren Sie den Code und führen Sie ihn aus.

C#-Beispielprojekt

Das Authentic Desktop-Paket enthält ein C#-Beispielprojekt, das Sie im Unterordner `API\C#` des Ordners `Examples` finden:

Windows 7, Windows 8, Windows 10, Windows 11	`C:\Benutzer\<Benutzername>\Dokumente\Altova\Authentic\2023\%APPNAME%>Examples`

Sie können das Projekt innerhalb von Visual Studio 2012/2013/2015/2017/2019/2022 kompilieren und ausführen.

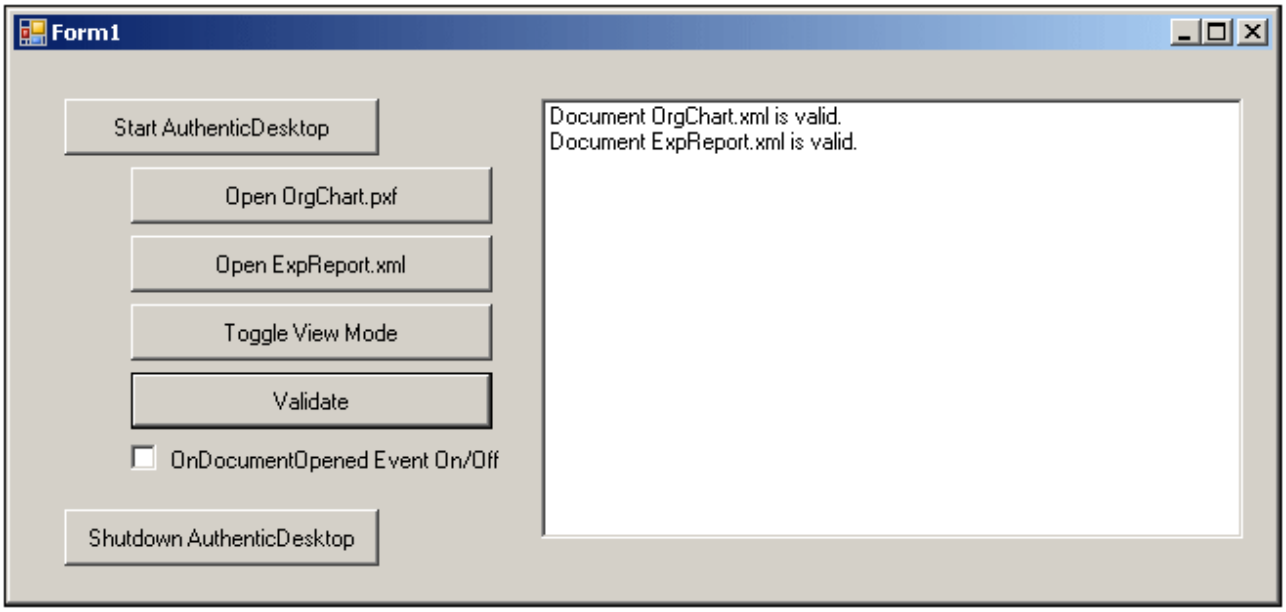
Anhand der unten stehenden Codefragmente wird gezeigt, wie Sie Grundfunktionen der Applikation verwenden können. Der Code ist dem Code im C#-Beispielprojekt im Ordner API Examples Ihres Applikationsordners ähnlich, kann sich aber geringfügig davon unterscheiden.

Platformkonfiguration

Wenn Sie ein 64-Bit-Betriebssystem haben und eine 32-Bit-Installation von Authentic Desktop verwenden, müssen Sie die x86-Plattform im Konfigurationsmanager der Projektmappe hinzufügen und das Beispiel mit dieser Konfiguration erstellen. Im Dialogfeld "Neue Projektmappenplattform" kann eine neue x86-Plattform (für die aktive Projektmappe in Visual Studio) erstellt werden (**Build | Konfigurations-Manager | Aktive Projektmappenplattform: | <Neu...>**).

Funktionen im Codebeispiel unten

In diesem Beispiel sehen Sie eine einfache Benutzeroberfläche (*Abbildung unten*) mit Schaltflächen, über die grundlegende Authentic Desktop-Operationen aufgerufen werden:



- Start Authentic Desktop 358: Startet Authentic Desktop, das als Automation Server registriert ist, oder aktiviert das Programm, wenn Authentic Desktop bereits ausgeführt wird.
- Open OrgChart.pxf 360: Navigiert zu einem der mit Authentic Desktop installierten Beispieldokumente und öffnet es. Wenn das Dokument bereits offen ist, wird es zum aktiven Dokument.
- Open ExpReport.xml 360: Öffnet ein weiteres Beispieldokument.
- Toggle View Mode 361:: Wechselt in der Ansicht aller offenen Dokumente zwischen der Browser- und der Authentic-Ansicht. Im Code wird gezeigt, wie Sie durch offene Dokumente iterieren können.
- Validate 361: Validiert das aktive Dokument und zeigt das Ergebnis in einem Meldungsfeld an. Im Code wird gezeigt, wie Fehler und COM-Ausgabeparameter behandelt werden.
- Shutdown Authentic Desktop 358: Beendet Authentic Desktop.

Sie können den Code (des Codefragments unten oder des C#-Beispiels aus dem Ordner API Examples) beliebig modifizieren und ausführen.

Kompilieren und Ausführen des Beispiels

Doppelklicken Sie im Ordner API Examples auf die Datei `AutomateAuthenticDesktop_VS2008.sln` bzw. auf die Datei `AutomateAuthenticDesktop_VS2010.sln` (um sie in Visual Studio 2012/2013/2015/2017/2019/2022 zu öffnen). Alternativ dazu können Sie die Datei auch von Visual Studio aus öffnen (mit dem Befehl **Datei | Öffnen | Projekt/Projektmappe**). Um das Beispiel zu kompilieren und auszuführen wählen Sie **Debuggen | Debuggen starten** bzw. **Debuggen | Starten ohne Debugging**.

Code des Beispiels

Nachstehend sehen Sie den C#-Code der Grundfunktionen des im `AutomateAuthenticDesktop` Beispiel erstellten Formulars (`Form1.cs`). Beachten Sie, dass ich der unten aufgelistete Code geringfügig von dem Code im Formular unter API Examples unterscheiden kann. Der Code ist zum besseren Verständnis mit Kommentaren versehen. Abhängig davon, welche Application API-Funktionen sie aufrufen, sind Teile dieses Codes auch separat in Unterabschnitten dieses Abschnitts beschrieben.

Der Code besteht im Wesentlichen aus einer Reihe von Handlern für die Schaltflächen der oben gezeigten Benutzeroberfläche.

```
namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        // An instance of AuthenticDesktop is accessed via its automation interface.
        XMLSpyLib.Application AuthenticDesktop;

        // Location of examples installed with AuthenticDesktop
        String strExamplesFolder;

        private void Form1_Load(object sender, EventArgs e)
        {
            // Locate examples installed with AuthenticDesktop.
```

 © 2016-2022 Altova GmbH

```
            // REMARK: You might need to adapt this if you have a different major version
of the product.
            strExamplesFolder = Environment.GetEnvironmentVariable("USERPROFILE") + "\\My
Documents\\Altova\\Authentic2012\\AuthenticExamples\\";
        }

        // Handler for the "Start AuthenticDesktop" button
        private void StartAuthenticDesktop_Click(object sender, EventArgs e)
        {
            if (AuthenticDesktop == null)
            {
                Cursor.Current = Cursors.WaitCursor;

                // If there is no AuthenticDesktop instance, create one and make it
visible.
                AuthenticDesktop = new XMLSpyLib.Application();
                AuthenticDesktop.Visible = true;

                Cursor.Current = Cursors.Default;
            }
            else
            {
                // If an AuthenticDesktop instance is already running, make sure it's
visible.
                if (!AuthenticDesktop.Visible)
                    AuthenticDesktop.Visible = true;
            }
        }

        // Handler for the "Open OrgChart.pxf" button
        private void openOrgChart_Click(object sender, EventArgs e)
        {
            // Make sure there's a running Authentic Desktop instance, and that it's
visible
            StartAuthenticDesktop_Click(null, null);

            // Open a sample file installed with the product.
            AuthenticDesktop.Documents.OpenFile(strExamplesFolder + "OrgChart.pxf", false);
        }

        // Handler for the "Open ExpReport.xml" button
        private void openExpReport_Click(object sender, EventArgs e)
        {
            // Make sure there's a running Authentic Desktop instance, and that it's
visible
            StartAuthenticDesktop_Click(null, null);

            // Open a sample file installed with the product.
            AuthenticDesktop.Documents.OpenFile(strExamplesFolder + "ExpReport.xml",
false);
        }

        // Handler for the "Toggle View Mode" button
        private void toggleView_Click(object sender, EventArgs e)
        {
```

```
            // Make sure there's a running Authentic Desktop instance, and that it's
visible
            StartAuthenticDesktop_Click(null, null);

            // Iterate through all open documents and toggle the current view between Text
View and Authentic View.
            foreach (XMLSpyLib.Document doc in AuthenticDesktop.Documents)
                if (doc.CurrentViewMode == XMLSpyLib.SPYViewModes.spyViewAuthentic)
                    doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewBrowser);
                else
                    doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewAuthentic);
        }

        // Handler for the "Shut down AuthenticDesktop" button
        // Shut down application instance by explicitely releasing the COM object.
        private void shutdownAuthenticDesktop_Click(object sender, EventArgs e)
        {
            if (AuthenticDesktop != null)
            {
                // Allow shut down of AuthenticDesktop by releasing the UI
                AuthenticDesktop.Visible = false;

                // Explicitly release the COM object
                try
                {
                    while
(System.Runtime.InteropServices.Marshal.ReleaseComObject(AuthenticDesktop) > 0) ;
                }
                finally
                {
                    // Avoid subsequent access to this object.
                    AuthenticDesktop = null;
                }
            }
        }

        // Handler for the "Validate" button
        private void validate_Click(object sender, EventArgs e)
        {
            // COM errors get returned to C# as exceptions. Use a try/catch block to handle
them.
            try
            {
                // Method 'IsValid' is one of the few functions that use output parameters.
                // Use 'object' type for these parameters.
                object strErrorText = "";
                object nErrorNumber = 0;
                object errorData = null;

                if (!AuthenticDesktop.ActiveDocument.IsValid(ref strErrorText, ref
nErrorNumber, ref errorData))
                {
                    // The COM call succeeds but the document is not valid.
                    // A detailed description of the problem is returned in strErrorText,
nErrorNumber and errorData.
```

```
                    listBoxMessages.Items.Add("Document " +
AuthenticDesktop.ActiveDocument.Name + " is not valid.");
                    listBoxMessages.Items.Add("\tErrorText  : " + strErrorText);
                    listBoxMessages.Items.Add("\tErrorNumber: " + nErrorNumber);
                    listBoxMessages.Items.Add("\tElement    : " + (errorData != null ?
((XMLSpyLib.XMLData)errorData).TextValue : "null"));
                }
                else
                {
                    // The COM call succeeds and the document is valid.
                    listBoxMessages.Items.Add("Document " +
AuthenticDesktop.ActiveDocument.Name + " is valid.");
                }
            }
            catch (Exception ex)
            {
                // The COM call was not successful.
                // Probably no application instance has been started or no document is
open.
                listBoxMessages.Items.Add("Error validating active document: " +
ex.Message);
            }
        }

        delegate void addListBoxItem_delegate(string sText);
        // Called from the UI thread
        private void addListBoxItem(string sText)
        {
            listBoxMessages.Items.Add(sText);
        }
        // Wrapper method to call UI control methods from a worker thread
        void syncWithUIthread(Control ctrl, addListBoxItem_delegate methodToInvoke, String
sText)
        {
            // Control.Invoke: Executes on the UI thread, but calling thread waits for
completion before continuing.
            // Control.BeginInvoke: Executes on the UI thread, and calling thread doesn't
wait for completion.
            if (ctrl.InvokeRequired)
                ctrl.BeginInvoke(methodToInvoke, new Object[] { sText });
        }

        // Event handler for OnDocumentOpened event
        private void handleOnDocumentOpened(XMLSpyLib.Document i_ipDocument)
        {
            String sText = "";

            if (i_ipDocument.Name.Length > 0)
                sText = "Document " + i_ipDocument.Name + " was opened!";
            else
                sText = "An empty document was created.";

            // Synchronize the calling thread with the UI thread because
            // COM events are triggered from a working thread
            addListBoxItem_delegate methodToInvoke = new
addListBoxItem_delegate(addListBoxItem);
```

```
            // Call syncWithUIthread with the following arguments:
            // 1 - listBoxMessages - list box control to display messages from COM events
            // 2 - methodToInvoke  - a C# delegate which points to the method which will be
called from the UI thread
            // 3 - sText           - the text to be displayed in the list box
            syncWithUIthread(listBoxMessages, methodToInvoke, sText);
        }


        private void checkBoxEventOnOff_CheckedChanged(object sender, EventArgs e)
        {
            if (AuthenticDesktop != null)
            {
                if (checkBoxEventOnOff.Checked)
                    AuthenticDesktop.OnDocumentOpened += new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);
                else
                    AuthenticDesktop.OnDocumentOpened -= new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);
            }
        }
    }
}
```

14.3.1.2.3.1 Hinzufügen einer Referenz zur Authentic Desktop API

Um die Typbibliothek der Applikations-API als Referenz in einem .NET-Projekt hinzuzufügen, gehen Sie folgendermaßen vor: Klicken Sie, während das .NET-Projekt geöffnet ist, auf den Befehl **Projekt | Verweis hinzufügen**. Navigieren Sie zur Typbibliothek `Authentic.tlb` im Authentic Desktop Applikationsordner.

Deklarieren Sie eine Variable zum Aufrufen der Authentic Desktop API

```
// An instance of Authentic Desktopis accessed via its automation interface.
XMLSpyLib.ApplicationAuthentic Desktop;
```

14.3.1.2.3.2 Starten und Beenden der Applikation

In den nachstehenden Codefragmenten wurden den Schaltflächen im Beispiel AutomateAuthenticDesktop 352 die Methoden `StartAuthenticDesktop_Click` und `ShutdownAuthenticDesktop_Click` zugewiesen. Sie starten bzw. beenden die Applikation. Dieses Beispiel befindet sich im Unterordner C# des Ordners API Examples (*siehe Datei Form1.cs)*:

Windows 7, Windows 8, Windows 10, Windows 11	`C:\Benutzer\<Benutzername>\Dokumente\Altova\Authentic\2023\%APPNAME%>Examples`

Sie können das Projekt von Visual Studio 2012/2013/2015/2017/2019/2022 aus kompilieren und ausführen.

 © 2016-2022 Altova GmbH

Starten von Authentic Desktop

Im folgenden Codefragment aus dem Beispiel AutomateAuthenticDesktop 352 wird gezeigt, wie man die Applikation startet.

```
 // Handler for the "Start AuthenticDesktop" button
        private void StartAuthenticDesktop_Click(object sender, EventArgs e)
        {
            if (AuthenticDesktop == null)
            {
                Cursor.Current = Cursors.WaitCursor;

                // If there is no AuthenticDesktop instance, create one and make it
visible.
                AuthenticDesktop = new XMLSpyLib.Application();
                AuthenticDesktop.Visible = true;

                Cursor.Current = Cursors.Default;
            }
            else
            {
                // If an instance of Authentic Desktop is already running, make sure it's
visible
                if (!AuthenticDesktop.Visible)
                    AuthenticDesktop.Visible = true;
            }
        }
```

Beenden von Authentic Desktop

Im folgenden Codefragment aus dem Beispiel AutomateAuthenticDesktop 352 wird gezeigt, wie man die Applikation beendet.

```
   // Handler for the "Shut down AuthenticDesktop" button
        // Shut down application instance by explicitely releasing the COM object.
        private void shutdownAuthenticDesktop_Click(object sender, EventArgs e)
        {
            if (AuthenticDesktop != null)
            {
                // Allow shut down of AuthenticDesktop by releasing the UI
                AuthenticDesktop.Visible = false;

                // Explicitly release the COM object
                try
                {
                    while
(System.Runtime.InteropServices.Marshal.ReleaseComObject(AuthenticDesktop) > 0) ;
                }
                finally
                {
                    // Avoid subsequent access to this object.
                    AuthenticDesktop = null;
```

```
                }
            }
        }
```

14.3.1.2.3.3 Öffnen von Dokumenten

In den unten stehenden Codefragmenten (aus dem Beispiel AutomateAuthenticDesktop[352]) wird gezeigt, wie zwei Dateien über zwei separate Methoden, die zwei Schaltflächen der Benutzeroberfläche zugewiesen wurden, geöffnet werden. Beide Methoden bedienen sich desselben Zugriffsmechanismus auf die Application API: Documents.OpenFile(string, boolean)[495].

Das AutomateAuthenticDesktop[352] (*siehe Datei form1.cs)*
befindet sich im Unterordner C# des Ordners API Examples:

Windows 7, Windows 8, Windows 10, Windows 11	`C:\Benutzer\<Benutzername>\Dokumente\Altova\Authentic\2023\%APPNAME%>Examples`

Sie können das Projekt von Visual Studio 2012/2013/2015/2017/2019/2022 aus kompilieren und ausführen.

Codefragment

```
       // Handler for the "Open OrgChart.pxf" button
        private void openOrgChart_Click(object sender, EventArgs e)
        {
            // Make sure there's a running Authentic Desktop instance, and that it's
visible
            StartAuthenticDesktop_Click(null, null);

            // Open a sample file installed with the product.
            AuthenticDesktop.Documents.OpenFile(strExamplesFolder + "OrgChart.pxf", false);
        }

        // Handler for the "Open ExpReport.xml" button
        private void openExpReport_Click(object sender, EventArgs e)
        {
            // Make sure there's a running Authentic Desktop instance, and that it's
visible
            StartAuthenticDesktop_Click(null, null);

            // Open a sample file installed with the product.
            AuthenticDesktop.Documents.OpenFile(strExamplesFolder + "ExpReport.xml",
false);
        }
```

Die zuletzt geöffnete Datei ist die aktive Datei.

© 2016-2022 Altova GmbH

14.3.1.2.3.4 Iterieren durch offene Dokumente

Im unten gezeigten Codefragment (aus dem Beispiel AutomateAuthenticDesktop[352]; *siehe Datei form1.cs*) wird gezeigt, wie man durch offene Dokumente iteriert. Anschließend wird in der Iterationsschleife eine Bedingung überprüft und es wird zwischen der Browser- und der Authentic-Ansicht gewechselt.

```
       // Handler for the "Toggle View Mode" button
        private void toggleView_Click(object sender, EventArgs e)
        {
            // Make sure there's a running Authentic Desktop instance, and that it's
visible
            StartAuthenticDesktop_Click(null, null);

            // Iterate through all open documents and toggle the current view between
Browser View and Authentic View.
            foreach (XMLSpyLib.Document doc in AuthenticDesktop.Documents)
                if (doc.CurrentViewMode == XMLSpyLib.SPYViewModes.spyViewAuthentic)
                    doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewBrowser);
                else
                    doc.SwitchViewMode(XMLSpyLib.SPYViewModes.spyViewAuthentic);
        }
```

DasAutomateAuthenticDesktop Beispiel[352] befindet sich im Unterordner C# des Ordners API Examples:

Windows 7, Windows 8, Windows 10, Windows 11	`C:\Benutzer\<Benutzername>\Dokumente\Altova\Authentic\2023\%APPNAME%>Examples`

Sie können das Projekt von Visual Studio 2012/2013/2015/2017/2019/2022 aus kompilieren und ausführen.

14.3.1.2.3.5 Fehler und COM-Ausgabeparameter

Im unten gezeigten Codefragment (aus dem Beispiel AutomateAuthenticDesktop[352]) wird gezeigt, wie man Fehler und COM-Ausgabeparameter behandelt. In der Methode AuthenticDesktop.ActiveDocument.IsValid(ref strErrorText, ref nErrorNumber, ref errorData)[482] werden Ausgabeparameter verwendet, mit Hilfe derer im unten gezeigten Codefragment der Text einer Fehlermeldung generiert wird.

Das AutomateAuthenticDesktop Beispiel[352] (*siehe Datei form1.cs)* befindet sich im Unterordner C# des Ordners API Examples:

Windows 7, Windows 8, Windows 10, Windows 11	`C:\Benutzer\<Benutzername>\Dokumente\Altova\Authentic\2023\%APPNAME%>Examples`

Sie können das Projekt von Visual Studio 2012/2013/2015/2017/2019/2022 aus kompilieren und ausführen.

Codefragment

```
        // Handler for the "Validate" button
        private void validate_Click(object sender, EventArgs e)
        {
            // COM errors get returned to C# as exceptions. Use a try/catch block to handle
them.
            try
            {
                // Method 'IsValid' is one of the few functions that use output parameters.
                // Use 'object' type for these parameters.
                object strErrorText = "";
                object nErrorNumber = 0;
                object errorData = null;

                if (!AuthenticDesktop.ActiveDocument.IsValid(ref strErrorText, ref
nErrorNumber, ref errorData))
                {
                    // The COM call succeeds but the document is not valid.
                    // A detailed description of the problem is returned in strErrorText,
nErrorNumber and errorData.
                    listBoxMessages.Items.Add("Document " +
AuthenticDesktop.ActiveDocument.Name + " is not valid.");
                    listBoxMessages.Items.Add("\tErrorText  : " + strErrorText);
                    listBoxMessages.Items.Add("\tErrorNumber: " + nErrorNumber);
                    listBoxMessages.Items.Add("\tElement    : " + (errorData != null ?
((XMLSpyLib.XMLData)errorData).TextValue : "null"));
                }
                else
                {
                    // The COM call succeeds and the document is valid.
                    listBoxMessages.Items.Add("Document " +
AuthenticDesktop.ActiveDocument.Name + " is valid.");
                }
            }
            catch (Exception ex)
            {
                // The COM call was not successful.
                // Probably no application instance has been started or no document is
open.
                listBoxMessages.Items.Add("Error validating active document: " +
ex.Message);
            }
        }
```

14.3.1.2.3.6 Events

Im unten gezeigten Codefragment (aus dem Beispiel AutomateAuthenticDesktop 352) wird der Code für zwei Event Handler aufgelistet. Das Beispiel AutomateAuthenticDesktop 352 (*siehe Datei form1.cs)* befindet sich im Unterordner C# des Ordners API Examples:

© 2016-2022 Altova GmbH

Windows 7, Windows 8, Windows 10, Windows 11	C:\Benutzer\<Benutzername>\Dokumente\Altova\Authentic\2023\%APPNAME%>Examples

Sie können das Projekt von Visual Studio 2012/2013/2015/2017/2019/2022 aus kompilieren und ausführen.

Codefragment

```
    delegate void addListBoxItem_delegate(string sText);
        // Called from the UI thread
        private void addListBoxItem(string sText)
        {
            listBoxMessages.Items.Add(sText);
        }
        // Wrapper method to call UI control methods from a worker thread
        void syncWithUIthread(Control ctrl, addListBoxItem_delegate methodToInvoke, String
sText)
        {
            // Control.Invoke: Executes on the UI thread, but calling thread waits for
completion before continuing.
            // Control.BeginInvoke: Executes on the UI thread, and calling thread doesn't
wait for completion.
            if (ctrl.InvokeRequired)
                ctrl.BeginInvoke(methodToInvoke, new Object[] { sText });
        }

        // Event handler for OnDocumentOpened event
        private void handleOnDocumentOpened(XMLSpyLib.Document i_ipDocument)
        {
            String sText = "";

            if (i_ipDocument.Name.Length > 0)
                sText = "Document " + i_ipDocument.Name + " was opened!";
            else
                sText = "An empty document was created.";

            // Synchronize the calling thread with the UI thread because
            // COM events are triggered from a working thread
            addListBoxItem_delegate methodToInvoke = new
addListBoxItem_delegate(addListBoxItem);
            // Call syncWithUIthread with the following arguments:
            // 1 - listBoxMessages - list box control to display messages from COM events
            // 2 - methodToInvoke  - a C# delegate which points to the method which will be
called from the UI thread
            // 3 - sText           - the text to be displayed in the list box
            syncWithUIthread(listBoxMessages, methodToInvoke, sText);
        }


        private void checkBoxEventOnOff_CheckedChanged(object sender, EventArgs e)
        {
            if (AuthenticDesktop != null)
            {
                if (checkBoxEventOnOff.Checked)
```

```
                    AuthenticDesktop.OnDocumentOpened += new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);
                else
                    AuthenticDesktop.OnDocumentOpened -= new
XMLSpyLib._IApplicationEvents_OnDocumentOpenedEventHandler(handleOnDocumentOpened);
            }
        }
```

14.3.1.2.4 Java

Sie können mit Hilfe von Java-Code auf die Applikations-API zugreifen. Damit Sie direkt vom Java-Code aus Zugriff auf den APPNAME%> Automation Server haben, müssen sich die unten aufgelisteten Bibliotheken im `classpath` befinden. Sie sind im Authentic Desktop Applikationsordner im Ordner `JavaAPI` installiert.

- `AltovaAutomation.dll`: ein JNI Wrapper für Altova Automation Server (Im Fall von 64-Bit-Versionen `AltovaAutomation_x64.dll`)
- `AltovaAutomation.jar`: Java-Klassen zum Aufrufen von Altova Automation Servern
- `AuthenticAPI.jar`: Java-Klassen, die als Wrapper für die Authentic Desktop Automation-Schnittstelle dienen
- `AuthenticAPI_JavaDoc.zip`: eine Javadoc Datei, die die Hilfedokumentation zur Java API enthält

Anmerkung: Um die Java API verwenden zu können, müssen sich die DLL und Jar-Dateien im Java Classpath befinden.

Java-Beispielprojekt

Im Lieferumfang Ihres Produkts ist ein Java-Beispielprojekt enthalten. Sie können das Java-Projekt nach Belieben testen und verwenden. Nähere Informationen zum Java-Beispielprojekt finden Sie im Abschnitt Java-Beispielprojekt 365.

Regeln für das Mappen der Application API-Namen auf Java

Für das Mapping zwischen der Application API und dem Java Wrapper gelten die folgenden Regeln:

- ***Klassen und Klassennamen***
  Für jedes Interface des Authentic Desktop Automation Interface gibt es eine Java-Klasse mit dem Namen des Interface.

- ***Methodennamen***
  Die Methodennamen im Java Interface sind dieselben wie die in den COM Interfaces, beginnen aber aufgrund der Java-Namenskonventionen mit einem Kleinbuchstaben. Zum Aufrufen von COM-Eigenschaften können dem Eigenschaftsnamen vorangestellte Java-Methoden mit `get` und `set` verwendet werden. Wenn eine Eigenschaft keinen Schreibzugriff ermöglicht, steht keine Setter-Methode zur Verfügung. Beispiel: Für die Eigenschaft `Name` des `Document` Interface stehen die Java-Methoden `getName` und `setName` zur Verfügung.

- ***Enumerationen***
  Für jede im Automation Interface definierte Enumeration ist eine Java-Enumeration desselben Namens und mit denselben Werten definiert.

- ***Events und Event Handler***

 © 2016-2022 Altova GmbH

Für jedes Interface im Automation Interface, das Events unterstützt, steht ein Java-Interface desselben Namens plus 'Event' zur Verfügung. Um das Überladen von Einzel-Events zu vereinfachen, gibt es eine Java-Klasse mit Standardimplementierungen für alle Events. Der Name dieser Java-Klasse ist der Name des Event Interface plus 'DefaultHandler'. Beispiel:
Application: Java-Klasse zum Aufrufen der Applikation
ApplicationEvents: Events Interface für die Applikation
ApplicationEventsDefaultHandler: Standard-Handler für ApplicationEvents

Ausnahmen für Mapping-Regeln

Zu den oben aufgelisteten Regeln gibt es die folgenden Ausnahmen:

Interface	Java-Name
Document, Methode SetEncoding	setFileEncoding
AuthenticView, Methode Goto	gotoElement
AuthenticRange, Methode Goto	gotoElement
AuthenticRange, Methode Clone	cloneRange

Dieser Abschnitt

In diesem Abschnitt wird erklärt, wie einige grundlegende Authentic Desktop-Funktionen über Java-Code aufgerufen werden können. Der Abschnitt ist in die folgenden Unterabschnitte gegliedert:

- Java-Beispielprojekt 365
- Starten und Beenden der Applikation 369
- Einfacher Dokumentaufruf 370
- Iterationen 370
- Verwendung von Out-Parametern 371
- Event Handler 371

14.3.1.2.4.1 Java-Beispielprojekt

Das Authentic Desktop-Installationspaket enthält ein Java-Beispielprojekt, das Sie im Unterordner API\Java des Ordners Examples finden:

Windows 7, Windows 8, Windows 10, Windows 11	C:\Benutzer\<Benutzername>\Dokumente\Altova\Authentic\2023\%APPNAME%>Examples

Dieser Ordner enthält Java-Beispiele für die Authentic Desktop API. Sie können das Beispielprojekt mit Hilfe der Batch-Datei BuildAndRun.bat, direkt über die Befehlszeile testen oder Sie können es in Eclipse kompilieren und ausführen. Anleitungen dafür finden Sie weiter unten.

Dateiliste

Der Ordner für die Java-Beispiele enthält alle zum Ausführen des Beispielprojekts erforderlichen Dateien. Diese Dateien sind unten aufgelistet. Wenn Sie eine 64-Bit-Version der Applikation verwenden, enthalten einige Dateinamen _x64. Diese Dateinamen sind mit `(_x64)` gekennzeichnet.

`AltovaAutomation(_x64).dll`	Java-COM Bridge: DLL-Teil
`AltovaAutomation.jar`	Java-COM Bridge: Java-Bibliotheksteil
`AuthenticAPI.jar`	Java-Klassen der Authentic Desktop API
`RunAuthenticDesktop.java`	Java-Beispielquellcode
`BuildAndRun.bat`	Batch-Datei zum Kompilieren und Ausführen des Beispielcodes über die Befehlszeile. Es wird ein Ordner benötigt, in dem sich die Java Virtual Machine als Parameter befindet.
`.classpath`	Hilfedatei Eclipse-Projekt
`.project`	Eclipse-Projektdatei
`Authentic_JavaDoc.zip`	`Javadoc` Datei, die die Hilfedokumentation für die Java API enthält

Funktionen in diesem Beispiel

In diesem Beispielprojekt wird Authentic Desktop gestartet und einige Operationen wie das Öffnen und Schließen von Dokumenten werden ausgeführt. Authentic Desktop bleibt danach geöffnet. Sie müssen die Applikation manuell schließen.

- Start Authentic Desktop 358: Startet Authentic Desktop, das als Automation Server registriert ist, oder aktiviert das Programm, wenn Authentic Desktop bereits ausgeführt wird.
- Open example files 370: Navigiert zu einem der mit Authentic Desktop installierten Beispieldokumente und öffnet es.
- Iterieren durch und Ändern des Ansichtsmodus 370: Wechselt von der Ansicht aller offenen Dokumente in die Browser-Ansicht. Im Code wird auch gezeigt, wie man durch offene Dokumente iteriert.
- Iteration, Validierung und Ausgabeparameter 371: Validiert das aktive Dokument und zeigt das Ergebnis in einem Meldungsfeld an. Im Code wird gezeigt, wie Ausgabeparameter verwendet werden.
- Event-Behandlung 371: Zeigt, wie XMLSpy Events behandelt werden.
- Shutdown Authentic Desktop 358: Beendet Authentic Desktop.

Sie können das Beispiel beliebig modifizieren und es ausführen.

Ausführen des Beispiels über die Befehlszeile

Um das Beispiel von der Befehlszeile aus auszuführen, öffnen Sie ein Eingabeaufforderungsfester, gehen Sie zum Ordner Java des Ordners API Examples (*Pfad siehe oben*) und geben Sie folgende Zeile ein:

```
buildAndRun.bat "<Path-to-the-Java-bin-folder>"
```

 © 2016-2022 Altova GmbH

Der Java Binary-Ordner muss von einer JDK 1.5 oder höheren Version auf Ihrem Rechner sein. Drücken Sie die **Eingabetaste.** Der Java-Quellcode in `RunAuthenticDesktop.java` wird kompiliert und anschließend ausgeführt.

Laden des Beispiels in Eclipse

Öffnen Sie Eclipse und wählen Sie den Befehl **Import | Existing Projects into Workspace** um die Eclipse-Projektdatei (`.project`) im Ordner Java des Ordners API Examples (*Pfad siehe oben*) zu Eclipse hinzuzufügen. Daraufhin wird das Projekt `RunAuthenticDesktop` in Ihrem Package Explorer oder Navigator angezeigt. Wählen Sie das Projekt aus und klicken Sie anschließend auf **Run as | Java Application** um das Beispiel auszuführen.

Anmerkung: Sie können einen Klassennamen oder eine Methode der Java API auswählen und F1 drücken, um Hilfe zu dieser Klasse oder Methode zu erhalten.

Java-Quellcode

Im Folgenden finden Sie den mit Kommentaren versehenen Java-Quellcode aus der Beispieldatei `RunAuthenticDesktop.java`.

```
01 // Access general JAVA-COM bridge classes
02 import com.altova.automation.libs.*;
03
04 // Access AuthenticDesktop Java-COM bridge
05 import com.altova.automation.AuthenticDesktop.*;
06 import com.altova.automation.AuthenticDesktop.Enums.SPYViewModes;
07
08 /**
09  * A simple example that starts AuthenticDesktop COM server and performs a view
operations on it.
10  * Feel free to extend.
11  */
12 public class RunAuthenticDesktop
13 {
14   public static void main(String[] args)
15   {
16     // An instance of the application.
17     Application authenticDesktop = null;
18
19     // Instead of COM error-handling, use Java exception mechanism.
20     try
21     {
22       // Start AuthenticDesktop as COM server.
23       authenticDesktop = new Application();
24       // COM servers start up invisible so we make it visible
25       authenticDesktop.setVisible(true);
26
27       // Locate samples installed with the product.
28       String strExamplesFolder = System.getenv("USERPROFILE") + "\\My Documents\\Altova\
\Authentic2012\\AuthenticExamples\\";
29
30       // Open two files from the product samples.
31       authenticDesktop.getDocuments().openFile(strExamplesFolder + "OrgChart.pxf",
false);
```

```
32       authenticDesktop.getDocuments().openFile(strExamplesFolder + "ExpReport.xml",
false);
33
34       // Iterate through all open documents and set the View Mode to 'Text'.
35       for (Document doc:authenticDesktop.getDocuments())
36         if ( doc.getCurrentViewMode() != SPYViewModes.spyViewText)
37           doc.switchViewMode(SPYViewModes.spyViewText);
38
39       // An alternative iteration mode is index-based. COM indices are typically zero-
based.
40       Documents documents = authenticDesktop.getDocuments();
41       for (int i = 1; i <= documents.getCount(); i++)
42       {
43         Document doc = documents.getItem(i);
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.
48         JVariant validationErrorText = new JVariant.JStringVariant("");
validationErrorText.setByRefFlag();
49         JVariant validationErrorCount = new JVariant.JIntVariant(0);
validationErrorCount.setByRefFlag();
50         JVariant validationErrorXMLData = new JVariant.JIDispatchVariant(0);
validationErrorXMLData.setByRefFlag();
51         if (!doc.isValid(validationErrorText, validationErrorCount,
validationErrorXMLData))
52           System.out.println("Document " + doc.getName() + " is not wellformed - " +
validationErrorText.getStringValue());
53         else
54           System.out.println("Document " + doc.getName() + " is wellformed.");
55       }
56
57       // The following lines attach to the document events using a default
implementation
58       // for the events and override one of its methods.
59       // If you want to override all document events it is better to derive your
listener class
60       // from DocumentEvents and implement all methods of this interface.
61       Document doc = authenticDesktop.getActiveDocument();
62       doc.addListener(new DocumentEventsDefaultHandler()
63       {
64         @Override
65         public boolean onBeforeCloseDocument(Document i_ipDoc) throws
AutomationException
66         {
67           System.out.println("Document " + i_ipDoc.getName() + " requested closing.");
68
69           // Allow closing of document
70           return true;
71         }
72       });
73       doc.close(true);
74       doc = null;
75
76       System.out.println("Watch AuthenticDesktop!");
77     }
```

 © 2016-2022 Altova GmbH

```
78     catch (AutomationException e)
79     {
80       // e.printStackTrace();
81     }
82     finally
83     {
84       // Make sure that AuthenticDesktop can shut down properly.
85       if (authenticDesktop != null)
86         authenticDesktop.dispose();
87
88       // Since the COM server was made visible and still is visible, it will keep
running
89       // and needs to be closed manually.
90       System.out.println("Now close AuthenticDesktop!");
91     }
92   }
93 }
```

14.3.1.2.4.2 Starten und Beenden der Applikation

Im nachstehenden Code wird gezeigt, wie man die Applikation startet und beendet.

Starten der Applikation

Bevor Sie die Applikation starten müssen die entsprechenden Klassen importiert werden (*siehe unten*).

```
01 // Access general JAVA-COM bridge classes
02 import com.altova.automation.libs.*;
03
04 // Access AuthenticDesktop Java-COM bridge
05 import com.altova.automation.AuthenticDesktop.*;
06 import com.altova.automation.AuthenticDesktop.Enums.SPYViewModes;
07
08 /**
09  * A simple example that starts AuthenticDesktop COM server and performs a view
operations on it.
10  * Feel free to extend.
11  */
12 public class RunAuthenticDesktop
13 {
14   public static void main(String[] args)
15   {
16     // An instance of the application.
17     Application authenticDesktop = null;
18
19     // Instead of COM error-handling, use Java exception mechanism.
20     try
21     {
22       // Start AuthenticDesktop as COM server.
23       authenticDesktop = new Application();
24       // COM servers start up invisible so we make it visible
25       authenticDesktop.setVisible(true);
26
```

```
27 ...
28     }
29   }
30 }
```

Beenden der Applikation

Die Applikation kann wie unten gezeigt beendet werden.

```
1 {
2       // Make sure that AuthenticDesktop can shut down properly.
3       if (authenticDesktop != null)
4         authenticDesktop.dispose();
5
6       // Since the COM server was made visible and still is visible, it will keep running
7       // and needs to be closed manually.
8       System.out.println("Now close AuthenticDesktop!");
9 }
```

14.3.1.2.4.3 Einfacher Dokumentzugriff

Im unten stehenden Code wird gezeigt, wie man ein Dokument öffnet.

```
1  // Locate samples installed with the product.
2  String strExamplesFolder = System.getenv("USERPROFILE") + "\\My Documents\\Altova\
\Authentic2012\\AuthenticExamples\\";
3
4  // Open two files from the product samples.
5  authenticDesktop.getDocuments().openFile(strExamplesFolder + "OrgChart.pxf", false);
6  authenticDesktop.getDocuments().openFile(strExamplesFolder + "ExpReport.xml", false);
```

14.3.1.2.4.4 Iterationen

Im nachstehenden Code wird gezeigt, wie man durch offene Dokumente iteriert.

```
01 // Iterate through all open documents and set the View mode to 'Browser'.
02 for (Document doc:authenticDesktop.getDocuments())
03   if ( doc.getCurrentViewMode() != SPYViewModes.spyViewBrowser)
04           doc.switchViewMode(SPYViewModes.spyViewBrowser);
05
06 // An alternative iteration mode is index-based. COM indices are typically zero-based.
07 Documents documents = authenticDesktop.getDocuments();
08 for (int i = 1; i <= documents.getCount(); i++)
09   {
10     Document doc = documents.getItem(i);
11     ...
12   }
```


© 2016-2022 Altova GmbH

14.3.1.2.4.5 Verwendung von Out-Parametern

Im unten stehenden Code wird durch offene Dokumente iteriert, wobei jedes davon validiert wird. Für jede Validierung wird mit Hilfe der Ausgabeparameter der Methode "Validation" eine Meldung generiert.

```
01 // Iterate through all open documents and set the View mode to 'Text'.
02 for (Document doc:authenticDesktop.getDocuments())
03   if ( doc.getCurrentViewMode() != SPYViewModes.spyViewText)
04        doc.switchViewMode(SPYViewModes.spyViewText);
05
06 // An alternative iteration mode is index-based. COM indices are typically zero-based.
07 Documents documents = authenticDesktop.getDocuments();
08 for (int i = 1; i <= documents.getCount(); i++)
09 {
10   Document doc = documents.getItem(i);
11
12 // Validation is one of the few methods that have output parameters.
13 // The class JVariant is the correct type for parameters in these cases.
14 // To get values back, mark them with the by-reference flag.
15 JVariant validationErrorText = new JVariant.JStringVariant("");
validationErrorText.setByRefFlag();
16 JVariant validationErrorCount = new JVariant.JIntVariant(0);
validationErrorCount.setByRefFlag();
17 JVariant validationErrorXMLData = new JVariant.JIDispatchVariant(0);
validationErrorXMLData.setByRefFlag();
18   if (!doc.isValid(validationErrorText, validationErrorCount, validationErrorXMLData))
19     System.out.println("Document " + doc.getName() + " is not wellformed - " +
validationErrorText.getStringValue());
20   else
21     System.out.println("Document " + doc.getName() + " is wellformed.");
22 }
```

14.3.1.2.4.6 Event Handler

Im nachstehenden Code wird gezeigt, wie die Applikation auf XMLSpy Events hört und diese verwendet.

```
01 // The following lines attach to the document events using a default implementation
02 // for the events and override one of its methods.
03 // If you want to override all document events, it is better to derive your listener
class
04 // from DocumentEvents and implement all methods of this interface.
05 Document doc = authenticDesktop.getActiveDocument();
06 doc.addListener(new DocumentEventsDefaultHandler()
07 {
08   @Override
09   public boolean onBeforeCloseDocument(Document i_ipDoc) throws AutomationException
10   {
11     System.out.println("Document " + i_ipDoc.getName() + " requested closing.");
12
13     // allow closing of document
```

```
14     return true;
15   }
16 });
17 doc.close(true);
18 doc = null;
```

14.3.2 Schnittstellen

Objekthierarchie
Application 372
SpyProject 543
SpyProjectItems 547
SpyProjectItem 544
Documents 493
Document 462
GridView 525
AuthenticView 428
AuthenticRange 399
AuthenticDataTransfer 394 (vormals DocEditDataTransfer)
AuthenticDataTransfer 394 (vormals DocEditDataTransfer)
TextView 551
XMLData 593
Dialogs 378
CodeGeneratorDlg 446
FileSelectionDlg 506
SchemaDocumentationDlg 529
GenerateSampleXMLDlg 519
DTDSchemaGeneratorDlg 497
FindInFilesDlg 508
DatabaseConnection 452
ExportSettings 503
TextImportExportSettings 548
ElementList 501
ElementListItem 502

Enumerations 604

Beschreibung
Dieses Kapitel enthält die Referenz der Authentic Desktop1.5 Typbibliothek.

Die meisten der hier beschriebenen Beispiele sind in VisualBasic verfasst. Bei diesen Codefragmenten wird davon ausgegangen, dass eine Variable namens **objSpy vom Typ Application** definiert und gesetzt ist. Einige der Codebeispiele wurden auch in JavaScript geschrieben.

14.3.2.1 Application

Methoden
GetDatabaseImportElementList 379
GetDatabaseSettings 380

 © 2016-2022 Altova GmbH

GetDatabaseTables 380
ImportFromDatabase 383
CreateXMLSchemaFromDBStructure 378

GetTextImportElementList 381
GetTextImportExportSettings 382
ImportFromText 385

ImportFromWord 386

ImportFromSchema 384

GetExportSettings 381

NewProject 387
OpenProject 387

AddMacroMenuItem 376
ClearMacroMenu 377

ShowForm 390

ShowApplication 390

URLDelete 391
URLMakeDirectory 391

AddXSLT_XQParameter 377
GetXSLT_XQParameterCount 383
GetXSLT_XQParameterName 383
GetXSLT_XQParameterXPath 383
RemoveXSLT_XQParameter 389

FindInFiles 379

Quit 388

Eigenschaften
Application 377
Parent 388

ActiveDocument 376
Documents 379

CurrentProject 378

Dialogs 378

WarningNumber 392
WarningText 392

Status 391
MajorVersion 386

MinorVersion 387
Edition 379
IsAPISupported 386
ServicePackVersion 390

Beschreibung
Application ist die Root für alle anderen Objekte. Es ist das einzige Objekt, das Sie mit CreateObject(VisualBasic) oder anderen ähnlichen COM-Funktionen erstellen können.

Beispiel

```
DimobjSpy AsApplication
SetobjSpy = CreateObject("XMLSpy.Application")
```

14.3.2.1.1 Events

14.3.2.1.1.1 OnBeforeOpenDocument

Event: OnBeforeOpenDocument(*objDialog* als FileSelectionDlg 506)

Beschreibung
Dieses Event wird immer dann ausgelöst, wenn ein Dokument mit dem Menübefehl OpenFile oder OpenURL geöffnet wird. Es wird nach dem Auswählen einer Dokumentdatei aber vor dem Öffnen des Dokuments gesendet. Das Dialogfeldobjekt zur Auswahl der Datei wird mit dem Namen der ausgewählten Dokumentdatei initialisiert. Sie können diese Auswahl ändern. Um mit dem Öffnen des Dokuments fortzufahren, belassen Sie die Eigenschaft FileSelectionDlg.DialogAction 507 von *io_objDialog* auf dem Standardwert spyDialogOK 607. Um das Öffnen des Dokuments abzubrechen, setzen Sie die Eigenschaft auf spyDialogCancel 607.

Beispiele
Im Folgenden finden Sie Beispiele dafür, wie ein Skript für dieses Event erstellt werden kann.

XMLSpy scripting environment - VBScript:

```
Function On_BeforeOpenDocument(objDialog)
End Function
```

XMLSpy scripting environment - JScript:

```
function On_BeforeOpenDocument(objDialog)
{
}
```

XMLSpy IDE Plugin:

```
IXMLSpyPlugIn.OnEvent(26, ...)   // nEventId = 26
```


© 2016-2022 Altova GmbH

14.3.2.1.1.2 OnBeforeOpenProject

Event: OnBeforeOpenProject(*objDialog* als FileSelectionDlg[506])

Beschreibung
Dieses Event wird ausgelöst, wenn eine Projektdatei ausgewählt, aber bevor sie geöffnet wird. Das Dialogfeldobjekt zur Auswahl der Datei wird mit dem Namen der ausgewählten Projektdatei initialisiert. Sie können diese Auswahl ändern. Um mit dem Öffnen des Projekts fortzufahren, belassen Sie die Eigenschaft FileSelectionDlg.DialogAction[507] von *io_objDialog* auf dem Standardwert spyDialogOK[607]. Um das Öffnen das Projekts abzubrechen, setzen Sie die Eigenschaft auf spyDialogCancel[607].

Beispiele
Im Folgenden sehen Sie einige Beispiele, wie ein Skript für dieses Event erstellt werden kann.

XMLSpy scripting environment - VBScript:
Function On_BeforeOpenProject(*objDialog*)
End Function

XMLSpy scripting environment - JScript:
function On_BeforeOpenProject(*objDialog*)
{
}

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent(25, ...) // nEventId = 25

14.3.2.1.1.3 OnDocumentOpened

Event: OnDocumentOpened(*objDocument* als Document[462])

Beschreibung
Dieses Event wird ausgelöst, wenn ein Dokument in Authentic Desktopgeöffnet wird. Voraussetzung ist, dass eine Datei über das Dialogfeld OpenFile oder OpenURL geöffnet oder eine neu Datei erstellt oder eine Datei in Authentic Desktopgezogen wird. Das neue Dokument wird als Parameter übergeben. Die Operation kann nicht abgebrochen werden.

Beispiele
Im Folgenden finden Sie Beispiele dafür, wie ein Skript für dieses Event erstellt werden kann.

XMLSpy Skripting-Umgebung - VBScript:
Function On_OpenDocument(*objDocument*)
End Function

XMLSpy Skripting-Umgebung - JScript:
function On_OpenDocument(*objDocument*)
{
}

XMLSpy IDE Plugin:

IXMLSpyPlugIn.OnEvent(7, ...) // nEventId = 7

14.3.2.1.1.4 OnProjectOpened

Event: OnProjectOpened(*objProject* als SpyProject[543])

Beschreibung
Dieses Event wird ausgelöst, wenn ein Projekt in Authentic Desktopgeöffnet wird. Das neue Projekt wird als Parameter übergeben.

Beispiele
Im Folgenden finden Sie Beispiele dafür, wie ein Skript für dieses Event erstellt werden kann.

XMLSpy Skripting-Umgebung - VBScript:
Function On_OpenProject(*objProject*)
End Function

XMLSpy Skripting-Umgebung - JScript:
function On_OpenProject(*objProject*)
{
}

XMLSpy IDE Plugin:
IXMLSpyPlugIn.OnEvent(6, ...) // nEventId = 6

14.3.2.1.2 ActiveDocument

Eigenschaft: ActiveDocumentals Document[462]

Beschreibung
Referenz auf das aktive Dokument. Wenn kein Dokument geöffnet ist, ist ActiveDocumentnull (nothing).

Fehler
1111 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.3 AddMacroMenuItem

Methode: AddMacroMenuItem(*strMacro*als String,*strDisplayText*als String)

Beschreibung
Fügt einen Menübefehl zum Menü **Extras** hinzu. Dieser neue Menübefehl ruft das von strMacrodefinierte Makro auf. Siehe auch "Beispiel: Skripting-Projekt[310]".

 © 2016-2022 Altova GmbH

Fehler

1111 Das application-Objekt ist nicht mehr gültig.
1100 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
1108 Die Anzahl der Makroelemente ist auf 16 beschränkt.

14.3.2.1.4 AddXSLT_XQParameter

Methode: AddXSLT_XQParameter(name als String, XPath als String)

Beschreibung
Fügt einen XSLT- oder XQuery-Parameter hinzu. Der Name und Wert des Parameters sind die beiden Argumente der Methode.

Fehler

1111 Das application-Objekt ist nicht mehr gültig.
1100 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
1124 Der XPath-Ausdruck wurde nicht definiert.
1125 Kein QName.
1126 Der definierte XPath ist nicht gültig. Der Grund dafür ist angehängt.
1127 Es existiert bereits ein Parameter mit dem angegebenen Namen.

14.3.2.1.5 Application

Eigenschaft: Applicationals Application 372 (schreibgeschützt)

Beschreibung
Ruft das Authentic Desktopapplication-Objekt auf.

Fehler

1111 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.6 ClearMacroMenu

Methode: ClearMacroMenu()

Rückgabewert
Keiner

Beschreibung
Entfernt alle Menübefehle, die durch Aufruf von `AddMacroMenuItem` 376 hinzugefügt wurden, aus dem Menü **Extras**. Siehe auch Beispiel:Skripting-Projekt 310.

Fehler
1111 Das application-Objekt ist nicht mehr gültig.

14.3.2.1.7 CreateXMLSchemaFromDBStructure

Methode: CreateXMLSchemaFromDBStructure(pImportSettingsals DatabaseConnection[452], pTablesals ElementList[501])

Beschreibung
CreateXMLSchemaFromDBStructureerstellt anhand der in pImportSettingsdefinierten Datenbank für die in pTables definierten Tabellen ein neues/neue Schema-Dokument(e), das/die die Struktur der Datenbanktabellen beschreibt/beschreiben.

Der Parameter pTablesdefiniert, welche Tabellenstrukturen das XML-Schema-Dokument enthalten soll. Dieser Parameter kann NULL sein und definiert in diesem Fall, dass alle Tabellenstrukturen exportiert werden sollen.

Siehe auch GetDataBaseTables[380].

Fehler
1112 Es wurde eine ungültige Datenbank definiert.
1120 Der Datenbankimport ist fehlgeschlagen.

14.3.2.1.8 CurrentProject

Eigenschaft: CurrentProjectals SpyProject[543]

Beschreibung
Referenz auf das aktive Dokument. Wenn kein Projekt geöffnet ist, ist CurrentProjectgleich Null (nothing).

Fehler
1111 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.9 Dialogs

Eigenschaft: Dialogsals Dialogs[458](schreibgeschützt)

Beschreibung
Ruft die integrierten Dialogfelder von Authentic Desktop auf.

Fehler
1111 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

 © 2016-2022 Altova GmbH

14.3.2.1.10 Documents

Eigenschaft: Documentsals Documents 493

Beschreibung
Sammlung aller offenen Dokumente.

Fehler
1111 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.11 Edition

Eigenschaft: Editionals String

Beschreibung
Gibt die Edition der Applikation, also z.B. "`Altova Authentic Desktop Enterprise Edition`" für die Enterprise Editionzurück.

Fehler
1111 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.12 FindInFiles

Method: FindInFiles(*pSettings*als FindInFilesDlg 508) als FindInFilesResults 518

Beschreibung
Gibt ein FindInFilesResults 518 Objekt zurück, das Informationen über die Dateien enthält, die den angegebenen Kriterien entsprechen.

Fehler
1111 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.13 GetDatabaseImportElementList

Methode: GetDatabaseImportElementList(*pImportSettings*als DatabaseConnection 452)als ElementList 501

Beschreibung
Die Funktion gibt eine Sammlung von ElementListItems zurück, wobei die ElementListItem.Name 503 Eigenschaften die Namen der Felder enthalten, die für den Import ausgewählt werden können und die ElementListItem.ElementKind 502 Eigenschaften, je nachdem, welcher Wert in DatabaseConnection.AsAttributes 453 übergeben wird, entweder auf *spyXMLDataAttr* oder *spyXMLDataElement* initialisiert werden. Diese Liste dient als Filter dafür, was schließlich durch einen zukünftigen Aufruf von

ImportFromDatabase[383] importiert wird. Verwenden Sie ElementList.RemoveElement[502] um Felder vom Export auszuschließen.

Für die Herstellung der Verbindung zur Datenbank muss eines der Objekte DatabaseConnection.File[455], DatabaseConnection.ADOConnection[452] oder DatabaseConnection.ODBCConnection[456] sowie DatabaseConnection.SQLSelect[457] ausgefüllt werden. Verwenden Sie die Eigenschaft DatabaseConnection.AsAttributes[453] um ElementListItem.ElementKind[502] der erzeugten Elementliste entweder auf *spyXMLDataAttr* oder *spyXMLDataElement* zu initialisieren.

Beispiel
Siehe Beispiel unter ImportFromDatabase[383].

Fehler

1111	Das application-Objekt ist nicht mehr gültig.
1100	Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
1107	Import aus Datenbank ist fehlgeschlagen.
1112	Es wurde eine ungültige Datenbank angegeben.
1114	Select-Anweisung fehlt.
1119	Die Liste der Datenbankelemente konnte nicht importiert werden.

14.3.2.1.14 GetDatabaseSettings

Methode: GetDatabaseSettings() als DatabaseConnection[452]

Beschreibung
GetDatabaseSettingserstellt ein neues Objekt für Datenbankeinstellungen. Mit Hilfe des Objekts werden die Datenbankverbindungsparameter für die Methoden GetDatabaseTables[380], GetDatabaseImportElementList[379], ImportFromDatabase[383], ImportFromSchema[384] und ExportToDatabase[474] definiert.

Beispiel
Siehe Beispiel von ImportFromDatabase[383].

Fehler

1111	Das application-Objekt ist nicht mehr gültig.
1100	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.15 GetDatabaseTables

Methode: GetDatabaseTables(*pImportSettings*als DatabaseConnection[372])als ElementList[501]

Beschreibung
GetDatabaseTablesliest die Tabellennamen aus der in *pImportSettings*definierten Datenbank. Für die Datenbankverbindung muss eine der folgenden Eigenschaften ausgefüllt werden DatabaseConnection.File[455], DatabaseConnection.ADOConnection[452] oder DatabaseConnection.ODBCConnection[456]. Alle anderen Eigenschaften werden ignoriert.
Die Funktion gibt eine Sammlung von ElementListItems zurück, wobei die ElementListItem.Name[503] Eigenschaften die Namen der in der angegebenen Datenbank gespeicherten Tabellen enthält. Die restlichen Eigenschaften von ElementListItem[502] werden nicht verwendet.

 © 2016-2022 Altova GmbH

Fehler

1111 Das application-Objekt ist nicht mehr gültig.
1100 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
1112 Es wurde eine ungültige Datenbank angegeben
1113 Fehler beim Lesen der Datenbanktabellendaten.
1118 Die Abfrage der Datenbanktabelle ist fehlgeschlagen.

Beispiel

```
        DimobjImpSettings AsDatabaseConnection
SetobjImpSettings = objSpy.GetDatabaseSettings
objImpSettings.ADOConnection = TxtADO.Text

'store table names in list box
ListTables.Clear

DimobjList AsElementList
DimobjItem AsElementListItem
OnErrorGoToErrorHandler
SetobjList = objSpy.GetDatabaseTables(objImpSettings)

        ForEachobjItem InobjList
                ListTables.AddItem objItem.Name
        Next
```

14.3.2.1.16 GetExportSettings

Methode: GetExportSettings()als ExportSettings 503 (schreibgeschützt)

Beschreibung

GetExportSettingserstellt ein neues Objekt allgemeiner Exporteinstellungen. Mit Hilfe dieses Objekts werden die Parameter an die Exportfunktionen übergeben und das Verhalten der Exportaufrufe wird definiert. Siehe auch die Exportfunktionen aus Document 462.

Fehler

1111 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.17 GetTextImportElementList

Methode: GetTextImportElementList(pImportSettingsals TextImportExportSettings 548)als ElementList 501

Beschreibung

GetTextImportElementListruft Importinformationen über die in pImportSettings definierte Textdatei ab. Die Funktion gibt eine Sammlung von ElementListItems zurück, wobei die ElementListItem.Name 503 Eigenschaften die Namen der in der Datei gefundenen Felder enthält. Die Werte der restlichen Eigenschaften werden nicht definiert.

Wenn die Textdatei keine Spaltenüberschriften enthält, setzen Sie pImportSettings.HeaderRow[550] auf false. Die erzeugte Elementliste enthält allgemeine Spaltennamen wie 'Field1' usw.

Fehler

1111 Das application-Objekt ist nicht mehr gültig.
1100 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
1107 Import aus Datenbank ist fehlgeschlagen.
1115 Fehler beim Import der Textelementliste. Für die Importdatei kann kein Parser erstellt werden.
1116 Fehler beim Import der Textelementliste.

Beispiel

```
' ---------------------------------------------------------
' VBA client code fragment - import selected fields from text file
' ---------------------------------------------------------
        DimobjImpSettings AsTextImportExportSettings
        SetobjImpSettings = objSpy.GetTextImportExportSettings

        objImpSettings.ImportFile = "C:\ImportMe.txt"
        objImpSettings.HeaderRow = False

        DimobjList AsElementList
        SetobjList = objSpy.GetTextImportElementList(objImpSettings)

        'exclude first column
        objList.RemoveItem 1

        DimobjImpDoc AsDocument
        OnErrorResume Next
        SetobjImpDoc = objSpy.ImportFromText(objImpSettings, objList)
        CheckForError
```

14.3.2.1.18 GetTextImportExportSettings

Methode: GetTextImportExportSettings() als TextImportExportSettings[548] (schreibgeschützt)

Beschreibung

GetTextImportExportSettingserstellt ein neues Objekt der allgemeinen Import- udn Exporteinstellungen für Textdateien. Siehe auch Beispiel zu Application.GetTextImportElementList[381].

Fehler

1111 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.1.19 GetXSLT_XQParameterCount

Methode: GetXSLT_XQParameterCount()als Long

Beschreibung
Gibt die Anzahl der XSLT- und XQuery-Parameter zurück.

Fehler
1111 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.20 GetXSLT_XQParameterName

Methode: GetXSLT_XQParameterName(index as Long)als String

Beschreibung
Gibt den Namen des durch den angegebenen Index identifizierten XSLT- oder XQuery-Parameters zurück.

Fehler
1111 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.21 GetXSLT_XQParameterXPath

Methode: GetXSLT_XQParameterXPath(index als Long)als String

Beschreibung
Gibt den XPath-Ausdruck des durch den angegebenen Index identifizierten XSLT- oder XQuery-Parameters zurück.

Fehler
1111 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.22 ImportFromDatabase

Methode: ImportFromDatabase(pImportSettingsals DatabaseConnection[452],pElementListals ElementList[501]) als Document[462]

Rückgabewert
Erstellt ein neues Dokument, das die aus der Datenbank importierten Daten enthält.

Beschreibung

ImportFromDatabaseimportiert Daten aus einer Datenbank, wie in pImportSettingsdefiniert und erstellt ein neues Dokument, das diese Daten enthält. Eine der folgenden Eigenschaften muss dazu ausgefüllt werden: DatabaseConnection.File[455], DatabaseConnection.ADOConnection[452]oder DatabaseConnection.ODBCConnection[456]und DatabaseConnection.SQLSelect[457]. Zusätzlich dazu können Sie mittels DatabaseConnection.AsAttributes[453], DatabaseConnection.ExcludeKeys[454], DatabaseConnection.IncludeEmptyElements[455]und NumberDateTimeFormat[456]weitere Parameter für den Import eingeben.

Der Parameter pElementListdefiniert, ob Felder der ausgewählten Daten in das neu erstellte Dokument geschrieben werden sollen und welche davon als Elemente und welche als Attribute erstellt werden sollen. Dieser Parameter kann NULL sein, d.h. alle ausgewählten Felder werden als XML-Elemente importiert.

Erforderliche Schritte, die vor dem Import von Daten aus einer Datenbank durchzuführen sind, werden unter GetDatabaseSettings[380]und GetDatabaseImportElementList[379]beschrieben.

Fehler

1111	Das application-Objekt ist nicht mehr gültig.
1100	Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
1107	Import aus Datenbank ist fehlgeschlagen.
1112	Es wurde eine ungültige Datenbank angegeben.
1114	Select-Anweisung fehlt.
1117	Transformation in XML ist fehlgeschlagen.
1120	Datenbankimport ist fehlgeschlagen.

Beispiel

```
DimobjImpSettings AsDatabaseConnection
SetobjImpSettings = objSpy.GetDatabaseSettings

objImpSettings.ADOConnection = strADOConnection
objImpSettings.SQLSelect = "SELECT * FROM MyTable"

DimobjDoc AsDocument
OnErrorResume Next
SetobjDoc = objSpy.ImportFromDatabase(objImpSettings,
objSpy.GetDatabaseImportElementList(objImpSettings))
' CheckForError here
```

14.3.2.1.23 ImportFromSchema

Methode: ImportFromSchema(*pImportSettings*als DatabaseConnection[452],*strTable*als String,*pSchemaDoc*als Document[462])als Document[462]

Rückgabewert

Erstellt ein neues Dokument, das die aus der Datenbank gemäß der Schemadefinition in *pSchemaDoc*importierten Daten enthält.

Beschreibung

 © 2016-2022 Altova GmbH

ImportFromSchema importiert Daten aus einer in pImportSettingsdefinierten Datenbank. Dazu muss eine der folgenden Eigenschaften ausgefüllt werden DatabaseConnection.File[455], DatabaseConnection.ADOConnection[452]oder DatabaseConnection.ODBCConnection[456]. Zusätzlich dazu können Sie mittels DatabaseConnection.AsAttributes[453], DatabaseConnection.ExcludeKeys[454]und NumberDateTimeFormat[456]weitere Parameter für den Import eingeben. Alle anderen Parameter werden ignoriert.

ImportFromSchema verwendet zum Auswählen der Daten keine explizite SQL-Anweisung. Statt dessen muss eine Strukturdefinition des Dokuments in Form eines XML-Schema-Dokuments in *pSchemaDoc*erstellt werden. Anhand dieser Definition wird automatisch die Datenbank-Select-Anweisung abgeleitet. Definieren Sie in *strTable*den Tabellennamen der Import-Root, die der Root-Node im neuen Dokument werden soll.

Erforderliche Schritte, die vor dem Import von Daten aus einer Datenbank anhand einer Schemadefinition durchzuführen sind, werden unter GetDatabaseSettings[380]und GetDatabaseTables[380]beschrieben. Verwenden Sie zum Erstellen einer Schemadefinitionsdatei den Befehl "Datenbankschema erstellen" aus dem Authentic Desktop-Menü "Konvertieren".

Fehler

1111	Das application-Objekt ist nicht mehr gültig.
1100	Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
1107	Import aus Datenbank ist fehlgeschlagen.
1112	Es wurde eine ungültige Datenbank angegeben.
1120	Datenbankimport ist fehlgeschlagen.
1121	Es konnte kein Validator für das angegebene Schema erstellt werden.
1122	Fehler beim Parsen des Schemas für den Datenbankimport.

14.3.2.1.24 ImportFromText

Methode: ImportFromText(*pImportSettings*als TextImportExportSettings[548],*pElementList*als ElementList[501]) als Document[462]

Beschreibung

ImportFromTextimportiert die Textdatei, wie in pImportSettingsdefiniert. Der Parameter pElementListkann als Importfilter verwendet werden. Übergeben Sie entweder die durch einen vorherigen Aufruf von GetTextImportElementList[381]zurückgegebene Liste oder den Wert nullum alle Spalten zu importieren. Um den Import unnötiger Spalten zu vermeiden, entfernen Sie die entsprechenden Feldnamen mit Hilfe von ElementList.RemoveElement[502]aus pElementListbevor Sie ImportFromTextaufrufen.
Die Methode gibt das neu erstellte Dokument mit den importierten Daten zurück. Dieses Dokument ist dasselbe wie das aktive Dokument von Authentic Desktop.

Fehler

1111	Das application-Objekt ist nicht mehr gültig.
1100	Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
1107	Import aus Textdatei ist fehlgeschlagen
1117	Transformation in XML ist fehlgeschlagen.

Beispiel

```
' ---------------------------------------------------------
' VBA client code fragment - import from text file
```

```
' ----------------------------------------------------------
        DimobjImpSettings AsTextImportExportSettings
        SetobjImpSettings = objSpy.GetTextImportExportSettings

        objImpSettings.ImportFile = strFileName
        objImpSettings.HeaderRow = False

        DimobjImpDoc AsDocument
        OnErrorResume Next
        SetobjImpDoc = objSpy.ImportFromText(objImpSettings,
objSpy.GetTextImportElementList(objImpSettings))

        CheckForError
```

14.3.2.1.25 ImportFromWord

Methode: ImportFromWord(*strFile*als String) als Document 462

Beschreibung
ImportFromWordimportiert das MS Word-Dokument strFileein ein neues XML-Dokument.

Fehler

1111	Das application-Objekt ist nicht mehr gültig.
1100	Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben. Import aus Dokument ist fehlgeschlagen.

14.3.2.1.26 IsAPISupported

Eigenschaft: IsAPISupportedals Boolean

Beschreibung
Gibt zurück, ob die API in dieser Version unterstützt wird oder nicht.

Fehler

1111	Das application-Objekt ist nicht mehr gültig.
1100	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.27 MajorVersion

Eigenschaft: MajorVersionals Integer

Beschreibung
Gibt die Hauptreleasenummer der Applikation zurück, z.B. `15`für 2013 Versionen und `16`für 2014 Versionen.

Fehler

1111	Das application-Objekt ist nicht mehr gültig.

© 2016-2022 Altova GmbH

1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.28 MinorVersion

Eigenschaft: MinorVersionas Integer

Beschreibung
Gibt die Nebenrelease-Versionsnummer der Applikation zurück.

Fehler
1111 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.29 NewProject

Methode: NewProject(*strPath*als String,*bDiscardCurrent*als Boolean)

Beschreibung
NewProjecterstellt ein neues Projekt.

Wenn es bereits ein offenes Projekt gibt, das geändert wurde undbDiscardCurrent"false" ist, schlägt NewProject()fehl.

Fehler
1111 Das application-Objekt ist nicht mehr gültig.
1102 Es ist bereits ein Projekt geöffnet, aber *bDiscardCurrent*ist *true*.
1103 Das neue Projekt konnte nicht erstellt werden.

14.3.2.1.30 OpenProject

Methode: OpenProject(*strPath*als String,*bDiscardCurrent*als Boolean,*bDialog*als Boolean)

Parameter
strPath
Der Pfad und Dateiname des zu öffnenden Projekts. Kann leer sein, wenn bDialog"true" ist.

bDiscardCurrent
Aktuell geöffnetes Projekt verwerfen. Änderungen daran gehen verloren.

bDialog
Dialogfelder für Benutzereingabe anzeigen.

Rückgabewert
Keiner

Beschreibung

OpenProjectöffnet ein vorhandenes Projekt. Wenn bereits ein Projekt geöffnet ist, das geändert wurde und bDiscardCurrent"false" ist, schlägt OpenProject()fehl.

Fehler

1111 Das application-Objekt ist nicht mehr gültig.
1100 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
1101 Das angegebene Projekt kann nicht geöffnet werden.
1102 Es ist bereits ein Projekt geöffnet, aber *bDiscardCurrent*ist *true*.

14.3.2.1.31 Parent

Eigenschaft: Parent als Application 372 (schreibgeschützt)

Beschreibung
Ruft das Authentic Desktopapplication-Objekt auf.

Fehler

1111 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.32 Quit

Method: Quit()

Rückgabewert
Keiner

Beschreibung
Diese Methode beendet Authentic Desktop. Alle geänderten Dokumente werden ohne Speicherung der Änderungen geschlossen. Dies gilbt auch für ein offenes Projekt.

Wenn Authentic Desktop automatisch von einem Client-Programm als Automation Server gestartet wird, wird die Applikation bei Schließen des Client-Programms nicht automatisch geschlossen, falls noch ein Projekt oder Dokument geöffnet ist. Verwenden Sie die Quit-Methode, um sicherzustellen, dass das Programm automatisch beendet wird.

Fehler

1111 Das application-Objekt ist nicht mehr gültig.

14.3.2.1.33 ReloadSettings

Methode: ReloadSettings

Rückgabewert

Beschreibung
Die Applikationseinstellungen werden aus der Registrierungsdatei neu geladen.

 © 2016-2022 Altova GmbH

Verfügbar mit Typbibliothek Version 1.5

Fehler
1111 Das application-Objekt ist nicht mehr gültig.

14.3.2.1.34 RemoveXSLT_XQParameter

Methode: RemoveXSLT_XQParameter(index als Long)

Beschreibung
Entfernt den durch den angegebenen Index identifizierten XSLT- oder XQuery-Parameter.

Fehler
1111 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.35 RunMacro

Methode: RunMacro(*strMacro*als String)

Rückgabewert

Beschreibung
Ruft das angegebene Makro entweder von den Projektskripts (falls vorhanden) oder den globalen Skripts aus auf.

Verfügbar mit Typbibliothek Version 1.5

Fehler
1111 Das application-Objekt ist nicht mehr gültig.

14.3.2.1.36 ScriptingEnvironment

Eigenschaft: ScriptingEnvironment als IUnknown (schreibgeschützt)

Beschreibung
Referenz auf jede aktive Skripting-Umgebung. Mit Hilfe dieser Eigenschaft kann die TypeLibrary der XMLSpyFormEditor.exe Applikation, die als aktuelle Skripting-Umgebung verwendet wird, aufgerufen werden.

Verfügbar mit Typbibliothek Version 1.5

Fehler
1111 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.37 ServicePackVersion

Eigenschaft: ServicePackVersionals Long

Beschreibung
Gibt die Service Pack-Versionsnummer der Applikation zurück, also z.B. 1 für 2010 R2 SP1

Fehler
1111 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.38 ShowApplication

Methode: ShowApplication(*bShow*als Boolean)

Rückgabewert
Keiner

Beschreibung
Die Methode blendet Authentic Desktop ein(bShow = True) oder aus (bShow = False).

Fehler
1110 Das application-Objekt ist nicht mehr gültig.

14.3.2.1.39 ShowFindInFiles

Methode: ShowFindInFiles(pSettingsals FindInFilesDlg[508]) als Boolean

Rückgabewert
Gibt "false" zurück, wenn der Benutzer die Schaltfläche "Abbrechen" gedrückt hat, andernfalls "true".

Beschreibung
Zeigt das Dialogfeld FindInFiles mit den vordefinierten Einstellungen an. Die benutzerdefinierten Änderungen an den Einstellungen werden im übergebenen Dialogfeldobjekt gespeichert.

Fehler
1111 Das application-Objekt ist nicht mehr gültig.
1100 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.

14.3.2.1.40 ShowForm

Methode: ShowForm(*strFormName*als String) als Long

Rückgabewert
Gibt Null zurück, wenn der Benutzer eine Abbrechen-Schaltfläche betätigt hat, oder die Formularaufrufe TheView.Cancel().

 © 2016-2022 Altova GmbH

Beschreibung
Zeigt das Formular strFormNamean.

Formulare, Events und Event Handler können mit der Skripting Umgebung erstellt werden. Wählen Sie im Menü **Extras**den Befehl "Skripting-Umgebung...", um die Skripting-Umgebung aufzurufen..

Fehler

1111	Das application-Objekt ist nicht mehr gültig.
1100	Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben..

14.3.2.1.41 Status

Eigenschaft: Statusals ENUMApplicationStatus 604

Beschreibung
Gibt den aktuellen Status der laufenden Applikation zurück.

Fehler

1111	Das application-Objekt ist nicht mehr gültig.
1100	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.42 URLDelete

Methode: URLDelete(*strURL*als String,*strUser*als String,*strPassword*als String)

Rückgabewert
Keiner

Beschreibung
Die Methode löscht die Datei an der URL strURL.

Fehler

1111	Das application-Objekt ist nicht mehr gültig.
1109	Fehler beim Löschen der Datei an der angegebenen URL.

14.3.2.1.43 URLMakeDirectory

Methode: URLMakeDirectory(*strURL*als String,*strUser*als String,*strPassword*als String)

Rückgabewert
Keiner

Beschreibung
Die Methode erstellt ein neues Verzeichnis an der URL strURL.

Fehler

1111 Das application-Objekt ist nicht mehr gültig.
1100 Es wurde ein ungültiger Parameter angegeben.

14.3.2.1.44 Visible

Eigenschaft: Visibleals VARIANT_BOOL

Beschreibung
Setzt oder ruft das Sichbarkeitsattribut von Authentic Desktopab. Diese Standard-Automation-Eigenschaft macht die Verwendung von ShowApplication 390 überflüssig.

Fehler
1110 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.45 WarningNumber

Eigenschaft: WarningNumberals integer

Beschreibung
Einige Methoden füllen die Eigenschaft WarningNumbermit zusätzlichen Informationen, wenn ein Fehler auftritt.

Derzeit wird diese Eigenschaft nur mit Documents.OpenFile 495 befüllt.

Fehler
1111 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.1.46 WarningText

Eigenschaft: WarningTextals String

Beschreibung
Einge Methoden befüllen die Eigenschaft WarningTextmit zusätzlichen Informationen, wenn ein Fehler auftritt.

Derzeit wird diese Eigenschaft nur mit Documents.OpenFile 495 befüllt.

Fehler
1111 Das application-Objekt ist nicht mehr gültig.
1100 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.2 AuthenticContextMenu

Über die Kontextmenü-Schnittstelle können die in Authentic angezeigten Kontextmenüs angepasst werden. Die Schnittstelle hat die in diesem Abschnitt aufgelisteten Methoden.

 © 2016-2022 Altova GmbH

14.3.2.2.1 CountItems

Methode: CountItems `()` nItems als long

Rückgabewert
Gibt die Anzahl der Menüoptionen zurück.

Fehler
2501 Ungültiges Objekt.

14.3.2.2.2 DeleteItem

Methode: DeleteItem `(IndexPosition als long)`

Rückgabewert
Löscht den Menüeintrag mit der im ersten Parameter übergebenen Indexposition.

Fehler
2501 Ungültiges Objekt
2502 Ungültiger Index

14.3.2.2.3 GetItemText

Methode: GetItemText `(IndexPosition als long)` Menübefehlname als String

Rückgabewert
Ruft den Namen des Menübefehls mit der im ersten Parameter übergebenen Indexposition ab.

Fehler
2501 Ungültiges Objekt
2502 Ungültiger Index

14.3.2.2.4 InsertItem

Methode: InsertItem `(IndexPosition als long, MenuItemName als string, MacroName als string)`

Rückgabewert
Fügt an der durch den ersten Parameter definierten Stelle im Menü einen benutzerdefinierten Menübefehl mit dem im zweiten Parameter definierten Namen ein. Der Menübefehl startet ein Makro, daher muss ein gültiger Makroname angegeben werden.

Errors

2501 Ungültiges Objekt
2502 Ungültiger Index
2503 Kein Makro dieses Namens vorhanden
2504 Interner Fehler

14.3.2.2.5 SetItemText

Methode: SetItemText `(IndexPosition als long, MenuItemName als string)`

Rückgabewert
Definiert den Namen des Menübefehls, der sich an der im ersten Parameter übergebenen Indexposition befindet.

Fehler

2501 Ungültiges Objekt
2502 Ungültiger Index

14.3.2.3 AuthenticDataTransfer

Umbenannt von DocEditDataTransfer in AuthenticDataTransfer

Das DocEditView Objekt wird in OldAuthenticView umbenannt.
DocEditSelection wird in AuthenticSelection umbenannt.
DocEditEvent wird in AuthenticEvent umbenannt.
DocEditDataTransfer wird in AuthenticDataTransfer umbenannt.

Es wird empfohlen, diese Objekte - mit Ausnahme von AuthenticDataTransfer - nicht mehr zu verwenden. Wir werden die bestehenden Funktionalitäten vorerst auf unbestimmte Zeit weiter unterstützen, doch werden keine neuen Funktionen zu diesen Schnittstellen hinzugefügt.

Beispiele zum Migrieren von DocEdit in Authentic finden Sie unter der Beschreibung der verschiedenen Methoden und Eigenschaften der einzelnen DocEdit-Objekte.

Methoden

getData 395

Eigenschaften

dropEffect 395
ownDrag 395
type 395

 © 2016-2022 Altova GmbH

Beschreibung

Die Events OnDragOverund OnBeforeDropliefern mit einer Instanz des Typs AuthenticDataTransferInformationen über das gezogene Objekt. Sie enthält eine Beschreibung des gezogenen Objekts und seines Inhalts. Letzterer steht entweder als String oder als Pointer auf ein COM-Objekt, das die IUnkown-Schnittstelle unterstützt, zur Verfügung.

14.3.2.3.1 dropEffect

Eigenschaft: dropEffectals long

Beschreibung
Die Eigenschaft speichert den Drop-Effekt aus dem Standard-Event Handler. Sie können den Drop-Effekt definieren, wenn Sie diesen Wert ändern und für den Event Handler TRUE zurückgeben.

Fehler
2101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.3.2 getData

Methode: getData()als Variant

Beschreibung
Ruft die Daten im Zusammenhang mit dem gezogenen Objekt ab. Abhängig von AuthenticDataTransfer.type 395 handelt es sich hierbei entweder um einen String oder um einen COM Interface Pointer vom Typ IUnknown.

Fehler
2101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.3.3 ownDrag

Eigenschaft: ownDragals Boolean (schreibgeschützt)

Beschreibung
Die Eigenschaft ist TRUE,wenn die aktuelle Quelle für das Ziehen des Objekts von innerhalb der Authentic View-Ansicht kommt.

Fehler
2101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.3.4 type

Eigenschaft: typeals String (schreibgeschützt)

Beschreibung
Enthält den Datentyp, den Sie mit der DocEditDataTransfer.getData 395 erhalten.

Derzeit werden die folgenden Datentypen unterstützt:

OWN	Daten aus der Authentic View-Ansicht selbst
TEXT	Unformatierter Text
UNICODETEXT	Unformatierter Text als UNICODE

Fehler

2101 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.4 AuthenticEventContext

Über die `EventContext` Schnittstellen haben Sie Zugriff auf viele Eigenschaften des Kontexts, in dem ein Makro ausgeführt wird.

14.3.2.4.1 EvaluateXPath

Methode: EvaluateXPath (strExpression als string) als strValue als string

Rückgabewert

Diese Methode wertet den XPath-Ausdruck im Kontext des Node aus, in dem das Event ausgelöst wurde, und gibt einen String zurück.

Beschreibung

`EvaluateXPath()` führt einen XPath-Ausdruck im angegebenen Event-Kontext aus. Das Ergebnis wird als String zurückgegeben. Im Fall einer Sequenz ist der String durch Leerzeichen getrennt.

Fehler

2201 Ungültiges Objekt.
2202 Kein Kontext.
2209 Ungültiger Parameter.
2210 Interner Fehler.
2211 XPath-Fehler.

14.3.2.4.2 GetEventContextType

Methode: GetEventContextType `()` Type als `AuthenticEventContextType` Enumeration

Rückgabewert

Gibt den Kontext-Node-Typ zurück.

Beschreibung

Mit Hilfe von `GetEventContextType` kann der Benutzer feststellen, ob sich das Makro in einem XML-Node oder in einem atomaren XPath Element-Kontext befindet. Die Enumeration `AuthenticEventContextType` ist folgendermaßen definiert:

 © 2016-2022 Altova GmbH

authenticEventContextXML,
authenticEventContextAtomicItem,
authenticEventContextOther

Wenn der Kontext ein normaler XML-Node ist, erhalten Sie mit der Funktion `GetXMLNode()` Zugriff darauf (falls nicht, wird `NULL` zurückgegeben).

Fehler

2201 Ungültiges Objekt.
2202 Kein Kontext.
2209 Ungültiger Parameter.

14.3.2.4.3 GetNormalizedTextValue

Methode: GetNormalizedTextValue`()` strValue als String

Rückgabewert
Gibt den Wert des aktuellen Node als String zurück.

Fehler

2201 Ungültiges Objekt.
2202 Kein Kontext.
2203 Ungültiger Kontext
2209 Ungültiger Parameter.

14.3.2.4.4 GetVariableValue

Methode: GetVariableValue`(strName als string) strValue als string`

Rückgabewert
Ruft den Wert der als Parameter übergebenen Variablen ab.

Beschreibung
`GetVariableValue` ruft den Wert der Variablen im Geltungsbereich des Kontexts ab.

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

Fehler

2201 Ungültiges Objekt.
2202 Kein Kontext.
2204 Keine Variable dieses Namens im Geltungsbereich
2205 Die Variable kann nicht ausgewertet werden
2206 Die Variable gibt eine Sequenz zurück
2209 Ungültiger Parameter

14.3.2.4.5 GetXMLNode

Methode: GetXMLNode() Node als XMLData Objekt

Rückgabewert
Gibt den XML-Kontext-Node oder `NULL` zurück

Fehler
2201 Ungültiges Objekt.
2202 Kein Kontext.
2203 Ungültiger Kontext
2209 Ungültiger Parameter.

14.3.2.4.6 IsAvailable

Methode: IsAvailable() als Boolean

Rückgabewert
Gibt "true" zurück, wenn `EventContext` definiert ist, andernfalls "false".

Fehler
2201 Ungültiges Objekt.

14.3.2.4.7 SetVariableValue

Methode: SetVariableValue`(strName als string, strValue als string)`

Rückgabewert
Definiert den Wert (den zweiten Parameter) der im ersten Parameter übergebenen Variablen.

Beschreibung
`SetVariableValue` definiert den Wert der Variablen im Geltungsbereich des Kontexts.

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

Fehler

 © 2016-2022 Altova GmbH

2201 Ungültiges Objekt.
2202 Kein Kontext.
2204 Keine Variable dieses Namens im Geltungsbereich
2205 Die Variable kann nicht ausgewertet werden
2206 Die Variable gibt eine Sequenz zurück
2207 Die Variable ist schreibgeschützt
2208 Keine Änderung zulässig

14.3.2.5 AuthenticRange

Die erste Tabelle enthält eine Liste der Eigenschaften und Methoden von AuthenticRange,die zum Navigieren durch das Dokument und zum Auswählen bestimmter Abschnitte verwendet werden.

Eigenschaften	Methoden	
Application 402	Clone 403	MoveBegin 421
FirstTextPosition 406	CollapsToBegin 403	MoveEnd 422
FirstXMLData 407	CollapsToEnd 403	NextCursorPosition 413
FirstXMLDataOffset 408	ExpandTo 406	PreviousCursorPosition 414
LastTextPosition 418	Goto 411	Select 424
LastXMLData 419	GotoNext 412	SelectNext 425
LastXMLDataOffset 420	GotoPrevious 413	SelectPrevious 425
Parent 423	IsEmpty 416	SetFromRange 427
	IsEqual 417	

Die folgende Tabelle enthält eine Liste der Methoden zur Bearbeitung von Inhalt. Die meisten dieser Methoden können über Kontextmenüs aufgerufen werden.

Eigenschaften	Bearbeitungsoperationen	Dynamische Tabellenoperationen
Text 428	Copy 404	AppendRow 401
	Cut 404	DeleteRow 404
	Delete 404	DuplicateRow 405
	IsCopyEnabled 416	InsertRow 415
	IsCutEnabled 416	IsFirstRow 417

	IsDeleteEnabled 416	IsInDynamicTable 417
	IsPasteEnabled 418	IsLastRow 417
	Paste 423	MoveRowDown 422
		MoveRowUp 422

Die folgenden Methoden liefern die Funktionen der Authentic-Eingabehilfefenster für Bereichsobjekte.

Operationen der Eingabehilfefenster		
Elemente	**Attribute**	**Entities**
CanPerformActionWith 402	GetElementAttributeValue 410	GetEntityNames 411
CanPerformAction 402	GetElementAttributeNames 409	InsertEntity 414
PerformAction 423	GetElementHierarchy 410	
	HasElementAttribute 414	
	IsTextStateApplied 418	
	SetElementAttributeValue 426	

Beschreibung
AuthenticRange Objekte sind die Cursor-Auswahl der Automation-Schnittstelle. Sie können damit auf jede Cursorposition in der Authentic-Ansicht zeigen oder einen Dokumentabschnitt auswählen. Die für AuthenticRange Objekte verfügbaren Operationen funktionieren anschließend an dieser Auswahl auf dieselbe Art wie die entsprechenden Operationen der Benutzeroberfläche an der aktuellen Auswahl des Benutzers. Der wichtigste Unterschied besteht darin, dass Sie beliebig viele AuthenticRange Objekte gleichzeitig verwenden können, während es auf der Benutzeroberfläche immer nur genau eine Cursorauswahl gibt.

Um zu Anfang ein Bereichsobjekt abzurufen, verwenden Sie AuthenticView.Selection 443, um einen Bereich zu erhalten, der der aktuellen Cursorauswahl auf der Benutzeroberfläche entspricht. Alternativ dazu stehen einige einfache Bereiche über die schreibgeschützten Eigenschaften AuthenticView.DocumentBegin 439, AuthenticView.DocumentEnd 439 und AuthenticView.WholeDocument 445 zur Verfügung. Die flexibelste Methode ist AuthenticView.Goto 441. Sie können damit in einem Aufruf zu einem bestimmten Bereich des Dokuments navigieren. Um eine komplexere Auswahl zu treffen, kombinieren Sie die obigen Methoden mit den verschiedenen Navigationsmethoden zu den in der ersten Tabelle auf dieser Seite aufgelisteten Bereichsobjekten.

Eine weitere Methode, einen Dokumentabschnitt auszuwählen, ist die Positionseigenschaften des Bereichsobjekts zu verwenden. Es stehen zwei Systeme zur Positionierung zur Verfügung, die beliebig miteinander kombiniert werden können:

- Für den Beginn und das Ende eines Bereichs können **absolute** Textcursorpositionen, beginnend mit der Position 0 am Dokumentanfang, gesetzt und abgerufen werden. Nähere Informationen dazu finden

 © 2016-2022 Altova GmbH

Sie unter FirstTextPosition[406] und LastTextPosition[418]. Diese Methode erfordert komplexe interne Berechnungen und sollte nur mit Vorsicht eingesetzt werden.

- Für den Beginn und das Ende eines Bereichs können das **XMLData-**Element und eine Textposition innerhalb dieses Elements gesetzt und abgerufen werden. Nähere Informationen dazu finden Sie unter FirstXMLData[407], FirstXMLDataOffset[408], LastXMLData[419] und LastXMLDataOffset[420]. Diese Methode ist äußerst effizient, erfordert aber Kenntnisse über die zugrunde liegende Dokumentstruktur. Sie kann zum Suchen von XMLData-Objekten und zum Durchführen von Operationen, die über die Benutzeroberfläche nicht aufgerufen werden können, verwendet werden.

Änderungen am Dokumentinhalt können durch verschiedene Methoden vorgenommen werden:

- Mit Hilfe der Text[428]-Eigenschaft können Sie den durch das Bereichsobjekt ausgewählten Dokumenttext abrufen. Wenn die Eigenschaft gesetzt ist, wird der ausgewählte Dokumenttext durch den neuen Text ersetzt.
- Die Standardbearbeitungsfunktionen Cut[404], Copy[404], Paste[423] und Delete[404].
- Tabellenoperationen für Tabellen, die dynamisch wachsen können.
- Methoden, die den Funktionalitäten der Authentic-Eingabehilfefenster zugeordnet sind.
- Aufrufe der XMLData[593] Objekte des zugrunde liegenden Dokuments, um diese direkt zu bearbeiten.

14.3.2.5.1 AppendRow

Methode: AppendRow()als Boolean

Beschreibung
Wenn sich der Anfang eines Bereichs innerhalb einer dynamischen Tabelle befindet, fügt diese Methode eine neue Zeile am Ende der ausgewählten Tabelle hinzu. Die Bereichsauswahl wird geändert, so dass sie auf den Beginn der neuen Zeile zeigt. Die Funktion gibt *true* zurück, wenn die Anhängeoperation erfolgreich war. Anderfalls wird *false* zurückgegeben.

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```
' -------------------------------------------------------
' Scripting environment - VBScript
' Append row at end of current dynamically growable table
' -------------------------------------------------------
DimobjRange
' we assume that the active document is open in authentic view mode
SetobjRange = Application.ActiveDocument.AuthenticView.Selection

' check if we can insert something
IfobjRange.IsInDynamicTable Then
        objRange.AppendRow
        ' objRange points to beginning of new row
        objRange.Select
EndIf
```

14.3.2.5.2 Application

Eigenschaft: Applicationals Application[372](schreibgeschützt)

Beschreibung
Ruft das Authentic Desktopapplication-Objekt auf.

Fehler

2001	The authentic range object or its related view object is no longer valid.Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig
2005	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.3 CanPerformAction

Methode: CanPerformAction(*eAction*als SPYAuthenticActions[605],*strElementName*als String) als Boolean

Beschreibung
CanPerformActionund die damit in Zusammenhang stehenden Methoden ermöglichen den Zugriff auf die Eingabehilfenfunktionen von Authentic. Diese Funktion gestattet die mühelose und einheitliche Bearbeitung des Dokumentinhalts, ohne dass Sie wissen müssen, wo genau die Änderung durchgeführt wird. Der Beginn des Bereichsobjekts dient dazu, die nächste gültige Stelle zu finden, an der die angegebene Aktion durchgeführt werden kann. Sobald diese Stelle gefunden wurde, gibt die Methode *True*zurück, andernfalls *False.*

Tipp: Um alle gültigen Elementnamen für eine bestimmte Aktion zu ermitteln, können Sie CanPerformActionWith[402]verwenden.

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
2007	Es wurde eine ungültige Aktion definiert.

Beispiele
Siehe PerformAction[423].

14.3.2.5.4 CanPerformActionWith

Methode: CanPerformActionWith(*eAction* als SPYAuthenticActions[605], *out_arrElementNames* als Variant)

Beschreibung
PerformActionWithund die damit in Zusammenhang stehenden Methoden ermöglichen den Zugriff auf die Eingabehilfenfunktionen von Authentic. Diese Funktion gestattet die mühelose und einheitliche Bearbeitung des Dokumentinhalts, ohne dass Sie wissen müssen, wo genau die Änderung durchgeführt wird.

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

 © 2016-2022 Altova GmbH

Tipp: Verwenden Sie zur Anwendung der Aktion CanPerformActionWith 402.

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
2007 Es wurde eine ungültige Aktion definiert.

Beispiele

Siehe PerformAction 423.

14.3.2.5.5 Clone

Methode: Clone() als AuthenticRange 399

Beschreibung

Gibt eine Kopie des Bereichsobjekts zurück.

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.6 CollapsToBegin

Methode: CollapsToBegin() als AuthenticRange 399

Beschreibung

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

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.7 CollapsToEnd

Methode: CollapsToEnd() as AuthenticRange 399

Beschreibung

Setzt den Beginn des Bereichsobjekts auf sein Ende. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.8 Copy

Methode: Copy() als Boolean

Beschreibung
Gibt *False*zurück, wenn der Bereich keine Abschnitte des Dokuments enthält, die kopiert werden können.
Gibt *True*zurück, wenn Text - und im Fall vollständig ausgewählter XML-Elemente auch die Elemente - in den Kopieren/Einfügen-Puffer kopiert wurde.

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.9 Cut

Methode: Cut() als Boolean

Beschreibung
Gibt *False*zurück, wenn der Bereich Dokumentabschnitte enthält, die nicht gelöscht werden dürfen.
Gibt *True*zurück, nachdem Text - und im Fall von vollständig ausgewählten XML-Elementen auch die Elemente aus dem Dokument gelöscht und im Kopieren/Einfügen-Puffer gespeichert wurden.-

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.10 Delete

Methode: Delete() als Boolean

Beschreibung
Gibt *False*zurück, wenn der Bereich Dokumentabschnitte enthält, die nicht gelöscht werden dürfen.
Gibt *True*zurück, nachdem Text und im Fall von vollständig ausgewählten XML-Elementen auch die Elemente aus dem Dokument gelöscht wurden.

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.11 DeleteRow

Methode: DeleteRow() als Boolean

 © 2016-2022 Altova GmbH

Beschreibung
Wenn sich der Anfang des Bereichs innerhalb einer dynamischen Tabelle befindet, löscht diese Methode die ausgewählte Zeile. Die Bereichsauswahl wird geändert, so dass sie auf das nächste Element nach der gelöschten Zeile zeigt. Die Funktion gibt *true* zurück, wenn die Operation erfolgreich war, andernfalls ist der Rückgabewert *false*.

Fehler
2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele
```
' ---------------------------------------------------------
' Scripting environment - VBScript
' Delete selected row from dynamically growing table
' ---------------------------------------------------------
DimobjRange
' we assume that the active document is open in authentic view mode
SetobjRange = Application.ActiveDocument.AuthenticView.Selection

' check if we are in a table
IfobjRange.IsInDynamicTable Then
        objRange.DeleteRow
EndIf
```

14.3.2.5.12 DuplicateRow

Methode: DuplicateRow() als Boolean

Beschreibung
Wenn sich der Anfang des Bereichs innerhalb einer dynamischen Tabelle befindet, fügt diese Methode eine Kopie der aktuellen Zeile nach der ausgewählten ein. Die Bereichsauswahl wird geändert, so dass sie auf den Anfang der nächsten Zeile zeigt. Die Funktion gibt *true* zurück, wenn die Operation erfolgreich war, andernfalls ist der Rückgabewert *false*.

Fehler
2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele
```
' ---------------------------------------------------------
' Scripting environment - VBScript
' duplicate row in current dynamically growable table
' ---------------------------------------------------------
DimobjRange
' we assume that the active document is open in authentic view mode
SetobjRange = Application.ActiveDocument.AuthenticView.Selection

' check if we can insert something
IfobjRange.IsInDynamicTable Then
```

```
        objRange.DuplicateRow
        ' objRange points to beginning of new row
        objRange.Select
EndIf
```

14.3.2.5.13 EvaluateXPath

Methode: EvaluateXPath (strExpression als string) strValue als string

Rückgabewert
Die Methode gibt einen String zurück

Beschreibung
`EvaluateXPath()` führt einen XPath-Ausdruck aus, wobei der Kontext-Node der Anfang der Bereichsauswahl ist. Das als String zurückgegebene Ergebnis ist im Fall einer Sequenz ein durch Leerzeichen getrennter String. Wenn der XML-Node nicht relevant ist, kann der Benutzer jeden Node angeben, wie z.B. `AuthenticView.XMLDataRoot`.

Fehler

2001	Ungültiges Objekt
2005	Ungültiger Parameter
2008	Interner Fehler
2202	Fehlender Kontext-Node
2211	XPath-Fehler

14.3.2.5.14 ExpandTo

Methode: ExpandTo (*eKind* als SPYAuthenticElementKind 606), als AuthenticRange 399

Beschreibung
Wählt das gesamte Element vom Typ eKindaus, das an der ersten Cursorposition des Bereichs beginnt oder diese enthält. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2003	Die Erweiterung des Bereichs würde über das Ende des Dokuments hinausgehen.
2005	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.15 FirstTextPosition

Eigenschaft: FirstTextPosition als Long

Beschreibung

© 2016-2022 Altova GmbH

Setzt oder ruft den am weitesten links gelegenen Positionsindex des Bereichsobjekts ab. Dieser Index ist immer kleiner oder gleich der LastTextPosition[418]. Die Indizierung beginnt am Dokumentanfang mit 0 und steigt dann mit jeder neuen Position, die der Textcursor einnehmen kann, schrittweise an. Die schrittweise Erhöhung der Textposition um 1 hat denselben Effekt, wie die nach rechts-Taste. Die schrittweise Verringerung der Textposition um 1 hat denselben Effekt, wie die nach lins-Taste.

Wenn Sie FirstTextPosition auf einen höheren Wert als den der aktuellen LastTextPosition[418] setzen, wird LastTextPosition[418] auf den neuen Wert von FirstTextPosition gesetzt.

Tipp: Seien Sie beim Setzen der Textcursorposition vorsichtig, da dies im Vergleich zur Cursorpositionierung mittes XMLData eine kostspielige Operation ist.

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
2006 Es wurde eine Textposition außerhalb des Dokuments definiert.

Beispiele

```
' ----------------------------------------
' Scripting environment - VBScript
' ----------------------------------------
DimobjAuthenticView
' we assume that the active document is open in authentic view mode
SetobjAuthenticView = Application.ActiveDocument.AuthenticView

nDocStartPosition = objAuthenticView.DocumentBegin.FirstTextPosition
nDocEndPosition = objAuthenticView.DocumentEnd.FirstTextPosition

' let's create a range that selects the whole document
' in an inefficient way
DimobjRange
' we need to get a (any) range object first
SetobjRange = objAuthenticView.DocumentBegin
objRange.FirstTextPosition = nDocStartPosition
objRange.LastTextPosition = nDocEndPosition

' let's check if we got it right
IfobjRange.isEqual(objAuthenticView.WholeDocument) Then
        MsgBox "Test using direct text cursor positioning was ok"
Else
        MsgBox "Ooops!"
EndIf
```

14.3.2.5.16 FirstXMLData

Eigenschaft: FirstXMLData als XMLData[593]

Beschreibung

Setzt oder ruft das erste XMLData Element im zugrunde liegenden Dokument ab, das durch die Bereichsauswahl teilweise oder zur Gänze ausgewählt ist. Der genaue Anfang der Auswahl wird durch das FirstXMLDataOffset[408] Attribut definiert.

Immer wenn Sie FirstXMLData auf ein neues Datenobjekt setzen, wird FirstXMLDataOffset[408] auf die erste Cursorposition innerhalb dieses Elements gesetzt. Nur XMLData-Objekte, die eine Cursorposition haben, können verwendet werden. Wenn Sie FirstXMLData / FirstXMLDataOffset[408] auf eine Position setzen, die größer ist als der aktuelle Wert von LastXMLData[419] / LastXMLDataOffset[420] wird zweiteres wird auf die neue Anfangsposition verschoben.

Tipp: Sie können das zugrunde liegende XML-Dokument in Fällen, in denen die mit dem AuthenticRange[399]-Objekt verfügbaren Methoden nicht ausreichen, mit Hilfe der Eigenschaften FirstXMLData[407] und LastXMLData[419] direkt aufrufen und bearbeiten

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht gültig.
2005	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
2008	Interner Fehler
2009	Das XMLData-Objekt kann nicht aufgerufen werden.

Beispiele

```
' ----------------------------------------------
' Scripting environment - VBScript
' show name of currently selected XMLData element
' ----------------------------------------------
DimobjAuthenticView
' we assume that the active document is open in authentic view mode
SetobjAuthenticView = Application.ActiveDocument.AuthenticView

DimobjXmlData
SetobjXMLData = objAuthenticView.Selection.FirstXMLData
' authentic view adds a 'text' child element to elements
' of the document which have content. So we have to go one
' element up.
SetobjXMLData = objXMLData.Parent
MsgBox "Current selection selects element " & objXMLData.Name
```

14.3.2.5.17 FirstXMLDataOffset

Eigenschaft: FirstXMLDataOffset as Long

Beschreibung

Setzt oder ruft den Versatz der Cursorposition innerhalb des Elements FirstXMLData[407] für den Anfang des Bereichs ab. Die Versatzpositionen basieren auf den von der Eigenschaft Text[428] zurückgegebenen Zeichen und beginnen mit 0. Verwenden Sie beim Definieren eines neuen Versatzes -1, um den Versatz an die letzte mögliche Position im Element zu setzen. Beachten Sie bitte besondes die folgenden Fälle:

- Die Textform von Einträgen in Auswahllisten, Kontrollkästchen und ähnlichen Steuerelementen kann anders aussehen, als angezeigt. Zwar basiert der Datenversatz auf diesem Text, doch gibt es nur zwei gültige Positionen für den Versatz, eine am Anfang und eine am Ende des Eintrags. Wenn Sie versuchen, den

 © 2016-2022 Altova GmbH

Versatz irgendwo mitten im Eintrag zu positionieren, hat dies zur Folge, dass der Versatz an das Ende des Eintrags gesetzt wird.

- Die Textform von XML-Entities kann sich in ihrer Länge von der Darstellung auf dem Bildschirm unterscheiden. Der Versatz basiert auf der Textform.

Wenn FirstXMLData / FirstXMLDataOffset[408] eine Position nach dem aktuellen LastXMLData[419] / LastXMLDataOffset[420] wählt, wird zweiteres an die neue Anfangsposition verschoben.

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht gültig.
2005 Es wurde ein ungültiger Versatz definiert.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```
' ---------------------------------------------
' Scripting environment - VBScript
' Select the complete text of an XMLData element
' using XMLData based selection and ExpandTo
' ---------------------------------------------
DimobjAuthenticView
' we assume that the active document is open in authentic view mode
SetobjAuthenticView = Application.ActiveDocument.AuthenticView

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

' the same can be achieved with the ExpandTo method
DimobjRange2
SetobjRange2 = objAuthenticView.Selection.ExpandTo(spyAuthenticTag)

' were we successful?
IfobjRange.IsEqual(objRange2) Then
        objRange.Select()
Else
        MsgBox "Oops"
EndIf
```

14.3.2.5.18 GetElementAttributeNames

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

Beschreibung

Ruft die Namen aller Attribute für das einschließende Element mit dem angegebenen Namen auf. Verwenden Sie die Element/Attribut-Paare, um den Attributwert mit den Methoden GetElementAttributeValue 410 und SetElementAttributeValue 426 zu setzen oder abzurufen.

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Invalid element name was specified.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

Siehe SetElementAttributeValue 426.

14.3.2.5.19 GetElementAttributeValue

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

Beschreibung

Ruft für das in strElementName angegebene Element den Wert des in strAttributeName definierten Attributs ab. Wenn das Attribut unterstütz wird, aber keinen Wert zugewiesen hat, wird der leere String zurückgegeben. Um die Namen der von einem Element unterstützten Attribute herauszufinden, verwenden Sie GetElementAttributeNames 409, oder HasElementAttribute 414.

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Es wurde ein ungültiger Elementname angegeben.
Es wurde ein ungültiger Attributname angegeben.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

Siehe SetElementAttributeValue 426.

14.3.2.5.20 GetElementHierarchy

Methode: GetElementHierarchy (*out_arrElementNames* als Variant)

Beschreibung

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

Die Namen der Elementhierarchie identifizieren zusammen mit dem Bereichsobjekt XMLData Elemente im Dokument. Die Attribute dieser Elemente können direkt von GetElementAttributeNames 409 und damit in Zusammenhang stehenden Methoden aufgerufen werden.

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

 © 2016-2022 Altova GmbH

14.3.2.5.21 GetEntityNames

Methode: GetEntityNames (*out_arrEntityNames* als Variant)

Beschreibung
Ruft die Namen alle definierten Entities ab. Die Liste der abgerufenen Entities hängt nicht von der aktuellen Auswahl oder der aktuellen Position ab. Verwenden Sie einen dieser Namen mit der Funktion InsertEntity 414.

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele
Siehe GetElementHierarchy 410 und InsertEntity 414.

14.3.2.5.22 GetVariableValue

Methode: GetVariableValue `(strName als string) strVal als string`

Rückgabewert
Ruft den Wert der als Parameter der Methode benannten Variablen ab.

Fehler

2001	Ungültiges Objekt.
2202	Kein Kontext.
2204	Es befindet sich keine Variable dieses Namens im Geltungsbereich
2205	Die Variable kann nicht ausgewertet werden
2206	Die Variable gibt eine Sequenz zurück
2209	Ungültiger Parameter

14.3.2.5.23 Goto

Methode: Goto (*eKind* als SPYAuthenticElementKind 606, *nCount* als Long, *eFrom* als SPYAuthenticDocumentPosition 605) als AuthenticRange 399

Beschreibung
Definiert den Bereich, der auf den Anfang des nCount Elements vom Typ eKind zeigt. Die Anfangsposition wird durch den Parameter eFrom definiert.

Mit Hilfe positiver Werte für nCount können Sie zum Ende des Dokuments navigieren. Mit Hilfe negativer Werte können Sie zum Beginn des Dokuments navigieren. die Methode gibt das geänderte Bereichsobjekt zurück.
Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2003 Das Ziel liegt hinter dem Ende des Dokuments.
2004 Das Ziel liegt vor dem Anfang des Dokuments.
2005 Es wurde eine ungültige Elementart definiert.
Es wurde eine ungültige Anfangsposition definiert.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.24 GotoNext

Methode: GotoNext (*eKind* als SPYAuthenticElementKind[606]) als AuthenticRange[399]

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

Fehler
2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2003 Das Ziel liegt hinter dem Ende des Dokuments.
2005 Es wurde eine ungültige Elementart definiert.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```
' ---------------------------------------------
' Scripting environment - VBScript
' Scan through the whole document word-by-word
' ---------------------------------------------
DimobjAuthenticView
' we assume that the active document is open in authentic view mode
SetobjAuthenticView = Application.ActiveDocument.AuthenticView

DimobjRange
SetobjRange = objAuthenticView.DocumentBegin
DimbEndOfDocument
bEndOfDocument = False

OnErrorResume Next
WhileNotbEndOfDocument
	objRange.GotoNext(spyAuthenticWord).Select
	If((Err.number - vbObjecterror) = 2003) Then
		bEndOfDocument = True
		Err.Clear
	ElseIf(Err.number <> 0) Then
		Err.Raise  ' forward error
	EndIf
Wend
```

 © 2016-2022 Altova GmbH

14.3.2.5.25 GotoNextCursorPosition

Methode: GotoNextCursorPosition() als AuthenticRange 399

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

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2003	Das Ziel liegt hinter dem Ende des Dokuments.
2005	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.26 GotoPrevious

Methode: GotoPrevious (*eKind* als SPYAuthenticElementKind 606) als AuthenticRange 399

Beschreibung
Setzt den Bereich auf den Beginn des Elements vom Typ eKind, welches sich vor dem Beginn des aktuellen Bereichs berfindet. Die Methode geibt das geänderte Bereichsobjekt zurück.

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2004	Das Ziel liegt vor dem Anfang des Dokuments.
2005	Es wurde eine ungültige Elementart definiert. Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```
' ---------------------------------------------
' Scripting environment - VBScript
' Scan through the whole document tag-by-tag
' ---------------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentEnd
Dim bBeginOfDocument
bBeginOfDocument = False

On Error Resume Next
While Not bBeginOfDocument
	objRange.GotoPrevious(spyAuthenticTag).Select
	If ((Err.number - vbObjecterror) = 2004) Then
		bBeginOfDocument = True
		Err.Clear
```

```
        ElseIf (Err.number <> 0) Then
                Err.Raise  ' forward error
        End If
Wend
```

14.3.2.5.27 GotoPreviousCursorPosition

Methode: GotoPreviousCursorPosition() als AuthenticRange 399

Beschreibung
Setzt den Bereich auf die Cursorposition unmittelbar vor der aktuellen Position. Gibt das geänderte Objekt zurück.

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2004	Das Ziel liegt vor dem Anfang des Dokuments.
2005	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.28 HasElementAttribute

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

Beschreibung
Überprüft, ob das einschließende Element mit dem Namen strElementName das in strAttributeName definierte Attribut unterstützt.

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005	Es wurde ein ungültiger Elementname angegeben. Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.29 InsertEntity

Methode: InsertEntity (*strEntityName* als String)

Beschreibung
Ersetzt die Bereichsauswahl durch die angegebene Entity. Die angegebene Entity muss einer der von GetEntityNames 411 zurückgegebenen Entity-Namen sein.

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005	Es wurde ein unbekannter Name für den Eintrag angegeben.

Beispiele

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


© 2016-2022 Altova GmbH

```
' Scripting environment - VBScript
' Insert the first entity in the list of available entities
' ----------------------------------------------------------
DimobjRange
' we assume that the active document is open in authentic view mode
SetobjRange = Application.ActiveDocument.AuthenticView.Selection

' first we get the names of all available entities as they
' are shown in the entry helper of XMLSpy
DimarrEntities
objRange.GetEntityNames arrEntities

' we insert the first one of the list
IfUBound(arrEntities) >= 0 Then
        objRange.InsertEntity arrEntities(0)
Else
        MsgBox "Sorry, no entities are available for this document"
EndIf
```

14.3.2.5.30 InsertRow

Methode: InsertRow() als Boolean

Beschreibung
Wenn sich der Beginn des Bereichs innerhalb einer dynamischen Tabelle befindet, fügt diese Methode eine neue Zeile vor der aktuellen ein. Der Auswahlbereich wird geändert, sodass er auf den Beginn der neu eingefügten Zeile zeigt. Die Funktion gibt *true* zurück, wenn die Einfügeoperation erfolgreich war, anderfalls ist das Ergebnis *false*.

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```
' ----------------------------------------------------------
' Scripting environment - VBScript
' Insert row at beginning of current dynamically growing table
' ----------------------------------------------------------
DimobjRange
' we assume that the active document is open in authentic view mode
SetobjRange = Application.ActiveDocument.AuthenticView.Selection

' check if we can insert something
IfobjRange.IsInDynamicTable Then
        objRange.InsertRow
        ' objRange points to beginning of new row
        objRange.Select
 EndIf
```

14.3.2.5.31 IsCopyEnabled

Eigenschaft: IsCopyEnabled als Boolean (schreibgeschützt)

Beschreibung
Überprüft, ob die Kopieroperation für diesen Bereich unterstützt wird.

Fehler
2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.32 IsCutEnabled

Eigenschaft: IsCutEnabled als Boolean (schreibgeschützt)

Beschreibung
Überprüft, ob die Ausschneideoperation für diesen Bereich unterstützt wird.

Fehler
2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.33 IsDeleteEnabled

Eigenschaft: IsDeleteEnabled als Boolean (schreibgeschützt)

Beschreibung
Überprüft, ob die Löschoperation für diesen Bereich unterstützt wird.

Fehler
2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.34 IsEmpty

Methode: IsEmpty() als Boolean

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

Fehler
2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.5.35 IsEqual

Methode: IsEqual (*objCmpRange* als AuthenticRange 399) als Boolean

Beschreibung
Überprüft, ob der Beginn und das Ende beider Bereiche identisch sind.

Fehler
2001 Eines der beiden verglichenen Bereichsobjekte ist ungültig.
2005 Es wurde eine ungültige Adresse für einen Rückgabeparameter angegeben.

14.3.2.5.36 IsFirstRow

Eigenschaft: IsFirstRow als Boolean (schreibgeschützt)

Beschreibung
Überprüft, ob der Bereich in der ersten Zeile einer Tabelle ist. Welche Tabelle berücksichtigt wird, ist vom Ausmaß des Bereichs abhängig. Wenn die Auswahl über eine einzige Tabellenzeile hinausgeht, wird überprüft, ob diese Tabelle das erste Element in einer einbettenden Tabelle ist. Nähere Informationen dazu finden Sie im Benutzerhandbuch unter Eingabehilfen.

Fehler
2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.37 IsInDynamicTable

Methode: IsInDynamicTable() als Boolean

Beschreibung
Überprüft, ob sich der gesamte Bereich innerhalb einer Tabelle befindet, die die verschiedenen Zeilenoperationen wie Einfügen, Anhängen, Kopieren usw. unterstützt.

Fehler
2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.38 IsLastRow

Eigenschaft: IsLastRow als Boolean (schreibgeschützt)

Beschreibung
Überprüft, ob der Bereich in der letzten Zeile einer Tabelle ist. Welche Tabelle berücksichtigt wird, hängt vom Ausmaß des Bereichs ab. Wenn die Auswahl über eine einzige Tabellenzeile hinausgeht, wird überprüft, ob

diese Tabelle das letzte Element in einer einbettenden Tabelle ist. Nähere Informationen dazu finden Sie im Benutzerhandbuch unter Eingabehilfen.

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.39 IsPasteEnabled

Eigenschaft: IsPasteEnabled als Boolean (schreibgeschützt)

Beschreibung
Überprüft, ob die Einfügeoperatione für diesen Bereich unterstützt wird.

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.40 IsSelected

Eigenschaft: IsSelected als Boolean

Beschreibung
Gibt true() zurück, wenn die Auswahl vorhanden ist. Der Auswahlbereich kann dennoch leer sein. Dies passiert z.B. wenn der Cursor gesetzt ist.

14.3.2.5.41 IsTextStateApplied

Methode: IsTextStateApplied (*i_strElementName* als String) als Boolean

Beschreibung
Überprüft, ob der gesamte ausgewählte Text in ein XML-Element mit dem Namen i_strElementName eingebettet ist. Ein typisches Beispiel für den Parameter i_strElementName ist "strong", "bold" oder "italic".

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.42 LastTextPosition

Eigenschaft: LastTextPosition als Long

 © 2016-2022 Altova GmbH

Beschreibung
Setzt oder ruft den am weitesten rechts gelegenen Positionsindex des Bereichsobjekts ab. Dieser Index ist immer größer oder gleich FirstTextPosition[406]. Die Indizierung beginnt am Dokumentanfang mit 0 und steigt dann mit jeder neuen Position, die der Textcursor einnehmen kann, schrittweise an. Die schrittweise Erhöhung der Textposition um 1 hat denselben Effekt, wie die nach rechts-Taste. Die schrittweise Verringerung der Textposition um 1 hat denselben Effekt, wie die nach lins-Taste.

Wenn Sie LastTextPosition auf einen niedrigeren Wert als den der aktuellen FirstTextPosition[406] setzen, wird FirstTextPosition[406] auf den neuen Wert von LastTextPosition gesetzt.

Tipp: Seien Sie beim Setzen der Textcursorposition vorsichtig, da dies im Vergleich zur Cursorpositionierung mittes XMLData eine kostspielige Operation ist.

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht gültig.
2005	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
2006	Es wurde eine Textposition außerhalb des Dokuments definiert.

Beispiele

```
' ----------------------------------------
' Scripting environment - VBScript
' ----------------------------------------
DimobjAuthenticView
' we assume that the active document is open in authentic view mode
SetobjAuthenticView = Application.ActiveDocument.AuthenticView

nDocStartPosition = objAuthenticView.DocumentBegin.FirstTextPosition
nDocEndPosition = objAuthenticView.DocumentEnd.FirstTextPosition

' let's create a range that selects the whole document
' in an inefficient way
DimobjRange
' we need to get a (any) range object first
SetobjRange = objAuthenticView.DocumentBegin
objRange.FirstTextPosition = nDocStartPosition
objRange.LastTextPosition = nDocEndPosition

' let's check if we got it right
IfobjRange.isEqual(objAuthenticView.WholeDocument) Then
        MsgBox "Test using direct text cursor positioning was ok"
Else
        MsgBox "Oops!"
EndIf
```

14.3.2.5.43 LastXMLData

Eigenschaft: LastXMLData als XMLData[593]

Beschreibung

Setzt oder ruft das letzte XMLData Element im zugrunde liegenden Dokument ab, das durch die Bereichsauswahl teilweise oder zur Gänze ausgewählt ist. Das genaue Ende der Auswahl wird durch das LastXMLDataOffset[420] Attribut definiert.

Immer wenn Sie LastXMLData auf ein neues Datenobjekt setzen, wird LastXMLDataOffset[420] auf die letzte Cursorposition innerhalb dieses Elements gesetzt. Nur XMLData-Objekte, die eine Cursorposition haben, können verwendet werden. Wenn Sie LastXMLData / LastXMLDataOffset[420] auf eine Position setzen, die niedriger ist als der aktuelle Wert von FirstXMLData[407] / FirstXMLDataOffset[408] wird zweiteres wird auf die neue Endposition verschoben.

Tipp: Sie können das zugrunde liegende XML-Dokument in Fällen, in denen die mit dem AuthenticRange[399]-Objekt verfügbaren Methoden nicht ausreichen, mit Hilfe der Eigenschaften FirstXMLData[407] und LastXMLData[419] direkt aufrufen und bearbeiten

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
2008 Interner Fehler
2009 Das XMLDataObjekt kann nicht aufgerufen werden.

14.3.2.5.44 LastXMLDataOffset

Eigenschaft: LastXMLDataOffset als Long

Beschreibung

Setzt oder ruft den Versatz der Cursorposition innerhalb des Elements LastXMLData[419] für das Ende des Bereichs ab.

Die Versatzpositionen basieren auf den von der Eigenschaft Text[428] zurückgegebenen Zeichen und beginnen mit 0. Verwenden Sie beim Definieren eines neuen Versatzes -1, um den Versatz an die letzte mögliche Position im Element zu setzen. Beachten Sie bitte besondes die folgenden Fälle:

- Die Textform von Einträgen in Auswahllisten, Kontrollkästchen und ähnlichen Steuerelementen kann anders aussehen, als angezeigt. Zwar basiert der Datenversatz auf diesem Text, doch gibt es nur zwei gültige Positionen für den Versatz, eine am Anfang und eine am Ende des Eintrags. Wenn Sie versuchen, den Versatz irgendwo mitten im Eintrag zu positionieren, hat dies zur Folge, dass der Versatz an das Ende des Eintrags gesetzt wird.

- Die Textform von XML-Entities kann sich in ihrer Länge von der Darstellung auf dem Bildschirm unterscheiden. Der Versatz basiert auf der Textform.

Wenn LastXMLData[419] / LastXMLDataOffset[420] eine Position vor dem aktuellen FirstXMLData[407] / FirstXMLDataOffset[408] wählt, wird zweiteres an die neue Endposition verschoben.

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht gültig.
2005 Es wurde ein ungültiger Versatz definiert.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

 © 2016-2022 Altova GmbH

Beispiele

```
' ---------------------------------------------
' Scripting environment - VBScript
' Select the complete text of an XMLData element
' using XMLData based selection and ExpandTo
' ---------------------------------------------
DimobjAuthenticView
' we assume that the active document is open in authentic view mode
SetobjAuthenticView = Application.ActiveDocument.AuthenticView

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

' the same can be achieved with the ExpandTo method
DimobjRange2
SetobjRange2 = objAuthenticView.Selection.ExpandTo(spyAuthenticTag)

' were we successful?
IfobjRange.IsEqual(objRange2) Then
        objRange.Select()
Else
        MsgBox "Ooops"
EndIf
```

14.3.2.5.45 MoveBegin

Methode: MoveBegin (*eKind* als SPYAuthenticElementKind[606], *nCount* als Long) als AuthenticRange[399]

Beschreibung

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

Verwenden Sie positive Zahlen für nCount, um die Position zum Ende des Dokuments hin zu verschieben und negative Zahlen um die Position zum Anfang des Dokuments hin zu verschieben. Das Ende des Bereichs bleibt gleich, es sei denn, der neue Anfang hat einen höheren Wert als das Ende. In diesem Fall wird das Ende an den neuen Anfang verschoben. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2003	Das Ziel liegt hinter dem Ende des Dokuments.
2004	Das Ziel liegt vor dem Anfang des Dokuments.
2005	Es wurde eine ungültige Elementart definiert. Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.46 MoveEnd

Methode: MoveEnd (*eKind* als SPYAuthenticElementKind 606, *nCount* als Long) als AuthenticRange 399

Beschreibung
Verschiebt das Ende des Bereichs an den Beginn des nCount Elements vom Typ eKind. Die Zählung begint am aktuellen Ende des Bereichsobjekts.

Verwenden Sie positive Zahlen für nCount, um die Position zum Ende des Dokuments hin zu verschieben und negative Zahlen um die Position zum Anfang des Dokuments hin zu verschieben. Der Anfang des Bereichs bleibt gleich, es sei denn, das neue Ende hat einen niedrigeren Wert als der Anfang. In diesem Fall wird der Anfang an das neue Ende verschoben. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2003	Das Ziel liegt hinter dem Ende des Dokuments.
2004	Das Ziel liegt vor dem Anfang des Dokuments.
2005	Es wurde eine ungültige Elementart definiert. Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.47 MoveRowDown

Methode: MoveRowDown() als Boolean

Beschreibung
Wenn sich der Anfang des Bereichs innerhalb einer dynamischen Tabelle befindet und eine Zeile auswählt, die nicht die letzte Zeile in dieser Tabelle ist, tauscht diese Methode diese Zeile gegen die unmittelbar darunter gelegene Zeile aus. Die Auswahl des Bereichs verschiebt sich mit der Zeile, ändert sich aber ansonsten nicht. Die Funktion gibt *true* zurück, wenn der Verschiebevorgang erfolgreich war, andernfalls wird *false* zurückgegeben.

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.48 MoveRowUp

Method: MoveRowUp() as Boolean

Beschreibung
Wenn sich der Anfang des Bereichs innerhalb einer dynamischen Tabelle befindet und eine Zeile auswählt, die nicht die erste Zeile in dieser Tabelle ist, tauscht diese Methode diese Zeile gegen die unmittelbar darüber gelegene Zeile aus. Die Auswahl des Bereichs verschiebt sich mit der Zeile, ändert sich aber ansonsten nicht. Die Funktion gibt *true* zurück, wenn der Verschiebevorgang erfolgreich war, andernfalls wird *false* zurückgegeben

 © 2016-2022 Altova GmbH

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.49 Parent

Eigenschaft: Parent als AuthenticView 428 (schreibgeschützt)

Beschreibung
Ruft die Ansicht auf, die dieses Bereichsobjekt enthält.

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.50 Paste

Methode: Paste() als Boolean

Beschreibung
Gibt *False*zurück, wenn der Kopieren/Einfügen-Puffer leer ist oder sein Inhalt die aktuelle Auswahl nicht ersetzen kann.

Löscht andernfalls die aktuelle Auswahl, fügt den Inhalt des Kopieren/Einfügen-Puffers ein und gibt *True*zurück.

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.5.51 PerformAction

Methode: PerformAction (*eAction* als SPYAuthenticActions 605, *strElementName* als String) als Boolean

Beschreibung
PerformActionund die dazugehörigen Methoden verschaffen Ihnen Zugriff auf die Eingabehilfenfunktionen von Authentic. Diese Funktion gestattet die einfach und einheitliche Bearbeitung des Dokumentinhalts, ohne dass Sie wissen müssen, wo genau die Änderung stattfinden soll. Der Anfang des Bereichsobjekts dient dazu, die nächste gültige Stelle zu finden, an der die angegebene Aktion durchgeführt werden kann. Wenn keine derartige Stelle gefunden wird, gibt die Methode *False*. zurück. Andernfalls wird das Dokument geändert und der Bereich zeigt auf den Anfang der Änderung.

Tipp: Um Elementnamen zu finden, die als der zweite Parameter übergeben werden können, verwenden Sie CanPerformActionWith 402.

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
2007 Es wurde eine ungültige Aktion definiert.

Beispiele

```
' ---------------------------------------------
' Scripting environment - VBScript
' Insert the innermost element
' ---------------------------------------------
DimobjRange
' we assume that the active document is open in authentic view mode
SetobjRange = Application.ActiveDocument.AuthenticView.Selection

' we determine the elements that can be inserted at the current position
DimarrElements()
objRange.CanPerformActionWith spyAuthenticInsertBefore, arrElements

' we insert the first (innermost) element
IfUBound(arrElements) >= 0 Then
        objRange.PerformAction spyAuthenticInsertBefore, arrElements(0)
        ' objRange now points to the beginning of the inserted element
        ' we set a default value and position at its end
        objRange.Text = "Hello"
        objRange.ExpandTo(spyAuthenticTag).CollapsToEnd().Select
Else
        MsgBox "Can't insert any elements at current position"
EndIf
```

14.3.2.5.52 Select

Methode: Select()

Beschreibung

Macht diesen Bereich zur aktuellen Auswahl der Benutzeroberfläche. Dasselbe Ergebnis können Sie mit: '*objRange.Parent.Selection* = *objRange*' erzielen.

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.

Beispiele

```
' ---------------------------------------
' Scripting environment - VBScript
' ---------------------------------------
DimobjAuthenticView
' we assume that the active document is open in authentic view mode
SetobjAuthenticView = Application.ActiveDocument.AuthenticView

' set current selection to end of document
```

 © 2016-2022 Altova GmbH

objAuthenticView.DocumentEnd.Select()

14.3.2.5.53 SelectNext

Methode: SelectNext (*eKind* als SPYAuthenticElementKind[606]) als AuthenticRange[399]

Beschreibung
Wählt das Element vom Typ eKind nach dem aktuellen Ende des Bereichs aus. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2003	Das Ziel liegt hinter dem Ende des Dokuments.
2005	Es wurde eine ungültige Elementart definiert. Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```
' ---------------------------------------------
' Scripting environment - VBScript
' Scan through the whole document word-by-word
' ---------------------------------------------
DimobjAuthenticView
' we assume that the active document is open in authentic view mode
SetobjAuthenticView = Application.ActiveDocument.AuthenticView

DimobjRange
SetobjRange = objAuthenticView.DocumentBegin
DimbEndOfDocument
bEndOfDocument = False

OnErrorResume Next
WhileNotbEndOfDocument
	objRange.SelectNext(spyAuthenticWord).Select
	If((Err.number - vbObjecterror) = 2003) Then
		bEndOfDocument = True
		Err.Clear
	ElseIf(Err.number <> 0) Then
		Err.Raise  ' forward error
	EndIf
Wend
```

14.3.2.5.54 SelectPrevious

Methode: GotoPrevious (*eKind* als SPYAuthenticElementKind[606]) als AuthenticRange[399]

Beschreibung
Wählt das Element vom Typ eKind vor dem aktuellen Anfang des Bereichs aus. Die Methode gibt das geänderte Bereichsobjekt zurück.

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2004 Das Ziel liegt vor dem Anfang des Dokuments
2005 Es wurde eine ungültige Elementart definiert.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```
' ---------------------------------------------
' Scripting environment - VBScript
' Scan through the whole document tag-by-tag
' ---------------------------------------------
Dim objAuthenticView
' we assume that the active document is open in authentic view mode
Set objAuthenticView = Application.ActiveDocument.AuthenticView

Dim objRange
Set objRange = objAuthenticView.DocumentEnd
Dim bBeginOfDocument
bBeginOfDocument = False

On Error Resume Next
While Not bBeginOfDocument
	objRange.SelectPrevious(spyAuthenticTag).Select
	If ((Err.number - vbObjecterror) = 2004) Then
		bBeginOfDocument = True
		Err.Clear
	ElseIf (Err.number <> 0) Then
		Err.Raise  ' forward error
	End If
Wend
```

14.3.2.5.55 SetElementAttributeValue

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

Beschreibung

Definiert den Wert des in strAttributeName definierten Attributs für das mit strElementName identifizierte Element. Wenn das Attribut unterstützt wird, aber keinen Wert zugewiesen hat, wird der leere String zurückgegeben. Um die Namen der von einem Element unterstützten Attribute zu ermitteln, verwenden Sie GetElementAttributeNames 409 oder HasElementAttribute 414.

Fehler

2001 Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005 Es wurde ein ungültiger Elementname angegeben.
Es wurde ein ungültiger Attributname definiert.

 © 2016-2022 Altova GmbH

Es wurde ein ungültiger Attributwert definiert.

Beispiele

```
' ----------------------------------------
' Scripting environment - VBScript
' Get and set element attributes
' ----------------------------------------
DimobjRange
' we assume that the active document is open in authentic view mode
SetobjRange = Application.ActiveDocument.AuthenticView.Selection

' first we find out all the elements below the beginning of the range
DimarrElements
objRange.GetElementHierarchy arrElements

IfIsArray(arrElements) Then
	IfUBound(arrElements) >= 0 Then
		' we use the top level element and find out its valid attributes
		DimarrAttrs()
		objRange.GetElementAttributeNames arrElements(0), arrAttrs

		IfUBound(arrAttrs) >= 0 Then
			' we retrieve the current value of the first valid attribute
			DimstrAttrVal
			strAttrVal = objRange.GetElementAttributeValue (arrElements(0), arrAttrs(0))
			msgbox "current value of " & arrElements(0) & "//" & arrAttrs(0) & " is: " & strAttrVal

			' we change this value and read it again
			strAttrVal = "Hello"
			objRange.SetElementAttributeValue arrElements(0), arrAttrs(0), strAttrVal
			strAttrVal = objRange.GetElementAttributeValue (arrElements(0), arrAttrs(0))
			msgbox "new value of " & arrElements(0) & "//" & arrAttrs(0) & " is: " & strAttrVal
		EndIf
	EndIf
EndIf
```

14.3.2.5.56 SetFromRange

Methode: SetFromRange (*objSrcRange* als AuthenticRange[399])

Beschreibung

Setzt den Bereich des Objekts auf dieselbe Anfangs- und Endposition wie objSrcRange.

Fehler

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

14.3.2.5.57 SetVariableValue

Methode: SetVariableValue`(strName als string, strValue als string)`

Rückgabewert
Definiert den Wert (zweiten Parameter) der im ersten Parameter genannten Variablen.

Fehler

2201	Ungültiges Objekt.
2202	Kein Kontext.
2204	Es befindet sich keine Variable dieses Namens im Geltungsbereich.
2205	Die Variable kann nicht ausgewertet werden.
2206	Die Variable gibt eine Sequenz zurück.
2207	Die Variable ist schreibgeschützt.
2208	Es ist keine Änderung zulässig.

14.3.2.5.58 Text

Eigenschaft: Text als String

Beschreibung
Setzt oder ruft den durch ein Bereichsobjekt ausgewählten Textinhalt ab.

Die Anzahl der abgerufenen Zeichen muss nicht unbedingt identisch sein, da es Textcursorpositionen zwischen dem Beginn und dem Ende des ausgewählten Bereichs gibt. Die meisten Dokumentelemente unterstützen eine Cursor-Endposition, die sich von der Cursor-Beginnposition des folgenden Elements unterscheidet. Dropdown-Listen haben nur eine Cursorposition, können jedoch Strings jeder Länge auswählen. Im Fall von Optionsfeldern und Kontrollkästchen enthält die Texteigenschaft den String des entsprechenden XML-Elements.

Wenn mit dem Bereich mehr als ein Element ausgewählt wird, ist der Text die Verkettung der Einzeltexte. XML Entities werden erweitert, sodass '&' als '&' erwartet wird.

Wenn der Text auf den leeren String gesetzt wird, werden keine XML-Elemente gelöscht. Verwenden Sie stattdessen Cut[404], Delete[404] oder PerformAction[423].

Fehler

2001	Das Authentic-Bereichsobjekt oder das damit in Zusammenhang stehende Ansichtsobjekt ist nicht mehr gültig.
2005	Es wurde eine ungültige Adresse für einen Rückgabeparameter angegeben.

14.3.2.6 AuthenticView

Eigenschaften	Methoden	Events
Application[437]	Goto[441]	OnBeforeCopy[429]

© 2016-2022 Altova GmbH

AsXMLString 437
DocumentBegin 439
DocumentEnd 439
Event 440
MarkupVisibility 442
Parent 443
Selection 443
XMLDataRoot 446
WholeDocument 445
IsRedoEnabled 442
IsUndoEnabled 442
Print 443
Redo 443
Undo 444
UpdateXMLInstanceEntities 445
OnBeforeCut 430
OnBeforeDelete 430
OnBeforeDrop 431
OnBeforePaste 431
OnDragOver 432
OnKeyBoardEvent 433
OnMouseEvent 434
OnSelectionChanged 434

Beschreibung
Über AuthenticView und seine untergeordneten Objekte AuthenticRange 399 und AuthenticDataTransfersteht Ihnen eine Schnittstelle für die Authentic View-Ansicht zur Verfügung, über die Sie Dokumentinhalt einfach und konsistent ändern können. Diese Schnittstellen treten nun anstelle der folgenden nun als **nicht mehr verwendet** markierten Schnittstellen:
OldAuthenticView(der alte Name war DocEditView)
AuthenticSelection(der alte Name war DocEditSelection, wird nun ersetzt durch AuthenticRange 399)
AuthenticEvent(der alte Name war DocEditEvent)

Mit AuthenticView erhalten Sie einfachen Zugriff auf spezielle Funktionen wie Drucken, den mehrstufigen Rückgängig-Puffer und die aktuelle Cursorauswahl oder - position.

AuthenticView verwendet zur einfachen Navigation im Dokument und zur flexiblen Auswahl logischer Textelemente Objekte vom Typ AuthenticRange 399. Verwenden Sie für einfache Auswahldefinitionen die Eigenschaften DocumentBegin 439, DocumentEnd 439 oder WholeDocument 445 und die Goto 441 Methode für komplexere Auswahldefinitionen. Informationen zum Navigieren relativ zu einem bestimmten Dokumentbereich finden Sie unter den Methoden und Eigenschaften des AuthenticRange 399 Objekts.

14.3.2.6.1 Events

14.3.2.6.1.1 OnBeforeCopy

Event: OnBeforeCopy()als Boolean

Skripting-Umgebung - VBScript:
```
Function On_AuthenticBeforeCopy()
        ' On_AuthenticBeforeCopy = False  ' to disable operation
End Function
```

Skripting-Umgebung - JScript:
```
function On_AuthenticBeforeCopy()
{
        // return false;  /* to disable operation */
}
```

IDE Plugin:
IXMLSpyPlugIn.OnEvent(21, ...) // nEventId = 21

Beschreibung
Dieses Event wird ausgelöst, bevor eine Kopieroperation an einem Dokument durchgeführt wird. Bei Rückgabe von *True* (oder nichts) wird die Kopieroperation gestattet, bei Rückgabe von *False* wird das Kopieren deaktiviert.

14.3.2.6.1.2 OnBeforeCut

Event: OnBeforeCut()als Boolean

Skripting-Umgebung - VBScript:
```
Function On_AuthenticBeforeCut()
	' On_AuthenticBeforeCut = False  ' to disable operation
End Function
```

Skripting-Umgebung - JScript:
```
function On_AuthenticBeforeCut()
{
	// return false;  /* to disable operation */
}
```

IDE Plugin:
IXMLSpyPlugIn.OnEvent(20, ...) // nEventId = 20

Beschreibung
Dieses Event wird ausgelöst, bevor eine Ausschneideoperation an einem Dokument durchgeführt wird. Geben Sie *True* (oder nichts) zurück, um den Vorgang zu gestatten und *False* um den Vorgang zu deaktivieren.

14.3.2.6.1.3 OnBeforeDelete

Event: OnBeforeDelete()als Boolean

Skripting-Umgebung - VBScript:
```
Function On_AuthenticBeforeDelete()
	' On_AuthenticBeforeDelete = False  ' to disable operation
End Function
```
Skripting-Umgebung - JScript:
```
function On_AuthenticBeforeDelete()
{
	// return false;  /* to disable operation */
}
```

IDE Plugin:
IXMLSpyPlugIn.OnEvent(22, ...) // nEventId = 22

Beschreibung
Dieses Event wird ausgelöst, bevor eine Löschoperation an einem Dokument durchgeführt wird. Geben Sie *True* (oder nichts) zurück, um den Vorgang zu gestatten und *False* um den Vorgang zu deaktivieren.

 © 2016-2022 Altova GmbH

14.3.2.6.1.4 OnBeforeDrop

Event: OnBeforeDrop (*i_nXPos*als Long,*i_nYPos*als Long,*i_ipRange* als AuthenticRange 399, *i_ipData*als cancelBoolean

Skripting-Umgebung - VBScript:

```
Function On_AuthenticBeforeDrop(nXPos, nYPos, objRange, objData)
        ' On_AuthenticBeforeDrop = False  ' to disable operation
End Function
```

Skripting-Umgebung - JScript:

```
function On_AuthenticBeforeDrop(nXPos, nYPos, objRange, objData)
{
        // return false;  /* to disable operation */
}
```

IDE Plugin:

IXMLSpyPlugIn.OnEvent(11, ...) // nEventId = 11

Beschreibung

Dieses Event wird ausgelöst, nachdem ein Objekt mit der Maus in das Applikationsfenster gezogen wurde. Alle mit dem Event in Zusammehang stehenden Informationen werden als Parameter übergeben.

Die ersten beiden Parameter definieren die Mausposition zum Zeitpunkt des Auftretens des Event. Der Parameter *objRange* übergibt ein Bereichsobjekt, das das XML-Element unterhalb der Mausposition auswählt. Der Wert dieses Parameters kann *NULL* sein. Überprüfen Sie dies, bevor Sie das Bereichsobjekt aufrufen. Mit Hilfe des Parameters *objData* können Sie Informationen über das gezogene Objekt aufrufen.

Geben Sie *False* zurück, um den Drop-Vorgang abzubrechen. Geben Sie *True* (oder nichts) zurück, um mit einem normalen Vorgang fortzusetzen.

14.3.2.6.1.5 OnBeforePaste

Event: OnBeforePaste (*objData* als Variant, *strType* als String) als Boolean

Skripting-Umgebung - VBScript:

```
Function On_AuthenticBeforePaste(objData,strType)
        ' On_AuthenticBeforePaste = False  ' to disable operation
End Function
```

Skripting-Umgebung - JScript:

```
function On_AuthenticBeforePaste(objData,strType)
{
        // return false;  /* to disable operation */
}
```

IDE Plugin:

IXMLSpyPlugIn.OnEvent(19, ...) // nEventId = 19

Beschreibung
Dieses Event wird ausgelöst, bevor eine Einfügeoperation am Dokument ausgeführt wird. Der Parameter *strType* ist entweder "TEXT", "UNICODETEXT" oder "IUNKNOWN". In den ersten beiden Fällen enthält *objData* eine String-Darstellung des Objekts, das eingefügt wird. Im letzteren Fall enthält *objData* einen Pointer auf eine IUnknown COM-Schnittstelle.

Geben Sie *True* (oder nichts) zurück, um die Einfügeoperation zu gestatten. Geben Sie *False* zurück, um die Operation zu deaktivieren.

14.3.2.6.1.6 OnBeforeSave

Event: OnBeforeSave (SaveAs flag) als Boolean

Beschreibung: Mit `OnBeforeSave` haben Sie die Möglichkeit den Benutzer vor einem Überschreiben des bestehenden Dokuments zu warnen oder das Dokument mit einem Schreibschutz zu versehen, wenn bestimmte Bedingungen nicht erfüllt werden. Das Event wird ausgelöst, bevor das Dialogfeld "Datei" angezeigt wird.

14.3.2.6.1.7 OnDragOver

Event: OnDragOver(*nXPos* als Long, *nYPos* als Long, *eMouseEvent* als SPYMouseEvent 610, *objRange* als AuthenticRange 399, *objData* als AuthenticDataTransfer) als Boolean

Skripting-Umgebung - VBScript:
```
Function On_AuthenticDragOver(nXPos, nYPos, eMouseEvent, objRange, objData)
        ' On_AuthenticDragOver = False  ' to disable operation
End Function
```

Skripting-Umgebung - JScript:
```
function On_AuthenticDragOver(nXPos, nYPos, eMouseEvent, objRange, objData)
{
        // return false;  /* to disable operation */
}
```

IDE Plugin:
```
IXMLSpyPlugIn.OnEvent(10, ...)   // nEventId = 10
```

Beschreibung
Dieses Event wird ausgelöst, wenn ein Objekt, von innerhalb oder außerhalb der Authentic View-Ansicht mit der Maus über das Applikationsfenster gezogen wird. Alle mit dem Event in Zusammenhang stehenden Informationen werden als Parameter übergeben.

Die ersten drei Parameter definieren die Mausposition, den Status der Maustaste und den Status virtueller Schaltflächen zum Zeitpunkt, zu dem das Event auftritt. Der Parameter *objRange* übergibt ein Bereichsobjekt, das das XML-Element unterhalb der Mausposition auswählt. Der Wert dieses Parameters kann *NULL* sein. Überprüfen Sie dies, bevor Sie das Bereichsobjekt aufrufen. Mit Hilfe des Parameters *objData* können Sie Informationen über das gezogene Objekt aufrufen.

 © 2016-2022 Altova GmbH

Geben Sie *False* zurück, um den Drag-Vorgang abzubrechen. Geben Sie *True* (oder nichts) zurück, um mit einem normalen Vorgang fortzusetzen.

14.3.2.6.1.8 OnKeyboardEvent

Event: OnKeyboardEvent (*eKeyEvent*als SPYKeyEvent[609], *nKeyCode*als Long, *nVirtualKeyStatus*als Long) als Boolean

Skripting-Umgebung - VBScript:

```
Function On_AuthenticKeyboardEvent(eKeyEvent, nKeyCode, nVirtualKeyStatus)
        ' On_AuthenticKeyboardEvent = True  ' to cancel bubbling of event
End Function
```

Skripting-Umgebung - JScript:

```
function On_AuthenticKeyboardEvent(eKeyEvent, nKeyCode, nVirtualKeyStatus)
{
        // return true;  /* to cancel bubbling of event */
}
```

IDE Plugin:

IXMLSpyPlugIn.OnEvent(30, ...) // nEventId = 30

Beschreibung

Dieses Event wird für *WM_KEYDOWN*, *WM_KEYUP* und *WM_CHAR* Windows-Meldungen ausgelöst.

Der eigentliche Meldungstyp steht im *eKeyEvent* Parameter zur Verfügung. Der Status virtueller Tasten ist im Parameter *nVirtualKeyStatus* kombiniert. Verwenden Sie die im Enumerations-Datentyp SPYVirtualKeyMask[617] definierten Bit-Masken, um eine Überprüfung der unterschiedlichen Tasten und ihrer Kombinationen durchzuführen.

14.3.2.6.1.9 OnLoad

Event: OnLoad ()

Beschreibung: Mit `OnLoad` können einige Funktionalitäten der Authentic-Ansicht eingeschränkt werden, wie im Beispiel unten gezeigt:

```
functionOn_AuthenticLoad(  )
{
    // We are disabling all entry helpers in order to prevent user from manipulating XML tree
    AuthenticView.DisableElementEntryHelper();
    AuthenticView.DisableAttributeEntryHelper();

    // We are also disabling the markup buttons for the same purpose
    AuthenticView.SetToolbarButtonState('AuthenticMarkupSmall',authenticToolbarButtonDisabled );
    AuthenticView.SetToolbarButtonState('AuthenticMarkupLarge',authenticToolbarButtonDisabled );
    AuthenticView.SetToolbarButtonState('AuthenticMarkupMixed',authenticToolbarButtonDisabled );
}
```

Im Beispiel wird der Status der Symbolleisten-Schaltflächen "Kleine Markup-Symbole", "Große Markup-Symbole" und "Gemischte Markup-Symbole" mit Hilfe der Schaltflächen-Identifier bearbeitet. Siehe vollständige Liste[435].

14.3.2.6.1.10 OnMouseEvent

Event: OnMouseEvent (*nXPos* als Long, *nYPos* als Long, *eMouseEvent* als SPYMouseEvent[610], *objRange* als AuthenticRange[399]) als Boolean

Skripting-Umgebung - VBScript:

```
Function On_AuthenticMouseEvent(nXPos, nYPos, eMouseEvent, objRange)
        ' On_AuthenticMouseEvent = True  ' to cancel bubbling of event
End Function
```

Skripting-Umgebung - JScript:

```
function On_AuthenticMouseEvent(nXPos, nYPos, eMouseEvent, objRange)
{
        // return true;  /* to cancel bubbling of event */
}
```

IDE Plugin:

```
IXMLSpyPlugIn.OnEvent(31, ...)   // nEventId = 31
```

Beschreibung

Dieses Event wird für jede Mausbewegung und jede Mausschaltflächen-Windows-Meldung ausgelöst.

Der eigentliche Meldungstyp und der Status der Maustaste stehen im *eMouseEvent* Parameter zur Verfügung. Verwenden Sie die im Enumerations-Datentyp SPYMouseEvent[610] definierten Bit-Masken, um eine Überprüfung auf die unterschiedlichen Meldungen, Tastenstatus und ihre Kombinationen durchzuführen.

Der Parameter *objRange* definiert den Teil des Dokuments, der sich an der aktuellen Mauscursorposition befindet. Das Bereichsobjekt wählt immer einen vollständigen Tag des Dokuments aus. (Dies kann sich in zukünftigen Versionen ändern, wenn ein genauerer Positionierungsmechanismus zur Verfügung steht). Wenn sich an der aktuellen Position kein auswählbarer Teil des Dokuments befindet, ist das Bereichsobjekt *Null*.

14.3.2.6.1.11 OnSelectionChanged

Event: OnSelectionChanged (*objNewSelection* als AuthenticRange[399])

Skripting-Umgebung - VBScript:

```
Function On_AuthenticSelectionChanged(objNewSelection)
End Function
```

Skripting-Umgebung - JScript:

```
function On_AuthenticSelectionChanged(objNewSelection)
{
}
```


© 2016-2022 Altova GmbH

IDE Plugin:
IXMLSpyPlugIn.OnEvent(23, ...) // nEventId = 23

Beschreibung
Dieses Event wird ausgelöst, wenn die sich die Auswahl auf der Benutzeroberfläche ändert.

14.3.2.6.1.12 OnToolbarButtonClicked

Event: OnToolbarButtonClicked (Button identifier)

Beschreibung: `OnToolbarButtonClicked` wird ausgelöst, wenn der Benutzer auf eine Symbolleisten-Schaltfläche klickt. Über den Parameter Schaltflächen-Identifier, kann ermittelt werden, auf welche Schaltfläche geklickt wurde. Unten sehen Sie die Liste der vordefinierten Schaltflächen-Identifier:

- AuthenticPrint
- AuthenticPrintPreview
- AuthenticUndo
- AuthenticRedo
- AuthenticCut
- AuthenticCopy
- AuthenticPaste
- AuthenticClear
- AuthenticMarkupHide
- AuthenticMarkupLarge
- AuthenticMarkupMixed
- AuthenticMarkupSmall
- AuthenticValidate
- AuthenticChangeWorkingDBXMLCell
- AuthenticSave
- AuthenticSaveAs
- AuthenticReload
- AuthenticTableInsertRow
- AuthenticTableAppendRow
- AuthenticTableDeleteRow
- AuthenticTableInsertCol
- AuthenticTableAppendCol
- AuthenticTableDeleteCol
- AuthenticTableJoinCellRight
- AuthenticTableJoinCellLeft
- AuthenticTableJoinCellAbove
- AuthenticTableJoinCellBelow
- AuthenticTableSplitCellHorizontally
- AuthenticTableSplitCellVertically
- AuthenticTableAlignCellContentTop
- AuthenticTableCenterCellVertically
- AuthenticTableAlignCellContentBottom
- AuthenticTableAlignCellContentLeft
- AuthenticTableCenterCellContent
- AuthenticTableAlignCellContentRight
- AuthenticTableJustifyCellContent
- AuthenticTableInsertTable

- AuthenticTableDeleteTable
- AuthenticTableProperties
- AuthenticAppendRow
- AuthenticInsertRow
- AuthenticDuplicateRow
- AuthenticMoveRowUp
- AuthenticMoveRowDown
- AuthenticDeleteRow
- AuthenticDefineEntities
- AuthenticXMLSignature

Der Benutzer kann für seine benutzerdefinierten Schaltflächen seine eigenen Identifier hinzufügen. Beachten Sie bitte, dass der Benutzer aufpassen muss, da die Identifier nicht auf Eindeutigkeit überprüft werden. Dieselben Identifier können zur Kennzeichnung von Schaltflächen in `Set/GetToolbarState()` COM API-Aufrufen verwendet werden. Durch Hinzufügen von Code für unterschiedliche Schaltflächen hat der Benutzer die Möglichkeit, das Verhalten der Symbolleiste der Authentic-Ansicht vollständig neu zu definieren, indem er seine eigenen Methoden zur Tabellenbearbeitung usw. hinzufügt.

14.3.2.6.1.13 OnToolbarButtonExecuted

Event: OnToolbarButtonExecuted (Button identifier)

Beschreibung: `OnToolbarButtonClicked` wird ausgelöst, wenn ein Benutzer auf eine Symbolleisten-Schaltfläche klickt. Über den Parameter Schaltflächen-Identifier, kann ermittelt werden, auf welche Schaltfläche geklickt wurde. Siehe Liste der vordefinierten vordefinierten Schaltflächen-Identifier 435.

`OnToolbarButtonExecuted` wird ausgelöst, nachdem die Symbolleisten-Aktion ausgeführt wurde und eignet sich z.B. zum Hinzufügen von aktualisiertem Code, wie im Beispiel unten gezeigt:

```
//event fired when a toolbar button action was executed
functionOn_AuthenticToolbarButtonExecuted(varBtnIdentifier )
{
	// After whatever command user has executed - make sure to update toolbar button states
	UpdateOwnToolbarButtonStates();
}
```

In diesem Fall ist `UpdateOwnToolbarButtonStates` eine in den globalen Deklarationen definierte Benutzerfunktion.

14.3.2.6.1.14 OnUserAddedXMLNode

Event: OnUserAddedXMLNode (XML node)

Beschreibung: `OnUserAddedXMLNode` wird ausgelöst, wenn der Benutzer einen XML-Node als Primäraktion hinzufügt. Dies geschieht in Situationen, in denen der Benutzer eine der folgenden Aktionen ausführt:

- Klicken auf automatisch hinzugefügte Hyperlinks (siehe Beispiel OnUserAddedXMLNode.sps)
- Auswahl der Kontextmenübefehle "Einfügen…", "Einfügen nach…", "Einfügen vor…"
- Klicken auf "Zeile anhängen", beim Einfügen neuer Symbolleisten-Schaltflächen

 © 2016-2022 Altova GmbH

- Ausführen von "Einfügen nach...", "Einfügen vor..." Aktionen in Elementeingabehilfen (außerhalb von StyleVision)

Das Event wird **nicht** bei "Zeile duplizieren" oder, wenn der Node extern (z.B. über die COM API) hinzugefügt wurde, oder bei "Anwenden" (z.B. Textstatus-Symbole) oder in XML-Tabellenoperationen bzw. in Datenbankoperationen ausgelöst.

Der Event-Parameter ist das XML-Node-Objekt, das hinzugefügt wurde, wodurch der Benutzer die Möglichkeit hat, den hinzugefügten XML-Node zu bearbeiten. Ein ausführliches Beispiel für einen Event Handler finden Sie in der Datei `OnUserAddedXMLNode.sps`.

14.3.2.6.2 Application

Eigenschaft: Application als Application 372 (schreibgeschützt)

Beschreibung
Ruft das Authentic Desktopapplication-Objekt auf.

Fehler

2000	Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
2005	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.6.3 AsXMLString

Eigenschaft: AsXMLString als String

Beschreibung
Gibt den Dokumentinhalt als XML-String zurück oder definiert ihn als solchen. Wenn der Inhalt auf einen neuen Wert gesetzt wird, wird die gerade verwendete Schema- oder sps-Datei nicht geändert. Wenn der neue XMLStringder aktuellen Schemadatei nicht entspricht, wird Fehler 2011 zurückgegeben.

Fehler

2000	Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
2011	AsXMLStringwurde auf einen Wert gesetzt, der gemäß der aktuellen Schemadatei kein gültiger XML-Code ist.

14.3.2.6.4 ContextMenu

Eigenschaft: ContextMenu() als ContextMenu

Beschreibung
Über die Eigenschaft `ContextMenu` erhält der Benutzer Zugriff auf das Kontextmenü. Am besten lässt sich dies über den Event Handler `OnContextMenuActivated` durchführen.

Fehler

2000	Ungültiges Objekt.
2005	Ungültiger Parameter.

14.3.2.6.5 CreateXMLNode

Methode: CreateXMLNode (*nKind* als SPYXMLDataKind) als XMLData

Rückgabewert
Die Methode gibt das neue XMLData [593] Objekt zurück.

Beschreibung
Um ein neues XMLDataObjekt zu erstellen, verwenden Sie die Methode CreateXMLNode().

Fehler
2000 Ungültiges Objekt.
2012 XML-Node kann nicht erstellt werden.

14.3.2.6.6 DisableAttributeEntryHelper

Methode: DisableAttributeEntryHelper()

Beschreibung
DisableAttributeEntryHelper() deaktiviert die Attribut-Eingabehilfe in XMLSpy, Authentic Desktop und dem Authentic Browser Plug-in.

Fehler
2000 Ungültiges Objekt.

14.3.2.6.7 DisableElementEntryHelper

Methode: DisableElementEntryHelper()

Beschreibung
DisableElementEntryHelper() deaktiviert die Element-Eingabehilfe in XMLSpy, Authentic Desktop und dem Authentic Browser Plug-in.

Fehler
2000 Ungültiges Objekt.

14.3.2.6.8 DisableEntityEntryHelper

Methode: DisableEntityEntryHelper()

Beschreibung

 © 2016-2022 Altova GmbH

DisableEntityEntryHelper() deaktiviert die Entity-Eingabehilfe in XMLSpy, Authentic Desktop und dem Authentic Browser Plug-in.

Fehler
2000 Ungültiges Objekt.

14.3.2.6.9 DocumentBegin

Eigenschaft: DocumentBegin als AuthenticRange 399 (schreibgeschützt)

Beschreibung
Ruft ein Bereichsobjekt auf, das auf den Anfang des Dokuments zeigt.

Fehler
2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.6.10 DocumentEnd

Eigenschaft: DocumentEndals AuthenticRange 399(schreibgeschützt)

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

Fehler
2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.6.11 DoNotPerformStandardAction

Methode DoNotPerformStandardAction ()

Beschreibung
`DoNotPerformStandardAction()` dient als Abbrech-Bubble für Makros und bricht die weitere Ausführung ab, nachdem das Makro fertig ausgeführt wurde.

Fehler
2000 Ungültiges Objekt.

14.3.2.6.12 EvaluateXPath

Methode: EvaluateXPath (XMLData als XMLData 593, strExpression als string) strValue als string

Rückgabewert
Diese Methode gibt einen String zurück.

Beschreibung
`EvaluateXPath()` führt einen XPath-Ausdruck im angegebenen XML-Kontext-Node aus. Das Ergebnis wird als String zurückgegeben. Im Fall einer Sequenz ist der String durch Leerzeichen getrennt.

Fehler
2000 Ungültiges Objekt.
2005 Ungültiger Parameter.
2008 Interner Fehler.
2013 XPath-Fehler.

14.3.2.6.13 Event

Eigenschaft: Eventals AuthenticEvent(schreibgeschützt)

Beschreibung
Über diese Eigenschaft erhalten Sie auf dieselbe Art wie bei OldAuthenticView.eventZugriff auf die Parameter des letzten Event. Da nun alle Events für die Skripting-Umgebung und externe Clients mit Parameters verfügbar sind, sollte dieseEventEigenschaft nur von IDE-Plug-ins aus verwendet werden.

Fehler
2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.6.14 EventContext

Eigenschaft: EventContext() als EventContext

Beschreibung
Über die Eigenschaft `EventContext` 396 erhalten Sie Zugriff auf den Kontext des laufenden Makros. Nähere Informationen dazu finden Sie unter der Beschreibung zur `EventContext` 396 Schnittstelle.

Fehler
2000 Ungültiges Objekt.

14.3.2.6.15 GetToolbarButtonState

Methode: GetToolbarButtonState (`ButtonIdentifier` als string) als `AuthenticToolbarButtonState`

Rückgabewert
Die Methode gibt `AuthenticToolbarButtonState` zurück.

Beschreibung

 © 2016-2022 Altova GmbH

`GetToolbarButtonState` fragt den Status der Symbolleisten-Schaltfläche ab und gestattet dem Benutzer, die durch ihren Schaltflächen-Identifier gekennzeichnete Schaltfläche (siehe Liste oben 435) zu aktivieren bzw. zu deaktivieren. Auf diese Art können die Symbolleisten-Schaltflächen z.B. permanent deaktiviert werden. Eine andere Verwendung der Methode ist `SetToolbarButtonState` in den `OnSelectionChanged` Event Handler zu setzen, da die Symbolleisten-Schaltflächen regelmäßig aktualisiert werden, wenn sich die Auswahl im Dokument ändert.

Der Status der Symbolleisten-Schaltflächen wird unter der Liste der Enumerationen 606 aufgelistet.

Der Standardzustand bedeutet, dass die Aktivierung/Deaktivierung der Schaltfläche über AuthenticView erfolgt. Wenn der Benutzer den Zustand auf aktiviert oder deaktiviert setzt, bleibt die Schaltfläche so lange in diesem Zustand, solange der Benutzer den Zustand nicht ändert.

Fehler

2000	Ungültiges Objekt.
2005	Ungültiger Parameter.
2008	Interner Fehler.
2014	Ungültiger Schaltflächen-Identifier.

14.3.2.6.16 Goto

Methode: Goto(*eKind*als SPYAuthenticElementKind 606, *nCount*als Long, *eFrom* als *SPYAuthenticDocumentPosition* 605) als *AuthenticRange* 399

Beschreibung

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

Fehler

2000	Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
2003	Das Ziel liegt hinter dem Ende des Dokuments.
2004	Das Ziel liegt vor dem Anfang des Dokuments.
2005	Es wurde eine ungültige Elementart definiert. Die Ausgangsposition des Dokuments ist keine *spyAuthenticDocumentBegin* oder *spyAuthenticDocumentEnd* Position. Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```
' ----------------------------------------
' Scripting environment - VBScript
' ----------------------------------------
DimobjAuthenticView
' we assume that the active document is open in authentic view mode
SetobjAuthenticView = Application.ActiveDocument.AuthenticView

OnErrorResume Next
DimobjRange
' goto beginning of first table in document
```

```
SetobjRange = objAuthenticView.Goto (spyAuthenticTable, 1, spyAuthenticDocumentBegin)
If(Err.number = 0) Then
        objRange.Select()
Else
        MsgBox "No table found in document"
EndIf
```

14.3.2.6.17 IsRedoEnabled

Eigenschaft: IsRedoEnabled als Boolean (schreibgeschützt)

Beschreibung
True, wenn Wiederhol-Schritte verfügbar sind und Redo 443 möglich ist.

Fehler
2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.6.18 IsUndoEnabled

Eigenschaft: IsUndoEnabled als Boolean (schreibgeschützt)

Beschreibung
True, wenn Rückgängig-Schritte verfügbar sind und Undo 444 möglich ist.

Fehler
2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.6.19 MarkupVisibility

Eigenschaft: MarkupVisibilityals SPYAuthenticMarkupVisibility 606

Beschreibung
Setzt oder ruft die aktuelle Sichtbarkeit des Markup-Codes ab.

Fehler
2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
2005 Es wurde ein ungültiger Enumerationswert angegeben.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.6.20 Parent

Eigenschaft: Parent als Document 462 (schreibgeschützt)

Beschreibung
Ruft das in dieser Ansicht angezeigte Dokument auf.

Fehler
2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.6.21 Print

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

Beschreibung
Druckt das in dieser Ansicht angezeigte Dokument. Wenn *bWithPreview* auf *True* gesetzt ist, wird das Dialogfeld "Druckvorschau" angezeigt. Wenn *bPromptUser* auf *True* gesetzt ist, wird das Dialogfeld "Drucken" angezeigt. Wenn beide Parameter auf *False* gesetzt sind, wird das Dokument ohne weitere Eingabe durch den Benutzer gedruckt.

Fehler
2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.

14.3.2.6.22 Redo

Methode: Redo() als Boolean

Beschreibung
Stellt die durch den letzten Rückgängig-Befehl rückgängig gemachte Änderung wieder her.

Fehler
2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.6.23 Selection

Eigenschaft: Selection als AuthenticRange 399

Beschreibung
Setzt oder ruft die aktuelle Textauswahl auf der Benutzeroberfläche ab.

Fehler
2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
2002 Es ist keine Cursorauswahl aktiv.

2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```
' ----------------------------------------
' Scripting environment - VBScript
' ----------------------------------------
DimobjAuthenticView
' we assume that the active document is open in authentic view mode
SetobjAuthenticView = Application.ActiveDocument.AuthenticView

' if we are the end of the document, re-start at the beginning
If(objAuthenticView.Selection.IsEqual(objAuthenticView.DocumentEnd)) Then
	objAuthenticView.Selection = objAuthenticView.DocumentBegin
Else
	' objAuthenticView.Selection = objAuthenticView.Selection.GotoNextCursorPosition()
	' or shorter:
	objAuthenticView.Selection.GotoNextCursorPosition().Select
EndIf
```

14.3.2.6.24 SetToolbarButtonState

Methode: SetToolbarButtonState (`ButtonIdentifier` als string, `AuthenticToolbarButtonState` state)

Beschreibung

`SetToolbarButtonState` fragt den Status der Symbolleisten-Schaltfläche ab und gestattet dem Benutzer, die durch ihren Schaltflächen-Identifier gekennzeichnete Schaltfläche (siehe Liste oben 435) zu aktivieren bzw. zu deaktivieren. Auf diese Art können die Symbolleisten-Schaltflächen z.B. permanent deaktiviert werden. Eine andere Verwendung der Methode ist `SetToolbarButtonState` in den `OnSelectionChanged` Event Handler zu setzen, da die Symbolleisten-Schaltflächen regelmäßig aktualisiert werden, wenn sich die Auswahl im Dokument ändert.

Der Status der Symbolleisten-Schaltflächen wird unter der Liste der Enumerationen 606 aufgelistet.

Der Standardzustand bedeutet, dass die Aktivierung/Deaktivierung der Schaltfläche über AuthenticView erfolgt. Wenn der Benutzer den Zustand auf aktiviert oder deaktiviert setzt, bleibt die Schaltfläche so lange in diesem Zustand, solange der Benutzer den Zustand nicht ändert.

Fehler

2000 Ungültiges Objekt.
2008 Interner Fehler.
2014 Ungültiger Schaltflächen-Identifier.

14.3.2.6.25 Undo

Methode: Undo()als Boolean

Beschreibung

Macht die letzte Änderung des Dokuments innerhalb dieser Ansicht rückgängig.

 © 2016-2022 Altova GmbH

Fehler

2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.6.26 UpdateXMLInstanceEntities

Methode: UpdateXMLInstanceEntities()

Beschreibung

Aktualisiert die interne Darstellung der deklarierten Entities und befüllt die Eingabehilfen erneut. Zusätzlich dazu wird der Validator neu geladen, sodass die XML-Datei korrekt validiert werden kann. Beachten Sie bitte, dass dadurch auch Schemadateien neu geladen werden können.

Fehler

Die Methode gibt niemals einen Fehler zurück.

Beispiel

```
// ----------------------------------------
// Scripting environment - JavaScript
// ----------------------------------------
if(Application.ActiveDocument &&(Application.ActiveDocument.CurrentViewMode ==4))
{
	varobjDocType;
	objDocType = Application.ActiveDocument.DocEditView.XMLRoot.GetFirstChild(10);

	if(objDocType)
	{
		varobjEntity = Application.ActiveDocument.CreateChild(14);
		objEntity.Name = "child";
		objEntity.TextValue = "SYSTEM \"child.xml\"";
		objDocType.AppendChild(objEntity);

		Application.ActiveDocument.AuthenticView.UpdateXMLInstanceEntities();
	}
}
```

14.3.2.6.27 WholeDocument

Eigenschaft: WholeDocument als AuthenticRange 399 (schreibgeschützt)

Beschreibung

Ruft ein Bereichsobjekt ab, das das gesamte Dokument auswählt.

Fehler

2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.6.28 XMLDataRoot

Eigenschaft: XMLDataRoot als XMLData 593 (schreibgeschützt)

Beschreibung
Gibt das am höchsten gestufte XMLData-Element des aktuellen Dokuments zurück oder definiert dieses. Dieses Element beschreibt normalerweise die Dokumentstruktur und wäre von der Art spyXMLDataXMLDocStruct, spyXMLDataXMLEntityDocStruct oder spyXMLDataDTDDocStruct.

Fehler
2000 Das Authentic-Ansichtsobjekt ist nicht mehr gültig.
2005 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.7 CodeGeneratorDlg

Nur verfügbar/aktiv in der Enterprise Edition. Bei Aufruf von einer anderen Version aus wird ein Fehler zurückgegeben.

Eigenschaften und Methoden

Standard-Automation-Eigenschaften
Application 447
Parent 450

Eigenschaften zur Auswahl der Programmiersprache
ProgrammingLanguage 451
TemplateFileName 451

Einstellungen für C++-Code
CPPSettings_DOMType 447
CPPSettings_LibraryType 448
CPPSettings_UseMFC 449
CPPSettings_GenerateVC6ProjectFile 447
CPPSettings_GenerateVSProjectFile 448

Einstellungen für C#-Code
CSharpSettings_ProjectType 449

Behandlung von Dialogfeldern für die oben genannten Codegenerierungseigenschaften
PropertySheetDialogAction 451

Eigenschaften zur Auswahl des Ausgabepfads
OutputPath 449
OutputPathDialogAction 450

Darstellung des Ergebnisses
OutputResultDialogAction 450

Beschreibung

 © 2016-2022 Altova GmbH

Mit Hilfe dieses Objekts können Sie die Generierung von Programmcode für Schemadateien konfigurieren. Die Methode GenerateProgramCode 478 erwartet einen CodeGeneratorDlg als Parameter, um die Codegenerierung sowie die damit in Zusammenhang stehenden Interaktionen des Benutzers zu konfigurieren.

14.3.2.7.1 Application

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Eigenschaft: Applicationals Application 372 (schreibgeschützt)

Beschreibung
Ruft das Authentic Desktopapplication-Objekt auf.

Fehler
2200 Das Objekt ist nicht mehr gültig.
2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.7.2 CPPSettings_DOMType

Eigenschaft: CPPSettings_DOMTypeals SPYDOMType 607
Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung
Definiert eine der Einstellungen für die Konfiguration der Generierung von C++-Code.

Fehler
2200 Das Objekt ist nicht mehr gültig.
2201 Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

14.3.2.7.3 CPPSettings_GenerateVC6ProjectFile

Eigenschaft: CPPSettings_GenerateVC6ProjectFileals Boolean
Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung
Definiert eine der Einstellungen für die Konfiguration der Generierung von C++-Code.

Fehler
2200 Das Objekt ist nicht mehr gültig.
2201 Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

14.3.2.7.4 CPPSettings_GenerateGCCMakefile

Eigenschaft: CPPSettings_GenerateGCCMakefileals Boolean
Steht nur in der Enterprise Edition zur Verfügung. Bei Aufruf von einer anderen Version aus, wird eine Fehlermeldung zurückgegeben.

Beschreibung
Erstellt Makefiles zum Kompilieren des generierten Codes unter Linux mit GCC.

Fehler

2200	Das Objekt ist nicht mehr gültig.
2201	Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

14.3.2.7.5 CPPSettings_GenerateVSProjectFile

Eigenschaft: CSharpSettings_GenerateVSProjectFileals SPYProjectType 612
Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung
Definiert eine der Einstellungen für die Konfiguration der Generierung von C++-Code. Nur spyVisualStudio2005Project(=4), spyVisualStudio2008Project (=5) und `spyVisualStudio2010Project (=6)` sind gültige Projekttypen.

Fehler

2200	Das Objekt ist nicht mehr gültig.
2201	Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

14.3.2.7.6 CPPSettings_LibraryType

Eigenschaft: CPPSettings_LibraryTypeals SPYLibType 610
Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung
Definiert eine der Einstellungen für die Konfiguration der Generierung von C++-Code.

Fehler

2200	Das Objekt ist nicht mehr gültig.
2201	Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

© 2016-2022 Altova GmbH

14.3.2.7.7 CPPSettings_UseMFC

Eigenschaft: CPPSettings_UseMFCals Boolean
Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung
Definiert eine der Einstellungen für die Konfiguration der Generierung von C++-Code.

Fehler

2200	Das Objekt ist nicht mehr gültig.
2201	Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

14.3.2.7.8 CSharpSettings_ProjectType

Eigenschaft: CSharpSettings_ProjectTypeals SPYProjectType 612
Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung
Definiert die einzige Einstellung für die Konfiguration der Generierung von C#-Code.

Fehler

2200	Das Objekt ist nicht mehr gültig.
2201	Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

14.3.2.7.9 OutputPath

Eigenschaft: OutputPathals String
Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung
Wählt das Basisverzeichnis für den gesamten generierten Code aus.

Fehler

2200	Das Objekt ist nicht mehr gültig.
2201	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.7.10 OutputPathDialogAction

Eigenschaft: OutputPathDialogActionals SPYDialogAction[607]
Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung
Definiert, wie das untergeordnete Dialogfeld zur Auswahl des Ausgabepfads für die Codegenerierung behandelt wird. Setzen Sie diese Wert auf *spyDialogUserInput(2),* damit das Dialogfeld standardmäßig mit dem aktuellen Wert der Eigenschaft OutputPath[449] angezeigt wird. Verwenden Sie *spyDialogOK(0),* damit das Dialogfeld nicht angezeigt wird.

Fehler
2200 Das Objekt ist nicht mehr gültig.
2201 Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

14.3.2.7.11 OutputResultDialogAction

Eigenschaft: OutputResultDialogActionals SPYDialogAction[607]
Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung
Definiert, wie das untergeordnete Dialogfeld behandelt wird, das fragt, ob das Ergebnis der Codegenerierung angezeigt werden soll. Setzen Sie diesen Wert auf *spyDialogUserInput(2),* damit das Dialogfeld angezeigt wird. Verwenden Sie *spyDialogOK(0),* damit das Dialogfeld dem Benutzer nicht angezeigt wird.

Fehler
2200 Das Objekt ist nicht mehr gültig.
2201 Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

14.3.2.7.12 Parent

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Eigenschaft: Parentals Dialogs[458] (schreibgeschützt)

Beschreibung
Ruft das übergeordnete Element des Objekts auf.

Fehler
2200 Das Objekt ist nicht mehr gültig.
2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

 © 2016-2022 Altova GmbH

14.3.2.7.13 ProgrammingLanguage

Eigenschaft: ProgrammingLanguageals ProgrammingLanguage 611
Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung
Wählt die Ausgabesprache aus, in der der Code generiert werden soll.

Achtung: Wenn Sie diese Eigenschaft auf C++, C# oder Java setzen, wird auch die Eigenschaft TemplateFileName 451 auf die entsprechende mit Authentic Desktopinstallierte Vorlagendatei gesetzt. Wenn Sie C++, C# oder Java-Code basierend auf Ihren eigenen Vorlagen generieren möchten, definieren Sie zuerst die Programmiersprache und wählen Sie dann Ihre Vorlagendatei aus.

Fehler

2200	Das Objekt ist nicht mehr gültig.
2201	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.7.14 PropertySheetDialogAction

Eigenschaft: PropertySheetDialogActionals SPYDialogAction 607
Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung
Definiert, wie das untergeordnete Dialogfeld, mit dem die Codegenerierung konfiguriert wird, behandelt wird. Setzen Sie diesen Wert auf *spyDialogUserInput(2)*um das Dialogfeld mit den aktuellen Werten als Standardwerten anzuzeigen. Verwenden Sie *spyDialogOK(0),*damit das Dialogfeld nicht angezeigt wird.

Fehler

2200	Das Objekt ist nicht mehr gültig.
2201	Es wurde eine ungültige Aktion als Parameter übergeben oder es wurde eine ungültige Adresse für den Rückgabeparameter angegeben.

14.3.2.7.15 TemplateFileName

Eigenschaft: TemplateFileNameals String
Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung
Wählt die Vorlagendatei für die Codegenerierung aus. Im Ordner "SPL" Ihres Authentic DesktopInstallationsverzeichnisses sind Vorlagendateien für C++, C# oder Java enthalten.

Wenn Sie diese Eigenschaft auf eine der Codegenerierungsvorlagendateien Ihrer Authentic DesktopInstallation setzen, wird die Eigenschaft ProgrammingLanguage 451 automatisch auf den entsprechenden Wert gesetzt.

Fehler

2200 Das Objekt ist nicht mehr gültig.
2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.8 DatabaseConnection

Eigenschaften für den Import und Export
File 455 oder
ADOConnection 452 oder
ODBCConnection 456

Eigenschaften nur für den Import
DatabaseKind 454
SQLSelect 457
AsAttributes 453
ExcludeKeys 454
IncludeEmptyElements 455
NumberDateTimeFormat 456
NullReplacement 456
CommentIncluded 453

Eigenschaften nur für den Export
CreateMissingTables 453
CreateNew 454
TextFieldLen 457
DatabaseSchema 454

Eigenschaften für die Generierung eines XML-Schemas anhand der Datenbankstruktur
PrimaryKeys 456
ForeignKeys 455
UniqueKeys 458
SchemaExtensionType 457
SchemaFormat 457
ImportColumnsType 455

Beschreibung
DatabaseConnectiondefiniert die Parameter für die Datenbankverbindung.

Bitte beachten Sie, dass sich die Eigenschaften der DatabaseConnection Schnittstelle auf die Einstellungen des Import- und Exportdialogfelds von Authentic Desktop beziehen.

14.3.2.8.1 ADOConnection

Eigenschaft: ADOConnectionals String

Beschreibung
Die Eigenschaft ADOConnectionenthält einen Connection String. Verwenden Sie entweder diese Eigenschaft oder ODBCConnection 456 oder File 455 ,um eine Datenbank zu referenzieren.

Fehler

 © 2016-2022 Altova GmbH

Es werden keine Fehlercodes zurückgegeben.

Beispiel

```
DimobjSpyConn AsDatabaseConnection
SetobjSpyConn = objSpy.GetDatabaseSettings

DimobjADO AsDataLinks
SetobjADO = CreateObject("DataLinks")

IfNot(objADO Is Nothing) Then
        DimobjConn AsConnection
        SetobjConn = objADO.PromptNew
        objSpyConn.ADOConnection = objConn.ConnectionString
EndIf
```

14.3.2.8.2 AsAttributes

Eigenschaft: AsAttributesals Boolean

Beschreibung
Setzen Sie AsAttributesauf "true", wenn alle Importfelder initialisiert werden sollen, um als Attribute importiert zu werden. Der Standardwert ist "false" und initialisiert alle Felder, um als Elemente importiert zu werden. Diese Eigenschaft wird nur in Aufrufen von Application.GetDatabaseImportElementList 379 verwendet.

Fehler
Es werden keine Fehlercodes zurückgegeben.

14.3.2.8.3 CommentIncluded

Eigenschaft: CommentIncludedals Boolean

Beschreibung
Diese Eigenschaft legt fest, ob zusätzliche Kommentare zum generierten XML-Code hinzugefügt werden. Der Standardwert ist true. Diese Eigenschaft wird nur beim Import aus einer Datenbank verwendet.

Fehler
Es werden keine Fehlercodes zurückgegeben.

14.3.2.8.4 CreateMissingTables

Eigenschaft: CreateMissingTablesals Boolean

Beschreibung
Wenn CreateMissingTables"true" ist, werden Tabellen, die nicht bereits in der Exportdatenbank definiert sind, beim Export erstellt. Die Standardeinstellung ist true. Diese Eigenschaft wird nur beim Export in Datenbanken verwendet.

Fehler
Es werden keine Fehlercodes zurückgegeben.

14.3.2.8.5 CreateNew

Eigenschaft: CreateNewals Boolean

Beschreibung
Setzen Sie CreateNewauf "true", wenn Sie beim Export eine neue Datenbank erstellen möchten. Eine möglicherweise bereits vorhandene Datenbank wird dabei überschrieben. Siehe auch DatabaseConnection.File 455. Der Standardwert ist "false". Diese Eigenschaft wird nur beim Export in eine Datenbank verwendet.

Fehler
Es werden keine Fehlercodes zurückgegeben.

14.3.2.8.6 DatabaseKind

Eigenschaft: DatabaseKindals SPYDatabaseKind 607

Beschreibung
Wählen Sie aus, welche Art von Datenbank Zugriff erhält. Der Standardwert ist spyDB_Unspecified(7)und ist in den meisten Fällen ausreichend. Diese Eigenschaft wird nur beim Import von Datenbanken verwendet.

Fehler
Es werden keine Fehlercodes zurückgegeben.

14.3.2.8.7 DatabaseSchema

Eigenschaft: DatabaseSchemaals String

Beschreibung
Diese Eigenschaft definiert das Schema, das für den Export in schemafähige Datenbanken verwendet wird. Der Standardwert ist "". Diese Eigenschaft wird nur beim Export in eine Datebank verwendet.

Fehler
Es werden keine Fehlercodes zurückgegeben.

14.3.2.8.8 ExcludeKeys

Eigenschaft: ExcludeKeysals Boolean

Beschreibung
Setzen Sie ExcludeKeysauf "true", wenn Sie alle Schlüsselspalten vom Import ausnehmen möchten. Der Standardwert ist "false". Diese Eigenschaft wird nur beim Import von Datenbanken verwendet.

© 2016-2022 Altova GmbH

Fehler
Es werden keine Fehlercodes zurückgegeben.

14.3.2.8.9 File

Eigenschaft: Fileals String

Beschreibung
Die Eigenschaft Filedefiniert den Pfad für die Datenbank beim Export oder Import. Diese Eigenschaft kann nur in Verbindung mit einer Microsoft Access-Datenbank verwendet werden. Verwenden Sie entweder diese Eigenschaft oder ODBCConnection 456 oder ADOConnection 452 um die Datenbank zu referenzieren.

Fehler
Es werden keine Fehlercodes zurückgegeben.

14.3.2.8.10 ForeignKeys

Eigenschaft: ForeignKeysals Boolean

Beschreibung
Gibt an, ob der Sekundärschlüssel-Constraint erstellt wird oder nicht. Diese Eigenschaft wird nur bei Erstellung eines XML-Schemas anhand einer Datenbankstruktur verwendet.

Fehler
Es werden keine Fehlercodes zurückgegeben.

14.3.2.8.11 ImportColumnsType

Eigenschaft: ImportColumnsTypeals SPYImportColumnsType 609

Beschreibung
Definiert, ob Spalteninformationen aus der Datenbank im Schema als Element oder Attribut gespeichert werden. Diese Eigenschaft wird nur bei Erstellung eines XML-Schemas anhand einer Datenbankstruktur verwendet.

Fehler
Es werden keine Fehlercodes zurückgegeben.

14.3.2.8.12 IncludeEmptyElements

Eigenschaft: IncludeEmptyElementsals Boolean

Beschreibung
Setzen Sie IncludeEmptyElementsauf "false" wenn alle leeren Elemente ausgenommen werden sollen. Der Standardwert ist "true". Diese Eigenschaft wird nur beim Import von Datenbanken verwendet.

Fehler

Es werden keine Fehlercodes zurückgegeben.

14.3.2.8.13 NullReplacement

Eigenschaft: NullReplacementals String

Beschreibung
Diese Eigenschaft enthält den Textwert, der beim Import von leeren Elementen (Nullwerten) verwendet wird. Der Standardwert ist "". Diese Eigenschaft wird nur beim Import von Datenbanken verwendet.

Fehler
Es werden keine Fehlercodes zurückgegeben.

14.3.2.8.14 NumberDateTimeFormat

Eigenschaft: NumberDateTimeFormat als SPYNumberDateTimeFormat 611

Beschreibung
Die Eigenschaft NumberDateTimeFormat definiert das Format von Datums- und Uhrzeitwerten. Die Standardeinstellung ist spySystemLocale 611. Diese Eigenschaft wird nur beim Import von Datenbanken verwendet.

Fehler
Es werden keine Fehlercodes zurückgegeben.

14.3.2.8.15 ODBCConnection

Eigenschaft: ODBCConnectionals String

Beschreibung
Die Eigenschaft ODBCConnectionenthält einen ODBC Connection String. Verwenden Sie entweder diese Eigenschaft oder ADOConnection 452 oder File 455 um eine Datenbank zu referenzieren.

Fehler
Es werden keine Fehlercodes zurückgegeben.

14.3.2.8.16 PrimaryKeys

Eigenschaft: PrimaryKeysals Boolean

Beschreibung
Definiert, ob ein Primärschlüssel-Constraint erstellt wird oder nicht. Die Standardeinstellung ist "true". Diese Eigenschaft wird nur verwendet, wenn ein XML-Schema anhand einer Datenbankstruktur erstellt wird.

Fehler
Es werden keine Fehlercodes zurückgegeben.

© 2016-2022 Altova GmbH

14.3.2.8.17 SchemaExtensionType

Eigenschaft: SchemaExtensionTypeals SPYSchemaExtensionType 614

Beschreibung
Definiert den bei der Schemagenerierung verwendeten Schema Extension-Typ. Diese Eigenschaft wird nur verwendet, wenn ein XML-Schema anhand einer Datenbankstruktur erstellt wird.

Fehler
Es werden keine Fehlercodes zurückgegeben.

14.3.2.8.18 SchemaFormat

Eigenschaft: SchemaFormatals SPYSchemaFormat 614

Beschreibung
Definiert das bei der Schemagenerierung verwendete Schemaformat. Diese Eigenschaft wird nur verwendet, wenn ein XML-Schema anhand einer Datenbankstruktur erstellt wird.

Fehler
Es werden keine Fehlercodes zurückgegeben.

14.3.2.8.19 SQLSelect

Eigenschaft: SQLSelectals String

Beschreibung
Die SQL-Abfrage für den Import ist in der Eigenschaft SQLSelectgespeichert. Diese Eigenschaft wird nur beim Import von Datenbanken verwendet.

Fehler
Es werden keine Fehlercodes zurückgegeben.

14.3.2.8.20 TextFieldLen

Eigenschaft: TextFieldLenals long

Beschreibung
Die Eigenschaft TextFieldLen definiert die Länge der beim Export erstellten Textfelder. Standardwert ist 255. Diese Eigenschaft wird nur beim Export in eine Datenbank verwendet.

Fehler
Es werden keine Fehlercodes zurückgegeben.

14.3.2.8.21 UniqueKeys

Eigenschaft: UniqueKeysals Boolean

Beschreibung
Definiert, ob ein Eindeutiger-Schlüsse-Constraint erstellt wird oder nicht. Die Standardeinstellung ist "true". Diese Eigenschaft wird nur verwendet, wenn ein XML-Schema anhand einer Datenbankstruktur erstellt wird.

Fehler
Es werden keine Fehlercodes zurückgegeben.

14.3.2.9 Dialogs

Eigenschaften und Methoden

Standard-Automation-Eigenschaften
Application 458
Parent 459

Verschiedene Dialog-Objekte
CodeGeneratorDlg 459
FileSelectionDlg 459
SchemaDocumentationDlg 460
GenerateSampleXMLDlg 460
DTDSchemaGeneratorDlg 460
FindInFilesDlg 460

Beschreibung
Über das Objekt "Dialogs" haben Sie Zugriff auf die verschiedenen integrierten Dialogfelder von Authentic Desktop. Mit Hilfe dieser Dialog-Objekte können die Felder von Benutzerdialogfeldern initialisiert werden, bevor sie dem Benutzer angezeigt werden oder es kann von Ihrem Programm eine vollständige Eingabe durch den Benutzer simuliert werden.

14.3.2.9.1 Application

Eigenschaft: Applicationals Application 372 (schreibgeschützt)

Beschreibung
Ruft das Authentic Desktopapplication-Objekt auf.

Fehler

2300	Das Objekt ist nicht mehr gültig.
2301	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.9.2 CodeGeneratorDlg

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Eigenschaft: CodeGeneratorDlgals CodeGeneratorDlg 446 (schreibgeschützt)

Beschreibung
Ruft eine neue Instanz eines Dialogfeldobjekts für die Codegenerierung ab. Sie benötigen dieses Objekt, um die nötigen Parameter an die Codegenerierungsmethoden zu übergeben. Als Anfangswerte werden die zuletzt bei diesem Dialogfeld verwendeten genommen.

Fehler

2300	Das Dialogs-Objekt oder eines seiner übergeordneten Objekte ist nicht mehr gültig.
2301	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.9.3 FileSelectionDlg

Eigenschaft: FileSelectionDlgals FileSelectionDlg 506 (schreibgeschützt)

Beschreibung
Ruft eine neue Instanz eines Dialogfeldobjekts zur Dateiauswahl auf.

Dialogfeldobjekte zur Dateiauswahl werden mit einigen Events an Sie übergeben, die das Öffnen oder Speichern von Dokumenten und Projekten anzeigen.

Fehler

2300	Das Dialogs-Objekt oder eines seiner übergeordneten Objekte ist nicht mehr gültig.
2301	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.9.4 Parent

Eigenschaft: Parentals Application 372 (schreibgeschützt)

Beschreibung
Ruft das Authentic Desktopapplication-Objekt auf.

Fehler

2300	Das Objekt ist nicht mehr gültig.
2301	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.9.5 SchemaDocumentationDlg

Eigenschaft: SchemaDocumentationDlgals SchemaDocumentationDlg 529 (schreibgeschützt)

Beschreibung
Ruft eine neue Instanz eines Dialogfeldobjekts ab, das die Generierung von Schemadokumentation parametrisiert. Informationen zur Verwendung finden Sei unter Document.GenerateSchemaDocumentation 479.

Fehler

2300 Das Dialogs-Objekt oder eines seiner übergeordneten Objekte ist nicht mehr gültig.
2301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.9.6 GenerateSampleXMLDlg

Eigenschaft: GenerateSampleXMLDlgals GenerateSampleXMLDlg 519 (schreibgeschützt)

Beschreibung
Ruft eine neue Instanz eines Dialogfeldobjekts ab, das die Generierung einer XML-Beispieldatei anhand eines W3C-Schemas oder einer DTD parametrisiert. Informationen zur Verwendung finden Sie unter GenerateSampleXML 478.

Fehler

2300 Das Dialogs-Objekt oder eines seiner übergeordneten Objekte ist nicht mehr gültig.
2301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.9.7 DTDSchemaGeneratorDlg

Eigenschaft: DTDSchemaGeneratorDlgals DTDSchemaGeneratorDlg 497 (schreibgeschützt)

Beschreibung
Ruft eine neue Instanz eines Dialogfeldobjekts ab, das die Generierung eines Schemas oder einer DTD parametrisiert. Informationen zur Verwendung finden Sie unter Document.GenerateDTDOrSchemaEx 478.

Fehler

2300 Das Dialogs-Objekt oder eines seiner übergeordneten Objekte ist nicht mehr gültig.
2301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.9.8 FindInFilesDlg

Eigenschaft: FindInFilesDlgas FindInFilesDlg 508 (schreibgeschützt)

Beschreibung
Ruft eine neue Instanz eines Dialogfeldobjekts ab, das die Suche oder Ersetzung von Stings in Dateien parametrisiert. Informationen zur Verwendung finden Sie unter Application.FindInFiles 379.

© 2016-2022 Altova GmbH

Fehler

2300 Das Dialogs-Objekt oder eines seiner übergeordneten Objekte ist nicht mehr gültig.
2301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.9.9 WSDLDocumentationDlg

Eigenschaft: WSDLDocumentationDlgals WSDLDocumentationDlg 557 (schreibgeschützt)

Beschreibung
Ruft eine neue Instanz eines Dialogfeldobjekts ab, das die Generierung von WSDL-Dokumentation parametrisiert. Informationen zur Verwendung finden Sie unter Document.GenerateWSDLDocumentation 480.

Fehler

2300 Das Dialogs-Objekt oder eines seiner übergeordneten Objekte ist nicht mehr gültig.
2301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.9.10 WSDL20DocumentationDlg

Eigenschaft: WSDL20DocumentationDlgals WSDL20DocumentationDlg 569 (schreibgeschützt)

Beschreibung
Ruft eine neue Instanz eines Dialogfeldobjekts ab, das die Generierung von WSDL 2.0-Dokumenten parametrisiert. Informationen zur Verwendung finden Sie unter Document.GenerateWSDL20Documentation 479.

Fehler

2300 Das Dialogs-Objekt oder eines seiner übergeordneten Objekte ist nicht mehr gültig.
2301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.9.11 XBRLDocumentationDlg

Eigenschaft: XBRLDocumentationDlg als XBRLDocumentationDlg 581 (schreibgeschützt)

Beschreibung
Ruft eine neue Instanz eines Dialogfeldobjekts ab, das die Generierung von XBRL-Dokumenten parametrisiert. Informationen zur Verwendung finden Sie unter Document.GenerateXBRLDocumentation 479.

Fehler

2300 Das Dialogs-Objekt oder eines seiner übergeordneten Objekte ist nicht mehr gültig.
2301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.10 Document

Eigenschaften und Methoden

Standard-Automation-Eigenschaften
Application 466
Parent 485

Verschiedene Dokumenteigenschaften und Methoden
SetActiveDocument 488
Encoding 473
SetEncoding (obsolete) 488
Suggestions 490

XML-Validierung
IsWellFormed 485
IsValid 482
SetExternalIsValid 489

Dokumentkonvertierung und -transformation
AssignDTD 467
AssignSchema 467
AssignXSL 467
AssignXSLFO 467
ConvertDTDOrSchema 469
ConvertDTDOrSchemaEx 470
GenerateDTDOrSchema 477
GenerateDTDOrSchemaEx 478
FlattenDTDOrSchema 476
CreateSchemaDiagram 471
ExecuteXQuery 474
TransformXSL 491
TransformXSLEx 491
TransformXSLFO 491
GenerateProgramCode 478 (nur Enterprise Edition)
GenerateSchemaDocumentation 479
GenerateSampleXML 478
ConvertToWSDL20 470

Dokumentexport
GetExportElementList 481
ExportToText 475
ExportToDatabase 474
CreateDBStructureFromXMLSchema 471
GetDBStructureList 481

Dateispeicherung und - benennung
FullName 477
Name 485
Path 486

 © 2016-2022 Altova GmbH

GetPathName (obsolete) 481
SetPathName (obsolete) 489
Title 491
IsModified 482
Saved 487
SaveAs 487
Save 486
SaveInString 487
SaveToURL 488
Close 469

Aufruf einer Ansicht
CurrentViewMode 472
SwitchViewMode 490
AuthenticView 468
GridView 482
DocEditView (obsolete) 472

Zugriff auf XMLData
RootElement 486
DataRoot 472
CreateChild 471
UpdateViews 492
StartChanges 490
EndChanges 473
UpdateXMLData 492

Beschreibung
Document-Objekte repräsentieren in Authentic Desktop geöffnete XML-Dokumente.

Verwenden Sie eine der folgenden Eigenschaften, um Dokumente aufzurufen, die bereits in Authentic Desktop geöffnet sind:
Application.ActiveDocument 376
Application.Documents 379

Verwenden Sie eine der folgenden Methoden, um ein neues Dokument in Authentic Desktop zu öffnen:
Documents.OpenFile 495
Documents.OpenURL 496
Documents.OpenURLDialog 496
Documents.NewFile 494
Documents.NewFileFromText 495
SpyProjectItem.Open 545
Application.ImportFromDatabase 383
Application.ImportFromSchema 384
Application.ImportFromText 385
Application.ImportFromWord 386
Document.ConvertDTDOrSchema 469
Document.GenerateDTDOrSchema 477

14.3.2.10.1 Events

14.3.2.10.1.1 OnBeforeSaveDocument

Event: OnBeforeSaveDocument(*objDocument* als Document[462], *objDialog* als FileSelectionDlg[506])

Skripting-Umgebung - VBScript:

```
Function On_BeforeSaveDocument(objDocument, objDialog)
End Function

' old handler - now obsolete
' return string to save to new file name
' return empty string to cancel save operation
' return nothing to save to original name
Function On_SaveDocument(objDocument, strFilePath)
End Function
```

Skripting-Umgebung - JScript:

```
function On_BeforeSaveDocument(objDocument, objDialog)
{
}

// old handler - now obsolete
// return string to save to new file name
// return empty string to cancel save operation
// return nothing to save to original name
function On_SaveDocument(objDocument, strFilePath)
{
}
```

IDE Plugin:

IXMLSpyPlugIn.OnEvent(27, ...) // nEventId = 27

Beschreibung

Dieses Event wird bei jedem Versuch, ein Dokument zu speichern ausgelöst. Das Dialogfeldobjekt zur Dateiauswahl wird mit dem für die Dokumentdatei gewählten Namen initialisiert. Sie können diese Auswahl ändern. Um mit dem Speichern des Dokuments fortzufahren, belassen Sie die Eigenschaft FileSelectionDlg.DialogAction[507] von *io_objDialog* auf ihrem Standardwert spyDialogOK[607]. Um das Speichern des Dokuments abzubrechen, setzen Sie diese Eigenschaft auf spyDialogCancel[607].

14.3.2.10.1.2 OnBeforeCloseDocument

Event: OnBeforeCloseDocument(*objDocument* als Document[462])als Boolean

Skripting-Umgebung - VBScript:

```
Function On_BeforeCloseDocument(objDocument)
        ' On_BeforeCloseDocument = False  ' to prohibit closing of document
End Function
```


© 2016-2022 Altova GmbH

Skripting-Umgebung - JScript:

```
function On_BeforeCloseDocument(objDocument)
{
        // return false;  /* to prohibit closing of document */
}
```

IDE Plugin:

IXMLSpyPlugIn.OnEvent(28, ...) // nEventId = 28

Beschreibung

Dieses Event wird bei jedem Versuch ein Dokument zu schließen ausgelöst. Um das Schließen des Dokuments zu verhindern, muss der Wert "false" zurückgegeben. werden.

14.3.2.10.1.3 OnBeforeValidate

Event: OnBeforeValidate(*objDocument* als Document 462, *bOnLoading* als Boolean, *bOnCommand* als Boolean)als Boolean

Skripting-Umgebung - VBScript:

```
Function On_BeforeValidate(objDocument, bOnLoading, bOnCommand)
        On_BeforeValidate = bCancelDefaultValidation     'set by the script if necessary
End Function
```

Skripting-Umgebung - JScript:

```
function On_BeforeValidate(objDocument, bOnLoading, bOnCommand)
{
        returnbCancelDefaultValidation    //set by the script if necessary
}
```

IDE Plugin:

IXMLSpyPlugIn.OnEvent(32, ...) // nEventId = 32

Beschreibung

Dieses Event wird ausgelöst, bevor das Dokument validiert wird. Die Standardvalidierung kann durch Rückgabe von "false" vom Event Handler unterdrückt werden. In diesem Fall sollte das Script auch mit Hilfe der Methode SetExternalIsValid 489 das Validierungsergebnis definieren.

bOnLoadingist "true", wenn das Event bei der Anfangsvalidierung beim Laden des Dokuments ausgelöst wird.

bOnCommandist "true", wenn der Benutzer in der Symbolleiste oder im Menü den Befehl "Validieren" ausgewählt hat.

Verfügbar mit Typbibliothek Version 1.5

14.3.2.10.1.4 OnCloseDocument

Event: OnCloseDocument(*objDocument* als Document 462)

Skripting-Umgebung - VBScript:
Function On_Close Document(*objDocument*)
End Function

Skripting-Umgebung - JScript:
function On_Close Document(*objDocument*)
{
}

IDE Plugin:
IXMLSpyPlugIn.OnEvent(8, ...) // nEventId = 8

Beschreibung
Dieses Event wird ausgelöst, wenn ein Dokument geschlossen wird. Ändern Sie das Dokument nicht von diesem Event aus.

14.3.2.10.1.5 OnViewActivation

Event: OnViewActivation(*objDocument* als Document[462], *eViewMode* als SPYViewModes[616], *bActivated* als Boolean)

Skripting-Umgebung - VBScript:
Function On_ViewActivation(*objDocument*, *eViewMode*, *bActivated*)
End Function

Skripting-Umgebung - JScript:
function On_ViewActivation(*objDocument*, *eViewMode*, *bActivated*)
{
}

IDE Plugin:
IXMLSpyPlugIn.OnEvent(29, ...) // nEventId = 29

Beschreibung
Dieses Event wird immer dann ausgelöst, wenn eine Ansicht eines Dokuments eingeblendet (d.h. zur aktiven Ansicht wird) oder ausgeblendet wird (d.h. eine andere Ansicht wird aktiv oder das Dokument wird geschlossen). Das erste Ansichtsaktivierungsevent nach dem Öffnen eines Dokuments kann jedoch nicht empfangen werden, da es kein document-Objekt gibt, von dem das Event abgerufen werden kann. Verwenden Sie stattdessen das Event Application.OnDocumentOpened[375].

14.3.2.10.2 Application

Eigenschaft: Applicationals Application[372](schreibgeschützt)

Beschreibung
Ruft das Authentic Desktopapplication-Objekt auf.

Fehler
1400 Das Objekt ist nicht mehr gültig.

 © 2016-2022 Altova GmbH

1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.10.3 AssignDTD

Methode: AssignDTD(*strDTDFile*als String, *bDialog* als Boolean)

Beschreibung
Die Methode platziert eine Referenz auf die DTD-Datei "strDTDFile" in das Dokument. Beachten Sie, dass kein Fehler auftritt, wenn die Datei nicht existiert oder kein Zugriff darauf besteht. Wenn bDialog "true" ist, zeigt Authentic Desktop ein Dialogfeld zum Definieren der Datei an.

Fehler
1400 Das Objekt ist nicht mehr gültig.
1409 Dem Dokument darf keine DTD zugewiesen werden.

14.3.2.10.4 AssignSchema

Methode: AssignSchema (*strSchemaFile* als String, *bDialog* als Boolean)

Beschreibung
Die Methode platziert eine Referenz auf die Schemadatei "strSchemaFile" in das Dokument. Beachten Sie, dass kein Fehler auftritt, wenn die Datei nicht existiert oder kein Zugriff darauf besteht. Wenn bDialog "true" ist, zeigt Authentic Desktop ein Dialogfeld zum Definieren der Datei an.

Fehler
1400 Das Objekt ist nicht mehr gültig.
1409 Dem Dokument darf keine Schemadatei zugewiesen werden.

14.3.2.10.5 AssignXSL

Methode: AssignXSL (*strXSLFile* als String, *bDialog* als Boolean)

Beschreibung
Die Methode platziert eine Referenz auf die XLS-Datei "strXSLFile" in das Dokument. Beachten Sie, dass kein Fehler auftritt, wenn die Datei nicht existiert oder kein Zugriff darauf besteht. Wenn bDialog "true" ist, zeigt Authentic Desktop ein Dialogfeld zum Definieren der Datei an.

Fehler
1400 Das Objekt ist nicht mehr gültig.
1409 Dem Dokument darf keine XSL-Datei zugewiesen werden

14.3.2.10.6 AssignXSLFO

Methode: AssignXSLFO (*strXSLFOFile* als String, *bDialog* als Boolean)

Beschreibung
Die Methode platziert eine Referenz auf die XSLFO-Datei "strXSLFile" in das Dokument. Beachten Sie, dass kein Fehler auftritt, wenn die Datei nicht existiert oder kein Zugriff darauf besteht. Wenn bDialog "true" ist, zeigt Authentic Desktop ein Dialogfeld zum Definieren der Datei an.

Fehler
1400 Das Objekt ist nicht mehr gültig.
1409 Dem Dokument darf keine XSL-Datei zugewiesen werden

14.3.2.10.7 AsXMLString

Eigenschaft: AsXMLStringals String

Beschreibung
Diese Eigenschaft dient zum Abrufen oder Definieren des Inhalts des Dokuments.

Fehler
1400 Das document-Objekt ist nicht mehr gültig.
1404 Das XMLData-Objekt kann nicht erstellt werden.
1407 Der Ansichtsmodus kann nicht gewechselt werden.

14.3.2.10.8 AuthenticView

Methode: AuthenticView als AuthenticView 428 (schreibgeschützt)

Beschreibung
Gibt ein Objekt zurück, über das Sie Zugriff auf die Eigenschaften und Methoden haben, die für die Authentic-Ansicht spezifisch sind. Das zurückgegebene Objekt ist nur gültig, wenn das aktuelle Dokument im Authentic-Ansichtsmodus geöffnet wird. Die Lebenszeit eines Objekts endet mit dem nächsten Ansichtswechsel. Jeder Versuch, Objekte oder eines ihrer Children danach aufrufen, hat einen Fehler zur Folge, der angibt, dass das Objekt ungültig ist.

Fehler
1400 Das Objekt ist nicht mehr gültig.
1417 Das Dokument muss im Authentic-Ansichtsmodus geöffnet sein.

Beispiele
```
' ----------------------------------------
' Scripting environment - VBScript
' secure access to authentic view object
' ----------------------------------------
DimobjDocument
SetobjDocument = Application.ActiveDocument
If(NotobjDocument Is Nothing) Then
	' we have an active document, now check for view mode
	If(objDocument.CurrentViewMode <> spyViewAuthentic) Then
		If(NotobjDocument.SwitchViewMode (spyViewAuthentic)) Then
			MsgBox "Active document does not support authentic view mode"
		Else
			' now it is safe to access the authentic view object
```

 © 2016-2022 Altova GmbH

```
                    DimobjAuthenticView
                    SetobjAuthenticView = objDocument.AuthenticView
                    ' now use the authentic view object

            EndIf
        EndIf
Else
        MsgBox "No document is open"
EndIf
```

14.3.2.10.9 Close

Methode: Close (*bDiscardChanges* als Boolean)

Beschreibung
Rufen Sie diese Methode auf, um ein Dokument zu schließen. Wenn bDiscardChanges"true" ist und das Dokument geändert wurde, wird das Dokument geschlossen aber nicht gespeichert.

Fehler
1400 Das Objekt ist nicht mehr gültig.
1401 Das Dokument muss zuerst gespeichert werden.

14.3.2.10.10 ConvertDTDOrSchema

Methode: ConvertDTDOrSchema (*nFormat* als SPYDTDSchemaFormat 608, *nFrequentElements* als SPYFrequentElements 608)

Parameter
nFormat
Setzt das Schemaausgabeformat auf DTD oder W3C.

nFrequentElements
Erstellt komplexe Elemente als Elemente oder complexTypes.

Beschreibung
ConvertDTDOrSchemanimmt das vorhandene Schemaformat und konvertiert es in ein anderes Format. Zur Feinabstimmung einer DTD/XSD-Konvertierung verwenden Sie ConvertDTDOrSchemaEx 470.

Fehler
1400 Das Objekt ist nicht mehr gültig.
1412 Fehler bei der Konvertierung. Im Fall einer Konvertierung einer DTD in eine DTD oder einer XSD- in eine XSD-Datei, werden die folgenden Fehler zurückgegeben: *Die Konvertierung einer DTD in eine andere DTD wird nicht unterstützt. Verwenden Sie stattdessen die Funktion "DTD oder Schema nivellieren"!* und *Die Konvertierung eines Schemas in ein anderes Schema wird nicht unterstützt. Verwenden Sie stattdessen die Funktion "DTD oder Schema nivellieren"!*

14.3.2.10.11 ConvertDTDOrSchemaEx

Methode: ConvertDTDOrSchemaEx (*nFormat* als SPYDTDSchemaFormat[608], *nFrequentElements* als SPYFrequentElements[608], sOutputPath als String, nOutputPathDialogAction als SPYDialogAction[607])

Parameters
nFormat
Setzt das Schemaausgabeformat auf DTD oder W3C.

nFrequentElements
Erstellt komplexe Elemente als Elemente oder complexTypes.

sOutputPath
Der Dateipfad für die neu generierte Datei.

nOutputPathDialogAction
Definiert die Dialogfeldinteraktion für diesen Aufruf.

Beschreibung
ConvertDTDOrSchemaExnimmt ein vorhandenes Schemaformat und konvertiert es in ein anderes Format.

Fehler

1400	Das Objekt ist nicht mehr gültig.
1412	Fehler bei der Konvertierung. Im Fall einer Konvertierung einer DTD in eine DTD oder einer XSD- in eine XSD-Datei, werden die folgenden Fehler zurückgegeben: *Die Konvertierung einer DTD in eine andere DTD wird nicht unterstützt. Verwenden Sie stattdessen die Funktion "DTD oder Schema nivellieren"!* und *Die Konvertierung eines Schemas in ein anderes Schema wird nicht unterstützt. Verwenden Sie stattdessen die Funktion "DTD oder Schema nivellieren"!*

14.3.2.10.12 ConvertToWSDL20

Methode: ConvertToWSDL20 (*sFilePath* als String, *bShowDialogs* als Boolean)

Parameter
sFilePath
Dieser Parameter definiert den Dateinamen der konvertierten WSDL-Datei. Falls auch Include-Dateien aus WSDL-Quelldateien konvertiert werden müssen, wird nur der Verzeichnisteil des angegebenen Pfads verwendet und die Dateinamen werden automatisch generiert.

bShowDialogs
Definiert, ob Dialogfelder zur Auswahl von Dateien/Ordnern angezeigt werden sollen.

Beschreibung
Konvertiert das WSDL 1.1-Dokument in eine WSDL 2.0-Datei. Konvertiert auch alle referenzierten WSDL-Dateien, die von diesem Dokument aus referenziert werden. Beachten Sie, dass diese Funktionalität auf die WSDL-Ansicht beschränkt ist. SieheDocument.CurrentViewMode[472]. und SPYViewModes[616].

Fehler

 © 2016-2022 Altova GmbH

1400 Das Dokument-Objekt ist nicht mehr gültig.
1407 Es wurden ungültige Parameter übergeben oder ein leerer Dateiname wurde als Ausgabeziel definiert.
1417 Das Dokument ist nicht in der WSDL-Ansicht geöffnet. Vielleicht ist es keine '.wsdl' Datei.
1421 Funktion steht in dieser Edition nicht zur Verfügung.
1433 Konvertierung von WSDL 1.1 in WSDL 2.0 ist fehlgeschlagen.

14.3.2.10.13 CreateChild

Methode: CreateChild (*nKind* als SPYXMLDataKind[618]) als XMLData[593]

Rückgabewert
Die Methode gibt das neue XMLDataObjekt zurück.

Beschreibung
Um ein neues XMLDataObjekt zu erstellen, verwenden Sie die Methdoe CreateChild().

Fehler
1400 Das Objekt ist nicht mehr gültig.
1404 Das XMLData-Objekt kann nicht erstellt werden.
1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.10.14 CreateDBStructureFromXMLSchema

Methode: CreateDBStructureFromXMLSchema (*pDatabase* als DatabaseConnection[452], *pTables* als ElementList[501], *bDropTableWithExistingName* als Boolean) als String

Beschreibung
CreateDBStructureFromXMLSchemaexportiert die angegebenen Tabellen in die angegebene Datenbank.Die Funktion gibt die SQL-Anweisungen zurück, die zur Durchführung der Änderungen erforderlich waren.

Siehe auch GetDBStructureList[481].

Fehler
1429 Datenbankauswahl fehlt.
1430 Der Dokumentexport ist fehlgeschlagen.

14.3.2.10.15 CreateSchemaDiagram

Method: CreateSchemaDiagram (*nKind* as SPYSchemaDefKind[613], *strName* as String, *strFile* as String)

Rückgabewert

Keiner.

Beschreibung
Die Methode erstellt ein Diagramm des Schematyps strName von der Art nKind und speichert die Ausgabedatei in strFile. Beachten Sie, dass diese Funktionalität auf die Schema-Ansicht beschränkt ist. Siehe Document.CurrentViewMode 472. und SPYViewModes 616.

Fehler
1400 Das Objekt ist nicht mehr gültig.
1414 Das Diagramm konnte nicht gespeichert werden.
1415 Es wurde ein ungültiger Schemadefinitionstyp angegeben.

14.3.2.10.16 CurrentViewMode

Methode: CurrentViewMode als SPYViewModes 616

Beschreibung
Die Eigenschaft enthält den aktuellen Ansichtsmodus des Dokuments. Siehe auch Document.SwitchViewMode 490.

Fehler
1400 Das Objekt ist nicht mehr gültig.
1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.10.17 DataRoot

Eigenschaft: DataRoot als XMLData 593 (schreibgeschützt)

Beschreibung
Über diese Methode haben Sie Zugriff auf das erste XMLData-Objekt des Dokuments vom Typ *spyXMLDataElement*. Dies ist normalerweise das Root-Element für alle Dokumentinhaltsdaten. Eine Beschreibung, wie Sie das Root-Element des gesamten Dokuments inkl. XML-Prologdaten abrufen, finden Sie unter XMLSpyDocument.RootElement 486. Wenn CurrentViewMode 472 nicht *spyViewGrid* oder *spyViewAuthentic* ist, ist möglicherweise ein UpdateXMLData 492 erforderlich, um Zugriff auf die neuesten XMLData 593 zu erhalten.

Fehler
1400 Das document-Objekt ist nicht mehr gültig.
1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.10.18 DocEditView

Methode: DocEditView als DocEditView

Beschreibung
Enthält eine Referenz auf das aktuelle Authentic ViewObjekt.

Fehler

 © 2016-2022 Altova GmbH

1400 Das Objekt ist nicht mehr gültig.
1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
1417 Das Dokument muss im Authentic-Ansichtsmodus geöffnet sein.

14.3.2.10.19 Encoding

Eigenschaft: Encoding als String

Beschreibung
Über diese Eigenschaft haben Sie Zugriff auf den Encoding-Wert des Dokuments. Diese Eigenschaft kann nur aufgerufen werden, wenn das Dokument in *spyViewGrid*, *spyViewText* oder *spyViewAuthentic* geöffnet ist. Um den aktuellen Ansichtsmodus eines Dokuments abzurufen, siehe Beschreibung CurrentViewMode 472.

Mit dieser Eigenschaft wird die Methode SetEncoding 488 nicht mehr benötigt.

Mögliche Werte sind zum Beispiel:

8859-1,
8859-2,
ASCII, ISO-646,
850,
1252,
1255,
SHIFT-JIS, MS-KANJI,
BIG5, FIVE,
UTF-7,
UTF-8,
UTF-16

Fehler
1400 Das document-Objekt ist nicht mehr gültig.
1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
1416 Die Operation wird im aktuellen Ansichtsmodus nicht unterstützt.

14.3.2.10.20 EndChanges

Methode: EndChanges()

Beschreibung
Verwenden Sie die Methode EndChangesum alle Änderungen seit dem letzten Aufruf von Document.StartChanges 490 anzuzeigen.

Fehler
1400 Das Objekt ist nicht mehr gültig.

14.3.2.10.21 ExecuteXQuery

Methode: ExecuteXQuery (*strXMLFileName* als String)

Beschreibung
Führt die im Dokument des Document-Objekts enthaltenen XQuery-Anweisungen aus. Je nachdem, welche Dateierweiterung das Dokument hat, wird entweder eine XQuery oder ein XQuery Update ausgeführt. Verwendet die im Argument definierte XML-Datei als XML-Zieldokument, das das XQuery-Dokument verarbeitet.

- Wenn das Dokument eine der XQuery-Dateierweiterungen hat, die im Dialogfeld "Optionen" von Authentic Desktopdefiniert sind, so wird eine XQuery-Ausführung durchgeführt. Standardmäßig sind in Authentic Desktop`.xq`, `.xql`und `.xquery`als XQuery-Dateierweiterungen definiert.
- Wenn das Dokument eine XQuery Update-Dateierweiterung hat, die im Dialogfeld "Optionen" von Authentic Desktopdefiniert ist, so wird eine XQuery Update-Aktion durchgeführt. Standardmäßig ist in Authentic Desktop`.xqu`als XQuery Update-Dateierweiterung definiert.

Wenn Ihr XQuery-Script keine XML-Quelle verwendet, setzen Sie den Parameter strXMLFileNameauf einen leeren String.

Fehler

1400	Das document-Objekt ist nicht mehr gültig.
1423	XQuery-Transformationsfehler.
1424	Es konnten nicht alle für die Operation benötigten Dateien geladen werden. Höchstwahrscheinlich existiert die in strXMLFileNamedefinierte Datei nicht oder ist ungültig.

14.3.2.10.22 ExportToDatabase

Methode: ExportToDatabase (*pFromChild* als XMLData[593], *pExportSettings* als ExportSettings[503], *pDatabase* als DatabaseConnection[452])

Beschreibung
ExportToDatabaseexportiert das XML-Dokument beginnend mit dem Element pFromChild. Der Parameter pExportSettingsdefiniert das Exportverhalten (siehe Application.GetExportSettings[381]). Der Parameter pDatabasedefiniert das Ziel des Exports (siehe Application.GetDatabaseSettings[380]). `UpdateXMLData()`[492] wird eventuell indirekt benötigt, da Sie `XMLData`[593]als Parameter an diese Funktion übergeben müssen.

Fehler

1400	Das Objekt ist nicht mehr gültig.
1407	Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
1416	Fehler beim Export.
1429	Datenbankauswahl fehlt.
1430	Dokumentexport fehlgeschlagen.

Beispiel

```
DimobjDoc AsDocument
```

 © 2016-2022 Altova GmbH

```
SetobjDoc = objSpy.ActiveDocument

'set the behaviour of the export with ExportSettings
DimobjExpSettings AsExportSettings
SetobjExpSettings = objSpy.GetExportSettings

'set the destination with DatabaseConnection
DimobjDB AsDatabaseConnection
SetobjDB = objSpy.GetDatabaseSettings

objDB.CreateMissingTables = True
objDB.CreateNew = True
objDB.File = "C:\Export.mdb"

objDoc.ExportToDatabase objDoc.RootElement, objExpSettings, objDB
IfErr.Number <> 0 Then
        a = MsgBox("Error: " & (Err.Number - vbObjectError) & Chr(13) &
                "Description: " & Err.Description)
EndIf
```

14.3.2.10.23 ExportToText

Methode: ExportToText (*pFromChild* als XMLData[593], *pExportSettings* als ExportSettings[503], *pTextSettings* als TextImportExportSettings[548])

Beschreibung

ExportToTextexportiert Tabellendaten aus dem Dokument beginnend mit pFromChildin eine oder mehrere Textdateien. Die Spalten der erzeugten Tabellen werden in alphabetischer Reihenfolge nach den Namen der Tabellenspalten generiert. Verwenden Sie GetExportElementList[481],um zu sehen, welche Daten exportiert werden. Der Parameter pExportSettingsdefiniert die näheren Einzelheiten für den Export. Setzen Sie die Eigenschaft ExportSettings.ElementList[504]auf die - möglicherweise geänderte - Liste, die von GetExportElementList[481]zurückgegeben wird, damit nicht alle enthaltenen Tabellen exportiert werden. Der Parameter pTextSettingsdefiniert die Optionen für den Textexport und -import. Sie müssen die Eigenschaft TextImportExportSettings.DestinationFolder[549]setzen, bevor Sie ExportToText aufrufen.`UpdateXMLData()`[492] wird eventuell indirekt benötigt, da Sie `XMLData`[593]als Parameter an diese Funktion übergeben müssen.

Fehler

1400	Das Objekt ist nicht mehr gültig.
1407	Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
1416	Fehler beim Export..
1430	Dokumentexport fehlgeschlagen.

Beispiel

```
' ---------------------------------------------------------
' VBA client code fragment - export document to text files
' ---------------------------------------------------------
        Dim objDoc As Document
        Set objDoc = objSpy.ActiveDocument
```

```
Dim objExpSettings As ExportSettings
Set objExpSettings = objSpy.GetExportSettings
objExpSettings.ElementList = objDoc.GetExportElementList(
                                                    objDoc.RootElement,
                                                    objExpSettings)

Dim objTextExp As TextImportExportSettings
Set objTextExp = objSpy.GetTextImportExportSettings
objTextExp.HeaderRow = True
objTextExp.DestinationFolder = "C:\Exports"

On Error Resume Next
objDoc.ExportToText objDoc.RootElement, objExpSettings, objTextExp

If Err.Number <> 0 Then
        a = MsgBox("Error: " & (Err.Number - vbObjectError) & Chr(13) & "Description: "
        & Err.Description)
End If
```

14.3.2.10.24 FlattenDTDOrSchema

Methode: FlattenDTDOrSchema (sOutputPath als String, nOutputPathDialogAction als SPYDialogAction[607])

Parameter

`sOutputPath`
Der Dateipfad der neu generierten Datei.

`nOutputPathDialogAction`
Definiert die Dialogfeldinteraktion für diesen Aufruf.

Beschreibung

`FlattenDTDOrSchema`generiert anhand einer vorhandenen DTD/eines vorhandenen Schemas eine nivellierte, flache Datei und speichert die generierte Datei unter dem angegebenen Pfad. Bei DTDs werden bei der Nivellierung Parameter Entities entfernt und anhand einer Sammlung von Modulen wird eine einzige DTD erzeugt; Mit `IGNORE` markierte Abschnitte werden unterdrückt und nicht verwendete Parameter Entities werden gelöscht. Bei Nivellierung eines XML-Schemas (i) werden die Komponenten aller inkludierten Schemas als globale Komponenten des aktiven Schemas hinzugefügt (ii) und die inkludierten Schemas gelöscht.

Errors

1400 Das Objekt ist nicht mehr gültig.
1412 Fehler bei der Konvertierung.

© 2016-2022 Altova GmbH

14.3.2.10.25 FullName

Eigenschaft: FullName als String

Beschreibung
Mit Hilfe dieser Eigenschaft kann der vollständige Dateiname - einschließlich des Pfads - unter dem das Dokument gespeichert wird, abgerufen oder definiert werden. Die Gültigkeit des Namens wird vor dem nächsten Speichervorgang nicht überprüft.

Diese Eigenschaft macht die Methoden GetPathName[481] und SetPathName[489] überflüssig.

Fehler
1400 Das document-Objekt ist nicht mehr gültig.
1402 Es wurde ein leerer String als vollständiger Dateiname definiert.

14.3.2.10.26 GenerateDTDOrSchema

Methode: GenerateDTDOrSchema (*nFormat* als SPYDTDSchemaFormat[608], *nValuesList* als integer, *nDetection* als SPYTypeDetection[615], *nFrequentElements* als SPYFrequentElements[608])

Parameter
nFormat
Setzt das Schemaausgabeformat auf DTD oder W3C.

nValuesList
Generiert nur diese Anzahl an enumerations-facets pro Typ. Setzen Sie den Wert auf -1 um die Anzahl nicht zu begrenzen.

nDetection
Definiert die Genauigkeit der Ermittlung eines simpleType.

nFrequentElements
Sollen die Typen für alle Elemente als global definiert werdenl? Verwenden Sie diesen Wert *spyGlobalComplexType* um sie als globale Elemente zu definiert. Verwenden Sie andernfalls den Wert *spyGlobalElements*.

Beschreibung
Mit Hilfe dieser Methode können Sie automatisch eine DTD oder ein Schema für das aktuelle XML-Dokument generieren.
Informationen zur Feinabstimmung der DTD / Schema-Generierung finden Sie unter GenerateDTDOrSchemaEx[478].
Beachten Sie, dass diese Funktionalität nicht nur in der ZIP-Ansicht verfügbar ist. Siehe Document.CurrentViewMode[472]. und SPYViewModes[616].

Fehler
1400 Das Objekt ist nicht mehr gültig.
1407 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.

14.3.2.10.27 GenerateDTDOrSchemaEx

Methode: GenerateDTDOrSchemaEx (*objDlg* als DTDSchemaGeneratorDlg[497]) als Document[462]

Beschreibung
Verwenden Sie diese Methode, um automatisch eine DTD oder ein Schema für das aktuelle XML-Dokument zu generieren. Mit Hilfe eines DTDSchemaGeneratorDlg[497] Objekts werden die Informationen an den Schema-/DTD-Generator übergeben. Der Generierungsvorgang kann so konfiguriert werden, dass der Benutzer entweder Eingaben machen kann oder dass der Prozess ohne weitere Eingaben durch den Benutzer ausgeführt wird.
Beachten Sie, dass diese Funktionalität nicht nur in der ZIP-Ansicht verfügbar ist. Siehe Document.CurrentViewMode[472]. und SPYViewModes[616].

Fehler

1400	Das Objekt ist nicht mehr gültig.
1407	Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.

14.3.2.10.28 GenerateProgramCode

Methode: GenerateProgramCode (*objDlg* als CodeGeneratorDlg[446])
Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Beschreibung
Generiert Java, C++ oder C#-Klassendateien anhand der XML-Schemadefinitionen in Ihrem Dokument. Die Daten werden mit Hilfe eines CodeGeneratorDlg[446] Objekts an den Code Generator übergeben. Der Generierungsvorgang kann so konfiguriert werden, dass der Benutzer entweder Eingaben machen kann oder dass der Prozess ohne weitere Eingaben durch den Benutzer ausgeführt wird.

Fehler

1400	Das document-Objekt ist nicht mehr gültig.
1407	Es wurde ein leerer Dateiname angegeben.
1421	Diese Funktion steht in dieser Edition nicht zur Verfügung.

14.3.2.10.29 GenerateSampleXML

Methode: GenerateSampleXML (*objDlg* als GenerateSampleXMLDlg[519]) als Document[462]

Beschreibung
Generiert eine XML-Beispieldatei, wenn das Dokument ein Schema oder eine DTD ist. Verwenden Sie Dialogs.GenerateSampleXMLDlg[460] um eine initialisierte Gruppe von Optionen aufzurufen.

Verfügbar mit Typbibliothek Version 1.5

Fehler

 © 2016-2022 Altova GmbH

1400 Das document-Objekt ist nicht mehr gültig.

14.3.2.10.30 GenerateSchemaDocumentation

Methode: GenerateSchemaDocumentation (*objDlg* als SchemaDocumentationDlg 529)

Beschreibung
Generiert Dokumentation für eine Schemadefinitionsdatei im HTML-, MS-Word- oder RTF-Format. Der Parameter "objDlg" dient zum Parametrisieren des Generierungsvorgangs. Verwenden Sie Dialogs.SchemaDocumentationDlg 460 um eine initialisierte Gruppe von Optionen aufzurufen. Sie müssen zumindest die Eigenschaft SchemaDocumentationDlg.OutputFile 536 definieren, bevor Sie mit der Generierung beginnen können. Beachten Sie, dass diese Funktion nur für die Schema-Ansicht zur Verfügung steht. Siehe Document.CurrentViewMode 472 und SPYViewModes 616.

Fehler
1400 Das document-Objekt ist nicht mehr gültig.
1407 Es wurden entweder ungültige Parameter übergeben oder es wurde ein leere Dateiname für die Ausgabe definiert.
1417 Das Dokument ist nicht in der Schemaansicht geöffnet. Vielleicht handelt es sich um keine '.xsd' Datei.
1421 Die Funktion steht in dieser Edition nicht zur Verfügung.
1422 Fehler bei der Generierung

14.3.2.10.31 GenerateWSDL20Documentation

Methode: GenerateWSDL2Documentation (*objDlg* als WSDL20DocumentationDlg 569)

Beschreibung
Generiert Dokumentation für eine WSDL-Definitionsdatei im HTML-, MS-Word- oder RTF-Format. Der Parameter "objDlg" dient zum Parametrisieren des Generierungsvorgangs. Verwenden Sie Dialogs.WSDL20DocumentationDlg 461 um eine initialisierte Gruppe von Optionen aufzurufen. Sie müssen zumindest die Eigenschaft WSDL20DocumentationDlg.OutputFile 575 definieren, bevor Sie mit der Generierung beginnen können. Beachten Sie, dass diese Funktionalität nur für die WSDL-Ansicht zur Verfügung steht.Siehe Document.CurrentViewMode 472 und SPYViewModes 616.

Fehler
1400 Das document-Objekt ist nicht mehr gültig.
1407 Es wurden entweder ungültige Parameter übergeben oder es wurde ein leerer Dateiname für die Ausgabe definiert.
1417 Das Dokument ist nicht in der Schemaansicht geöffnet. Vielleicht handelt es sich um keine '.xsd' Datei.
1421 Die Funktion steht in dieser Edition nicht zur Verfügung.
1422 Fehler bei der Generierung

14.3.2.10.32 GenerateWSDLDocumentation

Methode: GenerateWSDLDocumentation (*objDlg* als WSDLDocumentationDlg[557])

Beschreibung
Generiert Dokumentation für eine WSDL-Definitionsdatei im HTML-, MS-Word- oder RTF-Format. Der Parameter "objDlg" dient zum Parametrisieren des Generierungsvorgangs. Verwenden Sie Dialogs.SchemaDocumentationDlg[460] um eine initialisierte Gruppe von Optionen aufzurufen. Sie müssen zumindest die Eigenschaft SchemaDocumentationDlg.OutputFile[536] definieren, bevor Sie mit der Generierung beginnen können. Beachten Sie, dass diese Funktionalität nur für die WSDL-Ansicht zur Verfügung steht.Siehe Document.CurrentViewMode[472] und SPYViewModes[616].

Fehler

1400	Das document-Objekt ist nicht mehr gültig.
1407	Es wurden entweder ungültige Parameter übergeben oder es wurde ein leerer Dateiname für die Ausgabe definiert.
1417	Das Dokument ist nicht in der Schemaansicht geöffnet. Vielleicht handelt es sich um keine '.xsd' Datei.
1421	Die Funktion steht in dieser Edition nicht zur Verfügung.
1422	Fehler bei der Generierung

14.3.2.10.33 GenerateXBRLDocumentation

Methode: GenerateXBRLDocumentation (*objDlg* als XBRLDocumentationDlg[581])

Beschreibung
Generiert Dokumentation für eine XBRL-Definitionsdatei im HTML-, MS-Word- oder RTF-Format. Der Parameter "objDlg" dient zum Parametrisieren des Generierungsvorgangs. Verwenden Sie Dialogs.XBRLDocumentationDlg[461] um eine initialisierte Gruppe von Optionen aufzurufen. Sie müssen zumindest die Eigenschaft XBRLDocumentationDlg.OutputFile[587] definieren, bevor Sie mit der Generierung beginnen können. Beachten Sie, dass diese Funktionalität nur für die XBRL-Ansicht zur Verfügung steht.Siehe Document.CurrentViewMode[472] und SPYViewModes[616].

Fehler

1400	Das document-Objekt ist nicht mehr gültig.
1407	Es wurden entweder ungültige Parameter übergeben oder es wurde ein leerer Dateiname für die Ausgabe definiert.
1417	Das Dokument ist nicht in der Schemaansicht geöffnet. Vielleicht handelt es sich um keine '.xsd' Datei.
1421	Die Funktion steht in dieser Edition nicht zur Verfügung.
1422	Fehler bei der Generierung

© 2016-2022 Altova GmbH

14.3.2.10.34 GetDBStructureList

Methode: GetDBStructureList (*pDatabase* als DatabaseConnection[452]) als ElementList[501]

Beschreibung
GetDBStructureListerstellt anhand des Schema-Dokuments eine Sammlung von Elementen, für die Tabellen in der definierten Datenbank erstellt werden. Die Funktion gibt eine Sammlung von ElementListItemszurück, deren ElementListItem.Name[503]Eigenschaften die Namen der Tabellen enthalten.

Siehe auch CreateDBStructureFromXMLSchema[471].

Fehler

1400	Das Objekt ist nicht mehr gültig.
1427	Es konnte kein Parser für die definierte XML-Datei erstellt werden.
1428	Der Export der Elementliste ist fehlgeschlagen.
1429	Datenbankauswahl fehlt.

14.3.2.10.35 GetExportElementList

Methode: GetExportElementList (*pFromChild* als XMLData[593], *pExportSettings* als ExportSettings[503]) als ElementList[501]

Beschreibung
GetExportElementListerstellt eine Elementsammlung für den Export aus dem Dokument. Der Export wird durch die Einstellungen in pExportSettingsdefiniert und beginnt beim Element pFromChild.Die Funktion gibt eine Sammlung vonElementListItemszurück, wobei die Eigenschaften ElementListItem.Name[503]die Namen der Tabellen enthalten, die aus dem Dokument exportiert werden können. Die Eigenschaft ElementListItem.FieldCount[503]enthält die Anzahl der Spalten in der Tabelle. Die Eigenschaft ElementListItem.RecordCount[503]enthält die Anzahl der Datensätze in der Tabelle. Die Eigenschaft ElementListItem.ElementKind[502]wird nicht verwendet. `UpdateXMLData()`[492]wird eventuell indirekt benötigt, da Sie `XMLData`[593]als Parameter an diese Funktion übergeben müssen.

Fehler

1400	Das Objekt ist nicht mehr gültig.
1407	Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
1427	Es konnte kein Parser für die angegebene XML-Datei erstellt werden.
1428	Der Export der Elementliste ist fehlgeschlagen.

14.3.2.10.36 GetPathName (nicht mehr verwendet)

Wird ersetzt durch Document.FullName[477]

// ----- javascript sample -----

```
// instead of:
// strPathName = Application.ActiveDocument.GetPathName();
// use now:
strPathName = Application.ActiveDocument.FullName;
```

Methode: GetPathName() als String

Beschreibung
Die Methode GetPathNameruft den Pfad das aktiven Dokuments auf.

Siehe auch Document.SetPathName 489 (nicht mehr verwendet).

14.3.2.10.37 GridView

Eigenschaft: GridView als GridView 525

Beschreibung
Über diese Eigenschaft erhalten Sie Zugriff auf die Funktionalitäten der Grid-Ansicht für das Dokument.

Fehler
1400 Das Objekt ist nicht mehr gültig.
1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
1417 Das Dokument muss in der Grid-Ansicht geöffnet sein.

14.3.2.10.38 IsModified

Eigenschaft: IsModified als Boolean

Beschreibung
True, wenn das Dokument geändert wurde.

Fehler
1400 Das Objekt ist nicht mehr gültig.
1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.10.39 IsValid

Methode: HRESULT IsValid([in, out] VARIANT *strError, [in, out] VARIANT *nErrorPos, [in, out] VARIANT *pBadData, [out,retval] VARIANT_BOOL *bValid);

Rückgabewert
True, wenn das Dokument gültig ist, false, wenn es nicht gültig ist. Um `IsValid()` aufrufen zu können, muss die Benutzeroberfläche der Applikation sichtbar sein. (Wenn Sie eine Validierung durchführen wollen, ohne dass die Benutzeroberfläche angezeigt wird, verwenden Sie bitte Altova RaptorXML Server.)

Beschreibung
`IsValid` validiert das Dokument anhand des damit verknüpften Schemas bzw. der DTD. `strError` gibt dieselbe Fehlermeldung zurück, wie wenn Sie die Datei über die Benutzeroberfläche validieren.

 © 2016-2022 Altova GmbH

Fehler

1400 Das Objekt ist nicht mehr gültig.
1407 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
1408 Datei kann nicht validiert werden.

Beispiel

Im folgenden C++-Codefragment finden Sie ein Beispiel für die Verwendung der Methode **IsValid**.

```
#import "XMLSpy.tlb"


CComPtr< XMLSpyLib::IDocument12> ipDoc = ipXMLSpy->GetActiveDocument();
if ( ipDoc )
{
	// prepare in/out parameters for IsValid call
	CComVariant variantError;
	CComVariant variantErrorPos;
	CComVariant variantBadData;
	// IsValid always shows a dialog with the validation result. This cannot be turned
off.
	bool bIsValid = ipDoc->IsValid( &variantError, &variantErrorPos, &variantBadData )
== VARIANT_TRUE;

	if ( !bIsValid )
	{
		// retrieve values from out parameters
		CString strError = (V_VT( &variantError ) == VT_BSTR ?
V_BSTR( &variantError ) : _T( "" ));
		long npos = (V_VT( &variantErrorPos ) == VT_I4 ? V_I4( &variantErrorPos ) : -
1);
		CComQIPtr< XMLSpyLib::IXMLData > ipXMLBadData = (V_VT( &variantBadData ) ==
VT_DISPATCH ? V_DISPATCH( &variantBadData ) : nullptr);

		if ( ipXMLBadData )
			strError += CString( _T("\n\n Node: ") ) + (LPCWSTR)ipXMLBadData-
>GetName();

		if ( !strError.IsEmpty() )
			AfxMessageBox( "Validation failed - " + strError );
	}
}
```

14.3.2.10.40 IsValidEx

Methode: IsValidEx (*nXSDVersion* als SPYValidateXSDVersion 616, *nErrorLimit* als `int`, *nErrorFormat* als SPYValidateErrorFormat 616, out *strError* als Variante) als Boolean

Rückgabewert
"True", wenn das Dokument gültig ist, andernfalls "false'.

Beschreibung
`IsValidEx` validiert das Dokument anhand des damit verknüpften Schemas/der damit verknüpften DTD.

In-Parameter:
`nXSDVersion`, ein Enumerationswert von `SPYValidateXSDVersion` 616, womit die XDS-Version ausgewählt wird, anhand der die Validierung durchgeführt werden soll.
`nErrorLimit`, ein Ganzzahlwert. Der Wert muss `0` oder eine positive Ganzzahl sein oder `-1` für `unlimited`.
`nErrorFormat`, ein Enumerationswert von `SPYValidateErrorFormat` 616, womit die XDS-Version ausgewählt wird, anhand der die Validierung durchgeführt werden soll.

Out-Parameter:
`strError` ist die Fehlermeldung und mit der Meldung identisch, die man bei Validierung der Datei über die Benutzeroberfläche erhält.

Fehler

1400	Das Objekt ist nicht mehr gültig.
1407	Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.
1408	Datei kann nicht validiert werden.

Beispiel
Im folgenden C++-Codefragment finden Sie ein Beispiel für die Verwendung der Methode **`IsValidEx`**.

```
#import "XMLSpy.tlb"


CComPtr< XMLSpyLib::IDocument12> ipDoc = ipXMLSpy->GetActiveDocument();
if ( ipDoc )
{
	CComVariant variantErrorEx;
	bool bIsValidEx = ipDoc->IsValidEx( XMLSpyLib::spyValidateXSDVersion_AutoDetect,
100, XMLSpyLib::SPYValidateErrorFormat_LongXML, &variantErrorEx ) == VARIANT_TRUE;

	//	// retrieve values from out parameters
	CString strErrorEx = (V_VT( &variantErrorEx ) == VT_BSTR ?
V_BSTR( &variantErrorEx ) : _T( "" ));
	if ( !strErrorEx.IsEmpty() )
		AfxMessageBox( "Validation failed - " + strErrorEx );
}
```

 © 2016-2022 Altova GmbH

14.3.2.10.41 IsWellFormed

Methode: IsWellFormed (*pData* als XMLData[593], *bWithChildren* als Boolean, *strError* als Variant, *nErrorPos* als Variant, *pBadXMLData* als Variant) als Boolean

Rückgabewert
True, wenn das Dokument wohlgeformt ist.

Beschreibung
IsWellFormedüberprüft das Dokument beginnend beim Element pDataauf Wohlgeformtheit.

Wenn das Dokument nicht wohlgeformt ist, enthält strErroreine Fehlermeldung, nErrorPosdie Position der Datei und pBadXMLDataenthält eine Referenz auf das Element, das die Wohlgeformtheit verletzt. Diese Out-Parameter sind als VARIANTs definiert, um Scripting-Sprachen wie VBScript zu unterstützen.

Fehler
1400 Das Objekt ist nicht mehr gültig.
1407 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben

Beispiel

Siehe IsValid.

14.3.2.10.42 Name

Eigenschaft: Name als String (schreibgeschützt)

Beschreibung
Verwenden Sie diese Eigenschaft, um den Namen - ohne Pfad - der Dokumentdatei abzurufen. Um den Dateinamen für ein Dokument zu ändern, verwenden Sie die Eigenschaft FullName[477].

Fehler
1400 Das document-Objekt ist nicht mehr gültig.
1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.10.43 Parent

Eigenschaft: Parentals Documents[493](schreibgeschützt)

Beschreibung
Ruft das übergeordnete Objekt des Document-Objekts auf.

Fehler
1400 Das document-Objekt ist nicht mehr gültig.
1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Eigenschaft: Parent als Application[372] (schreibgeschützt)

14.3.2.10.44 Path

Eigenschaft: Path als String (schreibgeschützt)

Beschreibung
Verwenden Sie diese Eigenschaft, um den Pfad der Dokumentdatei - ohne Dateinamen - abzurufen. Um den Dateinamen und Pfad für ein Dokument zu ändern, verwenden Sie die Eigenschaft FullName[477].

Fehler

1400	Das document-Objekt ist nicht mehr gültig.
1407	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.10.45 RootElement

Eigenschaft: RootElement als XMLData[593] (schreibgeschützt)

Beschreibung
Über die Eigenschaft RootElementhaben Sie Zugriff auf das Root-Element der XML-Struktur des Dokuments einschließlich der XML-Prologdaten. Um das erste Element des Inhalts eines Dokuments aufzurufen, navigieren Sie zum ersten Child der Art *spyXMLDataElement*oder verwenden Sie die Eigenschaft Document.DataRoot[472]. Wenn CurrentViewMode[472] nicht *spyViewGrid* oder *spyViewAuthentic* ist, ist möglicherweise ein UpdateXMLData[492] erforderlich, um Zugriff auf die neuesten XMLData[593] zu erhalten.

Fehler

1400	Das document-Objekt ist nicht mehr gültig.
1407	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.10.46 Save

Methode: Save()

Beschreibung
Die Methode schreibt alle Änderungen das Dokuments in die dazugehörige Datei. Siehe auch Document.FullName[477].

Fehler

1400	Das document-Objekt ist nicht mehr gültig.
1407	Es wurde ein leerer Dateiname angegeben.
1403	Fehler beim Speichern der Datei. Wahrscheinlich ist der Dateiname ungültig.

© 2016-2022 Altova GmbH

14.3.2.10.47 SaveAs

Methode: SaveAs (*strFileName* als String)

Beschreibung
Speichert das Dokument in der angegebenen Datei.Wenn das Dokument erfolgreich gespeichert werden konnte, wird die Eigenschaft FullName[477] auf den angegebenen Dateinamen gesetzt.

Fehler

1400	Das document-Objekt ist nicht mehr gültig.
1407	Es wurde ein leerer Dateiname angegeben.
1403	Fehler beim Speichern der Datei. Wahrscheinlich ist der Dateiname ungültig.

14.3.2.10.48 Saved

Eigenschaft: Saved als Boolean (schreibgeschützt)

Beschreibung
Mit Hilfe dieser Eigenschaft können Sie überprüfen, ob das Dokument seit der letzten Änderung gespeichert wurde. Sie gibt die Negation von IsModified[482] zurück.

Fehler

1400	Das document-Objekt ist nicht mehr gültig.
1407	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.10.49 SaveInString

Methode: SaveInString (*pData* als XMLData[593], *bMarked* als Boolean) als String

Parameter
pData
XMLData Element als Start. Setzen Sie pData auf Document.RootElement[486], wenn die gesamte Datei kopiert werden soll.

bMarked
Wenn bMarked "true" ist, werden nur die in der Grid-Ansicht ausgewählten Elemente kopiert.

Rückgabewert
Gibt einen String mit den XML-Daten zurück.

Beschreibung
SaveInString beginnt am Element pData und konvertiert die XMLData Objekte in eine String-Darstellung. `UpdateXMLData()`[492] wird eventuell indirekt benötigt, da Sie `XMLData`[593] als Parameter an diese Funktion übergeben müssen.

Fehler

1400	Das Objekt ist nicht mehr gültig.

1407 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.

14.3.2.10.50 SaveToURL

Methode: SaveToURL (*strURL* als String, *strUser* als String, *strPassword* als String)

Rückgabewert

Beschreibung
SaveToURL()schreibt das Dokument in die URL strURL. Diese Methode definiert den permanenten Dateipfad des Dokuments nicht.

Fehler
1400 Das Objekt ist nicht mehr gültig.
1402 Es wurde eine ungültige URL angegeben.
1403 Fehler beim Speichern unter der URL.

14.3.2.10.51 SetActiveDocument

Methode: SetActiveDocument()

Beschreibung
Die Methode definiert das Dokument als das aktive und bringt es in den Vordergrund.

Fehler
1400 Das Objekt ist nicht mehr gültig.

14.3.2.10.52 SetEncoding (nicht mehr verwendet)

Wird ersetzt durch Document.Encoding 473

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.SetEncoding("UTF-16");
// use now:
Application.ActiveDocument.Encoding = "UTF-16";
```

Methode: SetEncoding (*strEncoding* als String)

Beschreibung
SetEncoding definiert die Kodierung des Dokuments wie der Menübefehl "Datei/Kodierung..." in Authentic Desktop. Mögliche Werte für strEncoding sind z.B.:

8859-1,
8859-2,
ASCII, ISO-646,

 © 2016-2022 Altova GmbH

850,
1252,
1255,
SHIFT-JIS, MS-KANJI,
BIG5, FIVE,
UTF-7,
UTF-8,
UTF-16

14.3.2.10.53 SetExternalIsValid

Methode: SetExternalIsValid (*bValid* als Boolean)

Parameter

bValid
Definiert das Ergebnis eines externen Validierungsvorgangs.

Beschreibung
Die mit dieser Methode definierte interne Information wird nur beim Abbruch der Standardvalidierung in jedem OnBeforeValidate 465 Handler abgefragt.

Verfügbar mit Typbibliothek Version 1.5

Fehler
1400 Das Objekt ist nicht mehr gültig.

14.3.2.10.54 SetPathName (nicht mehr verwendet)

Wird ersetzt durch Document.FullName 477

```
// ----- javascript sample -----
// instead of:
// Application.ActiveDocument.SetPathName("C:\\myXMLFiles\\test.xml");
// use now:
Application.ActiveDocument.FullName = "C:\\myXMLFiles\\test.xml";
```

Methode: SetPathName (*strPath* als String)

Beschreibung
Die Methode SetPathNamedefiniert den Pfad des aktiven Dokuments. SetPathNamekopiert den String nur und überprüft nicht, ob der Pfad gültig ist. Alle nachfolgenden Speicheroperationen erfolgen in dieser Datei.

14.3.2.10.55 StartChanges

Methode: StartChanges()

Beschreibung
Nach Ausführung von StartChanges aktualisiert Authentic Desktop seine Editor-Fenster erst, wenn Document.EndChanges 473 aufgerufen wird. Dadurch wird die Geschwindigkeit bei der Ausführung komplexer Operationen an der XML-Struktur verbessert.

Fehler
1400 Das Objekt ist nicht mehr gültig.

14.3.2.10.56 Suggestions

Eigenschaft: Suggestions als Array

Beschreibung
Diese Eigenschaft enthält die letzten gültigen Vorschläge des Benutzers für dieses Dokument. Die von XMLSpy generierten Vorschläge können bearbeitet werden, bevor sie dem Benutzer im OnBeforeShowSuggestions 552 Event angezeigt werden.

Fehler
1400 Das Objekt ist nicht mehr gültig.
1407 Ungültiger Parameter oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.

14.3.2.10.57 SwitchViewMode

Methode: SwitchViewMode (*nMode* als SPYViewModes 616) als Boolean

Rückgabewert
Gibt "true" zurück, wenn der Ansichtsmodus gewechselt wird.

Beschreibung
Die Methode definiert den aktuellen Ansichtsmodus des Dokuments in Authentic Desktop. Siehe auch Document.CurrentViewMode 472.

Fehler
1400 Das Objekt ist nicht mehr gültig.
1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
1417 Es wurde ein ungültiger Ansichtsmodus definiert.

14.3.2.10.58 TextView

Eigenschaft: TextView als TextView 551

Beschreibung

 © 2016-2022 Altova GmbH

Über diese Eigenschaft haben Sie Zugriff auf die Textansichtsfunktionalität des Dokuments.

Fehler
1400 Das Objekt ist nicht mehr gültig.
1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.10.59 Title

Eigenschaft: Title als String (schreibgeschützt)

Beschreibung
Titleenthält den Dateinamen des Dokuments. Um den Pfad und Dateinamen der Datei abzurufen, verwenden Sie FullName 477.

Fehler
1400 Das document-Objekt ist nicht mehr gültig.
1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.10.60 TransformXSL

Methode: TransformXSL()

Beschreibung
TransformXSL verarbeitet das XML-Dokument anhand der damit verknüpften XSL-Datei. Informationen, wie man in das Dokument eine Referenz auf eine XSL-Datei platziert, finden Sie unter Document.AssignXSL 467.

Fehler
1400 Das document-Objekt ist nicht mehr gültig.
1411 Fehler bei der Transformation.

14.3.2.10.61 TransformXSLEx

Method: TransformXSL(*nAction*as SPYDialogAction 607)

Beschreibung
TransformXSL verarbeitet das XML-Dokument über die damit verknüpfte XSL-Datei. Informationen, wie man in das Dokument eine Referenz auf eine XSL-Datei platziert, finden Sie unter Document.AssignXSL 467.

Fehler
1400 Das document-Objekt ist nicht mehr gültig.
1411 Fehler bei der Transformation.

14.3.2.10.62 TransformXSLFO

Methode: TransformXSLFO()

Beschreibung

TransformXSLFO verarbeitet das XML-Dokument anhand der damit verknüpften XSLFO-Datei. Informationen, wie man in das Dokument eine Referenz auf eine XSLFO-Datei platziert, finden Sie unter AssignXSLFO 467. Bevor Sie diese Methode verwenden, müssen Sie Authentic Desktop einen FOP-Prozessor zuweisen.

Fehler

1400 Das document-Objekt ist nicht mehr gültig.
1411 Fehler bei der Transformation.

14.3.2.10.63 TreatXBRLInconsistenciesAsErrors

Eigenschaft: TreatXBRLInconsistenciesAsErrors als Boolean

Beschreibung

Wenn diese Eigenschaft auf `true` gesetzt ist, gibt die Methode `Document.IsValid()` den Wert `false` bei XBRL-Instanzen zurück, die Inkonsistenzen mit der XBRL-Spezifikation enthalten. Der Standardwert dieser Eigenschaft ist `false`.

Errors

1400 Das document-Objekt ist nicht mehr gültig.
1407 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.10.64 UpdateViews

Methode: UpdateViews()

Beschreibung

Rufen Sie zum Neuzeichnen der Grid-Ansicht und der Strukturansicht UpdateViewsauf. Dies ist manchmal von Bedeutung, nachdem Sie die XMLData-Struktur eines Dokuments geändert haben. Diese Methode zeichnet die Textansicht von Authentic Desktop nicht neu.

Fehler

1400 Das document-Objekt ist nicht mehr gültig.

14.3.2.10.65 UpdateXMLData

Methode: UpdateXMLData() als Boolean

Beschreibung

Die XMLData 593-Struktur wird anhand der aktuellen Ansicht aktualisiert. Beachten Sie bitte, dass dieser Vorgang in der Textansicht fehlschlagen kann, wenn der aktuelle XML-Text nicht wohlgeformt ist. Dies ist nicht notwendig, wenn CurrentViewMode 472 *spyViewGrid*oder *spyViewAuthentic* ist, weil die XMLData 593 in diesen Ansichten automatisch aktualisiert werden.

Verfügbar mit Typbibliothek Version 1.5

 © 2016-2022 Altova GmbH

Fehler

1400 Das document-Objekt ist nicht mehr gültig.

14.3.2.11 Documents

Eigenschaften

Count 494
Item 494

Methoden

NewFile 494
OpenFile 495
OpenURL 496
OpenURLDialog 496
NewFileFromText 495

Beschreibung

Diese Objekt repräsentiert die gerade in Authentic Desktop geöffneten Dokumente. Verwenden Sie dieses Objekt, um weitere Dokumente zu öffnen oder durch die bereits offenen Dokumente zu iterieren.

Beispiele

```
' ----------------------------------------
' Scripting environment - VBScript
' iterate through open documents
' ----------------------------------------
DimobjDocuments
SetobjDocuments = Application.Documents

ForEachobjDoc InobjDocuments
        'do something useful with your document
        objDoc.SetActiveDocument()
Next

// ----------------------------------------
// Scripting environment - JScript
// close all open documents
// ----------------------------------------
for(var iter = newEnumerator (Application.Documents);
   ! iter.atEnd();
   iter.moveNext())
{
        // MsgBox ("Closing file " + iter.item().Name);
        iter.item().Close (true);
}
```

14.3.2.11.1 Count

Eigenschaft: Count als long

Beschreibung
Anzahl der offenen Dokumente.

14.3.2.11.2 Item

Methode: Item (*n* als long) als Document 462

Beschreibung
Ruft das Dokument mit dem Index n in dieser Sammlung ab. Der Index ist 1-basiert.

14.3.2.11.3 NewAuthenticFile

Methode: NewAuthenticFile (*strSPSPath* als String, *strXMLPath* als String) als Document 462

Parameter
strSPSPath
Der Pfad zum SPS-Dokument.

strXMLPath
Der neue Name des XML-Dokuments.

Rückgabewert
Die Methode gibt das neue Dokument zurück.

Beschreibung
NewAuthenticFile erstellt eine neue XML-Datei und öffnet es in der Authentic-Ansicht wobei es das SPS-Design strSPSPath verwendet.

14.3.2.11.4 NewFile

Methode: NewFile (*strFile* als String, *strType* als String) als Document 462

Parameter
strFile
Der vollständige Pfad der neuen Datei.

strType
Der Typ der neuen Datei als String (z.B. "xml", "xsd", ...)

Rückgabewert
Gibt die neue Datei zurück.

Beschreibung

 © 2016-2022 Altova GmbH

NewFileerstellt eine neue Datei vom Typ strType(d.h. "xml"). Die neu erstellte Datei ist auch das ActiveDocument.

14.3.2.11.5 NewFileFromText

Methode: NewFileFromText (*strText* als String, *strType* als String) als Document 462

Parameter
strText
Der Inhalt des neuen Dokuments ist unformatierter Text.

strType
Der zu erstellende Dokumenttyp (d.h. "xml").

Rückgabewert
Die Methode gibt das neue Dokument zurück.

Beschreibung
NewFileFromText erstellt ein neues Dokument mit strText als seinem Inhalt.

14.3.2.11.6 OpenAuthenticFile

Methode: OpenAuthenticFile (*strSPSPath* als String, *strXMLPath* als String) als Document 462

Parameter
strSPSPath
Der Pfad zum SPS-Dokument.

strXMLPath
Der Pfad zum XML-Dokument (kann leer sein).

Rückgabewert
Die Methode gibt das neue Dokument zurück.

Beschreibung
OpenAuthenticFile öffnet eine XML-Datei oder Datenbank in der Authentic-Ansicht wobei es das SPS-Design strSPSPath verwendet.

14.3.2.11.7 OpenFile

Methode: OpenFile (*strPath* als String, *bDialog* als Boolean) als Document 462

Parameter
strPath
Pfad und Dateiname der zu öffnenden Datei.

bDialog
Zeigt die Dialogfelder für die Benutzereingaben an..

Rückgabewert
Gibt die geöffnete Datei zurück, wenn die Methode erfolgreich war.

Beschreibung
OpenFileöffnet die Datei strPath. Wenn bDialogTRUE ist, wird ein Dateidialogfeld angezeigt.

Beispiel

```
DimobjDoc AsDocument
SetobjDoc = objSpy.Documents.OpenFile(strFile, False)
```

14.3.2.11.8 OpenURL

Methode: OpenURL (*strURL* als String, *nURLType* als SPYURLTypes 615, *nLoading* als SPYLoading 610, *strUser* als String, *strPassword* als String) als Document 462

Parameter
strURL
URL zum Öffnen eines Dokuments.

nURLType
Typ des zu öffnenden Dokuments. Setzen Sie den Wert auf -1, um den Typ automatisch zu ermitteln.

nLoading
Setzen Sie nLoading auf 0 (Null), wenn das Dokument aus dem Cache oder vom Proxy-Server geladen werden soll. Setzen Sie nLoading andernfalls auf 1.

strUser
Name des Benutzers, falls erforderlich. Kann leer bleiben.

strPassword
Passwort für die Authentifizierung. Kann leer bleiben.

Rückgabewert
Die Methode gibt das geöffnete Dokument zurück.

Beschreibung
OpenURL öffnet die URL strURL.

14.3.2.11.9 OpenURLDialog

Methode: OpenURLDialog (*strURL* als String, *nURLType* als SPYURLTypes 615, *nLoading* als SPYLoading 610, *strUser* als String, *strPassword* als String) als Document 462

Parameter
strURL
URL zum Öffnen eines Dokuments.

nURLType

 © 2016-2022 Altova GmbH

Typ des zu öffnenden Dokuments. Setzen Sie den Wert auf -1, um den Typ automatisch zu ermitteln.

nLoading
Setzen Sie nLoading auf 0 (Null), wenn das Dokument aus dem Cache oder vom Proxy-Server geladen werden soll. Setzen Sie nLoading andernfalls auf 1.

strUser
Name des Benutzers, falls erforderlich. Kann leer bleiben.

strPassword
Passwort für die Authentifizierung. Kann leer bleiben.

Rückgabewert
Die Methode gibt das geöffnete Dokument zurück.

Beschreibung
OpenURLDialogzeigt dem Benutzer das Dialogfeld "URL öffnen" an und setzt in die Eingabefelder die angegebenen Parameter ein.

14.3.2.12 DTDSchemaGeneratorDlg

Eigenschaften und Methoden

Standard-Automation-Eigenschaften
Application 497
Parent 500

DTDSchemaFormat 498
ValueList 501
TypeDetection 501
FrequentElements 498
MergeAllEqualNamed 499
ResolveEntities 500
AttributeTypeDefinition 498
GlobalAttributes 499
OnlyStringEnums 499
MaxEnumLength 499
OutputPath 500
OutputPathDialogAction 500

Beschreibung
Mit Hilfe dieses Objekts können Sie die Generierung eines Schemas oder einer DTD konfigurieren. Die Methode GenerateDTDOrSchemaEx 478 erwartet einen DTDSchemaGeneratorDlgals Parameter, um die Generierung sowie die damit in Zusammenhang stehenden Interaktionen mit dem Benutzer zu konfigurieren.

14.3.2.12.1 Application

Eigenschaft: Applicationals Application 372 (schreibgeschützt)

Beschreibung

Ruft das Authentic Desktopapplication-Objekt auf.

Fehler
3000 Das Objekt ist nicht mehr gültig.
3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.12.2 AttributeTypeDefinition

Eigenschaft: AttributeTypeDefinitionals SPYAttributeTypeDefinition 605

Beschreibung
Definiert, wie Attributdefinitionen zusammengeführt werden.

Fehler
3000 Das Objekt ist nicht mehr gültig.
3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.12.3 DTDSchemaFormat

Eigenschaft: DTDSchemaFormatals SPYDTDSchemaFormat 608

Beschreibung
Setzt das Schemaausgabeformat auf DTD oder W3C.

Fehler
3000 Das Objekt ist nicht mehr gültig.
3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.12.4 FrequentElements

Eigenschaft: FrequentElementsals SPYFrequentElements 608

Beschreibung
Sollen die Typen für alle Elemente als global definiert werden? Verwenden Sie diesen Wert *spyGlobalComplexType* um sie global zu definieren. Verwenden Sie andernfalls den Wert *spyGlobalElements*.

Fehler
3000 Das Objekt ist nicht mehr gültig.
3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.12.5 GlobalAttributes

Eigenschaft: GlobalAttributesals Boolean

Beschreibung
Sollen Attribute mit demselben Namen und Typ global aufgelöst werden?

Fehler
3000 Das Objekt ist nicht mehr gültig.
3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.12.6 MaxEnumLength

Eigenschaft: MaxEnumLengthals Integer

Beschreibung
Definiert die maximale Anzahl an Zeichen, die bei Enumerationsnamen zulässig ist. Wenn auch nur ein Wert länger als dieser ist, wird keine Enumeration generiert.

Fehler
3000 Das Objekt ist nicht mehr gültig.
3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.12.7 MergeAllEqualNamed

Eigenschaft: MergeAllEqualNamedals Boolean

Beschreibung
Sollen Typen aller Elemente mit demselben Namen in einem Typ zusammengeführt werden?

Fehler
3000 Das Objekt ist nicht mehr gültig.
3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.12.8 OnlyStringEnums

Eigenschaft: OnlyStringEnumsals Boolean

Beschreibung
Definiert, ob Enumerationen nur für unformatierte Strings oder für alle Arten von Werten erstellt werden sollen.

Fehler
3000 Das Objekt ist nicht mehr gültig.
3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.12.9 OutputPath

Eigenschaft: OutputPathals String

Beschreibung
Wählt den Dateinamen für das generierte Schema/die generierte DTD aus.

Fehler
3000 Das Objekt ist nicht mehr gültig.
3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.12.10 OutputPathDialogAction

Eigenschaft: OutputPathDialogActionals SPYDialogAction 607

Beschreibung
Definiert wie das untergeordnete Dialogfeld zur Auswahl des Ausgabepfads für das Schema/die DTD behandelt werden soll. Setzen Sie diesen Wert auf *spyDialogUserInput(2)*damit standardmäßig das Dialogfeld mit dem aktuellen Wert der Eigenschaft OutputPath 500 angezeigt wird. Verwenden Sie *spyDialogOK(0)*um dem Benutzer das Dialogfeld nicht anzuzeigen.

Fehler
3000 Das Objekt ist nicht mehr gültig.
3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.12.11 Parent

Eigenschaft: Parentals Dialogs 458 (schreibgeschützt)

Beschreibung
Ruft das dem Objekt übergeordnete Objekt auf.

Fehler
3000 Das Objekt ist nicht mehr gültig.
3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.12.12 ResolveEntities

Eigenschaft: ResolveEntitiesals Boolean

Beschreibung
Sollen alle Entities vor Beginn der Generierung aufgelöst werden? Wenn ja, wird ein Infoset erstellt.

© 2016-2022 Altova GmbH

Fehler
3000 Das Objekt ist nicht mehr gültig.
3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.12.13 TypeDetection

Eigenschaft: TypeDetectionals SPYTypeDetection 615

Beschreibung
Definiert die Genauigkeit, mit der ein simpleTyp ermittelt wird.

Fehler
3000 Das Objekt ist nicht mehr gültig.
3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.12.14 ValueList

Eigenschaft: ValueListals Integer

Beschreibung
Generiert nicht mehr als diese Anzahl von Enumeration-Facets pro Typ. Setzen Sie den Wert auf -1, um keine Einschränkung zu setzen.

Fehler
3000 Das Objekt ist nicht mehr gültig.
3001 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.13 ElementList

Eigenschaften
Count 502
Item 502

Methoden
RemoveElement 502

Beschreibung
Elementlisten werden beim Export und Import von Daten für verschiedene Zwecke verwendet. Je nach Verwendungszweck werden unterschiedliche Eigenschaften von ElementListItem 502 verwendet.

Die Methode kann die folgenden Eigenschaften enthalten
- eine Liste von Tabellennamen, die durch einen Aufruf von Application.GetDatabaseTables 380 zurückgegeben werden.
- eine Liste von Feldnamen, die durch einen Aufruf von Application.GetDatabaseImportElementList 379 oder Application.GetTextImportElementList 381 zurückgegeben werden,

- eine Feldnamenfilterliste, die in Application.ImportFromDatabase[383] und Application.ImportFromText[385] verwendet wird,
- eine Liste von Tabellennamen und der Anzahl ihrer Zeilen und Spalten, die durch Aufrufen von GetExportElementList[481] zurückgegeben werden oder
- eine in Document.ExportToDatabase[474] und Document.ExportToText[475] verwendete Feldnamenfilterliste.

14.3.2.13.1 Count

Eigenschaft: Countals long (schreibgeschützt)

Beschreibung
Anzahl der Elemente in dieser Sammlung.

14.3.2.13.2 Item

Methode: Item(nals long) als ElementListItem[502]

Beschreibung
Ruft das Element mit dem Index naus dieser Sammlung ab. Das erste Element hat den Index 1.

14.3.2.13.3 RemoveElement

Methodr: RemoveElement(Indexals long)

Beschreibung
RemoveElemententfernt den Elementindex aus der Sammlung. Das erste Element hat den Index 1.

14.3.2.14 ElementListItem

Eigenschaften
Name[503]

ElementKind[502]

FieldCount[503]
RecordCount[503]

Beschreibung
Ein Element in einer ElementList[501]. Die Verwendung seiner Eigenschaften ist vom Verwendungszweck der Elementliste abhängig. Nähere Informationen dazu siehe ElementList[501].

14.3.2.14.1 ElementKind

Eigenschaft: ElementKindals SPYXMLDataKind[618]

Beschreibung

 © 2016-2022 Altova GmbH

Definiert, ob ein Feld als XML-Element (Datenwert von spyXMLDataElement) oder als Attribut (Datenwert von spyXMLDataAttr) importiert werden soll.

14.3.2.14.2 FieldCount

Eigenschaft: FieldCountals long (schreibgeschützt)

Beschreibung
Anzahl der Felder (d.h. Spalten) in der durch dieses Element beschriebenen Tabelle. Diese Eigenschaft ist nur nach einem Aufruf von Document.GetExportElementList 481 gültig.

14.3.2.14.3 Name

Eigenschaft: Nameals String (schreibgeschützt)

Beschreibung
Name des Elements. Dies ist je nach Verwendungszweck der Elementliste entweder der Name einer Tabelle oder eines Felds.

14.3.2.14.4 RecordCount

Eigenschaft: RecordCountals long (schreibgeschützt)

Beschreibung
Anzahl der Datensätze (d.h. Zeilen) in der durch dieses Element beschriebenen Tabelle. Diese Eigenschaft ist nur nach einem Aufruf von Document.GetExportElementList 481 gültig.

14.3.2.15 ExportSettings

Eigenschaften

ElementList 504

EntitiesToText 504

ExportAllElements 504
SubLevelLimit 506

FromAttributes 505
FromSingleSubElements 505
FromTextValues 505

CreateKeys 504
IndependentPrimaryKey 505

Namespace 505

ExportCompleteXML 505
StartFromElement 506

Beschreibung
ExportSettingsenthält Optionen, die beim Export von XML-Daten in eine Datenbank oder Textdatei verwendet werden.

14.3.2.15.1 CreateKeys

Eigenschaft: CreateKeys als Boolean

Beschreibung
Diese Eigenschaft aktiviert bzw. deaktiviert die Erstellung von Schlüsseln (d.h. Primär- und Sekundärschlüssel). Die Standardeinstellung ist "True".

14.3.2.15.2 ElementList

Eigenschaft: ElementList als ElementList 501

Beschreibung
Standardeinstellung ist eine leere Liste. In dieser Elementliste wird definiert, welche Felder exportiert werden. Um die Liste der verfügbaren Felder abzurufen, verwenden Sie Document.GetExportElementList 481. Sie können Spalten vom Export ausnehmen, indem Sie Elemente mit ElementList.RemoveElement 502 aus dieser Liste entfernen, bevor Sie sie an Document.ExportToDatabase 474 oder Document.ExportToText 475 übergeben.

14.3.2.15.3 EntitiesToText

Eigenschaft: EntitiesToText als Boolean

Beschreibung
Definiert, ob XML-Entities in Text konvertiert oder unverändert belassen werden sollen. Die Standardeinstellung ist True.

14.3.2.15.4 ExportAllElements

Eigenschaft: ExportAllElements als Boolean

Beschreibung
Wenn diese Eigenschaft auf truegesetzt ist, werden alle Elemente im Dokument exportiert. Wenn sie auf falsegesetzt ist, wird die Anzahl der zu exportierenden Unterebenen mittels ExportSettings.SubLevelLimit 506 eingeschränkt. Die Standardeinstellung ist true.

© 2016-2022 Altova GmbH

14.3.2.15.5 ExportCompleteXML

Eigenschaft: ExportCompleteXML als Boolean

Beschreibung
Definiert, ob die komplette XML-Datei oder nur das durch StartFromElement 506 definierte Element und seine untergeordneten Elemente exportiert werden sollen. Standardeinstellung ist True.

14.3.2.15.6 FromAttributes

Eigenschaft: FromAttributes als Boolean

Beschreibung
Setzen Sie FromAttributes auf "false", wenn keine Exportdaten anhand von Attributen erstellt werden sollen. Die Standardeinstellung ist True.

14.3.2.15.7 FromSingleSubElements

Eigenschaft: FromSingleSubElements als Boolean

Beschreibung
Setzen Sie FromSingleSubElements auf "false", wenn keine Exportdaten von Elementen erstellt werden sollen. Die Standardeinstellung ist True.

14.3.2.15.8 FromTextValues

Eigenschaft: FromTextValues als Boolean

Beschreibung
Setzen Sie FromTextValues auf "false", wenn keine Exportdaten anhand von Textwerten erstellt werden sollen. Die Standardeinstellung ist "True".

14.3.2.15.9 IndependentPrimaryKey

Eigenschaft: IndependentPrimaryKey als Boolean

Beschreibung
Aktiviert bzw. deaktiviert die Erstellung unabhängiger Primärschlüsselzähler für alle Elemente. Wenn ExportSettings.CreateKeys 504 "False" ist, wird diese Eigenschaft ignoriert. Die Standardeinstellung ist "True".

14.3.2.15.10 Namespace

Eigenschaft: Namespace als SPYExportNamespace 608

Beschreibung
Mit der Standardeinstellung werden alle Namespace-Präfixe aus den Elementnamen entfernt. In einigen Datenbankformaten ist der Doppelpunkt kein zulässiges Zeichen. Die Standardeinstellung ist spyNoNamespace.

14.3.2.15.11 SubLevelLimit

Eigenschaft: SubLevelLimit als Integer

Beschreibung
Definiert die Anzahl der Unterebenen, die exportiert werden soll. Diese Eigenschaft, wird ignoriert, wenn ExportSettings.ExportAllElements 504 trueist.

14.3.2.16 FileSelectionDlg

Eigenschaften und Methoden

Standard-Automation-Eigenschaften
Application 506
Parent 507

Dialogfeldeigenschaften
FullName 507

Annahme oder Abbruch der Aktion, die das Event verursacht hat.
DialogAction 507

Beschreibung
Mit Hilfe dieses Dialogfeldobjekts können Sie Informationen über ein Event empfangen und Daten an den Event Handler auf dieselbe Art zurückgeben, wie über ein Benutzer-Dialogfeld. Mit Hilfe von FileSelectionDlg.FullName 507 können Sie den Dateipfad auswählen oder ändern und mit Hilfe der Eigenschaft FileSelectionDlg.DialogAction 507 können Sie die Aktion, die das Event verursacht hat, abbrechen oder akzeptieren.

14.3.2.16.1 StartFromElement

Eigenschaft: StartFromElement als String

Beschreibung
Definiert das Startelement für den Export. Diese Eigenschaft wird nur berücksichtigt, wenn ExportCompleteXML 505 "false" ist.

14.3.2.16.2 Application

Eigenschaft: Applicationals Application 372 (schreibgeschützt)

 © 2016-2022 Altova GmbH

Beschreibung
Ruft das Authentic Desktopapplication-Objekt auf.

Fehler
2400 Das Objekt ist nicht mehr gültig.
2401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.16.3 DialogAction

Eigenschaft: DialogActionals SPYDialogAction 607

Beschreibung
Wenn Sie möchten, dass Ihr Script die Dateiauswahl ohne Interaktion durch den Benutzer durchführt, simulieren Sie die Interaktion durch den Benutzer, indem Sie die Eigenschaft entweder auf *spyDialogOK(0)*oder *spyDialogCancel(1)*setzen.
Damit Ihr Script die Standardwerte einsetzt, der Benutzer das Dialogfeld aber sehen und darauf reagieren kann, verwenden Sie den Wert *spyDialogUserInput(2)*. Wenn Sie in einem Event Handler ein FileSelectionDlg Objekt empfangen, wird *spyDialogUserInput(2)*nicht unterstützt und als *spyDialogOK(0)*interpretiert.

Fehler
2400 Das Objekt ist nicht mehr gültig.
2401 Ungültiger Wert für die Dialogfeldaktion oder es wurde für den Rückgabeparameter eine ungültige Adresse angegeben.

14.3.2.16.4 FullName

Eigenschaft: FullNameals String

Beschreibung
Ruft den vollständigen Pfad der Datei auf, die durch das Dialogfeld ausgewählt wird. Bei den meisten Events, die ein FileSelectionDlg Objekt an Sie übergeben, können Sie diesen Wert ändern und dadurch die Aktion, die das Event verursacht hat (also Laden oder Speichern unter einem anderen Pfad), beeinflussen.

Fehler
2400 Das Objekt ist nicht mehr gültig.
2401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.16.5 Parent

Eigenschaft: Parentals Dialogs 458 (schreibgeschützt)

Beschreibung
Ruft das übergeordnete Objekt des Objekts auf.

Fehler
2400 Das Objekt ist nicht mehr gültig.
2401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17 FindInFilesDlg

Eigenschaften und Methoden

Standard-Automation-Eigenschaften
Application 509
Parent 510

Find 509
RegularExpression 511
Replace 511
DoReplace 509
ReplaceOnDisk 511
MatchWholeWord 510
MatchCase 510
SearchLocation 512
StartFolder 512
IncludeSubfolders 510
SearchInProjectFilesDoExternal 511
FileExtension 509
AdvancedXMLSearch 508
XMLElementNames 514
XMLElementContents 513
XMLAttributeNames 513
XMLAttributeContents 512
XMLComments 513
XMLCData 513
XMLPI 514
XMLRest 514
ShowResult 512

Beschreibung
Mit Hilfe dieses Objekts können Sie die Suche (oder die Erstetzung) von Strings in Dateien konfigurieren. Die Methode FindInFiles 379 erwartet einen FindInFilesDlgals Parameter.

14.3.2.17.1 AdvancedXMLSearch

Eigenschaft: AdvancedXMLSearchals Boolean

Beschreibung
Legt fest, ob die XML-Sucheigenschaften (XMLElementNames 514, XMLElementContents 513, XMLAttributeNames 513, XMLAttributeContents 513, XMLComments 513, XMLCData 513, XMLPI 514 und XMLRest 514) berücksichtigt werden. Die Standardeinstellung ist "false".

Fehler

3500	Das Objekt ist nicht mehr gültig.
3501	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.17.2 Application

Eigenschaft: Applicationals Application 372 (schreibgeschützt)

Beschreibung
Ruft das Authentic Desktopapplication-Objekt auf.

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.3 DoReplace

Eigenschaft: DoReplaceals Boolean

Beschreibung
Legt fest, ob der gefundene String durch den in Replace 511 definierten String ersetzt werden soll. Die Standardeinstellung ist "false"

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.4 FileExtension

Eigenschaft: FileExtensionals String

Beschreibung
Definiert den Dateifilter der Dateien, die bei der Suche berücksichtigt werden sollen. Mehrere Dateifilter müssen durch ein Semikolon voneinander getrennt werden (z.B. *.xml;*.dtd;a*.xsd). Verwenden Sie die Platzhalterzeichen * und ? um den Dateifilter zu definieren.

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.5 Find

Eigenschaft: Findals String

Beschreibung
Definiert den zu suchenden String.

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.6 IncludeSubfolders

Eigenschaft: IncludeSubfoldersals Boolean

Beschreibung
Legt fest, ob auch Unterordner durchsucht werden sollen. Die Standardeinstellung ist "True".

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.7 MatchCase

Eigenschaft: MatchCaseals Boolean

Beschreibung
Legt fest, ob die Groß- und Kleinschreibung bei der Suche beachtet werden soll. Die Standardeinstellung ist "True".

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.8 MatchWholeWord

Eigenschaft: MatchWholeWordals Boolean

Beschreibung
Legt fest, ob bei der Suche nur ganze Wörter oder Teile davon gesucht werden sollen. Die Standardeinstellung ist "false".

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.9 Parent

Eigenschaft: Parentals Dialogs 458 (schreibgeschützt)

Beschreibung
Ruft das übergeordnete Objekt des Objekts auf.

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.17.10 RegularExpression

Eigenschaft: RegularExpressionals Boolean

Beschreibung
Definiert, ob Find 509 eine Regular Expression enthält. Die Standardeinstellung ist "false".

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.11 Replace

Eigenschaft: Replaceals String

Beschreibung
Definiert den Ersetzungsstring. Der gefundene String wird nur ersetzt, wenn DoReplace 509 auf "true" gesetzt ist.

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.12 ReplaceOnDisk

Eigenschaft: ReplaceOnDiskals Boolean

Beschreibung
Legt fest, ob die Ersetzung direkt auf der Festplatte vorgenommen wird. Die geänderte Datei wird nicht geöffnet. Die Standardeinstellung ist "false".

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.13 SearchInProjectFilesDoExternal

Eigenschaft: SearchInProjectFilesDoExternalals Boolean

Beschreibung
Definiert, ob bei einer Projektsuche die externen Ordner im offenen Projektdurchsucht werden sollen. Die Standardeinstellung ist "false".

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.14 SearchLocation

Eigenschaft: SearchLocationals SPYFindInFilesSearchLocation 608

Beschreibung
Definiert den Ordner für die Suche. Die Standardeinstellung ist spyFindInFiles_Documents.

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.15 ShowResult

Eigenschaft: ShowResultals Boolean

Beschreibung
Definiert, ob das Ergebnis im Ausgabefenster "In Dateien suchen" angezeigt werden soll. Die Standardeinstellung ist "false".

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.16 StartFolder

Eigenschaft: StartFolderals String

Beschreibung
Definiert den Ordner, in dem die Suche auf der Festplatte beginnen soll.

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.17 XMLAttributeContents

Eigenschaft: XMLAttributeContentsals Boolean

Beschreibung
Legt fest ob Attributinhalt durchsucht werden soll, wenn AdvancedXMLSearch 508 true ist. Die Standardeinstellung ist "True".

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.17.18 XMLAttributeNames

Eigenschaft: XMLAttributeNamesals Boolean

Beschreibung
Definiert, ob Attributnamen durchsucht werden sollen, wenn AdvancedXMLSearch 508 true ist. Die Standardeinstellung ist "True".

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.19 XMLCData

Eigenschaft: XMLCDataals Boolean

Beschreibung
Legt fest, ob CData Tags durchsucht werden sollen, wenn AdvancedXMLSearch 508 true ist. Die Standardeinstellung ist "True".

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.20 XMLComments

Eigenschaft: XMLCommentsals Boolean

Beschreibung
Legt fest, ob Kommentare durchsucht werden sollen, wenn AdvancedXMLSearch 508 true ist. Die Standardeinstellung ist "True".

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.21 XMLElementContents

Eigenschaft: XMLElementContentsals Boolean

Beschreibung
Legt fest, ob der Elementinhalt durchsucht werden soll, wenn AdvancedXMLSearch 508 true ist. Die Standardeinstellung ist "True".

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.22 XMLElementNames

Eigenschaft: XMLElementNamesals Boolean

Beschreibung
Legt fest, ob Elementnamen durchsucht werden sollen, wenn AdvancedXMLSearch 508 true ist. Die Standardeinstellung ist "True".

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.23 XMLPI

Eigenschaft: XMLPIals Boolean

Beschreibung
Legt fest, ob XML-Verarbeitungsanweisungen durchsucht werden sollen, wenn AdvancedXMLSearch 508 true ist. Die Standardeinstellung ist "True".

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.17.24 XMLRest

Eigenschaft: XMLRestals Boolean

Beschreibung
Legt fest, ob der Rest der XML-Datei (der nicht durch die anderen XML-Sucheigenschaften berücksichtigt wird) durchsucht werden soll, wenn AdvancedXMLSearch 508 true ist. Die Standardeinstellung ist "True".

Fehler
3500 Das Objekt ist nicht mehr gültig.
3501 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.18 FindInFilesResult

Eigenschaften und Methoden

Standard-Automation-Eigenschaften
Application 515
Parent 516

Count 515
Item 515

 © 2016-2022 Altova GmbH

Path 516
Document 515

Beschreibung
Dieses Objekt repräsentiert eine Datei, die den Suchkriterien entsprochen hat. Es enthält eine Liste von FindInFilesResultMatch 516 Objekten, die die Suchposition beschreiben.

14.3.2.18.1 Application

Eigenschaft: Applicationals Application 372 (schreibgeschützt)

Beschreibung
Ruft das Authentic Desktopapplication-Objekt auf.

Fehler
3700 Das Objekt ist nicht mehr gültig.
3701 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.18.2 Count

Eigenschaft: Countals long (schreibgeschützt)

Beschreibung
Anzahl der Elemente in dieser Sammlung.

14.3.2.18.3 Document

Eigenschaft: Pathals Document 462 (schreibgeschützt)

Beschreibung
Diese Eigenschaft gibt das Document 462 -Objekt zurück, wenn die gefundene Datei bereits in XMLSpy geöffnet ist.

Fehler
3700 Das Objekt ist nicht mehr gültig.
3701 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.18.4 Item

Methode: Item(nals long) als FindInFilesResultMatch 516

Beschreibung
Ruft das Element mit dem Index naus dieser Sammlung ab. Das erste Element hat den Index 1.

14.3.2.18.5 Parent

Eigenschaft: Parentals FindInFilesResults 518 (schreibgeschützt)

Beschreibung
Ruft das dem Objekt übergeordnete Objekt auf.

Fehler
3700 Das Objekt ist nicht mehr gültig.
3701 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.18.6 Path

Eigenschaft: Pathals String (schreibgeschützt)

Beschreibung
Gibt den Pfad der Datei zurück, die den Suchkriterien entsprochen hat.

Fehler
3700 Das Objekt ist nicht mehr gültig.
3701 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.19 FindInFilesResultMatch

Eigenschaften und Methoden

Standard-Automation-Eigenschaften
Application 516
Parent 517

Line 517
Position 518
Length 517
LineText 517
Replaced 518

Beschreibung
Enthält die genaue Position des gesuchten String in der Datei.

14.3.2.19.1 Application

Eigenschaft: Applicationals Application 372 (schreibgeschützt)

Beschreibung
Ruft das Authentic Desktopapplication-Objekt auf.

 © 2016-2022 Altova GmbH

Fehler

3800 Das Objekt ist nicht mehr gültig.
3801 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.19.2 Length

Eigenschaft: Lengthals Long (schreibgeschützt)

Beschreibung

Gibt die Länge des gefundenen String zurück.

Fehler

3800 Das Objekt ist nicht mehr gültig.
3801 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.19.3 Line

Eigenschaft: Lineals Long (schreibgeschützt)

Beschreibung

Gibt die Zeilennummer des gefundenen Suchbegriffs zurück. Die Zeilennummerierung beginnt mit 0.

Fehler

3800 Das Objekt ist nicht mehr gültig.
3801 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.19.4 LineText

Eigenschaft: LineTextals String (schreibgeschützt)

Beschreibung

Gibt den Text der Zeile zurück.

Fehler

3800 Das Objekt ist nicht mehr gültig.
3801 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.19.5 Parent

Eigenschaft: Parentals FindInFilesResult 514 (schreibgeschützt)

Beschreibung

Ruft das dem Objekt übergeordnete Objekt auf.

Fehler
3800 Das Objekt ist nicht mehr gültig.
3801 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.19.6 Position

Eigenschaft: Positionals Long (schreibgeschützt)

Beschreibung
Gibt die Anfangsposition des gesuchten Begriffs in der Zeile zurück. Die Positionsnummerierung beginnt mit 0.

Fehler
3800 Das Objekt ist nicht mehr gültig.
3801 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.19.7 Replaced

Eigenschaft: Replacedals Boolean (schreibgeschützt)

Beschreibung
True, wenn der gefundene String ersetzt wurde.

Fehler
3800 Das Objekt ist nicht mehr gültig.
3801 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.20 FindInFilesResults

Eigenschaften und Methoden

Standard-Automation-Eigenschaften
Application 519
Parent 519

Count 519
Item 519

Beschreibung
Dies ist das Ergebnis der FindInFiles 379 Methode. Es handelt sich hiebei um eine Liste von FindInFilesResult 514 Objekten.

 © 2016-2022 Altova GmbH

14.3.2.20.1 Application

Eigenschaft: Applicationals Application 372 (schreibgeschützt)

Beschreibung
Ruft das Authentic Desktopapplication-Objekt auf.

Fehler
3600 Das Objekt ist nicht mehr gültig.
3601 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.20.2 Count

Eigenschaft: Countals long (schreibgeschützt)

Beschreibung
Anzahl der Elemente in dieser Sammlung.

14.3.2.20.3 Item

Methode: Item(nals long) als FindInFilesResult 514

Beschreibung
Ruft das Element mit dem Index naus dieser Sammlung ab. Das erste Element hat den Index 1.

14.3.2.20.4 Parent

Eigenschaft: Parentas Application 372 (schreibgeschützt)

Beschreibung
Ruft das dem Objekt übergeordnete Objekt auf.

Fehler
3600 Das Objekt ist nicht mehr gültig.
3601 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.21 GenerateSampleXMLDlg

Eigenschaften und Methoden

Standard-Automation-Eigenschaften
Application 520
Parent 523

NonMandatoryAttributes 522
NonMandatoryElements 523
RepeatCount 524
FillAttributesWithSampleData 521
FillElementsWithSampleData 521
ContentOfNillableElementsIsNonMandatory 521
TryToUseNonAbstractTypes 525
SchemaOrDTDAssignment 524
LocalNameOfRootElement 522
NamespaceURIOfRootElement 522
OptionsDialogAction 523

Nicht mehr unterstützte Eigenschaften
TakeFirstChoice - obsolete 524
FillWithSampleData - obsolete 522
Optimization - obsolete 523

Beschreibung
Dient zum Setzen der Parameter für die Generierung vom XML-Beispielinstanzen auf Basis eines W3C-Schemas oder einer DTD.

14.3.2.21.1 Application

Eigenschaft: Applicationals Application 372 (schreibgeschützt)

Beschreibung
Ruft das Authentic Desktopapplication-Objekt auf.

Fehler
2200 Das Objekt ist nicht mehr gültig.
2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.21.2 ChoiceMode

Eigenschaft: `ChoiceMode` als `SPYSampleXMLGenerationChoiceMode` 612

Beschreibung
Definiert, welche Elemente generiert werden sollen.

Fehler

2200 Das Objekt ist nicht mehr gültig.
2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.21.3 ConsiderSampleValueHints

Eigenschaft: `ConsiderSampleValueHints` als Boolean

Beschreibung
Wählt aus, ob `SampleValueHints` 524 verwendet werden sollen oder nicht.

Fehler

2200 Das Objekt ist nicht mehr gültig.
2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.21.4 ContentOfNillableElementsIsNonMandatory

Eigenschaft: ContentOfNillableElementsIsNonMandatoryals Boolean

Beschreibung
Wenn der Wert "true" ist, wird der Inhalt von Elementen die "nillable" sind, nicht als obligatorisch behandelt.

Fehler
2200 Das Objekt ist nicht mehr gültig.
2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.21.5 FillAttributesWithSampleData

Eigenschaft: FillAttributesWithSampleDataals Boolean

Beschreibung
Wenn der Wert "true" ist, haben Attribute Beispielinhalt.

Fehler
2200 Das Objekt ist nicht mehr gültig.
2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.21.6 FillElementsWithSampleData

Eigenschaft: FillElementsWithSampleDataals Boolean

Beschreibung
Wenn der Wert "true" ist, haben Elemente Beispielinhalt.

Fehler

2200 Das Objekt ist nicht mehr gültig.
2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.21.7 FillWithSampleData - nicht mehr verwendet

Eigenschaft: FillWithSampleDataals Boolean

Beschreibung
Diese Eigenschaft wird nicht mehr verwendet. Verwenden Sie stattdessen FillAttributesWithSampleData 521 und FillElementsWithSampleData 521, instead.

Fehler
0001 Die Eigenschaft steht nicht mehr zur Verfügung.

14.3.2.21.8 LocalNameOfRootElement

Eigenschaft: LocalNameOfRootElementals String

Beschreibung
Definiert den lokalen Namen des Root-Elements für die generierte XML-Beispieldatei.

Fehler
2200 Das Objekt ist nicht mehr gültig.
2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.21.9 NamespaceURIOfRootElement

Eigenschaft: NamespaceURIOfRootElementals String

Beschreibung
Definiert die Namespace URI des Root-Elements für die generierte XML-Beispieldatei.

Fehler
2200 Das Objekt ist nicht mehr gültig.
2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.21.10 NonMandatoryAttributes

Eigenschaft: NonMandatoryAttributesals Boolean

Beschreibung
Wenn der Wert "true" ist, werden in der XML-Beispielinstanzdatei auch nicht obligatorische Attribute erstellt.

Fehler
2200 Das Objekt ist nicht mehr gültig.
2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

 © 2016-2022 Altova GmbH

14.3.2.21.11 NonMandatoryElements

Eigenschaft: NonMandatoryElementsals Boolean

Beschreibung
Bei `true`werden Elemente, die nicht obligatorisch sind, in der XML-Beispielinstanzdatei erstellt.

Fehler
2200 Das Objekt ist nicht mehr gültig.
2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.21.12 Optimization - nicht mehr verwendet

Eigenschaft: Optimizationals SPYSampleXMLGenerationOptimization 612

Beschreibung
Verwenden Sie diese Eigenschaft nicht mehr. Verwenden Sie stattdessen ChoiceMode und NonMandatoryElements

Fehler

14.3.2.21.13 OptionsDialogAction

Eigenschaft: OptionsDialogActionals SPYDialogAction 607

Beschreibung
Damit des Script die Standardwerte einsetzt und der Benutzer das Dialogfeld sieht, und dort Eingaben machen kann, setzen Sie die Eigenschaft auf den Wert *spyDialogUserInput(2)*. Wenn Ihr Script alle Optionen im Dialogfeld "Schemadokumentation" definieren soll, ohne dass eine Interaktion durch den Benutzer erforderlich ist, verwenden Sie *spyDialogOK(0)*. Der Standardwert ist *spyDialogOK.*

Fehler
2200 Das Objekt ist nicht mehr gültig.
2201 Zur Definition der Eigenschaft wurde ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.21.14 Parent

Eigenschaft: Parentas Dialogs 458 (schreibgeschützt)

Beschreibung
Ruft das dem Objekt übergeordnete Objekt auf.

Fehler
2200 Das Objekt ist nicht mehr gültig.
2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.21.15 RepeatCount

Eigenschaft: RepeatCountals long

Beschreibung
Anzahl der Elemente, die für wiederholt vorkommende Typen erstellt werden soll.

Fehler
2200 Das Objekt ist nicht mehr gültig.
2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.21.16 SampleValueHints

Eigenschaft: `SampleValueHints` als `SPYSampleXMLGenerationSampleValueHints` 613

Beschreibung
Definiert, wie Daten für die generierte Beispieldatei ausgewählt werden solen.

Fehler
2200 Das Objekt ist nicht mehr gültig.
2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.21.17 SchemaOrDTDAssignment

Eigenschaft: SchemaOrDTDAssignmentals SPYSampleXMLGenerationSchemaOrDTDAssignment 613

Beschreibung
Definiert, wie eine Referenz auf das damit in Zusammenhang stehende Schema oder die DTD, welche sich im Dokument befindet, in eine XML-Beispieldatei umgewandelt wird.

Fehler
2200 Das Objekt ist nicht mehr gültig.
2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.21.18 TakeFirstChoice - nicht mehr verwendet

Eigenschaft: TakeFirstChoiceals Boolean

Beschreibung
Diese Eigenschaft wird nicht mehr verwendet.

Fehler
0001 Die Eigenschaft steht nicht mehr zur Verfügung.

© 2016-2022 Altova GmbH

14.3.2.21.19 TryToUseNonAbstractTypes

Eigenschaft: TryToUseNonAbstractTypesals Boolean

Beschreibung
Versucht bei "true" einen nicht abstrakten Typen für xsi:type zu verwenden, wenn das Element einen abstrakten Typen hat.

Fehler
2200 Das Objekt ist nicht mehr gültig.
2201 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.22 GridView

Methoden
Deselect 528
Select 528

SetFocus 528

Eigenschaften
CurrentFocus 528

IsVisible 528

Beschreibung
GridViewClass

14.3.2.22.1 Events

14.3.2.22.1.1 OnBeforeDrag

Event: OnBeforeDrag() als Boolean

Skripting-Umgebung - VBScript:
```
Function On_BeforeDrag()
        ' On_BeforeStartEditing = False  ' to prohibit dragging
End Function
```

Skripting-Umgebung - JScript:
```
function On_BeforeDrag()
{
        // return false;  /* to prohibit dragging */
}
```

IDE Plugin:
IXMLSpyPlugIn.OnEvent(4, ...) // nEventId = 4

Beschreibung
Dieses Event wird beim Versuch ausgelöst, ein XMLData-Element in der Grid-Ansicht mit der Maus zu verschieben. Geben Sie *false* zurück, um zu verhindern, dass das Datenelement an eine andere Position gezogen werden kann.

14.3.2.22.1.2 OnBeforeDrop

Event: OnBeforeDrop(*objXMLData* als XMLData[593]) als Boolean

Skripting-Umgebung - VBScript:
```
Function On_BeforeDrop(objXMLData)
        ' On_BeforeStartEditing = False  ' to prohibit dropping
End Function
```

Skripting-Umgebung - JScript:
```
function On_BeforeDrop(objXMLData)
{
        // return false;  /* to prohibit dropping */
}
```

IDE Plugin:
IXMLSpyPlugIn.OnEvent(5, ...) // nEventId = 5

Beschreibung
This event gets fired on an attempt to drop a previously dragged XMLData element on the grid view. Return *false* to prevent the data element to be moved from its original position to the drop destination position.
Dieses Event wird beim Versuch ausgelöst, ein zuvor mit der Maus gezogenes XMLData-Element in der Grid-Ansicht an dieser Stelle abzulegen. Geben Sie *false* zurück, um zu verhindern, dass das Datenelement von seiner ursprünglichen Position an eine andere Position gezogen werden kann.

14.3.2.22.1.3 OnBeforeStartEditing

Event: OnBeforeStartEditing(*objXMLData* als XMLData[593], *bEditingName*als Boolean)als Boolean

Skripting-Umgebung - VBScript:
```
Function On_BeforeStartEditing(objXMLData, bEditingName)
        ' On_BeforeStartEditing = False  ' to prohibit editing the field
End Function
```

Skripting-Umgebung - JScript:
```
function On_BeforeStartEditing(objXMLData, bEditingName)
{
        // return false;  /* to prohibit editing the field */
}
```

 © 2016-2022 Altova GmbH

IDE Plugin:
IXMLSpyPlugIn.OnEvent(1, ...) // nEventId = 1

Beschreibung
Dieses Event wird ausgelöst, bevor der Bearbeitungsmodus für eine Grid-Zelle aktiviert wird. Wenn der Parameter *bEditingName* "true" ist, wird der Namensteil des Elements bearbeitet, wenn der Wert "false" ist, wird der Werteteil bearbeitet.

14.3.2.22.1.4 OnEditingFinished

Event: OnEditingFinished(*objXMLData* als XMLData[593], *bEditingName*als Boolean)

Skripting-Umgebung - VBScript:
Function On_EditingFinished(*objXMLData*, *bEditingName*)
End Function

Skripting-Umgebung - JScript:
function On_EditingFinished(*objXMLData*, *bEditingName*)
{
}

IDE Plugin:
IXMLSpyPlugIn.OnEvent(2, ...) // nEventId = 2

Beschreibung
Dieses Event wird ausgelöst, wenn der Bearbeitungsmodus einer Grid-Zelle verlassen wird. Der Parameter *bEditingName* definiert, ob der Namensteil des Elements bearbeitet wurde.

14.3.2.22.1.5 OnFocusChanged

Event: OnFocusChanged(*objXMLData* als XMLData[593], *bSetFocus*als Boolean, *bEditingName*als Boolean)

Skripting-Umgebung - VBScript:
Function On_FocusChanged(*objXMLData*, *bSetFocus*, *bEditingName*)
End Function

Skripting-Umgebung - JScript:
function On_FocusChanged(*objXMLData*, *bSetFocus*, *bEditingName*)
{
}

IDE Plugin:
IXMLSpyPlugIn.OnEvent(3, ...) // nEventId = 3

Beschreibung

Dieses Event wird ausgelöst, wenn der Cursorfokus auf eine Grid-Zelle verschoben wird oder diese nicht mehr im Fokus ist. Wenn der Parameter *bEditingName* "*true*" ist, hat sich der Fokus auf dem Namensteil des Grid-Elements geändert. Anderfalls hat sich der Fokus auf dem Werteteil geändert.

14.3.2.22.2 CurrentFocus

Eigenschaft: CurrentFocusals XMLData 593

Beschreibung
Enthält das XML-Element mit dem aktuellen Fokus. Diese Eigenschaft ist schreibgeschützt.

14.3.2.22.3 Deselect

Methode: Deselect(*pData*als XMLData 593)

Beschreibung
Hebt die Auswahl des Elements pDatain der Grid-Ansicht auf.

14.3.2.22.4 IsVisible

Eigenschaft: IsVisibleals Boolean

Beschreibung
True, wenn die Grid-Ansicht die aktive Ansicht des Dokuments ist. Diese Eigenschaft ist schreibgeschützt.

14.3.2.22.5 Select

Methode: Select(*pData*als XMLData 593)

Beschreibung
Wählt das XML-Element pDatain der Grid-Ansicht aus.

14.3.2.22.6 SetFocus

Methode: SetFocus (*pFocusData* als XMLData 593)

Beschreibung
Setzt den Fokus auf das Element pFocusDatain der Grid-Ansicht.

© 2016-2022 Altova GmbH

14.3.2.23 SchemaDocumentationDlg

Eigenschaften und Methoden

Standard-Automation-Eigenschaften
Application 530
Parent 537

Eigenschaften bzgl. Interaktion und Sichtbarkeit
OutputFile 536
OutputFileDialogAction 536
OptionsDialogAction 536
ShowProgressBar 540
ShowResult 540

Optionen und Methoden für die Dokumentgenerierung
OutputFormat 537
EmbedDiagrams 531
DiagramFormat 531
MultipleOutputFiles 535
EmbedCSSInHTML 531
CreateDiagramsFolder 530
GenerateRelativeLinks 532

IncludeAll 532
IncludeIndex 534
IncludeGlobalAttributes 533
IncludeGlobalElements 533
IncludeLocalAttributes 534
IncludeLocalElements 534
IncludeGroups 533
IncludeComplexTypes 532
IncludeSimpleTypes 535
IncludeAttributeGroups 532
IncludeRedefines 535
IncludeReferencedSchemas 535

AllDetails 530
ShowDiagram 538
ShowNamespace 539
ShowType 541
ShowChildren 538
ShowUsedBy 542
ShowProperties 540
ShowSingleFacets 541
ShowPatterns 540
ShowEnumerations 539
ShowAttributes 538
ShowIdentityConstraints 539
ShowAnnotations 537
ShowSourceCode 541

Beschreibung
Dieses Objekt kombiniert alle Optionen für die Schemadokumentgenerierung, die auch über die Dialogfelder der Benutzeroberfläche von Authentic Desktopverfügbar sind. Die Dokumentgenerierungsoptionen werden mit den Werten initialisiert, die bei der letzten Generierung von Schemadokumentation verwendet wurden. Vor Verwendung des Objekts müssen Sie jedoch die EigenschaftSetOutputFile 536 auf einen gültigen Dateipfad setzen. Verwenden Sie OptionsDialogAction 536, OutputFileDialogAction 536 und ShowProgressBar 540 um den Grad der gewünschten Benutzerinteraktion festzulegen. Mit Hilfe von IncludeAll 532 und AllDetails 530 können Sie ganze Optionsgruppen auf einmal setzen. Durch Definition der einzelnen Eigenschaften können Sie die Optionen einzeln definieren.

14.3.2.23.1 AllDetails

Methode: AllDetails(i_bDetailsOnals Boolean)

Beschreibung
Mit Hilfe dieser Methode können Sie alle Detailoptionen aktivieren bzw. deaktivieren.

Fehler
2900 Das Objekt ist nicht mehr gültig.

14.3.2.23.2 Application

Eigenschaft: Applicationals Application 372 (schreibgeschützt)

Beschreibung
Ruft das Authentic Desktopapplication-Objekt auf.

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.3 CreateDiagramsFolder

Eigenschaft: CreateDiagramsFolderals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um ein Verzeichnis für die erstellten Bilder zu erstellen. Andernfalls werden die Diagramme neben der Dokumentation erstellt. Diese Eigenschaft steht nur zur Verfügung, wenn die Diagramme nicht eingebettet sind.Der Standardwert für die erste Ausführung ist true.

Fehler
2900 Das Objekt ist nicht mehr gültig.

© 2016-2022 Altova GmbH

2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.4 DiagramFormat

Eigenschaft: DiagramFormatals SPYImageKind 609

Beschreibung
Diese Eigenschaft definiert den generierten Diagrammbildtyp. Diese Eigenschaft steht für HTML-Dokumentation nicht zur Verfügung. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Verwendung der Eigenschaft ist PNG.

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.5 EmbedCSSInHTML

Eigenschaft: EmbedCSSInHTMLals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um die CSS-Daten in das generierte HTML-Dokument einzubetten. Andernfalls wird eine separate Datei erstellt und verknüpft.Der Standardwert für die erste Ausführung ist true

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.6 EmbedDiagrams

Eigenschaft: EmbedDiagramsals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um das Diagramm in das generierte Dokument einzubetten. Diese Eigenschaft steht für HTML-Dokumentation nicht zur Verfügung. Die Eigenschaft wird mit den Werten initialisiert, die beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurden. Der Standardwert für die erste Ausführung ist true.

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.7 GenerateRelativeLinks

Eigenschaft: GenerateRelativeLinksals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um relative Pfade zu lokalen Dateien zu erstellen. Diese Eigenschaft steht für die HTML-Dokumentation nicht zur Verfügung. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Die Standardeinstellung für die erste Ausführung ist "false".

Fehler
2900 The object is no longer valid.
2901 Invalid address for the return parameter was specified.

14.3.2.23.8 IncludeAll

Methode: IncludeAll(i_bIncludeals Boolean)

Beschreibung
Verwenden Sie diese Methode um alle Include-Optionen auszuwählen, bzw. die Auswahl für alle aufzuheben.

Fehler
2900 Das Objekt ist nicht mehr gültig.

14.3.2.23.9 IncludeAttributeGroups

Eigenschaft: IncludeAttributeGroupsals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf rue, um Attributgruppen in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.10 IncludeComplexTypes

Eigenschaft: IncludeComplexTypesals Boolean

Beschreibung

© 2016-2022 Altova GmbH

Setzen Sie diese Eigenschaft auf true, um complexTypes in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.11 IncludeGlobalAttributes

Eigenschaft: IncludeGlobalAttributesals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um globale Attribute in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.12 IncludeGlobalElements

Eigenschaft: IncludeGlobalElementsals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um globale Elemente in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.13 IncludeGroups

Eigenschaft: IncludeGroupsals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um Gruppen in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.14 IncludeIndex

Eigenschaft: IncludeIndexals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um einen Index in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.15 IncludeLocalAttributes

Eigenschaft: IncludeLocalAttributesas Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um lokale Attribute in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.16 IncludeLocalElements

Eigenschaft: IncludeLocalElementsals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um lokale Elemente in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.23.17 IncludeRedefines

Eigenschaft: IncludeRedefinesals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Redefines in die Schemadokumenation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.18 IncludeReferencedSchemas

Eigenschaft: IncludeReferencedSchemasals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um referenzierte Schemas in die Schemadokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.19 IncludeSimpleTypes

Eigenschaft: IncludeSimpleTypesals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um simpleTypes in die Schemadokumenation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.20 MultipleOutputFiles

Eigenschaft: MultipleOutputFilesals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um die Dokumentationsdateien aufzuteilen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist false.

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Es wurde zum Definieren der Eigenschaft ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.21 OptionsDialogAction

Eigenschaft: OptionsDialogActionals SPYDialogAction 607

Beschreibung
Damit Ihr Script die Standardwerte einsetzt und der Benutzer das Dialogfeld sieht, bzw. dort Einstellungen vornehmen kann, setzen Sie diese Eigenschaft auf den Wert *spyDialogUserInput(2)*. Wenn Ihr Script alle Optionen im Dialogfeld "Schemadokumentation" definieren soll, ohne dass eine Interaktion mit dem Benutzer stattfindet, verwenden Sie *spyDialogOK(0)*. Die Standardeinstellung ist *spyDialogOK.*

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Es wurde zum Definieren der Eigenschaft ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.22 OutputFile

Eigenschaft: SetOutputFileals String

Beschreibung
Der vollständige Pfad und Name der Datei, die die generierte Dokumentation enthalten soll. Bei einer HTML-Ausgabe werden zusätzliche '.png' Dateien auf Basis dieses Dateinamens generiert. Der Standardwert für diese Eigenschaft ist ein leerer String, der ersetzt werden muss, bevor Sie diesese Objekten in einem Aufruf von Document.GenerateSchemaDocumentation 479 verwenden.

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.23 OutputFileDialogAction

Eigenschaft: OutputFileDialogActionals SPYDialogAction 607

Beschreibung
Damit der Benutzer die Ausgabedatei in einem Dialogfeld zur Dateiauswahl auswählen kann, setzen Sie diese Eigenschaft auf *spyDialogUserInput(2)*. Wenn der in OutputFile 536 gespeicherte Wert verwendet werden soll

 © 2016-2022 Altova GmbH

und keinen Interaktion mit dem Benutzer vorgesehen ist, verwenden Sie *spyDialogOK(0)*. Der Standardwert ist *spyDialogOK.*

Fehler

2900 Das Objekt ist nicht mehr gültig.
2901 Es wurde zum Definieren der Eigenschaft ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.24 OutputFormat

Eigenschaft: OutputFormatals SPYSchemaDocumentationFormat 614

Beschreibung

Definiert, welche Art von Dokumentation generiert werden soll: HTML (Wert=`0`), MS-Word (Wert=`1`) oder RTF (Werte=`2`). Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist HTML.

Fehler

2900 Das Objekt ist nicht mehr gültig.
2901 Es wurde zum Definieren der Eigenschaft ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.25 Parent

Eigenschaft: Parentals Dialogs 458 (schreibgeschützt)

Beschreibung

Ruft das dem Objekt übergeordnete Objekt auf.

Fehler

2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.26 ShowAnnotations

Eigenschaft: ShowAnnotationsals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um die Annotationen zu einer Typdefinition in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.27 ShowAttributes

Eigenschaft: ShowAttributesals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um die Typdefinitionsattribute in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.28 ShowChildren

Eigenschaft: ShowChildrenals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um die untergeordneten Elemente einer Typdefinition als Links in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.29 ShowDiagram

Eigenschaft: ShowDiagramals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um die Typdefinitionen als Diagramme in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.23.30 ShowEnumerations

Eigenschaft: ShowEnumerationsals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um die in einer Typdefinition enthaltenen Enumerationen in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.31 ShowIdentityConstraints

Eigenschaft: ShowIdentityConstraints als Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um die Identity Constraints einer Typdefinition in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.32 ShowNamespace

Eigenschaft: ShowNamespaceals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um den Namespace der Typdefinitionen in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.33 ShowPatterns

Eigenschaft: ShowPatternsals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um die Patterns einer Typdefinition der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.34 ShowProgressBar

Eigenschaft: ShowProgressBarals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um das Fenster für die Dokumentgenerierung sichtbar zu machen. Verwenden Sie false, um es auszublenden. Der Standardwert ist false.

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.35 ShowProperties

Eigenschaft: ShowPropertiesals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um die Eigenschaften der Typdefinition in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.36 ShowResult

Eigenschaft: ShowResultals Boolean

Beschreibung

© 2016-2022 Altova GmbH

Setzen Sie diese Eigenschaft auf true, um das erzeugte Dokument automatisch zu öffnen, wenn die Generierung erfolgreich abgeschlossen wurde. HTML-Dokumentation wird in Authentic Desktopgeöffnet. Um Word-Dokumentation anzuzeigen, wird MS-Word gestartet. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.37 ShowSingleFacets

Eigenschaft: ShowSingleFacetsals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um die Facets einer Typdefinition in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.38 ShowSourceCode

Eigenschaft: ShowSourceCodeals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um den XML-Quellcode für die Typdefinitionen in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.39 ShowType

Eigenschaft: ShowTypeals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um den Typ von Typdefinitionen in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von

Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.40 ShowUsedBy

Eigenschaft: ShowUsedByals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um die used-by-Beziehung für Typdefinitionen in der Schemadokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateSchemaDocumentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.41 SPSFile

Eigenschaft: SPSFileals String

Beschreibung

Vollständiger Pfad der SPS-Datei, die zur Generierung der Dokumentation verwendet wird.

Fehler

2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.23.42 UseFixedDesign

Eigenschaft: UseFixedDesignals Boolean

Beschreibung

Gibt an, on die Dokumentation mit einem festgelegten Design oder mit einem Design erstellt werden soll, das durch eine SPS-Datei festgelegt wird (wobei StyleVision benötigt wird).

Fehler

2900 Das Objekt ist nicht mehr gültig.
2901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.24 SpyProject

Methoden
CloseProject 543
SaveProject 544
SaveProjectAs 544

Eigenschaften
RootItems 543
ProjectFile 543

Beschreibung
SpyProjectClass

14.3.2.24.1 CloseProject

Deklaration: CloseProject(*bDiscardChanges* als Boolean, *bCloseFiles* als Boolean, *bDialog* als Boolean)

Parameter
bDiscardChanges
Setzen Sie bDiscardChanges auf FALSE, wenn die Änderungen in den offenen Projektdateien und im Projekt gespeichert werden sollen.

bCloseFiles
Setzen Sie bCloseFiles auf TRUE, um alle offenen Projektdateien zu schließen.

bDialog
Zeigt die Dialogfelder für die Benutzereingabe an.

Beschreibung
CloseProjectschließt das aktuelle Projekt.

14.3.2.24.2 ProjectFile

Deklaration: ProjectFile als String

Beschreibung
Pfad und Dateiname des Projekts.

14.3.2.24.3 RootItems

Deklaration: RootItems als SpyProjectItems 547

Beschreibung
Root-Ebene der Sammlung der Projektelemente.

14.3.2.24.4 SaveProject

Deklaration: SaveProject

Beschreibung
SaveProjectspeichert das aktuelle Projekt.

14.3.2.24.5 SaveProjectAs

Deklaration: SaveProjectAs (strPath als String, bDialog als Boolean)

Parameter
strPath
Der vollständige Pfad mit dem Dateinamen der neuen Projektdatei.

bDialog
Wenn bDialogTRUE ist, wird ein Datei-Dialogfeld angezeigt.

Beschreibung
SaveProjectAsspeichertt die Projektdaten unter einem neuen Pfad.

14.3.2.25 SpyProjectItem

Methoden
Open 545

Eigenschaften
ChildItems 544
ParentItem 545
FileExtensions 545
ItemType 545
Name 545
Path 546
ValidateWith 546
XMLForXSLTransformation 546
XSLForXMLTransformation 546
XSLTransformationFileExtension 546
XSLTransformationFolder 546

Beschreibung
SpyProjectItemClass

14.3.2.25.1 ChildItems

Deklaration: ChildItems als SpyProjectItems 547

Beschreibung

 © 2016-2022 Altova GmbH

Wenn es sich bei dem Element um einen Ordner handelt, ist ChildItemsdie Sammlung des Ordnerinhalts.

14.3.2.25.2 FileExtensions

Deklaration: FileExtensions als String

Beschreibung
Dient zum Definieren der Dateierweiterungen, wenn das Projektelement ein Ordner ist.

14.3.2.25.3 ItemType

Deklaration: ItemType als SPYProjectItemTypes 611

Beschreibung
Die Eigenschaft ist schreibgeschützt.

14.3.2.25.4 Name

Deklaration: Name als String

Beschreibung
Name des Projektelements. Die Eigenschaft ist schreibgeschützt.

14.3.2.25.5 Open

Deklaration: Open als Document 462

Rückgabewert
Das Projektelement wird als Dokument geöffnet.

Beschreibung
Öffnet das Projektelement.

14.3.2.25.6 ParentItem

Deklaration: ParentItem als SpyProjectItem 544

Beschreibung
Das dem aktuellen Projektelement übergeordnete Element. Kann NULL (nichts) sein, wenn das Projektelement ein Element der obersten Ebene ist.

14.3.2.25.7 Path

Deklaration: Path als String

Beschreibung
Pfad des Projektelements. Diese Eigenschaft ist schreibgeschützt.

14.3.2.25.8 ValidateWith

Deklaration: ValidateWith als String

Beschreibung
Dient zum Definieren des Schemas/der DTD für die Validierung.

14.3.2.25.9 XMLForXSLTransformation

Deklaration: XMLForXSLTransformation als String

Beschreibung
Dient zum Definieren der XML-Datei für die XSL-Transformation.

14.3.2.25.10 XSLForXMLTransformation

Deklaration: XSLForXMLTransformation als String

Beschreibung
Dient zum Definieren der XSL-Datei für die XML-Transformation.

14.3.2.25.11 XSLTransformationFileExtension

Deklaration: XSLTransformationFileExtension als String

Beschreibung
Dient zum Definieren der Dateierweiterung für die Ausgabedateien der XSL-Transformation.

14.3.2.25.12 XSLTransformationFolder

Deklaration: XSLTransformationFolder als String

Beschreibung
Dient zum Definieren des Zielordners für die Ausgabedateien der XSL-Transformation.

© 2016-2022 Altova GmbH

14.3.2.26 SpyProjectItems

Methoden
AddFile 547
AddFolder 547
AddURL 547
RemoveItem 548

Eigenschaften
Count 548
Item 548

Beschreibung
SpyProjectItemsClass

14.3.2.26.1 AddFile

Deklaration: AddFile(*strPath*als String)

Parameters

strPath
Der vollständige Pfadname eines neuen Projektelements

Beschreibung
Die Methode fügt eine neue Datei zur Sammlung der Projektelemente hinzu.

14.3.2.26.2 AddFolder

Deklaration: AddFolder(*strName* als String)

Parameter

strName
Name des neuen Ordners

Beschreibung
Die Methode AddFolderfügt einen Ordner mit dem Namen strNamezur Sammlung der Projektelemente hinzu.

14.3.2.26.3 AddURL

Deklaration: AddURL(*strURL* als String, *nURLType* als SPYURLTypes 615, *strUser* als String, *strPassword* als String, *bSave* als Boolean)

Beschreibung

strURL

URL zum Öffnen als Dokument.

nURLType
Typ des zu öffnenden Dokuments. Setzen Sie die Wert zur automatischen Ermittlung auf -1.

strUser
Name des Benutzers, falls erforderlich. Kann leer bleiben.

strPassword
Passwort für die Authentifizierung. Kann leer bleiben.

bSave
Speichert die Benutzer- und Passwortinformationen.

Beschreibung
Die Methode fügt einen URL-Eintrag zur Projektsammlung hinzu.

14.3.2.26.4 Count

Deklaration: Count als long

Beschreibung
Diese Eigenschaft ruft die Anzahl der Projektelemente in der Sammlung ab. Die Eigenschaft ist schreibgeschützt.

14.3.2.26.5 Item

Deklaration: Item(*n* als long) als SpyProjectItem 544

Beschreibung
Ruft das nte Element der Sammlung der Projektelemente auf. Das erste Element hat den Index 1.

14.3.2.26.6 RemoveItem

Deklaration: RemoveItem(*pItem* als SpyProjectItem 544)

Beschreibung
`RemoveItem`löscht das Element `pItem`aus der Sammlung der Projektelemente.

14.3.2.27 TextImportExportSettings

Eigenschaften nur für den Import
ImportFile 550

Eigenschaften nur für den Export
DestinationFolder 549
FileExtension 550

 © 2016-2022 Altova GmbH

CommentIncluded 549
RemoveDelimiter 551
RemoveNewline 551

Eigenschaften für Import und Export
HeaderRow 550
FieldDelimiter 550
EnclosingCharacter 549
Encoding 549
EncodingByteOrder 550

Beschreibung
TextImportExportSettingsenthält Optionen, die Textimport- und -exportfunktionen gemeinsam haben.

14.3.2.27.1 CommentIncluded

Eigenschaft: CommentIncluded als Boolean

Beschreibung
Diese Eigenschaft legt fest, ob zusätzliche Kommentare zur generierten Textdatei hinzugefügt werden sollen. Die Standardeinstellung ist true. Diese Eigenschaft wird nur beim Export von Textdateien verwendet.

14.3.2.27.2 DestinationFolder

Eigenschaft: DestinationFolder als String

Beschreibung
Die Eigenschaft DestinationFolderlegt den Ordner fest, in dem die gespeicherten Dateien beim Textexport abgelegt werden.

14.3.2.27.3 EnclosingCharacter

Eigenschaft: EnclosingCharacter als SPYTextEnclosing 615

Beschreibung
Diese Eigenschaft definiert das Zeichen, das alle Feldwerte für den Import und Export umschließt. Der Standardwert ist spyNoEnclosing 615.

14.3.2.27.4 Encoding

Eigenschaft: Encoding als String

Beschreibung
Die Eigenschaft Encodingdefiniert die Zeichensatzcodierung für die zu importierenden und exportierenden Textdateien.

14.3.2.27.5 EncodingByteOrder

Eigenschaft: EncodingByteOrder als SPYEncodingByteOrder 608

Beschreibung
Die Eigenschaft EncodingByteOrderdefiniert die Bytefolge für Unicode-Zeichen. Standardeinstellung ist spyNONE 608.

14.3.2.27.6 FieldDelimiter

Eigenschaft: FieldDelimiter als SPYTextDelimiters 615

Beschreibung
Die Eigenschaft FieldDelimiterdefiniert die Trennzeichen zwischen den Feldern beim Import und Export. Standardeinstellung ist spyTabulator 615.

14.3.2.27.7 FileExtension

Eigenschaft: FileExtension als String

Beschreibung
Diese Eigenschaft definiert die Dateierweiterung für beim Textexport erstellte Dateien.

14.3.2.27.8 HeaderRow

Eigenschaft: HeaderRow als Boolean

Beschreibung
Die Eigenschaft HeaderRow wird beim Export und Import verwendet. Setzen Sie beim Import HeaderRow auf true, wenn die erste Zeile der Textdatei die Namen der Spalten enthält. Setzen Sie HeaderRow beim Export auf true, wenn die erste Zeile der erstellten Textdateien die Namen der Spalten enthalten soll. Der Standardwert ist true.

14.3.2.27.9 ImportFile

Eigenschaft: ImportFile als String

Beschreibung
Diese Eigenschaft dient dazu, die Textdatei für den Import zu definieren. Der String muss ein vollständig qualifizierter Pfad sein.

© 2016-2022 Altova GmbH

14.3.2.27.10 RemoveDelimiter

Eigenschaft: RemoveDelimiter als Boolean

Beschreibung
Die Eigenschaft RemoveDelimiterdefiniert, ob Zeichen im Text, die dem Trennzeichen entsprechen, entfernt werden sollen. Der Standardwert ist false. Diese Eigenschaft wird beim Export in Textdateien verwendet.

14.3.2.27.11 RemoveNewline

Eigenschaft: RemoveNewline als Boolean

Beschreibung
Die Eigenschaft RemoveNewlinedefiniert, ob Zeichen für den Beginn einer neuen Zeile im Text entfernt werden sollen. Die Standardeinstellung ist false. Die Eigenschaft wird nur beim Export in Textdateien verwendet.

14.3.2.28 TextView

Eigenschaften und Methoden

Application 553
Parent 555

LineFromPosition 554
PositionFromLine 555
LineLength 554
SelText 556
GetRangeText 553
ReplaceText 555
MoveCaret 555
GoToLineChar 553
SelectText 556
SelectionStart 556
SelectionEnd 556
Text 557
LineCount 554
Length 554

Beschreibung

14.3.2.28.1 Events

14.3.2.28.1.1 OnBeforeShowSuggestions

Event: OnBeforeShowSuggestions() als Boolean

Beschreibung
Dieses Event wird ausgelöst, bevor das Vorschlagsfenster angezeigt wird. Die Document[462]-Eigenschaft Suggestions[490] enthält einen String Array, der dem Benutzer empfohlen wird. Die angezeigten Empfehlungen können vom Benutzer während des Events bearbeitet werden. Davor muss der Eigenschaft Suggestions[490] ein leerer Array zugewiesen werden. Die beste Stelle dafür ist das OnDocumentOpened[375] Event. Damit das Vorschlagsfenster nicht angezeigt wird, geben Sie falsezurück. Um es weiter anzuzeigen, verwenden Sie den Werttrue.

Beispiele
Unten finden Sie Beispiele dafür, wie dieses Event in Form eines Scripts geschrieben werden kann.

Skripting-Umgebung - VBScript:
```
Function On_BeforeShowSuggestions()
End Function
```

Skripting-Umgebung - JScript:
```
function On_BeforeShowSuggestions()
{
}
```

IDE Plugin:
```
IXMLSpyPlugIn.OnEvent(33, ...)   // nEventId = 33
```

14.3.2.28.1.2 OnChar

Event: OnChar(nCharals Long, bExistSuggestionals Boolean) als Boolean

Beschreibung
Dieses Event wird bei jedem Tastenanschlag ausgelöst. Der Parameter nCharist die Taste, die gedrückt wurde und bExistSuggestionsgibt darüber Auskunft ob ein von Authentic Desktopgeneriertes Vorschlagsfenster nach dem Drücken dieser Taste angezeigt wird. Die Document[462] Eigenschaft Suggestions[490] enthält einen String Array, der dem Benutzer empfohlen wird. Die angezeigten Empfehlungen können vom Benutzer während des Events bearbeitet werden. Davor muss der Eigenschaft Suggestions[490] ein leerer Array zugewiesen werden. Die beste Stelle dafür ist das OnDocumentOpened[375] Event. Damit das Vorschlagsfenster nicht angezeigt wird, geben Sie false zurück. Um es weiter anzuzeigen, verwenden Sie den Wert true.
Es kann auch ein neues Vorschlagsfenster erstellt werden, wenn Authentic Desktop keines bereitstellt. Setzen Sie die Document[462] Eigenschaft Suggestions[490] auf einen String Array mit Ihren Empfehlungen und geben Sie true zurück.
Dieses Event wird vor dem OnBeforeShowSuggestions[552] Event ausgelöst. Wenn Sie festlegen, dass das Vorschlagsfenster nicht angezeigt werden soll, indem Sie den Wert auf false setzen, dann wird OnBeforeShowSuggestions[552] nicht ausgelöst.

 © 2016-2022 Altova GmbH

Beispiele
Unterhalb finden Sit Beispiele, wie ein Skript für dieses Event erstellt werden kann.

Skripting-Umgebung - VBScript:
```
Function On_Char(nChar, bExistSuggestions)
End Function
```

Skripting-Umgebung - JScript:
```
function On_Char(nChar, bExistSuggestions)
{
}
```

IDE Plugin:
```
IXMLSpyPlugIn.OnEvent(35, ...)   // nEventId = 35
```

14.3.2.28.2 Application

Eigenschaft: Applicationals Application 372 (schreibgeschützt)

Beschreibung
Ruft das Authentic Desktopapplication-Objekt auf.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.28.3 GetRangeText

Methode: GetRangeText(nStartals Long, nEndals Long)als String

Beschreibung
Gibt den Text im angegebenen Bereich zurück.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.28.4 GoToLineChar

Methode: GoToLineChar(nLineals Long, nCharals Long)

Beschreibung
Verschiebt den Cursor an die angegebene Zeilen- und Zeichenposition.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.28.5 Length

Eigenschaft: Lengthals Long

Beschreibung
Gibt die Anzahl der Zeichen des Dokuments zurück.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.28.6 LineCount

Eigenschaft: LineCountals Long

Beschreibung
Gibt die Anzahl der Zeilen im Dokument zurück.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.28.7 LineFromPosition

Methode: LineFromPosition(nCharPosals Long)als Long

Beschreibung
Gibt die Zeilennummer der Zeichenposition zurück.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.28.8 LineLength

Methode: LineLength(nLineals Long)als Long

Beschreibung
Gibt die Länge der Zeile zurück.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.28.9 MoveCaret

Methode: MoveCaret(nDiffals Long)

Beschreibung
Verschiebt den Cursor um nDiffZeichen.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.28.10 Parent

Eigenschaft: Parentals Document 462 (schreibgeschützt)

Beschreibung
Ruft das dem Objekt übergeordnete Objekt auf.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.28.11 PositionFromLine

Methode: PositionFromLine(nLineals Long) als Long

Beschreibung
Gibt die Anfangsposition der Zeile zurück.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.28.12 ReplaceText

Methode: ReplaceText(nPosFromals Long, nPosTillals Long, sTextals String)

Beschreibung
Ersetzt den Text im angegebenen Bereich.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.28.13 SelectionEnd

Eigenschaft: SelectionEndals Long

Beschreibung
Gibt die Endposition der Textauswahl zurück bzw. definiert diese.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.28.14 SelectionStart

Eigenschaft: SelectionStartals Long

Beschreibung
Gibt die Anfangsposition der Textauswahl zurück bzw. definiert diese.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.28.15 SelectText

Methode: SelectText(nPosFromals Long, nPosTillals Long)

Beschreibung
Wählt den Text im angegebenen Bereich aus.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.28.16 SelText

Eigenschaft: SelTextals String

Beschreibung
Gibt den ausgewählten Text zurück bzw. definiert diesen.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.28.17 Text

Eigenschaft: Textals String

Beschreibung
Gibt den Dokumenttext zurück bzw. definiert diesen.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29 WSDLDocumentationDlg

Eigenschaften und Methoden

Standard-Automation-Eigenschaften
Application 558
Parent 564

Eigenschaften zur Interaktion und Sichtbarkeit
OptionsDialogAction 563
OutputFile 563
OutputFileDialogAction 563
ShowProgressBar 567
ShowResult 567

Optionen und Methoden zur Dokumentgenerierung
OutputFormat 564
UseFixedDesign 569
SPSFile 569
EmbedDiagrams 559
DiagramFormat 559
MultipleOutputFiles 562
EmbedCSSInHTML 559
CreateDiagramsFolder 558

IncludeAll 560
IncludeBinding 560
IncludeImportedWSDLFiles 561
IncludeMessages 561
IncludeOverview 561
IncludePortType 562
IncludeService 562
IncludeTypes 562

AllDetails 558
ShowBindingDiagram 565
ShowExtensibility 565
ShowMessageParts 565
ShowPort 566

ShowPortTypeDiagram 566
ShowPortTypeOperations 566
ShowServiceDiagram 567
ShowSourceCode 568
ShowTypesDiagram 568
ShowUsedBy 568

Beschreibung
Dieses Objekt kombiniert alle Optionen für die WSDL-Dokumentgenerierung, die über Dialogfelder der Benutzeroberfläche von Authentic Desktopzur Verfügung stehen. Die Dokumentgenerierungsoptionen werden mit den Werten initialisiert, die bei der letzten Generierung von WSLD-Dokumentation verwendet wurden. Bevor Sie das Objekt jedoch verwenden, müssen Sie dieOutputFile 563 Eigenschaft auf einen gültigen Dateipfad setzen. Mit Hilfe von OptionsDialogAction 563, OutputFileDialogAction 563 und ShowProgressBar 567 können Sie den Grad der gewünschten Benutzerinteraktion definieren. Mit Hilfe von IncludeAll 560 und AllDetails 558 können Sie ganze Optionsgruppen auf einmal definieren. Oder Sie definieren die Eigenschaften einzeln, um die Feineinstellung vorzunehmen.

14.3.2.29.1 AllDetails

Methode: AllDetails(i_bDetailsOnals Boolean)

Beschreibung
Mit Hilfe dieser Methode können Sie alle detaillierten Optionen aktivieren bzw. deaktivieren.

Fehler
3900 Das Objekt ist nicht mehr gültig.

14.3.2.29.2 Application

Eigenschaft: Applicationals Application 372 (schreibgeschützt)

Beschreibung
Ruft das Authentic Desktopapplication-Objekt auf.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.3 CreateDiagramsFolder

Eigenschaft: CreateDiagramsFolderals Boolean

Beschreibung

 © 2016-2022 Altova GmbH

Setzen Sie diese Eigenschaft auf true, um ein Verzeichnis für die erstellten Bilder zu erstellen. Andernfalls werden die Diagramme neben der Dokumentation erstellt. Diese Eigenschaft steht nur zur Verfügung, wenn die Diagramme nicht eingebettet sind. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.4 DiagramFormat

Eigenschaft: DiagramFormatals SPYImageKind 609

Beschreibung

Diese Eigenschaft definiert den generierten Diagrammbildtyp. Diese Eigenschaft steht für HTML-Dokumentation nicht zur Verfügung. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Die Standardeinstellung für die erste Ausführung ist PNG.

Fehler

3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.5 EmbedCSSInHTML

Eigenschaft: EmbedCSSInHTMLals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um die CSS-Daten in das generierte HTML-Dokument einzubetten. Andernfalls wird eine separate Datei erstellt und verknüpft. Diese Eigenschaft steht nur für die HTML-Dokumentation zur Verfügung. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true

Fehler

3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.6 EmbedDiagrams

Eigenschaft: EmbedDiagramsals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um das Diagramm in das generierte Dokument einzubetten. Diese Eigenschaft steht für HTML-Dokumentation nicht zur Verfügung. Die Eigenschaft wird mit den Werten

initialisiert, die beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurden. Der Standardwert für die erste Ausführung ist true.

Fehler

3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.7 GlobalElementsAndTypesOnly

Eigenschaft: GlobalElementsAndTypesOnlyals Boolean

Beschreibung

Gibt einen Wert zurück bzw. definiert einen Wert, der angibt, ob eine komplette Schemadokumentation vorgenommen wird, oder ob nur globale Elemente und Typen dokumentiert werden.

Fehler

3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.8 IncludeAll

Methode: IncludeAll(i_bIncludeals Boolean)

Beschreibung

Verwenden Sie diese Methode um alle Include-Optionen auszuwählen, bzw. die Auswahl für alle aufzuheben.

Fehler

3900 Das Objekt ist nicht mehr gültig.

14.3.2.29.9 IncludeBinding

Eigenschaft: IncludeBindingals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um Bindings in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.29.10 IncludeImportedWSDLFiles

Eigenschaft: IncludeImportedWSDLFilesals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um importierte WSDL-Dateien in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.11 IncludeMessages

Eigenschaft: IncludeMessagesals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Messages in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.12 IncludeOverview

Eigenschaft: IncludeOverviewals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um eine Übersicht in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.13 IncludePortType

Eigenschaft: IncludePortTypeals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Port Types in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.14 IncludeService

Eigenschaft: IncludeServiceals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Services in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.15 IncludeTypes

Eigenschaft: IncludeTypesals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Typen in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.16 MultipleOutputFiles

Eigenschaft: MultipleOutputFilesals Boolean

Beschreibung

 © 2016-2022 Altova GmbH

Setzen Sie diese Eigenschaft auf true, um die Dokumentationsdateien aufzuteilen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist false.

Fehler

3900 Das Objekt ist nicht mehr gültig.
3901 Zum Definieren der Eigenschaft wurde ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.17 OptionsDialogAction

Eigenschaft: OptionsDialogActionals SPYDialogAction 607

Beschreibung

Damit Ihr Script die Standardwerte ausfüllen kann und der Benutzer das Dialogfeld sieht und darauf reagieren kann, setzen Sie diese Eigenschaft auf *spyDialogUserInput(2)*. Wenn alle Optionen im Dialogfeld "Schemadokumentation" vom Script definiert werden sollen und keine Interaktion mit dem Benutzer vorgesehen ist, verwenden Sie *spyDialogOK(0)*. Der Standardwert ist *spyDialogOK.*

Fehler

3900 Das Objekt ist nicht mehr gültig.
3901 Zum Definieren der Eigenschaft wurde ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.18 OutputFile

Eigenschaft: OutputFileals String

Beschreibung

Der vollständige Pfad und Name der Datei, die die generierte Dokumentation enthalten soll. Bei einer HTML-Ausgabe werden zusätzliche '.png' Dateien auf Basis dieses Dateinamens generiert. Der Standardwert für diese Eigenschaft ist ein leerer String, der ersetzt werden muss, bevor Sie diesese Objekten in einem Aufruf von Document.GenerateWSDLDocumentation 480 verwenden.

Fehler

3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.19 OutputFileDialogAction

Eigenschaft: OutputFileDialogActionals SPYDialogAction 607

Beschreibung

Damit der Benutzer die Ausgabedatei in einem Dialogfeld zur Dateiauswahl auswählen kann, setzen Sie diese Eigenschaft auf *spyDialogUserInput(2)*. Wenn der in OutputFile 563 gespeicherte Wert verwendet werden soll und keinen Interaktion mit dem Benutzer vorgesehen ist, verwenden Sie *spyDialogOK(0)*. Der Standardwert ist *spyDialogOK.*

Fehler

3900 Das Objekt ist nicht mehr gültig.
3901 Zum Definieren der Eigenschaft wurde ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.20 OutputFormat

Eigenschaft: OutputFormatals SPYSchemaDocumentationFormat 614

Beschreibung

Definiert, welche Art von Dokumentation generiert werden soll: HTML (Wert=`0`), MS-Word (Wert=`1`) oder RTF (Werte=`2`). Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist HTML.

Fehler

3900 Das Objekt ist nicht mehr gültig.
3901 Zum Definieren der Eigenschaft wurde ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.21 Parent

Eigenschaft: Parentals Dialogs 458 (schreibgeschützt)

Beschreibung

Ruft das dem Objekt übergeordnete Objekt auf.

Fehler

3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.22 SeparateSchemaDocument

Eigenschaft: SeparateSchemaDocumentals Boolean

Beschreibung

Gibt einen Wert zurück bzw. definiert einen Wert, der angibt, ob die Schemadokumentation in ein separates Dokument geschrieben werden soll.

Fehler

3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.29.23 ShowBindingDiagram

Eigenschaft: ShowBindingDiagramals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Binding-Diagramme in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.24 ShowExtensibility

Eigenschaft: ShowExtensibilityals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Service- und Binding-Erweiterungen der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.25 ShowMessageParts

Eigenschaft: ShowMessagePartsals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Message-Teile von Messagesn der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.26 ShowPort

Eigenschaft: ShowPortals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Service Ports in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.27 ShowPortTypeDiagram

Eigenschaft: ShowPortTypeDiagramals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Port Type-Diagramme in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.28 ShowPortTypeOperations

Eigenschaft: ShowPortTypeOperationsas Boolean

Beschreibung
Set this property to true, to show port type operations in the WSDL documentation. The property is initialized with the value used during the last call to Document.GenerateWSDLDocumentation 480. Der Standardwert für die erste Ausführung ist true.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.29.29 ShowProgressBar

Eigenschaft: ShowProgressBarals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um das Fenster, in dem die Fortschrittsanzeige für die Dokumentgeneration angezeigt wird, sichtbar zu machen. Mit falsewird es ausgeblendet. Die Standardeinstellung ist false.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.30 ShowResult

Eigenschaft: ShowResultals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um das erzeugte Dokument automatisch zu öffnen, wenn die Generierung erfolgreich abgeschlossen wurde. HTML-Dokumentation wird in Authentic Desktopgeöffnet. Um Word-Dokumentation anzuzeigen, wird MS-Word gestartet. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.31 ShowServiceDiagram

Eigenschaft: ShowServiceDiagramas Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Service-Diagramme in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.32 ShowSourceCode

Eigenschaft: ShowSourceCodeas Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Quellcode für die Includes in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.33 ShowTypesDiagram

Eigenschaft: ShowTypesDiagramas Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Typ-Diagramme in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.34 ShowUsedBy

Eigenschaft: ShowUsedByas Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um used-by-Beziehungen für Types, Bindings und Message-Definitionen in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.29.35 UseFixedDesign

Eigenschaft: UseFixedDesignals Boolean

Beschreibung
Gibt an, on die Dokumentation mit einem festgelegten Design oder mit einem Design erstellt werden soll, das durch eine SPS-Datei festgelegt wird (wobei StyleVision benötigt wird).

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.29.36 SPSFile

Eigenschaft: SPSFileals String

Beschreibung
Vollständiger Pfad der SPS-Datei, die zur Generierung der Dokumentation verwendet wird.

Fehler
3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30 WSDL20DocumentationDlg

Eigenschaften und Methoden

Standard-Automation-Eigenschaften
Application 571
Parent 576

Eigenschaften zur Interaktion und Sichtbarkeit
OptionsDialogAction 575
OutputFile 575
OutputFileDialogAction 576
ShowProgressBar 579
ShowResult 579

Optionen und Methoden zur Dokumentgenerierung
OutputFormat 576
UseFixedDesign 581
SPSFile 581
EmbedDiagrams 572
DiagramFormat 571
MultipleOutputFiles 575
EmbedCSSInHTML 571

CreateDiagramsFolder 571

IncludeAll 572
IncludeBinding 573
IncludeImportedWSDLFiles 573
IncludeInterface 573
IncludeOverview 574
IncludeService 574
IncludeTypes 574

AllDetails 570
ShowBindingDiagram 577
ShowExtensibility 577
ShowEndpoint 577
ShowFault 578
ShowInterfaceDiagram 578
ShowOperation 578
ShowServiceDiagram 579
ShowSourceCode 580
ShowTypesDiagram 580
ShowUsedBy 580

Beschreibung
Dieses Objekt kombiniert alle Optionen für die WSDL-Dokumentgenerierung, die über Dialogfelder der Benutzeroberfläche von Authentic Desktopzur Verfügung stehen. Die Dokumentgenerierungsoptionen werden mit den Werten initialisiert, die bei der letzten Generierung von WSLD-Dokumentation verwendet wurden. Bevor Sie das Objekt jedoch verwenden, müssen Sie dieOutputFile 575 Eigenschaft auf einen gültigen Dateipfad setzen. Mit Hilfe von OptionsDialogAction 575, OutputFileDialogAction 576 und ShowProgressBar 579 können Sie den Grad der gewünschten Benutzerinteraktion definieren. Mit Hilfe von IncludeAll 572 und AllDetails 570 können Sie ganze Optionsgruppen auf einmal definieren. Oder Sie definieren die Eigenschaften einzeln, um die Feineinstellung vorzunehmen.

14.3.2.30.1 AllDetails

Methode: AllDetails(i_bDetailsOnals Boolean)

Beschreibung
Mit Hilfe dieser Methode können Sie alle detaillierten Optionen aktivieren bzw. deaktivieren.

Fehler
4300 Das Objekt ist nicht mehr gültig.

© 2016-2022 Altova GmbH

14.3.2.30.2 Application

Eigenschaft: Applicationals Application[372](schreibgeschützt)

Beschreibung
Ruft das Authentic Desktopapplication-Objekt auf.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.3 CreateDiagramsFolder

Eigenschaft: CreateDiagramsFolderals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um ein Verzeichnis für die erstellten Bilder zu erstellen. Andernfalls werden die Diagramme neben der Dokumentation erstellt. Diese Eigenschaft steht nur zur Verfügung, wenn die Diagramme nicht eingebettet sind. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation[479] verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.4 DiagramFormat

Eigenschaft: DiagramFormatals SPYImageKind[609]

Beschreibung
Diese Eigenschaft definiert den generierten Diagrammbildtyp. Diese Eigenschaft steht für HTML-Dokumentation nicht zur Verfügung. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation[479]verwendet wurde. Die Standardeinstellung für die erste Ausführung ist PNG.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.5 EmbedCSSInHTML

Eigenschaft: EmbedCSSInHTMLals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um die CSS-Daten in das generierte HTML-Dokument einzubetten. Andernfalls wird eine separate Datei erstellt und verknüpft. Diese Eigenschaft steht nur für die HTML-Dokumentation zur Verfügung. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true

Fehler

4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.6 EmbedDiagrams

Eigenschaft: EmbedDiagramsals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um das Diagramm in das generierte Dokument einzubetten. Diese Eigenschaft steht für HTML-Dokumentation nicht zur Verfügung. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.7 GlobalElementsAndTypesOnly

Eigenschaft: GlobalElementsAndTypesOnlyals Boolean

Beschreibung

Gibt einen Wert zurück bzw. definiert einen Wert, der angibt, ob eine komplette Schemadokumentation vorgenommen wird, oder ob nur globale Elemente und Typen dokumentiert werden.

Fehler

3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.8 IncludeAll

Methode: IncludeAll(i_bIncludeals Boolean)

Beschreibung

Verwenden Sie diese Methode um alle Include-Optionen auszuwählen, bzw. die Auswahl für alle aufzuheben.

Fehler

4300 Das Objekt ist nicht mehr gültig.

© 2016-2022 Altova GmbH

14.3.2.30.9 IncludeBinding

Eigenschaft: IncludeBindingals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Bindings in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.10 IncludeImportedWSDLFiles

Eigenschaft: IncludeImportedWSDLFilesals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um importierte WSDL-Dateien in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.11 IncludeInterface

Eigenschaft: IncludeInterfaceals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Interfaces in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.12 IncludeOverview

Eigenschaft: IncludeOverviewals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um eine Übersicht in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.13 IncludeService

Eigenschaft: IncludeServiceals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Services in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.14 IncludeTypes

Eigenschaft: IncludeTypesals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Types in die WSDL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.30.15 MultipleOutputFiles

Eigenschaft: MultipleOutputFilesals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um die Dokumentationsdateien aufzuteilen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist false.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Zum Definieren der Eigenschaft wurde ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.16 OptionsDialogAction

Eigenschaft: OptionsDialogActionals SPYDialogAction 607

Beschreibung
Damit Ihr Script die Standardwerte ausfüllen kann und der Benutzer das Dialogfeld sieht und darauf reagieren kann, setzen Sie diese Eigenschaft auf *spyDialogUserInput(2)*. Wenn alle Optionen im Dialogfeld "Schemadokumentation" vom Script definiert werden sollen und keine Interaktion mit dem Benutzer vorgesehen ist, verwenden Sie *spyDialogOK(0)*. Der Standardwert ist *spyDialogOK.*

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Zum Definieren der Eigenschaft wurde ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.17 OutputFile

Eigenschaft: OutputFileals String

Beschreibung
Der vollständige Pfad und Name der Datei, die die generierte Dokumentation enthalten soll. Bei einer HTML-Ausgabe werden zusätzliche '.png' Dateien auf Basis dieses Dateinamens generiert. Der Standardwert für diese Eigenschaft ist ein leerer String, der ersetzt werden muss, bevor Sie diesese Objekten in einem Aufruf von Document.GenerateWSDL20Documentation 479 verwenden.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.18 OutputFileDialogAction

Eigenschaft: OutputFileDialogActionals SPYDialogAction 607

Beschreibung
Damit der Benutzer die Ausgabedatei in einem Dialogfeld zur Dateiauswahl auswählen kann, setzen Sie diese Eigenschaft auf *spyDialogUserInput(2)*. Wenn der in OutputFile 575 gespeicherte Wert verwendet werden soll und keinen Interaktion mit dem Benutzer vorgesehen ist, verwenden Sie *spyDialogOK(0)*. Der Standardwert ist *spyDialogOK*.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Zum Definieren der Eigenschaft wurde ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.19 OutputFormat

Eigenschaft: OutputFormatals SPYSchemaDocumentationFormat 614

Beschreibung
Definiert, welche Art von Dokumentation generiert werden soll: HTML (Wert=`0`), MS-Word (Wert=`1`) oder RTF (Werte=`2`). Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist HTML.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Zum Definieren der Eigenschaft wurde ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.20 Parent

Eigenschaft: Parentals Dialogs 458 (schreibgeschützt)

Beschreibung
Ruft das dem Objekt übergeordnete Objekt auf.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.21 SeparateSchemaDocument

Eigenschaft: SeparateSchemaDocumentals Boolean

Beschreibung

© 2016-2022 Altova GmbH

Gibt einen Wert zurück bzw. definiert einen Wert, der angibt, ob die Schemadokumentation in ein separates Dokument geschrieben werden soll.

Fehler

3900 Das Objekt ist nicht mehr gültig.
3901 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.22 ShowBindingDiagram

Eigenschaft: ShowBindingDiagramals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um Binding-Diagramme in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.23 ShowEndpoint

Eigenschaft: ShowEndpointals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um Sercice Endpoints in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.24 ShowExtensibility

Eigenschaft: ShowExtensibilityals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um Service- und Binding-Erweiterungen der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.25 ShowFault

Eigenschaft: ShowFaultals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Fault-Elemente in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.26 ShowInterfaceDiagram

Eigenschaft: ShowInterfaceDiagramals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Interface-Diagramme in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.27 ShowOperation

Eigenschaft: ShowOperationals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Interface- und Binding-Operationen in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.30.28 ShowProgressBar

Eigenschaft: ShowProgressBarals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, damit das Fenster mit der Fortschrittsanzeige für die Dokumentgenerierung angezeigt wird. Verwenden Sie `false`, um es auszublenden. Der Standardwert für ist `false`.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.29 ShowResult

Eigenschaft: ShowResultals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um das erzeugte Dokument automatisch zu öffnen, wenn die Generierung erfolgreich abgeschlossen wurde. HTML-Dokumentation wird in Authentic Desktopgeöffnet. Um Word-Dokumentation anzuzeigen, wird MS-Word gestartet. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.30 ShowServiceDiagram

Eigenschaft: ShowServiceDiagramas Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Service-Diagramme in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.31 ShowSourceCode

Eigenschaft: ShowSourceCodeas Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Quellcode für die Includes in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.32 ShowTypesDiagram

Eigenschaft: ShowTypesDiagramas Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Typ-Diagramme in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.33 ShowUsedBy

Eigenschaft: ShowUsedByas Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um used-by-Beziehungen für Types, Bindings und Message-Definitionen in der WSDL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateWSDL20Documentation 479 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.30.34 SPSFile

Eigenschaft: SPSFileals String

Beschreibung
Vollständiger Pfad der SPS-Datei, die zur Generierung der Dokumentation verwendet wird.

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.30.35 UseFixedDesign

Eigenschaft: UseFixedDesignals Boolean

Beschreibung
Gibt an, on die Dokumentation mit einem festgelegten Design oder mit einem Design erstellt werden soll, das durch eine SPS-Datei festgelegt wird (wobei StyleVision benötigt wird).

Fehler
4300 Das Objekt ist nicht mehr gültig.
4301 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31 XBRLDocumentationDlg

Eigenschaften und Methoden

Standard-Automation-Eigenschaften
Application 583
Parent 588

Eigenschaften zur Interaktion und Sichtbarkeit
OptionsDialogAction 586
OutputFile 587
OutputFileDialogAction 587
ShowProgressBar 591
ShowResult 591

Optionen und Methoden zur Dokumentgenerierung
OutputFormat 587
EmbedDiagrams 584
DiagramFormat 583
EmbedCSSInHTML 583
CreateDiagramsFolder 583

IncludeAll 584

IncludeOverview 586
IncludeNamespacePrefixes 585
IncludeGlobalElements 585
IncludeDefinitionLinkroles 585
IncludePresentationLinkroles 586
IncludeCalculationLinkroles 584

AllDetails 582
ShowDiagram 589
ShowSubstitutiongroup 592
ShowItemtype 589
ShowBalance 588
ShowPeriod 591
ShowAbstract 588
ShowNillable 590
ShowLabels 590
ShowReferences 591
ShowLinkbaseReferences 590

ShortQualifiedName 588
ShowImportedElements 589

Beschreibung
Dieses Objekt kombiniert alle Optionen für die XBRL-Dokumentgenerierung, die über Dialogfelder der Benutzeroberfläche von Authentic Desktopzur Verfügung stehen. Die Dokumentgenerierungsoptionen werden mit den Werten initialisiert, die bei der letzten Generierung von WSLD-Dokumentation verwendet wurden. Bevor Sie das Objekt jedoch verwenden, müssen Sie dieOutputFile 587 Eigenschaft auf einen gültigen Dateipfad setzen. Mit Hilfe von OptionsDialogAction 586, OutputFileDialogAction 587 und ShowProgressBar 591 können Sie den Grad der gewünschten Benutzerinteraktion definieren. Mit Hilfe von IncludeAll 584 und AllDetails 582 können Sie ganze Optionsgruppen auf einmal definieren. Oder Sie definieren die Eigenschaften einzeln, um die Feineinstellung vorzunehmen.

14.3.2.31.1 AllDetails

Methode: AllDetails(i_bDetailsOnals Boolean)

Beschreibung
Mit Hilfe dieser Methode können Sie alle detaillierten Optionen aktivieren bzw. deaktivieren.

Fehler
4400 Das Objekt ist nicht mehr gültig.

© 2016-2022 Altova GmbH

14.3.2.31.2 Application

Eigenschaft: Applicationals Application[372] (schreibgeschützt)

Beschreibung
Ruft das Authentic Desktopapplication-Objekt auf.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.3 CreateDiagramsFolder

Eigenschaft: CreateDiagramsFolderals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um ein Verzeichnis für die erstellten Bilder zu erstellen. Andernfalls werden die Diagramme neben der Dokumentation erstellt. Diese Eigenschaft steht nur zur Verfügung, wenn die Diagramme nicht eingebettet sind. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation[480] verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.4 DiagramFormat

Eigenschaft: DiagramFormatals SPYImageKind[609]

Beschreibung
Diese Eigenschaft definiert den generierten Diagrammbildtyp. Diese Eigenschaft steht für HTML-Dokumentation nicht zur Verfügung. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation[480] verwendet wurde. Die Standardeinstellung für die erste Ausführung ist PNG.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.5 EmbedCSSInHTML

Eigenschaft: EmbedCSSInHTMLals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um die CSS-Daten in das generierte HTML-Dokument einzubetten. Andernfalls wird eine separate Datei erstellt und verknüpft. Diese Eigenschaft steht nur für die HTML-

Dokumentation zur Verfügung. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true

Fehler

4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.6 EmbedDiagrams

Eigenschaft: EmbedDiagramsals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um das Diagramm in das generierte Dokument einzubetten. Diese Eigenschaft steht für HTML-Dokumentation nicht zur Verfügung. Die Eigenschaft wird mit den Werten initialisiert, die beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurden. Der Standardwert für die erste Ausführung ist true.

Fehler

4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.7 IncludeAll

Methode: IncludeAll(i_bIncludeals Boolean)

Beschreibung

Verwenden Sie diese Methode um alle Include-Optionen auszuwählen, bzw. die Auswahl für alle aufzuheben.

Fehler

4400 Das Objekt ist nicht mehr gültig.

14.3.2.31.8 IncludeCalculationLinkroles

Eigenschaft: IncludeCalculationLinkrolesals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um Calculation Linkroles in die XBRL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

© 2016-2022 Altova GmbH

4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.9 IncludeDefinitionLinkroles

Eigenschaft: IncludeDefinitionLinkrolesals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Definition Linkroles in die XBRL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.10 IncludeGlobalElements

Eigenschaft: IncludeGlobalElementsals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um globale Elemente in die XBRL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.11 IncludeNamespacePrefixes

Eigenschaft: IncludeNamespacePrefixesals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Namespace-Präfixe in die XBRL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.12 IncludeOverview

Eigenschaft: IncludeOverviewals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um eine Übersicht in die XBRL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.13 IncludePresentationLinkroles

Eigenschaft: IncludePresentationLinkrolesals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Presentation Linkroles in die XBRL-Dokumentation zu inkludieren. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.14 OptionsDialogAction

Eigenschaft: OptionsDialogActionals SPYDialogAction 607

Beschreibung
Damit Ihr Script die Standardwerte ausfüllen kann und der Benutzer das Dialogfeld sieht und darauf reagieren kann, setzen Sie diese Eigenschaft auf *spyDialogUserInput(2)*. Wenn alle Optionen im Dialogfeld "Schemadokumentation" vom Script definiert werden sollen und keine Interaktion mit dem Benutzer vorgesehen ist, verwenden Sie *spyDialogOK(0)*. Der Standardwert ist *spyDialogOK*.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Zum Definieren der Eigenschaft wurde ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.31.15 OutputFile

Eigenschaft: OutputFileals String

Beschreibung
Der vollständige Pfad und Name der Datei, die die generierte Dokumentation enthalten soll. Bei einer HTML-Ausgabe werden zusätzliche '.png' Dateien auf Basis dieses Dateinamens generiert. Der Standardwert für diese Eigenschaft ist ein leerer String, der ersetzt werden muss, bevor Sie diesese Objekten in einem Aufruf von Document.GenerateXBRLDocumentation 480 verwenden.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.16 OutputFileDialogAction

Eigenschaft: OutputFileDialogActionals SPYDialogAction 607

Beschreibung
Damit der Benutzer die Ausgabedatei in einem Dialogfeld zur Dateiauswahl auswählen kann, setzen Sie diese Eigenschaft auf *spyDialogUserInput(2)*. Wenn der in OutputFile 587 gespeicherte Wert verwendet werden soll und keinen Interaktion mit dem Benutzer vorgesehen ist, verwenden Sie *spyDialogOK(0)*. Der Standardwert ist *spyDialogOK.*

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Zum Definieren der Eigenschaft wurde ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.17 OutputFormat

Eigenschaft: OutputFormatals SPYSchemaDocumentationFormat 614

Beschreibung
Definiert, welche Art von Dokumentation generiert werden soll: HTML (Wert=`0`), MS-Word (Wert=`1`) oder RTF (Werte=`2`). Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist HTML.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Zum Definieren der Eigenschaft wurde ein ungültiger Wert verwendet.
Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.18 Parent

Eigenschaft: Parentals Dialogs 458 (schreibgeschützt)

Beschreibung
Ruft das dem Objekt übergeordnete Objekt auf.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.19 ShortQualifiedName

Eigenschaft: ShortQualifiedNameals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um in der XBRL-Dokumentation kurze qualifizierte Namen zu verwenden. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.20 ShowAbstract

Eigenschaft: ShowAbstractals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um abstrakte Elemente in der XBRL-Dokumentation zu verwenden. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.21 ShowBalance

Eigenschaft: ShowBalanceals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um balance-Elemente in der XBRL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von

 © 2016-2022 Altova GmbH

Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.22 ShowDiagram

Eigenschaft: ShowDiagramals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um Diagramme in der XBRL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.23 ShowImportedElements

Eigenschaft: ShowImportedElementsals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um importierte Elemente in der XBRL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.24 ShowItemtype

Eigenschaft: ShowItemtypeals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um Item Types in der XBRL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.25 ShowLabels

Eigenschaft: ShowLabelsals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Labels in der XBRL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.26 ShowLinkbaseReferences

Eigenschaft: ShowLinkbaseReferencesals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Linkbase-Referenzen in der XBRL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.27 ShowNillable

Eigenschaft: ShowNillableals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um nillable-Eigenschaften in der XBRL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.31.28 ShowPeriod

Eigenschaft: ShowPeriodals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um periods (Zeitdauer) in der XBRL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.29 ShowProgressBar

Eigenschaft: ShowProgressBarals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um das Fenster, in dem die Fortschrittsanzeige für die Dokumentgeneration angezeigt wird, sichtbar zu machen. Mit falsewird es ausgeblendet. Die Standardeinstellung ist false.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.30 ShowReferences

Eigenschaft: ShowReferencesals Boolean

Beschreibung
Setzen Sie diese Eigenschaft auf true, um Referenzen in der XBRL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler
4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.31 ShowResult

Eigenschaft: ShowResultals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um das erzeugte Dokument automatisch zu öffnen, wenn die Generierung erfolgreich abgeschlossen wurde. HTML-Dokumentation wird in Authentic Desktopgeöffnet. Um Word-Dokumentation anzuzeigen, wird MS-Word gestartet. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.32 ShowSubstitutiongroup

Eigenschaft: ShowSubstitutiongroupals Boolean

Beschreibung

Setzen Sie diese Eigenschaft auf true, um Substitution Groups in der XBRL-Dokumentation anzuzeigen. Die Eigenschaft wird mit dem Wert initialisiert, der beim letzten Aufruf von Document.GenerateXBRLDocumentation 480 verwendet wurde. Der Standardwert für die erste Ausführung ist true.

Fehler

4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.33 SPSFile

Eigenschaft: SPSFileals String

Beschreibung

Vollständiger Pfad der SPS-Datei, die zur Generierung der Dokumentation verwendet wird.

Fehler

4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.31.34 UseFixedDesign

Eigenschaft: UseFixedDesignals Boolean

Beschreibung

Gibt an, on die Dokumentation mit einem festgelegten Design oder mit einem Design erstellt werden soll, das durch eine SPS-Datei festgelegt wird (wobei StyleVision benötigt wird).

Fehler

4400 Das Objekt ist nicht mehr gültig.
4401 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.32 XMLData

Eigenschaften
Kind 603
Name 603
TextValue 604

HasChildren 601
MayHaveChildren 603
Parent 603

Methoden
GetFirstChild 598
GetNextChild 599
GetCurrentChild 598

InsertChild 601
InsertChildAfter 602
InsertChildBefore 602
AppendChild 594

EraseAllChildren 595
EraseChild 595
EraseCurrentChild 596

IsSameNode 602

CountChildren 594
CountChildrenKind 595

GetChild 597
GetChildAttribute 597
GetChildElement 597
GetChildKind 598
GetNamespacePrefixForURI 599

HasChildrenKind 601
SetTextValueXMLEncoded 604

Beschreibung
Über die XMLData-Schnittstelle haben Sie auf XML-Ebene direkten Zugriff auf das Dokument. Sie können die XML-Darstellung des Dokuments anzeigen und direkt bearbeiten. Beachten Sie jedoch die folgenden Einschränkungen:

- Die XMLData-Darstellung ist nur gültig, wenn das Dokument in der Grid-Ansicht oder der Authentic-Ansicht angezeigt wird.
- In der Authentic-Ansicht werden zusätzliche XMLData-Elemente automatisch als übergeordnete Elemente der einzelnen sichtbaren Dokumentelemente eingefügt. Normalerweise handelt es sich dabei um ein XMLData-Element von der Art spyXMLDataElement,wobei die Name 603 Eigenschaft auf 'Text' gesetzt ist.

- Wenn Sie die XMLData-Schnittstelle von einem anderen Ansichtsmodus aus verwenden, werden zwar keine Fehler angezeigt, aber die Änderungen werden in der Ansicht nicht angezeigt und gehen u.U. beim nächsten Ansichtswechsel verloren.

Beachten Sie bitte außerdem:

- Das Definieren eines neuen Textwerts für ein XML-Element ist möglich, wenn das Element keine Unterelemente hat, die keinen Text enthalten. Ein Textwert kann sogar definiert werden, wenn das Element Attribute enthält.
- Wenn Sie für ein XML-Element, das mehr als ein Text-Child-Element hat, einen neuen Textwert definieren, so wird das letzte davon gelöscht und durch ein neues Text-Child-Element ersetzt.
- Wenn der Textwert eines XML-Elements mit mehr als einem Text-Child-Element gelesen wird, so wird nur der Wert des ersten Text-Child-Elements zurückgegeben.

14.3.2.32.1 AppendChild

Deklaration: AppendChild (*pNewData* als XMLData 593)

Beschreibung
AppendChildhängt pNewDataals letztes Child-Element an das XMLData-Objekt an.

Fehler

1500	Das XMLData-Element ist nicht mehr gültig.
1505	Es wurde eine ungültige XMLData-Art definiert.
1506	Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
1507	Das Element kann keine untergeordneten Elemente haben.
1512	Zyklische Einfügung - ein neues Datenelement ist bereits Teil des Dokuments.
1514	Es wurde eine ungülte XMLData-Art für diese Position definiert.
1900	Das Dokument darf nicht geändert werden.

Beispiel

```
DimobjCurrentParent AsXMLData
DimobjNewChild AsXMLData

SetobjNewChild = objSpy.ActiveDocument.CreateChild(spyXMLDataElement)
SetobjCurrentParent = objSpy.ActiveDocument.RootElement

objCurrentParent.AppendChild objNewChild

SetobjNewChild = Nothing
```

14.3.2.32.2 CountChildren

Deklaration: CountChildren als long

Beschreibung
CountChildrenruft die Anzahl der Child-Elemente ab.

 © 2016-2022 Altova GmbH

Verfügbar mit Typbibliothek Version 1.5

Fehler
1500 Das XMLData-Objekt ist nicht mehr gültig.

14.3.2.32.3 CountChildrenKind

Deklaration: CountChildrenKind (*nKind* als SPYXMLDataKind[618]) als long

Beschreibung
CountChildrenKindruft die Anzahl der Child-Elemente dieser bestimmten Art ab.

Verfügbar mit Typbibliothek Version 1.5

Fehler
1500 Das XMLData-Objekt ist nicht mehr gültig.

14.3.2.32.4 EraseAllChildren

Deklaration: EraseAllChildren

Beschreibung
EraseAllChildrenlöscht alle zum XMLDataObjekt gehörigen Child-Elemente.

Fehler
1500 Das XMLData-Objekt ist nicht mehr gültig.
1900 Das Dokument darf nicht geändert werden.

Beispiel
In diesem Beispiel werden alle Elemente des aktiven Dokuments gelöscht.

```
DimobjCurrentParent AsXMLData

SetobjCurrentParent = objSpy.ActiveDocument.RootElement
objCurrentParent.EraseAllChildren
```

14.3.2.32.5 EraseChild

Methode: EraseChild (Child als XMLData[593])

Beschreibung
Löscht den angegebenen Child Node.

Fehler

1500 Ungültiges Objekt.
1506 Ungültige XML-Input-Datei.
1510 Ungültiger Parameter.

14.3.2.32.6 EraseCurrentChild

Deklaration: EraseCurrentChild

Beschreibung
EraseCurrentChildlöscht das aktuelle XMLDataChild-Objekt. Bevor Sie EraseCurrentChildaufrufen, müssen Sie einen internen Iterator mit XMLData.GetFirstChild[598] initialisieren. Nachdem Sie das aktuelle Child-Element gelöscht haben, erhöht EraseCurrentChild den internen Iterator des XMLData-Elements. Es wird kein Fehler zurückgegeben, wenn das letzte Child-Element gelöscht wird und der Iterator über das Ende der Child-Liste hinaus verschoben wird. Der nächste Aufruf von EraseCurrentChildgibt allerdings den Fehler 1503 zurück.

Fehler
1500 Das XMLData-Objekt ist nicht mehr gültig.
1503 Es wurde für dieses XMLData-Objekt kein Iterator initialisiert oder der Iterator zeigt hinter das letzte Child-Element.
1900 Das Dokument darf nicht geändert werden.

Beispiele
```
// ----------------------------------------
// Scripting environment - JScript
// erase all children of XMLData
// ----------------------------------------
// let's get an XMLData element, we assume that the
// cursor selects the parent of a list in grid view
var objList = Application.ActiveDocument.GridView.CurrentFocus;

// the following line would be shorter, of course
//	objList.EraseAllChildren ();

// but we want to demonstrate the usage of EraseCurrentChild
if((objList != null) && (objList.HasChildren))
{
	try
	{
		objEle = objList.GetFirstChild(-1);
		while(objEle != null)
			objList.EraseCurrentChild();
			// no need to call GetNextChild
	}
	catch(err)
		// 1503 - we reached end of child list
		{ if((err.number & 0xffff) != 1503) throw(err); }
}
```

 © 2016-2022 Altova GmbH

14.3.2.32.7 GetChild

Deklaration: GetChild (*position* als long) als XMLData 593

Rückgabewert
Gibt ein XML-Element als XMLDataObjekt zurück.

Beschreibung
GetChild() gibt eine Referenz auf das Child am angegebenen Index (nullbasiert) zurück.

Verfügbar mit Typbibliothek Version 1.5

Fehler
1500 Das XMLData-Objekt ist nicht mehr gültig.
1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.32.8 GetChildAttribute

Methode: GetChildAttribute (strName als string) Child als XMLData Objekt (`NULL` bei Fehler)

Beschreibung
Ruft das Attribut mit dem angegebenen Namen auf.

Fehler
1500 Ungültiges Objekt.
1510 Ungültiger Parameter.

14.3.2.32.9 GetChildElement

Methode: GetChildElement (strName als string, nIndex als long) Child als XMLData-Objekt (`NULL` bei Fehler)

Beschreibung
Ruft das nte Child Element des angegebenen Namens auf.

Fehler
1500 Ungültiges Objekt.
1510 Ungültiger Parameter.

14.3.2.32.10 GetChildKind

Deklaration: GetChildKind (*position* als long, *nKind* als SPYXMLDataKind[618]) als XMLData[593]

Rückgabewert
Gibt ein XML-Element als XMLDataObjekt zurück.

Beschreibung
GetChildKind() gibt eine Referenz auf ein Child dieser Art am angegebenen Index (nullbasiert) zurück. Der Positionsparameter ist relativ zur Anzahl der Child-Elemente der angegebenen Art und nicht zu allen Child-Elementen des Objekts.

Verfügbar mit Typbibliothek Version 1.5

Fehler
1500 Das XMLData-Objekt ist nicht mehr gültig.
1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.32.11 GetCurrentChild

Deklaration: GetCurrentChild als XMLData[593]

Rückgabewert
Gibt ein XML-Element als XMLDataObjekt zurück.

Beschreibung
GetCurrentChildruft das aktuelle Child-Element ab. Bevor Sie GetCurrentChildaufrufen, müssen Sie einen internen Iterator mit XMLData.GetFirstChild[598]initialisieren.

Fehler
1500 Das XMLData-Objekt ist nicht mehr gültig.
1503 Es wurde für dieses XMLData-Objekt kein Iterator initialisiert.
1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.32.12 GetFirstChild

Deklaration: GetFirstChild (*nKind* als SPYXMLDataKind[618]) als XMLData[593]

Rückgabewert
Gibt ein XML-Element als XMLDataObjekt zurück.

Beschreibung
GetFirstChildinitialisiert einen neuen Iterator und gibt das erste Child-Element zurück. Setzen Sie nKind= -1, um einen Iterator für alle Arten von Child-Elementen abzurufen.

© 2016-2022 Altova GmbH

Anmerkung: Der Iterator wird innerhalb des XMLData-Objekts gespeichert und wird zerstört, wenn das XMLData-Objekt zerstört wird. Stellen Sie sicher, dass Sie eine Referenz auf dieses Objekt beibehalten, solange Sie GetCurrentChild[598], GetNextChild[599] oder EraseCurrentChild[596] verwenden möchten.

Fehler

1500 Das XMLData-Objekt ist nicht mehr gültig.
1501 Es wurde eine ungültige XMLData-Art definiert.
1504 Das Element hat keine untergeordneten Elemente der angegebenen Art.
1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiel

Siehe Beispiel unter XMLData.GetNextChild[599].

14.3.2.32.13 GetNamespacePrefixForURI

Methode: GetNamespacePrefixForURI (strURI als string) strNS als string

Beschreibung

Gibt das Namespace-Präfix der vorgegebenen URI zurück.

Fehler

1500 Ungültiges Objekt.
1510 Ungültiger Parameter.

14.3.2.32.14 GetNextChild

Deklaration: GetNextChild als XMLData[593]

Rückgabewert

Gibt ein XML-Element als XMLDataObjekt zurück.

Beschreibung

GetNextChildgeht zum nächsten Child-Element dieses Elements. Bevor Sie GetNextChildaufrufen, müssen Sie einen internen Iterator mit XMLData.GetFirstChild[598] initialisieren.

Überprüfen Sie das letzte Child-Element des Elements, wie im Beispiel unten gezeigt.

Fehler

1500 Das XMLData-Objekt ist nicht mehr gültig.
1503 Für dieses XMLData-Objekt wurde kein Iterator initialisiert.
1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

Beispiele

```
' ---------------------------------------------
' VBA code snippet - iterate XMLData children
' ---------------------------------------------
OnErrorResume Next
SetobjParent = objSpy.ActiveDocument.RootElement
```

```
'get elements of all kinds
SetobjCurrentChild = objParent.GetFirstChild(-1)

Do
	'do something useful with the child

	'step to next child
	SetobjCurrentChild = objParent.GetNextChild
LoopUntil(Err.Number - vbObjectError = 1503)
```

```
// ---------------------------------------
// Scripting environment - JScript
// iterate through children of XMLData
// ---------------------------------------
try
{
	var objXMLData = ... // initialize somehow
	var objChild = objXMLData.GetFirstChild(-1);

	while(true)
	{
		// do something usefull with objChild

		objChild = objXMLData.GetNextChild();
	}
}
catch(err)
{
	if((err.number & 0xffff) == 1504)
		; // element has no children
	elseif((err.number & 0xffff) == 1503)
		;	// last child reached
	else
		throw(err);
}
```

14.3.2.32.15 GetTextValueXMLDecoded

Methode: GetTextValueXMLDecoded () als string

Beschreibung
Ruft den dekodierten Textwert der XML-Datei ab.

Fehler

1500	Ungültiges Objekt.
1510	Ungültiger Parameter.

© 2016-2022 Altova GmbH

14.3.2.32.16 HasChildren

Deklaration: HasChildren als Boolean

Beschreibung
Die Eigenschaft ist true, wenn das Objekt das übergeordnete Objekt von anderen XMLDataObjekten ist. Diese Eigenschaft ist schreibgeschützt.

Fehler
1500 Das XMLData-Objekt ist nicht mehr gültig.
1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.32.17 HasChildrenKind

Deklaration: HasChildrenKind (*nKind* als SPYXMLDataKind[618]) als Boolean

Beschreibung
Die Methode gibt true zurück, wenn das Objekt das übergeordnete Objekt von anderen XMLDataObjekten dieser Art ist.

Verfügbar mit Typbibliothek Version 1.5

Fehler
1500 Das XMLData-Objekt ist nicht mehr gültig.
1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.32.18 InsertChild

Deklaration: InsertChild(*pNewData* als XMLData[593])

Beschreibung
InsertChildfügt das neue Child-Element vor dem aktuellen Child-Element ein (Informationen zum Setzen des aktuellen Child-Elements finden Sie auch unter XMLData.GetFirstChild[598], XMLData.GetNextChild[599]).

Fehler
1500 Das XMLData-Objekt ist nicht mehr gültig.
1503 Es wurde für dieses XMLData-Objekt kein Iterator initialisiert.
1505 Es wurde eine ungültige XMLData-Art definiert.
1506 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.
1507 Das Element kann keine Child-Elemente haben.
1512 Zyklische Einfügung - ein neues Datenelement ist bereits Teil des Dokuments.
1514 Es wurde eine ungültige XMLData-Art für diese Position definiert.
1900 Das Dokument darf nicht geändert werden.

14.3.2.32.19 InsertChildAfter

Methode: InsertChildAfter (Node als XMLData, NewData als XMLData)

Beschreibung
Fügt einen neuen XML-Node (definiert im zweiten Parameter) nach dem (mit dem ersten Parameter) angegebenen Node ein.

Errors
1500 Ungültiges Objekt.
1506 Ungültige XML-Input-Datei
1507 Es sind keine Child-Nodes zulässig
1510 Ungültiger Parameter.
1512 Das Child wurde bereits hinzugefügt
1514 Ungültige Art an dieser Stelle

14.3.2.32.20 InsertChildBefore

Methode: InsertChildBefore (Node als XMLData, NewData als XMLData)

Beschreibung
Fügt einen neuen XML-Node (definiert im zweiten Parameter) vor dem (im ersten Parameter) angegebenen Node ein.

Errors
1500 Ungültiges Objekt.
1506 Ungültige XML-Input-Datei
1507 Es sind keine Child-Nodes zulässig
1510 Ungültiger Parameter.
1512 Das Child wurde bereits hinzugefügt
1514 Ungültige Art an dieser Stelle

14.3.2.32.21 IsSameNode

Deklaration: IsSameNode (*pNodeToCompare* als XMLData [593]) als Boolean

Beschreibung

Gibt true zurück, wenn pNodeToComparedenselben Node wie das Objekt selbst referenziert.

Fehler
1500 Das XMLData-Objekt ist nicht mehr gültig.
1506 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

© 2016-2022 Altova GmbH

14.3.2.32.22 Kind

Dellaration: Kind als SPYXMLDataKind 618

Beschreibung
Art dieses XMLDataObjekts. Diese Eigenschaft ist schreibgeschützt.

Fehler
1500 Das XMLData-Objekt ist nicht mehr gültig.
1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.32.23 MayHaveChildren

Deklaration: MayHaveChildren als Boolean

Beschreibung
Gibt an, ob Child-Elemente zu diesem XMLDataObjekt hinzugefügt werden dürfen.
Diese Eigenschaft ist schreibgeschützt.

Fehler
1500 Das XMLData-Objekt ist nicht mehr gültig.
1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.32.24 Name

Deklaration: Name als String

Beschreibung
Dient zum Bearbeiten und Abrufen des Namens des XMLDataObjekts.

Fehler
1500 Das XMLData-Objekt ist nicht mehr gültig.
1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.32.25 Parent

Deklaration: Parent als XMLData 593

Rückgabewert
Übergeordnetes Element als XMLDataObjekt. Nichts (oder NULL), wenn es kein übergeordnetes Element gibt.

Beschreibung
Diesem Element übergeordnetes Element. Diese Eigenschaft ist schreibgeschützt.

Fehler
1500 Das XMLData-Objekt ist nicht mehr gültig.
1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.2.32.26 SetTextValueXMLEncoded

Methode: SetTextValueXMLEncoded (*strVal* als String[618])

Beschreibung
Definiert den kodierten Textwert der XML-Datei.

Fehler
1500 Ungültiges Objekt.
1513 Keine Änderung zulässig.

14.3.2.32.27 TextValue

Deklaration: TextValue als String

Beschreibung
Dient zum Bearbeitung und Abrufen des Textwerts dieses XMLDataObjekts.

Fehler
1500 Das XMLData-Objekt ist nicht mehr gültig.
1510 Für den Rückgabeparameter wurde eine ungültige Adresse angegeben.

14.3.3 Enumerationen

Dies ist eine Liste aller von der Authentic Desktop API verwendeten Enumerationen. Wenn Ihre Skripting-Umgebung Enumerationen nicht unterstützt, verwenden Sie stattdessen die Zahlenwerte.

14.3.3.1 ENUMApplicationStatus

Beschreibung
Enumeration zur Definition des aktuellen Applikationsstatus.

Mögliche Werte:

eApplicationRunning	= 0
eApplicationAfterLicenseCheck	= 1
eApplicationBeforeLicenseCheck	= 2
eApplicationConcurrentLicenseCheckFailed	= 3
eApplicationProcessingCommandLine	= 4

© 2016-2022 Altova GmbH

14.3.3.2 SPYAttributeTypeDefinition

Beschreibung
Attributtypdefinition, die zur Generierung einer XML-Beispieldatei ausgewählt werden kann.
Dieser Typ wird mit den Methoden GenerateDTDOrSchema[477] und GenerateDTDOrSchemaEx[478] verwendet.

Mögliche Werte:

spyMergedGlobal	= 0
spyDistinctGlobal	= 1
spyLocal	= 2

14.3.3.3 SPYAuthenticActions

Beschreibung
Aktionen, die an AuthenticRange[399] Objekten durchgeführt werden können.

Mögliche Werte:

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

14.3.3.4 SPYAuthenticDocumentPosition

Beschreibung
Relative und absolute Position für die Navigation mit AuthenticRange[399] Objekten.

Mögliche Werte:

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

14.3.3.5 SPYAuthenticElementActions

Beschreibung
Aktionen die mit dem nicht mehr verwendeten Objekt GetAllowedElements verwendet werden können (Wird ersetzt durch AuthenticRange.CanPerformActionWith[402]).

Mögliche Werte:

k_ActionInsertAt	= 0
k_ActionApply	= 1

k_ActionClearSurr = 2
k_ActionAppend = 3
k_ActionInsertBefore = 4
k_ActionRemove = 5

14.3.3.6 SPYAuthenticElementKind

Beschreibung
Enumeration der verschiedenen Arten von Elementen, die für die Navigation und Auswahl innerhalb der AuthenticRange[399] und AuthenticView[428] Objekte verwendet werden.

Mögliche Werte:

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

14.3.3.7 SPYAuthenticMarkupVisibility

Beschreibung
Enumerationswerte zum Anpassen der Sichtbarkeit von Markup-Tags mit MarkupVisibility[442].

Mögliche Werte:

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

14.3.3.8 SPYAuthenticToolbarButtonState

Beschreibung
Die Status der Authentic-Symbolleisten-Schaltflächen werden durch die folgende Enumeration definiert:

Possible values:

authenticToolbarButtonDefault = 0
authenticToolbarButtonEnabled = 1
authenticToolbarButtonDisabled = 2

© 2016-2022 Altova GmbH

14.3.3.9 SPYDatabaseKind

Beschreibung
Were zur Auswahl verschiedener Datenbankarten für den Import. Informationen zur Verwendung finden Sie unter DatabaseConnection.DatabaseKind 454.

Mögliche Werte:

spyDB_Access	= 0
spyDB_SQLServer	= 1
spyDB_Oracle	= 2
spyDB_Sybase	= 3
spyDB_MySQL	= 4
spyDB_DB2	= 5
spyDB_Other	= 6
spyDB_Unspecified	= 7
spyDB_PostgreSQL	= 8
spyDB_iSeries	= 9

14.3.3.10 SPYDialogAction

Beschreibung
Werte zur Simulation verschiedener Interaktionen in Dialogfeldern. Informationen dazu, welche Dialogfelder zur Verfügung stehen, finden Sie unter Dialogs 458.

Mögliche Werte:

spyDialogOK	= 0	// simulate click on OK button
spyDialogCancel	= 1	// simulate click on Cancel button
spyDialogUserInput	= 2	//show dialog and allow user interaction

14.3.3.11 SPYDOMType

Beschreibung
Enumerationswerte zum Parametrisieren der Generierung von C++ Code anhand von Schemadefinitionen.

Mögliche Werte:

spyDOMType_msxml4	= 0	wird nicht mehr verwendet
spyDOMType_xerces	= 1	
spyDOMType_xerces3	= 2	
spyDOMType_msxml6	= 3	

`spyDOMType_xerces` bezeichnet die Verwendung von Xerces 2.x; `spyDOMType_xerces3` bezeichnet die Verwendung von Xerces 3.x.

14.3.3.12 SPYDTDSchemaFormat

Beschreibung
Enumeration zum Identifizieren der verschiedenen Schemaformate.

Mögliche Werte:

spyDTD	= 0
spyW3C	= 1

14.3.3.13 SPYEncodingByteOrder

Beschreibung
Enumerationswerte zur Definition der Codierungsbytefolge für den Textimport und -export.

Mögliche Werte:

spyNONE	= 0
spyLITTLE_ENDIAN	= 1
spyBIG_ENDIAN	= 2

14.3.3.14 SPYExportNamespace

Beschreibung
Enumerationstyp, um die Behandlung von Namespace Identifiers beim Export zu konfigurieren.

Mögliche Werte:

spyNoNamespace	= 0
spyReplaceColonWithUnderscore	= 1

14.3.3.15 SPYFindInFilesSearchLocation

Beschreibung
Die verschiedenen Optionen, wo eine Suche durchgeführt werden kann. Dieser Typ wird mit dem Dialogfeld FindInFilesDlg 508 verwendet.

Mögliche Werte:

spyFindInFiles_Documents	= 0
spyFindInFiles_Project	= 1
spyFindInFiles_Folder	= 2

14.3.3.16 SPYFrequentElements

Beschreibung
Enumerationswert zum Parametrisieren der Schemagenerierung.

Mögliche Werte:

© 2016-2022 Altova GmbH

spyGlobalElements = 0
spyGlobalComplexType = 1

14.3.3.17 SPYImageKind

Beschreibung
Enumerationswerte zum Parametrisieren des Bildtyps der generierten Dokumentation. Diese Werte werden in SchemaDocumentationDialog.DiagramFormat [531] verwendet.

Mögliche Werte:
spyImageType_PNG = 0
spyImageType_EMF = 1

14.3.3.18 SPYImportColumnsType

Beschreibung
Enumeration zur Definition der verschiedenen Importspaltenarten.

Mögliche Werte:

spyImportColumns_Element = 0
spyImportColumns_Attribute = 1

14.3.3.19 SPYKeyEvent

Beschreibung
Enumerationstyp zum Identifizieren der verschiedenen Tastatur-Events. Diese Events entsprechen den gleichnamigen Windows-Messages.

Mögliche Werte:
spyKeyDown = 0
spyKeyUp = 1
spyKeyPressed = 2

14.3.3.20 SPYKeyStatus

Beschreibung
Enumerationstyp zum Identifizieren des Tastenstatus.

Mögliche Werte:
spyLeftShiftKeyMask = 1
spyRightShiftKeyMask = 2
spyLeftCtrlKeyMask = 4
spyRightCtrlKeyMask = 8
spyLeftAltKeyMask = 16
spyRightAltKeyMask = 32

14.3.3.21 SPYLibType

Beschreibung
Enumerationswerte zum Parametrisieren der Generierung von C++ Code anhand von Schemadefinitionen.

Mögliche Werte:

```
spyLibType_static      = 0
spyLibType_dll         = 1
```

14.3.3.22 SPYLoading

Beschreibung
Enumerationswerte zum Definieren des Ladeverhaltens von URL-Dateien.

Mögliche Werte:

```
spyUseCacheProxy      = 0
spyReload             = 1
```

14.3.3.23 SPYMouseEvent

Beschreibung
Enumerationstyp, der die Mausstatus während eines Maus-Events definiert. Verwenden Sie diese Enumerationswerte als Bitmasken anstatt direkte Vergleiche damit durchzuführen.

Beispiele

```
' to check for ctrl-leftbutton-down in VB
If(i_eMouseEvent = (XMLSpyLib.spyLeftButtonDownMask OrXMLSpyLib.spyCtrlKeyDownMask)) Then
        ' react on ctrl-leftbutton-down
EndIf

' to check for double-click with any button in VBScript
If(((i_eMouseEvent AndspyDoubleClickMask) <> 0) Then
        ' react on double-click
EndIf
```

Mögliche Werte:

```
spyNoButtonMask          = 0
spyMouseMoveMask         = 1
spyLeftButtonMask        = 2
spyMiddleButtonMask      = 4
spyRightButtonMask       = 8
spyButtonUpMask          = 16
spyButtonDownMask        = 32
spyDoubleClickMask       = 64
```


© 2016-2022 Altova GmbH

```
spyShiftKeyDownMask        = 128
spyCtrlKeyDownMask         = 256
spyLeftButtonDownMask      = 34   // spyLeftButtonMask | spyButtonDownMask
spyMiddleButtonDownMask    = 36   // spyMiddleButtonMask | spyButtonDownMask
spyRightButtonDownMask     = 40   // spyRightButtonMask | spyButtonDownMask
spyLeftButtonUpMask        = 18   // spyLeftButtonMask | spyButtonUpMask
spyMiddleButtonUpMask      = 20   // spyMiddleButtonMask | spyButtonUpMask
spyRightButtonUpMask       = 24   // spyRightButtonMask | spyButtonUpMask
spyLeftDoubleClickMask     = 66   // spyRightButtonMask | spyButtonUpMask
spyMiddleDoubleClickMask   = 68   // spyMiddleButtonMask | spyDoubleClickMask
spyRightDoubleClickMask    = 72   // spyRightButtonMask | spyDoubleClickMask
```

14.3.3.24 SPYNumberDateTimeFormat

Beschreibung
Enumerationswert zum Konfigurieren von Datenbankverbindungen.

Mögliche Werte:

```
spySystemLocale      = 0
spySchemaCompatible  = 1
```

14.3.3.25 SPYProgrammingLanguage

Beschreibung
Enumerationswerte zur Auswahl der Programmiersprache für die Codegenerierung von Schemadefinitionen.

Ist nur in der Enterprise Edition verfügbar/aktiv. Bei Aufruf über eine andere Version wird ein Fehler zurückgegeben.

Mögliche Werte:

```
spyUndefinedLanguage   = -1
spyJava                = 0
spyCpp                 = 1
spyCSharp              = 2
```

14.3.3.26 SPYProjectItemTypes

Beschreibung
Enumerationswerte zum Identifizieren der verschiedenen Elemente in Projektelementlisten. Siehe SpyProjectItem.ItemType 545.

Mögliche Werte:

```
spyUnknownItem   = 0
spyFileItem      = 1
spyFolderItem    = 2
spyURLItem       = 3
```

14.3.3.27 SPYProjectType

Beschreibung
Enumerationswerte zum Parametrisieren der Generierung von C# Code anhand von Schemadefinitionen.

Mögliche Werte:

spyVisualStudioProject	= 0	Wird nicht mehr verwendet
spyVisualStudio2003Project	= 1	Wird nicht mehr verwendet
spyBorlandProject	= 2	Wird nicht mehr verwendet
spyMonoMakefile	= 3	Wird nicht mehr verwendet
spyVisualStudio2005Project	= 4	Wird nicht mehr verwendet
spyVisualStudio2008Project	= 5	Wird nicht mehr verwendet
spyVisualStudio2010Project	= 6	Auch für C++-Code
spyVisualStudio2013Project	= 7	Auch für C++-Code
spyVisualStudio2015Project	= 8	Auch für C++-Code
spyVisualStudio2017Project	= 9	Auch für C++-Code
spyVisualStudio2019Project	=10	Auch für C++-Code

14.3.3.28 SpySampleXMLGenerationChoiceMode

Beschreibung
Diese Enumeration wird in `GenerateSampleXMLDlg.ChoiceMode` 520 verwendet:

spySampleXMLGen_FirstBranch = 0
spySampleXMLGen_AllBranches = 1
spySampleXMLGen_ShortestBranch = 2

14.3.3.29 SPYSampleXMLGenerationOptimization (nicht mehr verwendet)

Diese Enumeration wird seit v2014 NICHT MEHR VERWENDET.

Beschreibung
Definiert die Elemente, die in der XML-Beispieldatei generiert werden.
Diese Enumeration wird in GenerateSampleXMLDlg 519 verwendet.

Mögliche Werte:

© 2016-2022 Altova GmbH

spySampleXMLGen_Optimized = 0
spySampleXMLGen_NonMandatoryElements = 1
spySampleXMLGen_Everything = 2

14.3.3.30 SpySampleXMLGenerationSampleValueHints

Beschreibung
Diese Enumeration wird in `GenerateSampleXMLDlg.SampleValueHints` 524 verwendet.

spySampleXMLGen_FirstFit = 0
spySampleXMLGen_RandomFit = 1
spySampleXMLGen_CycleThrough = 2

14.3.3.31 SPYSampleXMLGenerationSchemaOrDTDAssignment

Beschreibung
Definiert, welche Art von Referenz zum Schema/zur DTD zur generierten XML-Beispieldatei hinzugefügt werden soll.
Diese Enumeration wird in GenerateSampleXMLDlg 519 verwendet.

Mögliche Werte:
spySampleXMLGen_AssignRelatively = 0
spySampleXMLGen_AssignAbsolutely = 1
spySampleXMLGen_DoNotAssign = 2

14.3.3.32 SPYSchemaDefKind

Beschreibung
Enumerationstyp zur Auswahl von Schemadiagrammtypen.

Mögliche Werte:
spyKindElement = 0
spyKindComplexType = 1
spyKindSimpleType = 2
spyKindGroup = 3
spyKindModel = 4
spyKindAny = 5
spyKindAttr = 6
spyKindAttrGroup = 7
spyKindAttrAny = 8
spyKindIdentityUnique = 9

spyKindIdentityKey = 10
spyKindIdentityKeyRef = 11
spyKindIdentitySelector = 12
spyKindIdentityField = 13
spyKindNotation = 14
spyKindInclude = 15
spyKindImport = 16
spyKindRedefine = 17
spyKindFacet = 18
spyKindSchema = 19
spyKindCount = 20

14.3.3.33 SPYSchemaDocumentationFormat

Beschreibung
Enumerationswerte zum Parametrisieren der Generierung von Schemadokumentation. Diese Werte werden in SchemaDocumentationDialog.OutputFormat 537 verwendet.

Mögliche Werte:
spySchemaDoc_HTML = 0
spySchemaDoc_MSWord = 1
spySchemaDoc_RTF = 2
spxSchemaDoc_PDF =3

14.3.3.34 SPYSchemaExtensionType

Beschreibung
Enumeration zur Definition der verschiedenen Schema Extension-Typen.

Mögliche Werte:

spySchemaExtension_None = 0
spySchemaExtension_SQL_XML = 1
spySchemaExtension_MS_SQL_Server = 2
spySchemaExtension_Oracle = 3

14.3.3.35 SPYSchemaFormat

Beschreibung
Enumeration zur Definition der unterschiedlichen Schemaformattypen.

Mögliche Werte:

spySchemaFormat_Hierarchical = 0
spySchemaFormat_Flat = 1

 © 2016-2022 Altova GmbH

14.3.3.36 SPYTextDelimiters

Beschreibung
Enumerationswerte zum Definieren der Texttrennzeichen für den Textexport.

Mögliche Werte:

spyTabulator	= 0
spySemicolon	= 1
spyComma	= 2
spySpace	= 3

14.3.3.37 SPYTextEnclosing

Beschreibung
Enumerationswert zum Definieren der den Text einschließenden Zeichen beim Textimport und -export.

Mögliche Werte:

spyNoEnclosing	= 0
spySingleQuote	= 1
spyDoubleQuote	= 2

14.3.3.38 SPYTypeDetection

Beschreibung
Enumeration, um auszuwählen, wie die Typerkennung bei `GenerateDTDOrSchema` 477 und `GenerateDTDOrSchemaEx` 478 funktioniert.

Mögliche Werte:

spyBestPossible	= 0
spyNumbersOnly	= 1
spyNoDetection	= 2

14.3.3.39 SPYURLTypes

Beschreibung
Enumeration zum Definieren der verschiedenen URL-Typen.

Mögliche Werte:

spyURLTypeAuto	= -1
spyURLTypeXML	= 0
spyURLTypeDTD	= 1

14.3.3.40 SPYValidateXSDVersion

Beschreibung
Enumerationswerte zur Auswahl der zu verwendenden XSD-Version. Welche XSD-Version ausgewählt wird, hängt (i) vom Vorhandensein/Fehlen - und falls vorhanden, vom Wert - des `/xs:schema/@vc:minVersion` Attributs des XSD-Dokuments ab und (ii) dem Wert dieser Enumeration.

`spyValidateXSDVersion_1_0` wählt `XSD 1.0` aus, wenn `vc:minVersion` fehlt oder mit einem beliebigen Wert vorhanden ist.
`spyValidateXSDVersion_1_1` wählt `XSD 1.1` aus, wenn `vc:minVersion` fehlt oder mit einem beliebigen Wert vorhanden ist.
`spyValidateXSDVersion_AutoDetect` wählt `XSD 1.1` aus, wenn `vc:minVersion=1.1`. Wenn das `vc:minVersion` Attribut fehlt oder vorhanden ist und einen beliebigen anderen Wert als `1.1`, hat, so wird `XSD 1.0` ausgewählt.

Mögliche Werte

```
spyValidateXSDVersion_AutoDetect       = 0
spyValidateXSDVersion_1_1              = 1
spyValidateXSDVersion_1_0              = 2
```

14.3.3.41 SPYValidateErrorFormat

Beschreibung
Enumerationswerte zur Auswahl des Formats der Fehlermeldung.

Mögliche Werte

```
spyValidateErrorFormat_Text            = 0
spyValidateErrorFormat_ShortXML        = 1
spyValidateErrorFormat_LongXML         = 2
```

14.3.3.42 SPYViewModes

Beschreibung
Enumerationswerte, die die verschiedenen Ansichtsmodi für XML-Dokumente definieren. Der Modus *spyViewAuthentic(4)* identifiziert den Modus, der zwischenzeitlich DocEdit-Modus genannt wurde und nun Authentic-Modus heißt. Der Modus *spyViewJsonSchema* identifiziert einen Modus, der auf der Benutzeroberfläche der Schema-Design-Ansicht zugeordnet ist, intern aber unterschiedlich behandelt wird.

Mögliche Werte:

```
spyViewGrid            = 0
```


© 2016-2022 Altova GmbH

```
spyViewText          = 1
spyViewBrowser       = 2
spyViewSchema        = 3
spyViewContent       = 4     // nicht mehr verwendet
spyViewAuthentic     = 4
spyViewWSDL          = 5
spyViewZIP           = 6
spyViewEditionInfo   = 7
spyViewXBRL          = 8
spyViewJsonSchema    = 9
```

14.3.3.43 SPYVirtualKeyMask

Beschreibung
Enumeratiostyp für die am häufigsten verwendeten Tastaturmasken, die den Status der virtuellen Tasten identifizieren. Verwenden Sie diese Werte als Bitmasken, anstatt direkte Vergleiche damit durchzuführen. Wenn nötig können Sie mit Hilfe des 'logical or' Operators weitere Masken erstellen.

Beispiele

```
'  VBScript sample: check if ctrl-key is pressed
If((i_nVirtualKeyStatus AndspyCtrlKeyMask) <> 0)) Then
        ' ctrl-key is pressed
EndIf

'  VBScript sample: check if ONLY ctrl-key is pressed
If(i_nVirtualKeyStatus == spyCtrlKeyMask) Then
        ' exactly ctrl-key is pressed
EndIf

// JScript sample: check if any of the right virtual keys is pressed
if((i_nVirtualKeyStatus & (spyRightShiftKeyMask | spyRightCtrlKeyMask | spyRightAltKeyMask)) != 0)
{
        ; ' right virtual key is pressed
}
```

Mögliche Werte:

```
spyNoVirtualKeyMask      = 0
spyLeftShiftKeyMask      = 1
spyRightShiftKeyMask     = 2
spyLeftCtrlKeyMask       = 4
spyRightCtrlKeyMask      = 8
spyLeftAltKeyMask        = 16
spyRightAltKeyMask       = 32
spyShiftKeyMask          = 3     // spyLeftShiftKeyMask | spyRightShiftKeyMask
spyCtrlKeyMask           = 12    // spyLeftCtrlKeyMask | spyRightCtrlKeyMask
spyAltKeyMask            = 48    // spyLeftAltKeyMask | spyRightAltKeyMask
```

14.3.3.44 SPYXMLDataKind

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

Mögliche Werte:

spyXMLDataXMLDocStruct	= 0
spyXMLDataXMLEntityDocStruct	= 1
spyXMLDataDTDDocStruct	= 2
spyXMLDataXML	= 3
spyXMLDataElement	= 4
spyXMLDataAttr	= 5
spyXMLDataText	= 6
spyXMLDataCData	= 7
spyXMLDataComment	= 8
spyXMLDataPI	= 9
spyXMLDataDefDoctype	= 10
spyXMLDataDefExternalID	= 11
spyXMLDataDefElement	= 12
spyXMLDataDefAttlist	= 13
spyXMLDataDefEntity	= 14
spyXMLDataDefNotation	= 15
spyXMLDataKindsCount	= 16

© 2016-2022 Altova GmbH

14.4 ActiveX Integration

Die in diesem Abschnitt beschriebene Authentic Desktop-Benutzeroberfläche und deren Funktionalitäten können in benutzerdefinierte Applikationen integriert werden, die ActiveX Controls verwenden können. Mit Hilfe der ActiveX-Technologie können die verschiedensten Programmiersprachen wie z.B. C++, C#, VB.NET oder HTML für die Integration verwendet werden. (Beachten Sie, dass in HTML integrierte ActiveX-Komponenten mit Microsoft Internet Explorer-Versionen und Plattformen, die ActiveX unterstützen, ausgeführt werden müssen). Alle Komponenten sind vollständige OLE Controls, Die Integration in Java wird durch Wrapper-Klassen möglich gemacht.

Um ActiveX Controls in Ihren benutzerdefinierten Code zu integrieren, müssen Sie das Authentic Desktop-Integrationspaket installieren (siehe https://www.altova.com/de/components/download). Stellen Sie sicher, dass Sie zuerst Authentic Desktop installieren und dann erst das Authentic Desktop-Integrationspaket. Je nach Sprache und Plattform gelten andere Voraussetzungen (siehe Voraussetzungen 619).

Sie haben die Wahl zwischen zwei verschiedenen Ebenen der Integration: auf Applikations- und auf Dokumentebene.

Bei einer Integration auf Applikationsebene wird die komplette Benutzeroberfläche von Authentic Desktop (einschließlich aller Menüs, Symbolleisten, Fenster usw.) als ActiveX Control in Ihre benutzerdefinierte Applikation eingebettet. So könnte Ihre benutzerdefinierte Applikation im einfachsten Szenario z.B. aus nur einem Formular bestehen, in das die grafische Benutzeroberfläche von Authentic Desktop eingebettet ist. Diese Methode ist einfacher zu implementieren als die Integration auf Dokumentebene, ist aber möglicherweise nicht geeignet, wenn Sie die grafische Benutzeroberfläche von Authentic Desktop Ihren Anforderungen gemäß flexibel konfigurieren möchten.

Bei der Integration auf Dokumentebene wird Authentic Desktop Stück für Stück in Ihre eigene Applikation eingebettet. Dabei werden nicht nur das Authentic Desktop Haupt-Control, sondern auch das Dokument-Editor-Hauptfenster und optional zusätzliche Fenster implementiert. Bei dieser Methode haben Sie größere Flexibilität beim Konfigurieren der grafischen Benutzeroberfläche, es ist aber mehr Interaktion mit den ActiveX Controls der Sprache Ihrer Wahl erforderlich.

In den Abschnitten Integration auf Applikationsebene 622 und Integration auf Dokumentebene 624 werden die grundlegenden Schritte auf diesen Ebenen beschrieben. Im Abschnitt Beispiele zur ActiveX-Integration 627 finden Sie Beispiele in C#, HTML und Java. Diese sollen Ihnen dabei helfen, rasch die richtige Entscheidung zu treffen. Der Abschnitt Objektreferenz 648 enthält eine Beschreibung aller für die Integration verwendbaren COM-Objekte mit ihren Eigenschaften und Methoden.

Informationen zur Verwendung von Authentic Desktop als Visual Studio Plug-in finden Sie unter Authentic Desktop in Visual Studio 157.

14.4.1 Voraussetzungen

Um das Authentic Desktop ActiveX Control in eine benutzerdefinierte Applikation zu integrieren, müssen die folgenden Programme auf Ihre Computer installiert sein:

- Authentic Desktop

- Das Authentic Desktop Integrationspaket, das Sie von https://www.altova.com/de/components/download herunterladen können.

Um das 64-Bit ActiveX Control zu integrieren, installieren Sie die 64-Bit-Version von Authentic Desktop und dem Authentic Desktop Integrationspaket. Für Applikationen, die mit Hilfe von Visual Studio unter der Microsoft .NET-Plattform entwickelt wurden, müssen, wie unten erläutert, die 32- und die 64-Bit-Version von Authentic Desktop und dem Authentic Desktop Integrationspaket installiert sein.

Microsoft .NET (C#, VB.NET) mit Visual Studio

Um das Authentic Desktop ActiveX Control in eine unter Microsoft .NET entwickelte 32-Bit-Applikation zu integrieren, müssen die folgenden Programme auf Ihrem Computer installiert sein:

- Microsoft .NET Framework 4.0 oder höher
- Visual Studio 2012/2013/2015/2017/2019/2022
- Authentic Desktop 32-Bit und Authentic Desktop Integrationspaket 32-Bit
- Die ActiveX Controls müssen zur Visual Studio Toolbox hinzugefügt werden (siehe Hinzufügen der ActiveX Controls zur Toolbox 621).

Falls Sie das 64-Bit ActiveX Control integrieren möchten, sind zusätzlich zu den oben erwähnten Voraussetzungen die folgenden erforderlich:

- Authentic Desktop 32-Bit und das Authentic Desktop Integrationspaket 32-Bit müssen weiterhin installiert sein (dies ist erforderlich, damit das 32-Bit ActiveX Control dem Visual Studio Designer zur Verfügung steht, da Visual Studio unter 32-Bit läuft)
- Authentic Desktop 64-Bit und das Authentic Desktop Integrationspaket 64-Bit müssen installiert sein (stellt Ihrer benutzerdefinierten Applikation zur Laufzeit das eigentliche 64-Bit ActiveX Control zur Verfügung)
- Erstellen Sie in Visual Studio eine 64-Bit Build-Konfiguration und bauen Sie Ihre Applikation mit Hilfe dieser Konfiguration. Ein Beispiel dazu finden Sie unter Ausführen der C#-Beispiellösung 628.

Java

Um das Authentic Desktop ActiveX Control über die Eclipse-Entwicklungsumgebung in die Java-Applikation zu integrieren, müssen die folgenden Programme auf Ihrem Computer installiert sein:

- Java Runtime Environment (JRE) oder Java Development Kit (JDK) 7 oder höher
- Eclipse
- Authentic Desktop und das Authentic Desktop Integrationspaket

Anmerkung: Verwenden Sie zur Ausführung der 64-Bit-Version des Authentic Desktop ActiveX Control eine 64-Bit-Version von Eclipse sowie eine 64-Bit-Version von Authentic Desktop und dem Authentic Desktop Integrationspaket.

Authentic Desktop Integration und Bereitstellung auf Client-Rechnern

Wenn Sie eine .NET-Applikation erstellen und beabsichtigen, diese auf anderen Client-Rechnern zur Verfügung zu stellen, müssen auf dem/den Client-Rechner(n) die folgenden Programme installiert sein:

- Authentic Desktop
- Das Authentic Desktop Integrationspaket
- Der benutzerdefinierte Integrationscode oder die benutzerdefinierte Applikation.

 © 2016-2022 Altova GmbH

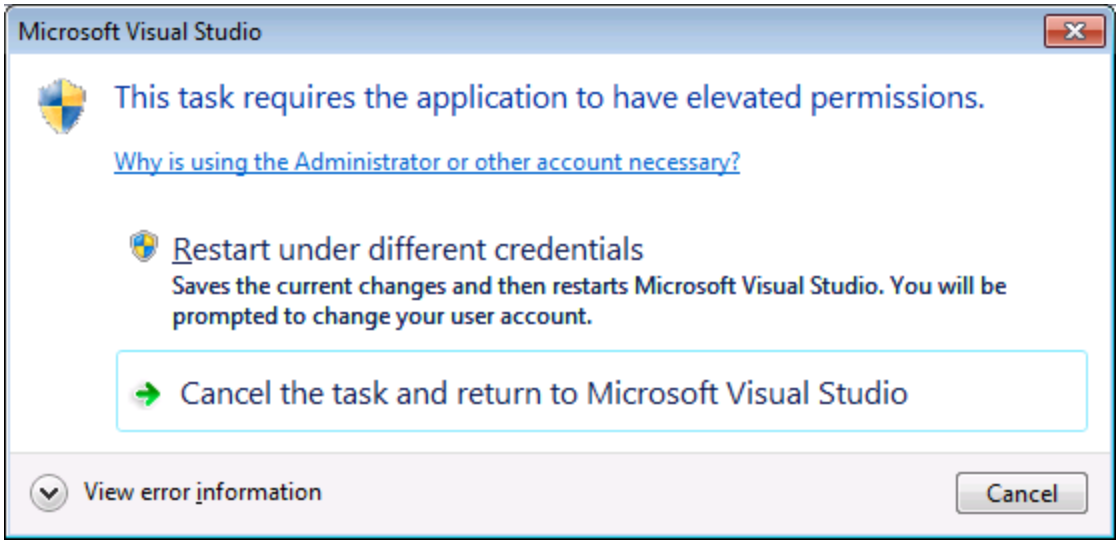
14.4.2 Hinzufügen der ActiveX Controls zur Toolbox

Um die Authentic Desktop ActiveX Controls in einer mit Visual Studio entwickelten Applikation verwenden zu können, fügen Sie diese folgendermaßen zu Ihrer Visual Studio Toolbox hinzu:

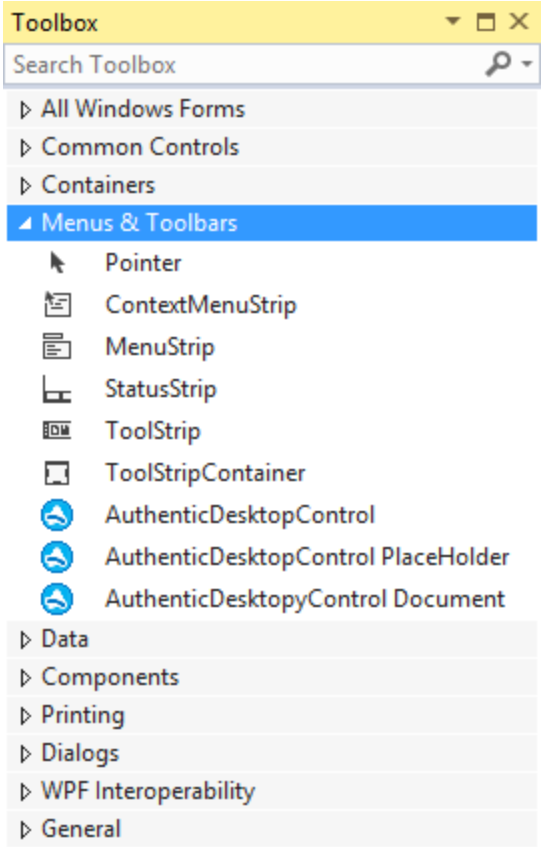
1. Klicken Sie im Menü **Tools** von Visual Studio auf **Choose Toolbox Items**.
2. Aktivieren Sie auf dem Register **COM Components** die Kontrollkästchen neben dem Authentic DesktopControl, dem Authentic DesktopControl-Dokument und dem Authentic DesktopControl-Platzhalter.

Falls die oben erwähnten Controls nicht zur Verfügung stehen, gehen Sie folgendermaßen vor:

1. Klicken Sie auf dem Register **COM Components** auf **Browse** und wählen Sie die Datei **AuthenticControl.ocx** aus dem Authentic Desktop-Installationsordner aus. Beachten Sie, dass das Authentic Desktop-Integrationspaket installiert sein muss, da die Datei sonst nicht zur Verfügung steht (siehe Voraussetzungen 619).
3. Sobald Sie aufgefordert werden, Visual Studio mit erweiterten Berechtigungen zu starten, klicken Sie auf **Restart under different credentials**.



Wenn die obigen Schritte erfolgreich ausgeführt wurden, stehen die Authentic Desktop ActiveX Controls in der Visual Studio Toolbox zur Verfügung.



Anmerkung: Bei einer Integration auf Applikationsebene wird nur das **AuthenticDesktopControl** ActiveX Control verwendet (siehe Integration auf Applikationsebene[622]). Die Controls **AuthenticDesktopControl Document** und **AuthenticDesktopControl Placeholder** werden für die Integration auf Dokumentebene verwendet (siehe Integration auf Dokumentebene[624]).

14.4.3 Integration auf Applikationsebene

Bei der Integration auf Applikationsebene können Sie die gesamte Benutzeroberfläche von Authentic Desktop in ein Fenster Ihrer Applikation einbetten. Bei dieser Art von Integration steht Ihnen die gesamte Benutzeroberfläche von Authentic Desktop einschließlich aller Menüs, Symbolleisten, der Statusleiste, der Dokumentfenster und Eingabehilfen zur Verfügung. Die Anpassung der Benutzeroberfläche der Applikation ist auf die Optionen eingeschränkt, die Authentic Desktop bietet. Dazu gehören die Neuanordnung und Anpassung der Größe der Eingabehilfen und die Anpassung von Menüs und Symbolleisten.

Das einzige ActiveX Control, das Sie integrieren müssen, ist `AuthenticDesktopControl`[651]. Bei Integration auf Applikationsebene dürfen `AuthenticDesktopControlDocument`[659] oder `AuthenticDesktopControlPlaceHolder`[666] ActiveX Controls nicht instantiiert oder aufgerufen werden.

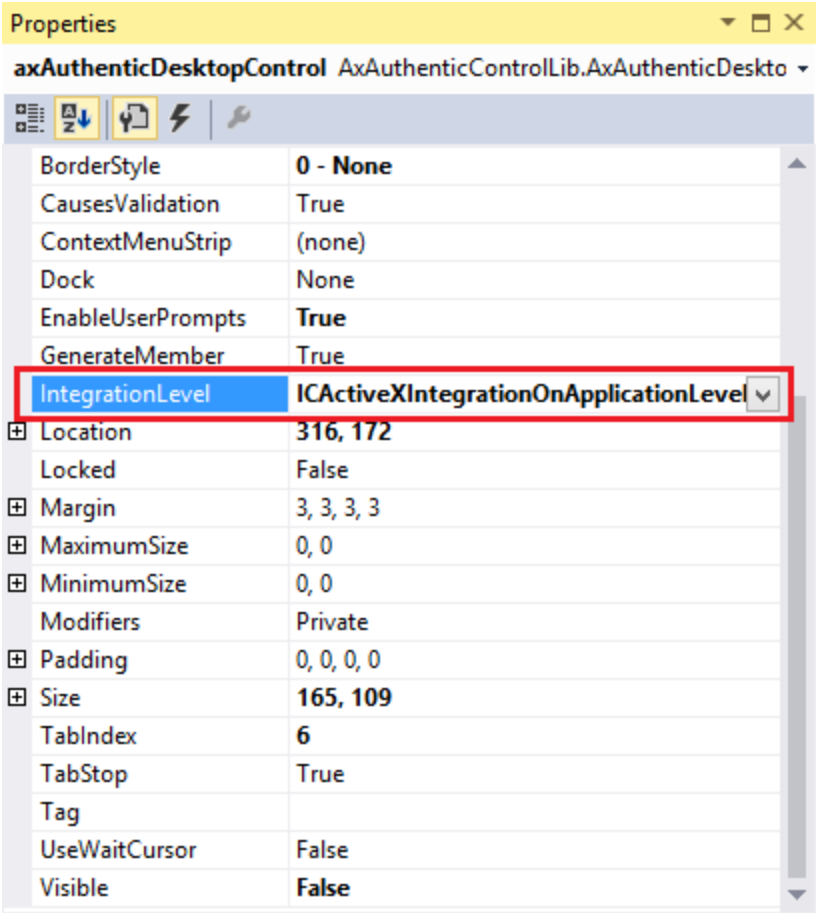
Wenn Sie Initialisierungen vornehmen müssen oder ein bestimmtes Verhalten von Authentic Desktop automatisieren wollen, verwenden Sie die für `AuthenticDesktopControl`[651] beschriebenen

 © 2016-2022 Altova GmbH

Eigenschaften, Methoden und Events. Um komplexere Authentic Desktop Funktionsaufrufe auszuführen, sollten Sie eventuell `AuthenticDesktopControl.Application` 652 verwenden.

Gehen Sie in C# oder VB.NET bei Verwendung von Visual Studio folgendermaßen vor, um eine aus einem Formular bestehende Applikation zu erstellen, in der die Authentic Desktop ActiveX Controls auf Applikationsebene integriert sind:

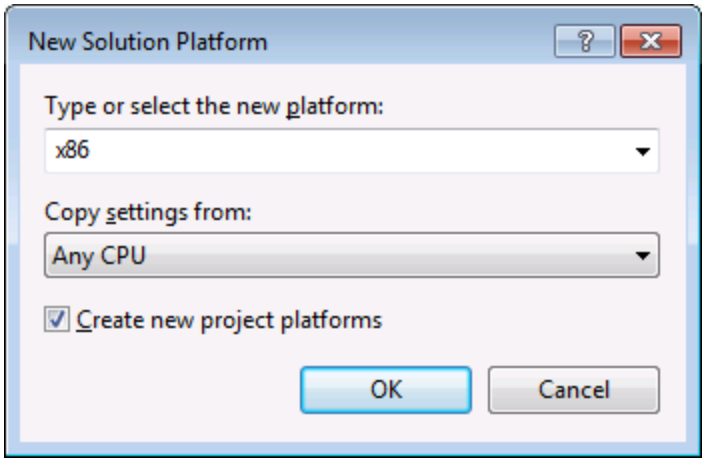
1. Überprüfen Sie, ob alle Voraussetzungen erfüllt werden (siehe Voraussetzungen 619).
2. Erstellen Sie ein Visual Studio Windows Forms-Projekt mit einem neuen leeren Formular.
3. Fügen Sie die ActiveX Controls zur Toolbox hinzu, falls Sie das noch nicht getan haben (siehe Hinzufügen der ActiveX Controls zur Toolbox 621).
4. Ziehen Sie das **AuthenticDesktopControl** aus der Toolbox in Ihr neues Formular.
5. Wählen Sie das **AuthenticDesktopControl** im Formular aus und definieren Sie im Fenster "Properties" für die Eigenschaft **IntegrationLevel** den Wert **ICActiveXIntegrationOnApplicationLevel**.



6. Erstellen Sie eine Build-Plattform-Konfiguration für die Plattform, unter der Sie die Projektmappe erstellen möchten (x86, x64). So erstellen Sie die Build-Konfiguration:

   a. Klicken Sie mit der rechten Maustaste in Visual Studio auf die Projektmappen und wählen Sie **Configuration Manager**.

b. Wählen Sie unter **Active solution platform** den Befehl **New...** und wählen Sie anschließend die x86- oder x64-Konfiguration aus (in diesem Beispiel **x86**).



Sie sind nun fertig und können die Projektmappe in Visual Studio erstellen und ausführen. Denken Sie daran, die Projektmappe mit der richtigen Konfiguration für Ihre Zielplattform (x86, x64) zu erstellen.

14.4.4 Integration auf Dokumentebene

Im Vergleich zur Integration auf Applikationsebene ist die Integration auf Dokumentebene komplexer, bietet aber dafür mehr Flexibilität beim Einbetten von Authentic Desktop-Funktionalitäten in Ihre Applikation mit Hilfe von ActiveX Controls. Auf diese Art haben Sie selektiven Zugriff auf die folgenden Teile der Authentic Desktop-Benutzeroberfläche:

- Dokument-Bearbeitungsfenster
- Projektfenster
- Infofenster
- Fenster "Meldungen"
- Eingabehilfenfenster (Elemente, Attribute, Entities)
- Ausgabefenster

Wie bereits im Abschnitt Integration auf Applikationsebene 622 erwähnt, ist für eine ActiveX-Integration auf Applikationsebene nur ein Control, nämlich das **AuthenticDesktopControl**, erforderlich. Für eine ActiveX-Integration auf Dokumentebene werden die Authentic Desktop-Funktionalitäten jedoch durch die folgenden ActiveX Controls zur Verfügung gestellt:

- AuthenticDesktopControl 651
- AuthenticDesktopControl Document 659
- AuthenticDesktopControl Placeholder 666

Diese Controls werden durch die Datei **AuthenticControl.ocx** aus dem Applikationsinstallationsordner von Authentic Desktop zur Verfügung gestellt. Wenn Sie die ActiveX-Integration mit Hilfe von Visual Studio erstellen, benötigen Sie über die Visual Studio Toolbox Zugriff auf diese Controls (siehe Hinzufügen der ActiveX Controls zur Toolbox 621).

Die grundlegenden Schritte, um die ActiveX Controls auf Dokumentebene in Ihre Applikation zu integrieren sind die folgenden:

 © 2016-2022 Altova GmbH

1. Instantiieren Sie in Ihrer Applikation zuerst **AuthenticDesktopControl**. Die Instantiierung dieses Control ist obligatorisch; Es ermöglicht die Unterstützung für die oben erwähnten Controls **AuthenticDesktopControl Document** und **AuthenticDesktopControl Placeholder**. Die Eigenschaft `IntegrationLevel` 653 muss auf **ICActiveXIntegrationOnDocumentLevel** (oder "1") gesetzt werden. Um das Control für den Benutzer auszublenden, setzen Sie seine Eigenschaft **Visible** auf **False**.

Anmerkung: Verwenden Sie für die Integration auf Dokumentebene nicht die **Open**-Methode des **AuthenticDesktopControl**, da dies zu unerwünschten Ergebnissen führen könnte. Verwenden Sie stattdessen die entsprechenden open-Methoden von **AuthenticDesktopControl Document** und **AuthenticDesktopControl PlaceHolder**.

2. Erstellen Sie mindestens eine Instanz von **AuthenticDesktopControl Document** in Ihrer Applikation. Dieses Control stellt Ihrer Applikation das Dokumentbearbeitungsfenster von Authentic Desktop zur Verfügung und kann bei Bedarf mehrmals instantiiert werden.

   Verwenden Sie die Methode **Open,** um eine vorhandene Datei zu laden. Um Funktionen im Zusammenhang mit einem Dokument aufzurufen, verwenden Sie die Methoden **Path** und **Save** oder Methoden und Eigenschaften, die über die Eigenschaft **Document** zur Verfügung stehen.

Anmerkung: Das Control unterstützt keinen schreibgeschützten Modus. Der Wert der Eigenschaft **ReadOnly** wird ignoriert.

3. Fügen Sie optional für jedes weitere Fenster (bei dem es sich nicht um das Dokumentfenster handelt) und das Ihrer Applikation zur Verfügung stehen soll, den **AuthenticDesktopControl Placeholder** zur Ihrer Applikation hinzu.

   Mit Hilfe von Instanzen von **AuthenticDesktopControl PlaceHolder** können Sie selektiv zusätzliche Fenster von Authentic Desktop in Ihre Applikation einbetten. Die Art des Fensters (z.B. Projektfenster) wird durch die Eigenschaft **PlaceholderWindowID** definiert. Um daher die Art des Fensters zu definieren, definieren Sie die Eigenschaft **PlaceholderWindowID**. Gültige Fenster-IDs finden Sie unter AuthenticDesktopControlPlaceholderWindow 669.

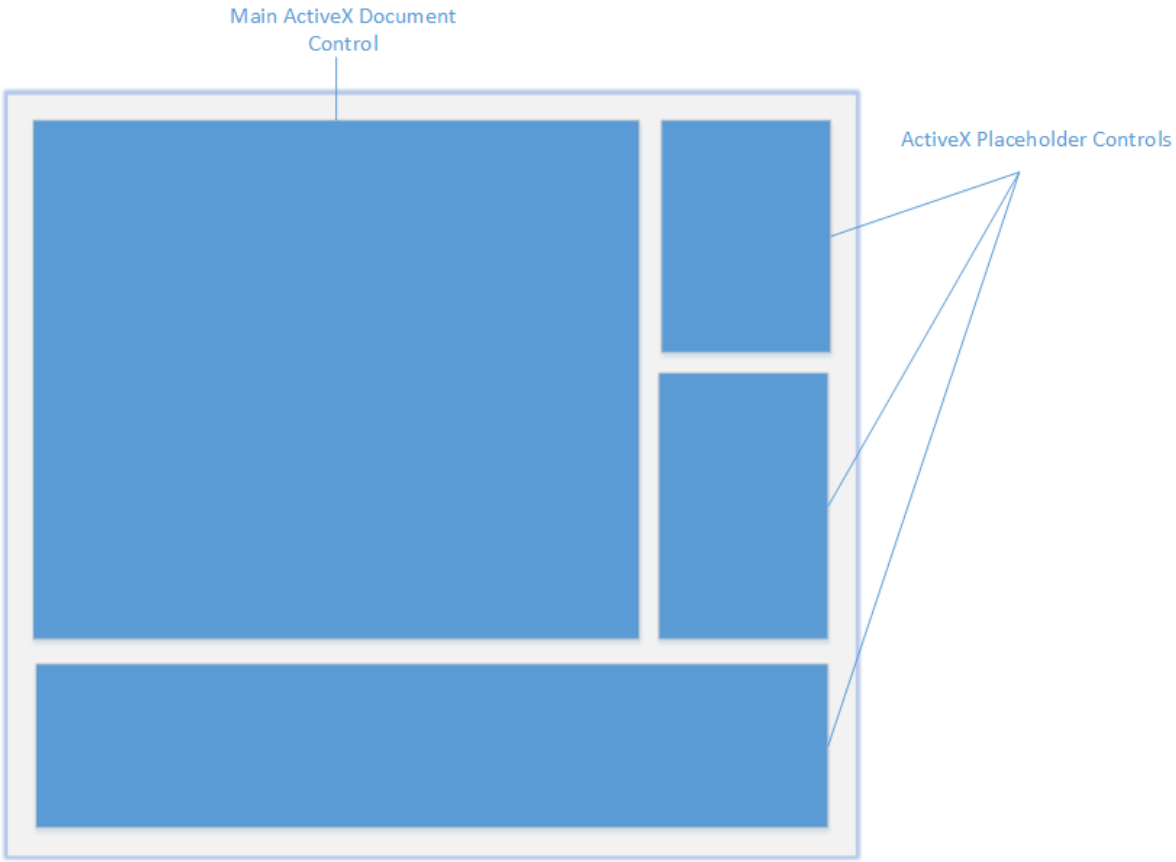
Anmerkung: Verwenden Sie für jeden Windows Identifier nur einen **AuthenticDesktopControl PlaceHolder**.

Für Placeholder Controls, die das Authentic Desktop-Projektfenster auswählen, stehen zusätzliche Methoden zur Verfügung. Mit Hilfe von **OpenProject** können Sie ein Authentic Desktop-Projekt laden. Mit Hilfe der Eigenschaft "Project" und der Methoden und Eigenschaften aus der Authentic Desktop Automation-Schnittstelle können Sie andere Operationen im Zusammenhang mit einem Projekt durchführen.

Ein Beispiel, wie Sie in C# oder VB.NET mit Visual Studio eine einfache, aus einem Formular bestehende Applikation, in der die Authentic Desktop ActiveX Controls auf Dokumentebene integriert sind, erstellen, sehen Sie in der Beschreibung unten. Beachten Sie, dass Ihre Applikation bei Bedarf auch komplexer sein kann. In der Anleitung unten sehen Sie jedoch, welche Voraussetzungen mindestens gegeben sein müssen, damit eine ActiveX-Integration auf Dokumentebene erfolgen kann.

1. Erstellen Sie ein neues Visual Studio Windows Forms-Projekt mit einem neuen leeren Formular.
2. Fügen Sie die ActiveX Controls zur Toolbox hinzu, falls dies noch nicht geschehen ist (siehe Hinzufügen der ActiveX Controls zur Toolbox 621).
3. Ziehen Sie das **AuthenticDesktopControl** 651 aus der Toolbox in Ihr neues Formular.

4. Setzen Sie die Eigenschaft **IntegrationLevel** des **AuthenticDesktopControl** auf **ICActiveXIntegrationOnDocumentLevel** und die Eigenschaft **Visible** auf **False**. Sie können dies entweder über den Code oder über das Fenster **Properties** tun.
5. Ziehen Sie das **AuthenticDesktopControl Document**[659] aus der Toolbox in das Formular. Dieses Control stellt das Dokument-Hauptfenster von Authentic Desktop für Ihre Applikation zur Verfügung. Eventuell müssen Sie die Größe des Fensters an die Ihres Dokuments anpassen.
6. Fügen Sie optional ein oder mehrere **AuthenticDesktopControl Placeholder**[666] Controls zum Formular hinzu (eines für jeden zusätzlichen Fenstertyp, der für Ihre Applikation benötigt wird, z.B. das Projektfenster). Solche zusätzlichen Placeholder Controls werden normalerweise entweder unterhalb oder rechts oder links vom Dokument-Haupt-Control platziert z.B.:



7. Setzen Sie die Eigenschaft **PlaceholderWindowID** jedes **AuthenticDesktopControl Placeholder** auf einen gültigen Fenster-Identifier. Eine Liste der gültigen Werte finden Sie unter AuthenticDesktopControlPlaceholderWindow[669].
8. Fügen Sie, wie unten gezeigt, Befehle zu Ihrer Applikation hinzu (Sie benötigen zumindest Befehle zum Öffnen, Speichern und Schließen von Dokumenten).

Abfragen von Authentic Desktop-Befehlen

Bei einer Integration auf Dokumentebene stehen in Ihrer Applikation kein Authentic Desktop-Menü- und keine Authentic Desktop-Symbolleiste zur Verfügung. Sie können statt dessen die benötigten Befehle aufrufen, deren Status anzeigen und diese programmatisch ausführen, wie folgt:

- Um alle verfügbaren Befehle abzurufen, verwenden Sie die Eigenschaft `CommandsList`[653] des **AuthenticDesktopControl**.

 © 2016-2022 Altova GmbH

- Um die Befehle, geordnet nach ihrer Menüstruktur abzurufen, verwenden Sie die Eigenschaft MainMenu[654].
- Um die Befehle, geordnet nach der Symbolleiste, in der sie vorkommen, abzurufen, verwenden Sie die Eigenschaft Toolbars[654].
- Um Befehle an Authentic Desktop zu senden, verwenden Sie die Methode Exec[655].
- Um abzufragen, ob ein Befehl gerade aktiviert oder deaktiviert ist, verwenden Sie die Methode QueryStatus[656].

Auf diese Art können Sie Authentic Desktop-Befehle flexibel in die Menüs und Symbolleisten Ihrer Applikation integrieren.

Über Ihre Installation von Authentic Desktop stehen Ihnen auch Symbole für in Authentic Desktop verwendete Befehle zur Verfügung. Im Ordner **<Applicationsordner>\Examples\ActiveX\Images** Ihrer Authentic Desktop-Installation finden Sie Symbole im GIF-Format. Die Dateinamen entsprechen den im Abschnitt Befehlsreferenz[641] aufgelisteten Befehlsnamen.

Allgemeines

Um das Verhalten von Authentic Desktop zu automatisieren, verwenden Sie die für **AuthenticDesktopControl**[651], **AuthenticDesktopControl Document**[659] und **AuthenticDesktopControl Placeholder**[666] beschriebenen Eigenschaften, Methoden und Ereignisse.

Für einen komplexeren Zugriff auf Authentic Desktop-Funktionalitäten sollten Sie die folgenden Eigenschaften verwenden:

- AuthenticDesktopControl.Application[652]
- AuthenticDesktopControlDocument.Document[661]
- AuthenticDesktopControlPlaceHolder.Project[667]

Über diese Eigenschaften erhalten Sie Zugriff auf die Authentic Desktop Automation-Schnittstelle (<%APPAPI %)

Anmerkung: Verwenden zum Öffnen eines Dokuments immer AuthenticDesktopControlDocument.Open[663] oder AuthenticDesktopControlDocument.New[662] für das entsprechenden Dokument-Control. Verwenden Sie zum Öffnen eines Projekts immer AuthenticDesktopControlPlaceHolder.OpenProject[668] in einem Placeholder Control, das ein Authentic Desktop-Projektfenster einbettet.

Beispiele, wie Sie die erforderlichen Controls in unterschiedlichen Programmierumgebungen instantiieren und aufrufen finden Sie unter Beispiele zur ActiveX-Integration[627].

14.4.5 Beispiele zur ActiveX-Integration

Dieser Abschnitt enthält Beispiele für die Integration von Authentic Desktop auf Dokumentebene über unterschiedliche Container-Umgebungen und Programmiersprachen. Der Quellcode für alle Beispiele steht im Ordner <ApplicationFolder>\Examples\ActiveX Ihrer Authentic Desktop Installation zur Verfügung.

14.4.5.1 C#

Im Ordner `<ApplicationFolder>\Examples\ActiveX\C#` finden Sie ein einfaches Beispiel für eine ActiveX-Integration mittels C# und Visual Studio. Bevor Sie den Quellcode kompilieren und das Beispiel ausführen, stellen Sie sicher, dass alle Voraussetzungen erfüllt werden (siehe Ausführen der C#-Beispiellösung 628).

14.4.5.1.1 Ausführen der C#-Beispiellösung

In der Visual Studio-Beispielprojektmappe im Ordner **<ApplicationFolder>\Examples\ActiveX\C#** wird gezeigt, wie die Authentic Desktop ActiveX Controls verwendet werden. Beachten Sie die folgenden Schritte, bevor Sie versuchen, diese Projektmappen zu erstellen und auszuführen:

Schritt 1: Überprüfen Sie die Voraussetzungen

Um die Beispielprojektmappen öffnen zu können, benötigen Sie Visual Studio 2010 oder höher. Eine Liste der Voraussetzungen finden Sie unter Voraussetzungen 619.

Schritt 2: Kopieren Sie das Beispiel in ein Verzeichnis, auf das Sie Schreibzugriff haben

Um Visual Studio nicht als Administrator ausführen zu müssen, kopieren Sie den Quellcode in ein Verzeichnis, auf das Sie Schreibzugriff haben, anstatt ihn vom Standardverzeichnis aus auszuführen.

Schritt 3: Überprüfen und definieren Sie alle erforderlichen Control-Eigenschaften

Die Beispielapplikation enthält eine Instanz von AuthenticDesktopControlDocument 659 und mehrere Instanzen von AuthenticDesktopControlPlaceHolder 666 Controls. Überprüfen Sie noch einmal, ob die folgenden Eigenschaften dieser Controls, wie in der Tabelle unten angegeben, konfiguriert sind:

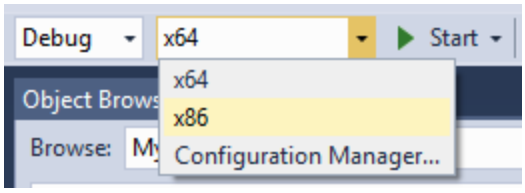
Control-Name	Eigenschaft	Eigenschaftswert
axAuthenticDesktopControl	IntegrationLevel	ICActiveXIntegrationOnDocumentLevel
axAuthenticDesktopControlHelperWndEntities	PlaceholderWindowID	2
axAuthenticDesktopControlHelperWndAttributes	PlaceholderWindowID	1
axAuthenticDesktopControlHelperWndElements	PlaceholderWindowID	0
axAuthenticDesktopControlHelperWndInfo	PlaceholderWindowID	18
axAuthenticDesktopControlHelperWndProject	PlaceholderWindowID	4

So können Sie die Eigenschaften eines ActiveX Control anzeigen oder definieren:

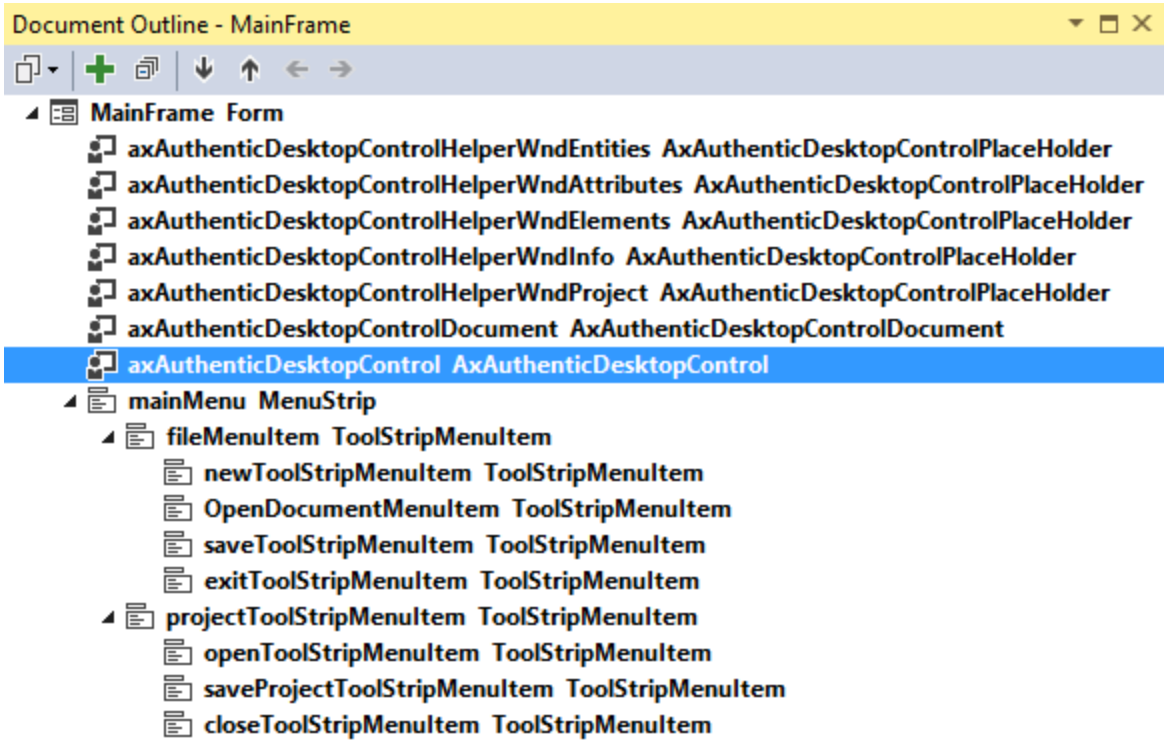
1. Öffnen Sie das Formular **MDIMain.cs** im Designer-Fenster.

 © 2016-2022 Altova GmbH

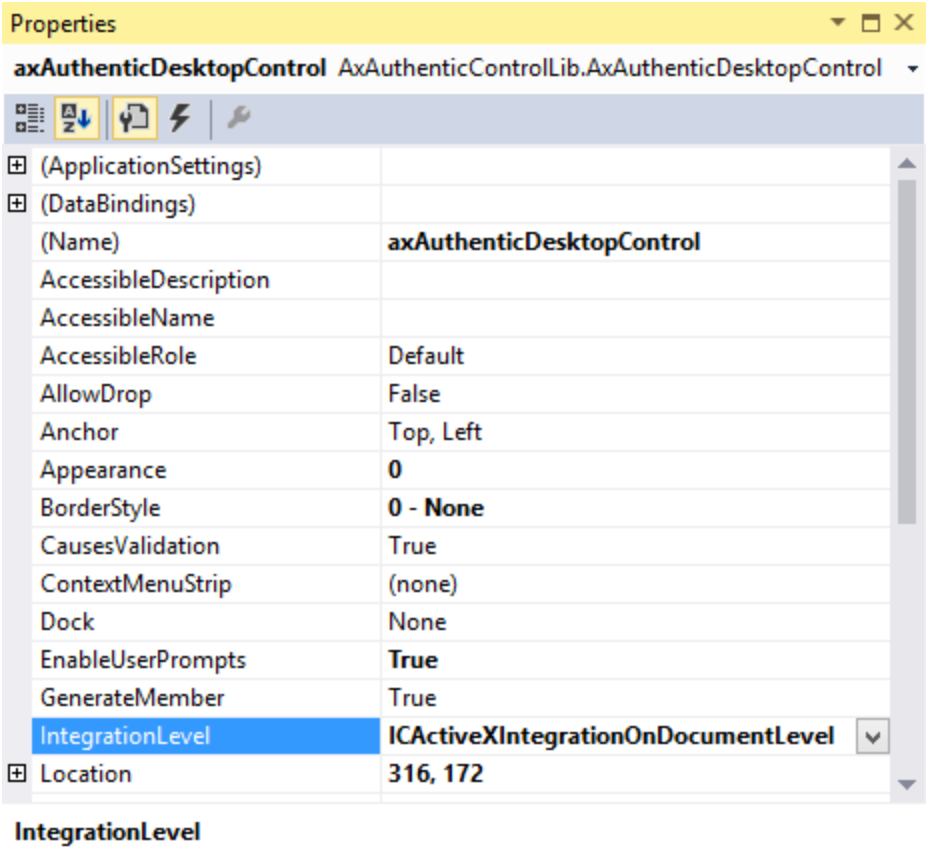
Anmerkung: Unter 64-Bit Windows müssen Sie die Build-Konfiguration der Visual Studio-Projektmappen eventuell in "x86" ändern, **bevor** Sie das Designer-Fenster öffnen. Wenn Sie das Beispiel als 64-Bit-Applikation erstellen müssen, schlagen Sie nach unter Voraussetzungen 619.



2. Öffnen Sie das **Document Outline**-Fenster von Visual Studio (Klicken Sie im Menü **View** auf **Other Windows | Document Outline**).



3. Klicken Sie im Fenster **Document Outline** auf ein ActiveX Control und bearbeiten Sie die benötigte Eigenschaft im Fenster **Properties**, z.B.:

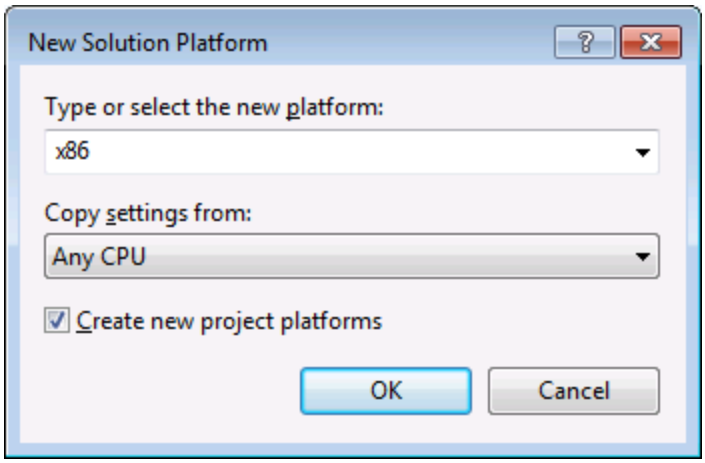


Schritt 4: Definieren Sie die Build-Plattform

- Erstellen Sie eine Build-Plattform-Konfiguration für die Plattform, unter der Sie die Projektmappe erstellen möchten (x86, x64). So erstellen Sie die Build-Konfiguration:

  a. Klicken Sie mit der rechten Maustaste auf die Projektmappe in Visual Studio und wählen Sie **Configuration Manager**.
  b. Wählen Sie unter **Active solution platform** den Befehl **New…** und wählen Sie anschließend die x86- oder x64-Konfiguration (in diesem Beispiel **x86**).

 © 2016-2022 Altova GmbH



Sie sind nun fertig und können die Projektmappe in Visual Studio erstellen und ausführen. Denken Sie daran, die Projektmappe mit der richtigen Konfiguration für Ihre Zielplattform (x86, x64) zu erstellen, da es sonst zu Laufzeitfehlern kommen kann.

14.4.5.2 Java

Authentic Desktop ActiveX-Komponenten können von Java-Code aus aufgerufen werden. Die Java-Integration wird von den unten aufgelisteten Bibliotheken zur Verfügung gestellt. Diese Bibliotheken stehen im Ordner `<ApplicationFolder>\Examples\JavaAPI` Ihrer Authentic Desktop Installation zur Verfügung, nachdem Sie Authentic Desktop und das Authentic Desktop Integrationspaket installiert haben (siehe auch Voraussetzungen 619).

- `AltovaAutomation.dll`: ein JNI Wrapper für Altova Automation Server (bei 32-Bit-Installationen von Authentic Desktop)
- `AltovaAutomation_x64.dll`: ein JNI-Wrapper für Altova Automation Server (bei 64-Bit-Installationen von Authentic Desktop)
- `AltovaAutomation.jar`: Java-Klassen zum Aufrufen von Altova Automation Servern
- `AuthenticActiveX.jar`: Java-Klassen, die als Wrapper für die Authentic ActiveX-Schnittstelle dienen
- `AuthenticActiveX_JavaDoc.zip`: eine Javadoc-Datei, die die Hilfedokumentation zur Java API enthält

Anmerkung: Um die Java ActiveX Integration verwenden zu können, müssen sich die .dll- und .jar-Dateien im Java Class-Suchpfad befinden.

Java-Beispielprojekt

Im Lieferumfang Ihres Produkts ist ein Java-Beispielprojekt enthalten. Sie können das Java-Projekt nach Belieben testen und verwenden. Nähere Informationen finden Sie im Abschnitt Java-Beispielprojekt 633.

Regeln für das Mappen der ActiveX Control-Namen auf Java

Eine Dokumentation zu den ActiveX Controls finden Sie in der Objektreferenz 648. Beachten Sie, dass sich die Objektbenennungskonventionen in Java etwas von anderen Sprachen unterscheiden. Für das Mapping zwischenden ActiveX Controls und dem Java Wrapper gelten die folgenden Regeln:

- ***Klassen und Klassennamen***
  Für jede Komponente des Authentic Desktop ActiveX Interface gibt es eine Java-Klasse mit dem Namen der Komponente.

- ***Methodennamen***
  Die Methodennamen im Java Interface sind dieselben wie die in den COM Interfaces, beginnen aber aufgrund der Java-Namenskonventionen mit einem Kleinbuchstaben. Zum Aufrufen von COM-Eigenschaften können dem Eigenschaftsnamen vorangestellte Java-Methoden mit `get` und `set` verwendet werden. Wenn eine Eigenschaft keinen Schreibzugriff ermöglicht, steht keine Setter-Methode zur Verfügung. Beispiel: Für die Eigenschaft `IntegrationLevel` des AuthenticDesktopControl stehen die Java-Methoden `getIntegrationLevel` und `setIntegrationLevel` zur Verfügung.

- ***Enumerationen***
  Für jede im ActiveX Interface definierte Enumeration ist eine Java-Enumeration desselben Namens und mit denselben Werten definiert.

- ***Events und Event Handler***
  Für jedes Interface im Automation Interface, das Events unterstützt, steht ein Java-Interface desselben Namens plus '`Event`' zur Verfügung. Um das Überladen von Einzel-Events zu vereinfachen, gibt es eine Java-Klasse mit Standardimplementierungen für alle Events. Der Name dieser Java-Klasse ist der Name des Event Interface plus '`DefaultHandler`'. Beispiel:
  `AuthenticDesktopControl`: Java-Klasse zum Aufrufen der Applikation
  `AuthenticDesktopControlEvents`: Events Interface für das AuthenticDesktopControl Control
  `AuthenticDesktopControlEventsDefaultHandler`: Standard-Handler für `AuthenticDesktopControlEvents`

Ausnahmen für Mapping-Regeln

Zu den oben aufgelisteten Regeln gibt es die folgenden Ausnahmen:

Interface	Java-Name
`AuthenticDesktopControlDocument`, Methode `New`	`newDocument`
`Document`, Methode `SetEncoding`	`setFileEncoding`
`AuthenticView`, Methode `Goto`	`gotoElement`
`AuthenticRange`, Methode `Goto`	`gotoElement`
`AuthenticRange`, Methode `Clone`	`cloneRange`

Dieser Abschnitt

In diesem Abschnitt wird erklärt, wie einige grundlegende Authentic Desktop ActiveX-Funktionen über Java-Code aufgerufen werden können. Der Abschnitt ist in die folgenden Unterabschnitte gegliedert:

 © 2016-2022 Altova GmbH

- Java-Beispielprojekt 633
- Erstellen der ActiveX Controls 635
- Laden der Daten in die Controls 636
- Behandlung von Events 636
- Menüs 637
- Behandlung von UI Update Events 638
- Erstellen einer XML-Baumstruktur 639

14.4.5.2.1 Java-Beispielprojekt

Das Authentic Desktop-Installationspaket enthält ein Java-Beispielprojekt, das Sie Ordner "ActiveX Examples" des Applikationsordners `<ApplicationFolder>\Examples\ActiveX\Java\` finden.

Im Java-Beispiel wird gezeigt, wie Sie das AuthenticDesktopControl in eine mit Java erstellte Desktop Applikation integrieren können. Sie können das Beispielprojekt mit Hilfe der Batch-Datei `BuildAndRun.bat`, direkt über die Befehlszeile testen oder Sie können es in Eclipse kompilieren und ausführen. Anleitungen dafür finden Sie weiter unten.

Dateiliste

Der Ordner für die Java-Beispiele enthält alle zum Ausführen des Beispielprojekts erforderlichen Dateien. Diese Dateien sind unten aufgelistet:

`.classpath`	Hilfedatei Eclipse-Projekt
`.project`	Eclipse-Projektdatei
`AltovaAutomation.dll`	Java-COM Bridge: DLL-Teil (für die 32-Bit-Installation)
`AltovaAutomation_x64.dll`	Java-COM Bridge: DLL-Teil (für die 64-Bit-Installation)
`AltovaAutomation.jar`	Java-COM Bridge: Java-Bibliotheksteil
`AuthenticActiveX.jar`	Java-Klassen des Authentic Desktop ActiveX Control
`AuthenticActiveX_JavaDoc.zip`	Javadoc Datei, die die Hilfedokumentation für die Java API enthält
`AuthenticDesktopContainer.java`	Java-Beispielquellcode
`AuthenticDesktopContainerEventHandler.java`	Java-Beispielquellcode
`BuildAndRun.bat`	Batch-Datei zum Kompilieren und Ausführen des Beispielcodes über die Befehlszeile. Es wird ein Ordner benötigt, in dem sich die Java Virtual Machine als Parameter befindet.
`XMLTreeDialog.java`	Java-Beispielquellcode

Funktionen in diesem Beispiel

Im Beispiel werden ein AuthenticDesktop Dokument-Editor-Fenster, das Projektfenster, das Info-Fenster und eine Authentic-Eingabehilfe in ein AWT-Rahmenfenster platziert. Das für Authentic definierte Menü "Datei" wird ausgelesen und ein AWT-Menü mit derselben Struktur wird erstellt. Über dieses Menü oder über das Projektfenster können Sie Dateien im Dokument-Editor öffnen und damit arbeiten.

Sie können das Beispiel nach Ihren Wünschen modifizieren.

In den Codefragmenten werden die folgenden spezifischen Schritte beschrieben:

- Erstellen der ActiveX Controls [635]: Startet Authentic Desktop, das als Automation Server registriert ist, bzw. aktiviert das Programm, wenn Authentic Desktop bereits ausgeführt wird.
- Laden der Daten in die Controls [636]: Navigiert zu einem der mit Authentic Desktop installierten Beispieldokumente und öffnet es.
- Grundlegendes zur Event-Behandlung [636]: Wechselt von der Ansicht aller offenen Dokumente in die Browser-Ansicht. Im Code wird auch gezeigt, wie man durch offene Dokumente iteriert.
- Menüs [637]: Validiert das aktive Dokument und zeigt das Ergebnis in einem Meldungsfeld an. Im Code wird gezeigt, wie Ausgabeparameter verwendet werden.
- Behandlung von UI Update Events [638]: Zeigt, wie Authentic Desktop Events behandelt werden.
-
- Erstellen einer XML-Baumstruktur [639]: Zeigt wie, Sie eine XML-Baumstruktur erstellen und sie für die modale Aktivierung vorbereiten.

Aktualisieren des Pfads zum Ordner "Examples"

Bevor Sie das zur Verfügung gestellte Beispiel ausführen, müssen Sie die **AuthenticDesktopContainer.java**-Datei möglicherweise bearbeiten und überprüfen, ob der folgende Pfad auf den tatsächlichen Ordner verweist, in dem die Authentic Desktop Beispieldateien auf Ihrem Betriebssystem gespeichert sind.

```
// Locate samples installed with the product.
final String strExamplesFolder = System.getenv( "USERPROFILE" ) + "\\Documents\\Altova\
\Authentic2023\\AuthenticExamples\\";
```

Ausführen des Beispiels über die Befehlszeile

So führen Sie das Beispiel über die Befehlszeile aus:

1. Überprüfen Sie, ob alle Voraussetzungen erfüllt werden (siehe Voraussetzungen [619]).
2. Öffnen Sie ein Eingabeaufforderungsfester, wechseln Sie in den Java-Beispielprojektordner und geben Sie folgende Zeile ein:

   ```
   buildAndRun.bat "<Pfad-zum-Java-bin-Ordner>"
   ```

1. Drücken Sie die **Eingabetaste**.

Der Java-Quellcode in `AuthenticDesktopContainer.java` wird kompiliert und anschließend ausgeführt.

Kompilieren und Ausführen des Beispiels in Eclipse

So importieren Sie das Java-Beispielprojekt in Eclipse:

 © 2016-2022 Altova GmbH

3. Überprüfen Sie, ob alle Voraussetzungen erfüllt werden (siehe Voraussetzungen 619).
1. Klicken Sie im Menü **File** auf **Import**.
2. Wählen Sie **Existing Projects into Workspace** und navigieren Sie zur Eclipse-Projektdatei unter `<ApplicationFolder>\Examples\ActiveX\Java\`. Da Sie möglicherweise keinen Schreibzugriff auf diesen Ordner haben, wird empfohlen, im Importdialogfeld das Kontrollkästchen **Copy projects into workspace** zu aktivieren.

Um die Beispielapplikation zu starten, klicken Sie mit der rechten Maustaste im Package Explorer auf das Paket und wählen Sie den Befehl **Run as | Java Application**.

Hilfe zu Java API-Klassen steht in Form von Kommentaren im Code sowie als Javadoc-Ansicht von Eclipse zur Verfügung. Um die Javadoc-Ansicht in Eclipse zu aktivieren, wählen Sie den Menübefehl **Window | Show View | JavaDoc**.

14.4.5.2.2 Erstellen der ActiveX Controls

Im unten gezeigten Code sehen Sie wie ActiveX Controls erstellt werden. Die Konstruktoren erstellen die Java Wrapper-Objekte. Durch Hinzufügen dieser aus Canvas stammenden Objekte zu einem Bereich oder Rahmen wird die Erstellung des in Wrapper verpackten ActiveX-Objekts ausgelöst.

```
01   /**
02    * Authentic Desktop manager control - always needed
03    */
04   public static AuthenticDesktopControl         authenticDesktopControl = null;
05
06   /**
07    * Authentic Desktop document editing control
08    */
09   public static AuthenticDesktopControlDocument     authenticDesktopDocument = null;
10
11   /**
12    * Tool windows - Authentic Desktop place-holder controls
13    */
14   private static AuthenticDesktopControlPlaceHolder   authenticDesktopInfoToolWindow =
null;
15   private static AuthenticDesktopControlPlaceHolder
authenticDesktopEHElementToolWindow = null;
16   private static AuthenticDesktopControlPlaceHolder   authenticDesktopProjectToolWindow
= null;
17
18   // Create the Authentic Desktop ActiveX control, The parameter determines that we want
     // to place document controls and place-holder controls individually.
19   // It gives us full control over the menu, as well.
20      authenticDesktopControl = new AuthenticDesktopControl(
        ICActiveXIntegrationLevel.ICActiveXIntegrationOnDocumentLevel.getValue() );
21
22      authenticDesktopDocument = new AuthenticDesktopControlDocument();
23      authenticDesktopDocument.setPreferredSize( new Dimension ( 640, 480 ) );
24      frame.add( authenticDesktopDocument, BorderLayout.CENTER );
25
26   // Create a project window and open the sample project in it
27      authenticDesktopProjectToolWindow = new AuthenticDesktopControlPlaceHolder(
        XMLSpyControlPlaceholderWindow.XMLSpyControlProjectWindowToolWnd.getValue() );
```

```
28      authenticDesktopProjectToolWindow.setPreferredSize( new Dimension( 200, 200 ) );
```
).

14.4.5.2.3 Laden der Daten in die Controls

Im unten gezeigten Code sehen Sie wie Daten in ActiveX Controls geladen werden können.

```
1    // Locate samples installed with the product.
2       final String strExamplesFolder = System.getenv( "USERPROFILE" ) +
        "\\Documents\\Altova\\Authentic2022\\AuthenticExamples\\";
3       authenticDesktopProjectToolWindow = new
AuthenticDesktopControlPlaceHolder( XMLSpyControlPlaceholderWindow.XMLSpyControlProjectWind
owToolWnd.getValue() );
```

14.4.5.2.4 Grundlegendes zur Event-Behandlung

Im unten gezeigten Code sehen Sie wie grundlegende Events behandelt werden. Bei Aufruf der AuthenticDesktopControl-Methode `open` oder beim Öffnen einer Datei über das Menü oder die Projektstruktur wird das `onOpenedOrFocused` Event an den dazugehörigen Event Handler gesendet. Im Prinzip wird dieses Event behandelt, indem die Datei durch Aufruf der `open`-Methode des Authentic DesktopDocumentControl geöffnet wird.

```
01 // Open the PXF file when button is pressed
02       btnOpenPxf.addActionListener( new ActionListener() {
03         public void actionPerformed(ActionEvent e) {
04           try {
05             authenticDesktopControl.open( strExamplesFolder + "OrgChart.pxf" );
06           } catch (AutomationException e1) {
07             e1.printStackTrace();
08           }
09         }
10       } );
11       public void onOpenedOrFocused( String i_strFileName, boolean
i_bOpenWithThisControl, boolean i_bFileAlreadyOpened ) throws AutomationException
12   {
13     // Handle the New/Open events coming from the Project tree or from the menus
14     if ( !i_bFileAlreadyOpened )
15     {
16       // This is basically an SDI interface, so open the file in the already existing
document control
17       try {
18         AuthenticDesktopContainer.authenticDesktopDocument.open( i_strFileName );
19         AuthenticDesktopContainer.authenticDesktopDocument.requestFocusInWindow();
20       } catch (Exception e) {
21         e.printStackTrace();
22       }
23     }
24   }
```


© 2016-2022 Altova GmbH

14.4.5.2.5 Menüs

Im unten gezeigten Code sehen Sie, wie Menüeinträge erstellt werden können. Jedes `Authentic DesktopCommand` Objekt holt ein entsprechendes `MenuItem` Objekt, wobei der `ActionCommand` auf die ID des Befehls gesetzt wird. Die von allen Menüeinträgen generierten Aktionen werden von derselben Funktion gehandelt. Diese kann bestimmte Handlings (wie z.B Neuinterpretieren der Schließfunktion) durchführen oder die Ausführung durch Aufruf seiner `exec`-Methode an das `AuthenticDesktopControl` Objekt delegieren. Das `menuMap` Objekt, das bei der Menüerstellung befüllt wird, wird später verwendet (siehe Abschnitt Behandlung von UI Update Events 638).

```
01  // Load the file menu when the button is pressed
02       btnMenu.addActionListener( new ActionListener() {
03         public void actionPerformed(ActionEvent e) {
04           try {
05             // Create the menubar that will be attached to the frame
06             MenuBar mb = new MenuBar();
07             // Load the main menu's first item - the File menu
08             XMLSpyCommand xmlSpyMenu =
xmlSpyControl.getMainMenu().getSubCommands().getItem( 0 );
09             // Create Java menu items from the Commands objects
10             Menu fileMenu = new Menu();
11             handlerObject.fillMenu( fileMenu, xmlSpyMenu.getSubCommands() );
12             fileMenu.setLabel( xmlSpyMenu.getLabel().replace( "&", "" ) );
13             mb.add( fileMenu );
14             frame.setMenuBar( mb );
15             frame.validate();
16           } catch (AutomationException e1) {
17             e1.printStackTrace();
18           }
19           // Disable the button when the action has been performed
20           ((AbstractButton) e.getSource()).setEnabled( false );
21         }
22       } ) ;
23 /** * Populates a menu with the commands and submenus contained in an XMLSpyCommands
object  */
24       public void fillMenu(Menu newMenu, XMLSpyCommands xmlSpyMenu) throws
AutomationException
25   {
26     // For each command/submenu in the xmlSpyMenu
27     for ( int i = 0 ; i < xmlSpyMenu.getCount() ; ++i  )
28     {
29       XMLSpyCommand xmlSpyCommand = xmlSpyMenu.getItem( i );
30       if ( xmlSpyCommand.getIsSeparator() )
31         newMenu.addSeparator();
32       else
33       {
34         XMLSpyCommands subCommands = xmlSpyCommand.getSubCommands();
35         // Is it a command (leaf), or a submenu?
36         if ( subCommands.isNull() || subCommands.getCount() == 0 )
37         {
38           // Command -> add it to the menu, set its ActionCommand to its ID and store in
in the menuMap
39           MenuItem mi = new MenuItem( xmlSpyCommand.getLabel().replace( "&", "" ) );
40           mi.setActionCommand( "" + xmlSpyCommand.getID() );
41           mi.addActionListener( this );
```

```
42            newMenu.add( mi );
43            menuMap.put( xmlSpyCommand.getID(), mi );
44          }
45          else
46          {
47            // Submenu -> create submenu and repeat recursively
48            Menu newSubMenu = new Menu();
49            fillMenu( newSubMenu, subCommands );
50            newSubMenu.setLabel( xmlSpyCommand.getLabel().replace( "&", "" ) );
51            newMenu.add( newSubMenu );
52          }
53        }
54      }
55    }
56
57     /**
58     * Action handler for the menu items
59     * Called when the user selects a menu item; the item's action command corresponds to
the command table for XMLSpy
60     */
61    public void actionPerformed( ActionEvent e )
62    {
63      try
64      {
65        int iCmd = Integer.parseInt( e.getActionCommand() );
66        // Handle explicitly the Close commands
67        switch ( iCmd )
68        {
69          case 57602:       // Close
70          case 34050:       // Close All
71            AuthenticDesktopContainer.initXmlSpyDocument();
72            break;
73          default:
74            AuthenticDesktopContainer.xmlSpyControl.exec( iCmd );
75            break;
76        }
77      }
78      catch ( Exception ex )
79      {
80        ex.printStackTrace();
81      }
82
83    }
```

14.4.5.2.6 Behandlung von UI Update Events

Im unten gezeigten Code, sehen Sie wie ein UI-Update Event Handler erstellt werden kann.

```
01     /**
02     * Call-back from the XMLSpyControl.
03     * Called to enable/disable commands
04     */
05    @Override
06    public void onUpdateCmdUI() throws AutomationException
07    {
```

 © 2016-2022 Altova GmbH

```
08      // A command should be enabled if the result of queryStatus contains the Supported
(1) and Enabled (2) flags
09      for ( java.util.Map.Entry<Integer, MenuItem> pair : menuMap.entrySet() )
10
      pair.getValue().setEnabled( AuthenticDesktopContainer.authenticDesktopControl.querySt
atus( pair.getKey() ) > 2 );
11    }
12    /**
13     * Call-back from the XMLSpyControl.
14     * Usually called while enabling/disabling commands due to UI updates
15     */
16    @Override
17    public boolean onIsActiveEditor( String i_strFilePath ) throws AutomationException
18    {
19      try {
20        return
AuthenticDesktopContainer.authenticDesktopDocument.getDocument().getFullName().equalsIgnore
Case( i_strFilePath );
21      } catch ( Exception e ) {
22        return false;
23      }
24    }
```

14.4.5.2.7 Erstellen einer XML-Baumstruktur

Im unten gezeigten Code wird ein XML-Datenobjekt in Form von Nodes in eine Baumstruktur geladen.

```
01 // access required XMLSpy Java-COM classes
02 import com.altova.automation.XMLSpy.XMLData;
03
04 // access AWT and Swing components
05 import java.awt.*;
06 import javax.swing.*;
07 import javax.swing.tree.*;
08
09 /**
10  * A simple example of a tree control loading the structure from an XMLData object.
11  * The class receives an XMLData object, loads its nodes in a JTree, and prepares
12  * for modal activation.
13  *
14  * Feel free to modify and extend this sample.
15  *
16  * @author Altova GmbH
17  */
18 class XMLTreeDialog extends JDialog
19 {
20   /**
21    * The tree control
22    */
23   private JTree myTree;
24
25   /**
26    * Root node of the tree control
27    */
```

```
28   private DefaultMutableTreeNode top ;
29
30   /**
31    * Constructor that prepares the modal dialog containing the filled tree control
32    * @param xml   The data to be displayed in the tree
33    * @param parent  Parent frame
34    */
35   public XMLTreeDialog( XMLData xml, Frame parent )
36   {
37     // Construct the modal dialog
38     super( parent, "XML tree", true );
39     // Arrange controls in the dialog
40     top = new DefaultMutableTreeNode("root");
41     myTree = new JTree(top);
42     setContentPane( new JScrollPane( myTree ) );
43     // Build up the tree
44     fillTree( top, xml );
45     myTree.expandRow( 0 );
46   }
47
48   /**
49    * Loads the nodes of an XML element under a given tree node
50    * @param node  Target tree node
51    * @param elem  Source XML element
52    */
53   private void fillTree( DefaultMutableTreeNode node, XMLData elem)
54   {
55     try
56     {
57       // There are several ways to iterate through child elements: either using the
getFirstChild/getNextChild,
58       // or by incrementing an index up to countChildren and calling getChild [as shown
below].
59       // If you only want to get childer of one kind, you should use
countChildrenKind/getChildKind,
60       // or provide a kind to the getFirstChild before iterating with the getNextChild.
61       int nSize = elem.countChildren() ;
62       for ( int i = 0 ; i < nSize ; ++i)
63       {
64         // Create a new tree node for each child element, and continue recursively
65         XMLData newElem = elem.getChild(i) ;
66         DefaultMutableTreeNode newNode = new DefaultMutableTreeNode( newElem.getName() )
;
67         node.add( newNode ) ;
68         fillTree( newNode, newElem ) ;
69       }
70     }
71     catch (Exception e)
72     {
73       e.printStackTrace();
74     }
75   }
76
77 }
```


© 2016-2022 Altova GmbH

14.4.6 Befehlsreferenz

In diesem Abschnitt werden die Namen und Identifier aller Menübefehle aufgelistet, die in Authentic Desktop zur Verfügung stehen. In den einzelnen Unterabschnitten werden jeweils die Befehle aus dem entsprechenden Menü von Authentic Desktop aufgelistet. Die Befehlstabellen sind folgendermaßen gegliedert:

- In der Spalte "Menübefehl" sehen Sie den Menütext des Befehls, wie er in Authentic Desktop, angezeigt wird, damit Sie die dem Befehl zugrunde liegende Funktionalität leichter erkennen.
- In der Spalte "Befehlsname" ist der String angegeben, anhand dessen ein Symbol desselben Namens aus dem Ordner **ActiveX\Images** des Authentic Desktop Installationsverzeichnis aufgerufen werden kann.
- In den "ID"-Spalten sehen Sie die numerischen Identifier der Spalte, die als Argument an die Methoden geliefert wird, die diesen Befehl ausführen oder abfragen.

Um einen Befehl auszuführen, verwenden Sie die Methoden `AuthenticDesktopControl.Exec` 655 oder `AuthenticDesktopControlDocument.Exec` 662. Um den Status eines Befehls abzufragen, verwenden Sie die Methoden `AuthenticDesktopControl.QueryStatus` 656 oder `AuthenticDesktopControlDocument.QueryStatus` 663.

Einige dieser Befehle werden je nach installierter Authentic Desktop Edition eventuell nicht unterstützt.

14.4.6.1 Menü "Datei"

Das Menü "Datei" enthält die folgenden Befehle:

Menübefehl	Befehlsname	ID
Neu...	ID_FILE_NEW	57600
Öffnen...	ID_FILE_OPEN	57601
Neu laden	IDC_FILE_RELOAD	34065
Kodierung...	IDC_ENCODING	34061
Schließen	ID_FILE_CLOSE	57602
Alle schließen	IDC_CLOSE_ALL	34050
Alle inaktiven schließen	IDC_CLOSE_OTHERS	34271
Speichern	ID_FILE_SAVE	57603
Speichern unter...	ID_FILE_SAVE_AS	57604
Alles speichern	ID_FILE_SAVE_ALL	34208
Als Mail senden...	ID_FILE_SEND_MAIL	57612
Drucken...	ID_FILE_PRINT	57607

Menübefehl	Befehlsname	ID
Druckvorschau	IDC_PRINT_PREVIEW	34104
Druckereinrichtung...	ID_FILE_PRINT_SETUP	57606
Letzte Datei	ID_FILE_MRU_FILE1	57616
Beenden	ID_APP_EXIT	57665

14.4.6.2 Menü "Bearbeiten"

Das Menü "Bearbeiten" enthält die folgenden Befehle:

Menübefehl	Befehlsname	ID
Rückgängig	ID_EDIT_UNDO	57643
Wiederherstellen	ID_EDIT_REDO	57644
Ausschneiden	ID_EDIT_CUT	57635
Kopieren	ID_EDIT_COPY	57634
Einfügen	ID_EDIT_PASTE	57637
Löschen	ID_EDIT_CLEAR	57632
Alles markieren	ID_EDIT_SELECT_ALL	57642
Suchen...	ID_EDIT_FIND	57636
Weitersuchen	ID_EDIT_REPEAT	57640
Ersetzen...	ID_EDIT_REPLACE	57641

14.4.6.3 Menü "Projekt"

Das Menü "Projekt" enthält die folgenden Befehle:

Menübefehl	Befehlsname	ID
Neues Projekt	IDC_ICPROJECTGUI_NEW	37200
Projekt öffnen...	IDC_ICPROJECTGUI_OPEN	37201
Projekt neu laden	IDC_ICPROJECTGUI_RELOAD	37202
Projekt schließen	IDC_ICPROJECTGUI_CLOSE	37203
Projekt speichern	IDC_ICPROJECTGUI_SAVE	37204

© 2016-2022 Altova GmbH

Menübefehl	Befehlsname	ID
Projekt speichern unter...	IDC_ICPROJECTGUI_SAVE_AS	37207
Versionskontrolle aktivieren	ID_SCC_ENABLE	38602
Dateien zu Projekt hinzufügen...	IDC_ICPROJECTGUI_ADD_FILES_TO_PROJECT	37205
Globale Ressource zu Projekt hinzufügen...	IDC_ICPROJECTGUI_ADD_GLOBAL_RESOURCE_TO_PROJECT	37239
URL zu Projekt hinzufügen...	IDC_ICPROJECTGUI_ADD_URL_TO_PROJECT	37206
Aktive Datei zu Projekt hinzufügen	IDC_ICPROJECTGUI_ADD_ACTIVE_FILE_TO_PROJECT	37208
Aktive und verwandte Dateien zu Projekt hinzufügen	IDC_ICPROJECTGUI_ADD_ACTIVE_AND_RELATED_FILES_TO_PROJECT	37209
Projektordner zu Projekt hinzufügen...	IDC_ICPROJECTGUI_ADD_FOLDER_TO_PROJECT	37210
Externes Verzeichnis zum Projekt hinzufügen...	IDC_ICPROJECTGUI_ADD_EXT_FOLDER_TO_PROJECT	37211
Externes Webverzeichnis zum Projekt hinzufügen...	IDC_ICPROJECTGUI_ADD_EXT_URL_FOLDER_TO_PROJECT	37212
Skript-Einstellungen...	IDC_PROJECT_SCRIPT_SETTINGS	34136
Eigenschaften...	IDC_ICPROJECTGUI_PROJECT_PROPERTIES	37223
Letztes Projekt	IDC_ICPROJECTGUI_RECENT	37224

14.4.6.4 Menü "XML"

Das Menü "XML" enthält die folgenden Befehle:

Menübefehl	Befehlsname	ID
Wohlgeformtheit prüfen	IDC_CHECK_WELL_FORM	34049
XML validieren	IDC_VALIDATE	32954

14.4.6.5 Menü "XSL/XQuery"

Das Menü "XSL/XQuery" enthält die folgenden Befehle:

Menübefehl	Befehlsname	ID
XSL-Transformation	IDC_TRANSFORM_XSL	33006
XSL:FO-Transformation	IDC_TRANSFORM_XSLFO	33007
XSL-Parameter / XQuery-Variablen...	IDC_TRANSFORM_XSL_PARAMS	33008

14.4.6.6 Menü "Authentic"

Das Menü "Authentic" enthält die folgenden Befehle:

Menübefehl	Befehlsname	ID
Neues Dokument...	IDC_AUTHENTIC_NEW_FILE	34036
Datenbank bearbeiten...	IDC_AUTHENTIC_EDIT_DB	34035
StyleVision Stylesheet bearbeiten...	IDC_EDIT_SPS	34060
Neue Zeile mit XML-Daten für Bearbeitung auswählen...	IDC_CHANGE_WORKING_DB_XML_CELL	32861
XML-Signatur...	IDC_AUTHENTICGUI_XMLSIGNATURE	32862
XML Entities definieren...	IDC_DEFINE_ENTITIES	32805
Markup-Symbole ausblenden	IDC_MARKUP_HIDE	32855
Kleine Markup-Symbole einblenden	IDC_MARKUP_SMALL	32858
Große Markup-Symbole einblenden	IDC_MARKUP_LARGE	32856
Gemischte Markup-Symbole einblenden	IDC_MARKUP_MIXED	32857
Fett ein/aus	IDC_AUTHENTICGUI_RICHEDIT_TOGGLEBOLD	32813
Kursiv ein/aus	IDC_AUTHENTICGUI_RICHEDIT_TOGGLEITALIC	32814
Unterstrichen ein/aus	IDC_AUTHENTICGUI_RICHEDIT_TOGGLEUNDERLINE	32815
Durchgestrichen ein/aus	IDC_AUTHENTICGUI_RICHEDIT_TOGGLESTRIKETHROUGH	32816
Vordergrundfarbe	IDC_AUTHENTICGUI_RICHEDIT_COLOR_FOREGROUND	32824
Hintergrundfarbe	IDC_AUTHENTICGUI_RICHEDIT_COLOR_BACKGROUND	32830

© 2016-2022 Altova GmbH

Menübefehl	Befehlsname	ID
Linksbündig	IDC_AUTHENTICGUI_RICHEDIT_ALIGN_LEFT	32818
Zentriert	IDC_AUTHENTICGUI_RICHEDIT_ALIGN_CENTER	32819
Rechtsbündig	IDC_AUTHENTICGUI_RICHEDIT_ALIGN_RIGHT	32820
Zeile anhängen	IDC_ROW_APPEND	32806
Zeile einfügen	IDC_ROW_INSERT	32809
Zeile kopieren	IDC_ROW_DUPLICATE	32808
Zeile nach oben	IDC_ROW_MOVE_UP	32811
Zeile nach unten	IDC_ROW_MOVE_DOWN	32810
Zeile löschen	IDC_ROW_DELETE	32807
HTML-Dokument generieren	IDC_PXF_GENERATE_HTML	34283
RTF-Dokument generieren	IDC_PXF_GENERATE_RTF	34284
PDF-Dokument generieren	IDC_PXF_GENERATE_PDF	34285
Word 2007+-Dokument generieren	IDC_PXF_GENERATE_DOCX	34286
Vertrauenswürdige Pfade...	IDC_TRUSTED_LOCATIONS	34288

14.4.6.7 Menü "Ansicht"

Das Menü "Ansicht" enthält die folgenden Befehle:

Menübefehl	Befehlsname	ID
Authentic-Ansicht	IDC_VIEW_CONTENT	34177
Browser-Ansicht	IDC_VIEW_BROWSER	34176
Einstellungen für Textansicht	IDC_TEXTVIEW_SETTINGS	34119

14.4.6.8 Menü "Browser"

Das Menü "Browser" enthält die folgenden Befehle:

Menübefehl	Befehlsname	ID
Zurück	IDC_STEP_BACK	32958

Menübefehl	Befehlsname	ID
Vorwärts	IDC_STEP_FORWARD	32957
Abbrechen	IDC_BROWSER_STOP	34047
Aktualisieren	IDC_BROWSER_REFRESH	34046
Maximale Größe	IDC_BROWSER_FONT_LARGEST	34041
Größer	IDC_BROWSER_FONT_LARGE	34040
Mittel	IDC_BROWSER_FONT_MEDIUM	34042
Kleiner	IDC_BROWSER_FONT_SMALL	34043
Minimale Größe	IDC_BROWSER_FONT_SMALLEST	34044

14.4.6.9 Menü "Extras"

Das Menü "Extras" enthält die folgenden Befehle:

Menübefehl	Befehlsname	ID
Rechtschreibung...	IDC_SPELL_CHECK	34154
Rechtschreiboptionen...	IDC_SPELL_OPTIONS	34155
Skript-Editor...	ID_SCRIPTFORMEDITOR_EDIT_PROJECT	39666
Keines	ID_SCRIPTFORMEDITOR_EXECUTE_MACRO_MENU_UPPDATE	39600
	IDC_TOOLS_ENTRY	34292
Globale Ressourcen	IDC_GLOBALRESOURCES	37401
	IDC_GLOBALRESOURCES_SUBMENUENTRY1	37408
Anpassen...	IDC_APP_TOOLS_CUSTOMIZE	32959
Optionen...	IDC_SETTINGS	34133
	ID_SCRIPTING_MACROITEMS	34249

14.4.6.10 Menü "Fenster"

Das Menü "Fenster" enthält die folgenden Befehle:

© 2016-2022 Altova GmbH

Menübefehl	Befehlsname	ID
Überlappend	ID_WINDOW_CASCADE	57650
Horizontal anordnen	ID_WINDOW_TILE_HORZ	57651
Vertikal anordnen	ID_WINDOW_TILE_VERT	57652
Projekt-Fenster	IDC_PROJECT_WINDOW	34128
Info-Fenster	IDC_INFO_WINDOW	34085
Eingabehilfen	IDC_ENTRY_HELPERS	34062
Ausgabefenster	IDC_OUTPUT_DIALOGBARS	34004
Projekt und Eingabehilfen	IDC_PROJECT_ENTRYHELPERS	34006
Alle ein/aus	IDC_ALL_BARS	34031

14.4.6.11 Menü "Hilfe"

Das Menü "Hilfe" enthält die folgenden Befehle:

Menübefehl	Befehlsname	ID
Inhaltsverzeichnis...	IDC_HELP_CONTENTS	32966
Index...	IDC_HELP_INDEX	32967
Suchen...	IDC_HELP_SEARCH	32969
Tastaturzuordnungen...	IDC_HELP_KEYMAPDLG	32968
Software-Aktivierung...	IDC_ACTIVATION	32970
Bestellformular...	IDC_OPEN_ORDER_PAGE	32971
Registrieren...	IDC_REGISTRATION	32972
Auf Updates überprüfen...	IDC_CHECK_FOR_UPDATES	32973
XMLSpy-Produktvergleich...	IDC_PRODUCT_COMPARISON	32955
Support Center...	IDC_OPEN_SUPPORT_PAGE	32961
Fragen und Antworten im Web...	IDC_OPEN_FAQ_PAGE	32962
Komponenten und Gratistools downloaden...	IDC_OPEN_COMPONENTS_PAGE	32963
Authentic im Internet...	IDC_OPEN_HOME_PAGE	32964
Authentic Training...	IDC_OPEN_TRAINING_PAGE	32965

Menübefehl	Befehlsname	ID
Über Authentic...	ID_APP_ABOUT	57664

14.4.7 Objektreferenz

Objekte:
Authentic DesktopCommand 648
Authentic DesktopCommands 650
AuthenticDesktopControl 651
AuthenticDesktopControlDocument 659
AuthenticDesktopControlPlaceHolder 666

Um Zugriff auf die Authentic Desktop Standardfunktionalitäten zu erhalten, können auch Objekte des **Authentic Desktop Automation Interface** aufgerufen werden. Nähere Informationen dazu finden Sie unter AuthenticDesktopControl.Application 652, AuthenticDesktopControlDocument.Document 661 und AuthenticDesktopControlPlaceHolder.Project 667.

14.4.7.1 Authentic DesktopCommand

Eigenschaften:
`ID` 649
`Label` 649
`Name` 650
`IsSeparator` 649
`ToolTip` 650
`StatusText` 650
`Accelerator` 649
`SubCommands` 650

Beschreibung:
Jedes Command Objekt kann einer von drei möglichen Typen sein: ein ausführbarer Befehl, ein Befehlscontainer (z.B. ein Menü, ein Untermenü oder eine Symbolleiste) oder ein Menütrennzeichen. Um herauszufinden, welche Art von Informationen im aktuellen `Command`-Objekt gespeichert sind, fragen Sie seine Eigenschaften `ID`, `IsSeparator` und `SubCommands` folgendermaßen ab.

Das Befehlsobjekt ist...	wenn...
ein ausführbarer Befehl	• `ID` größer als Null ist • `IsSeparator` "false" ist • `SubCommands` leer ist
ein Befehlscontainer	• `ID` Null ist • `IsSeparator` "false" ist • `SubCommands` eine Sammlung von `Command` Objekten enthält
ein Trennzeichen	• `ID` Null ist

 © 2016-2022 Altova GmbH

Das Befehlsobjekt ist...	wenn...
	• `IsSeparator` "true" ist

14.4.7.1.1 Accelerator

Eigenschaft: `Accelerator` als string

Beschreibung:
Gibt die für den Befehl definierte Zugriffstaste zurück. Wenn dem Befehl keine Zugriffstaste zugewiesen wurde, gibt diese Eigenschaft den leeren String zurück. Die String-Darstellung der Zugriffstaste hat das folgenden Format:

```
[ALT+][CTRL+][SHIFT+]key
```

`key` wird mittels der Windows Plattform SDK-Funktion `GetKeyNameText` konvertiert.

14.4.7.1.2 ID

Eigenschaft: `ID` als long

Beschreibung:
Mit dieser Eigenschaft wird der eindeutige Identifier des Befehls abgerufen. Die ID eines Befehls wird benötigt, um den Befehl (mittels `Exec` 655) auszführen oder seinen Status (mittels `QueryStatus` 656) abzurufen. Wenn der Befehl ein Container für andere Befehle (z.B. für ein Menü der obersten Ebene) oder ein Trennzeichen ist, ist die ID 0.

14.4.7.1.3 IsSeparator

Eigenschaft: `IsSeparator` als boolean

Beschreibung:
Die Eigenschaft gibt `true` zurück, wenn das Befehlsobjekt ein Menütrennzeichen ist; andernfalls wird `false` zurückgegebven. Siehe auch Command 648.

14.4.7.1.4 Label

Eigenschaft: `Label` als string

Beschreibung:
Diese Eigenschaft ruft den Text des Befehls, wie er auf der grafischen Benutzeroberfläche von Authentic Desktop angezeigt wird, ab. Wenn der Befehl ein Trennzeichen ist, ist "Label" ein leerer String. Diese

Eigenschaft kann für einige Symbolleistenbefehle, zu denen es keinen GUI-Text gibt, auch einen leeren String zurückgeben.

14.4.7.1.5 Name

Eigenschaft: `Name` als string

Beschreibung:
Diese Eigenschaft ruft den eindeutigen Namen des Befehls ab. Anhand dieses Werts kann die Symboldatei das Befehls, falls verfügbar, abgerufen werden. Die verfügbaren Symboldateien befinden sich im Ordner **<ApplicationFolder>\Examples\ActiveX\Images** Ihrer Authentic Desktop-Installation.

14.4.7.1.6 StatusText

Eigenschaft: `Label` als string

Beschreibung:
Der Statustext ist der Text, der in der Statusleiste von Authentic Desktop angezeigt wird, wenn der Befehl ausgewählt wird. Dies gilt nur für Befehlsobjekte, die keine Trennzeichen oder Container von anderen Befehlen sind. Andernfalls ist die Eigenschaft ein leerer String.

14.4.7.1.7 SubCommands

Eigenschaft: `SubCommands` als `Commands` 650

Beschreibung:
Die Eigenschaft `SubCommands` ruft die Sammlung von `Command` 648-Objekten, die dem aktuellen Befehl untergeordnet sind, auf. Die Eigenschaft gilt nur für Befehle, die Container für andere Befehle sind (Menüs, Untermenüs oder Symbolleisten). Bei solchen Container-Befehlen ist die `ID` auf `0` gesetzt und die Eigenschaft `IsSeparator` auf `false`.

14.4.7.1.8 ToolTip

Eigenschaft: `ToolTip` als string

Beschreibung:
Diese Eigenschaft ruft den Text, der als Tooltipp zu den einzelnen Befehlen angezeigt wird, ab. Wenn der Befehl keinen Tooltipp-Text hat, so gibt die Eigenschaft einen leeren String zurück.

14.4.7.2 Authentic DesktopCommands

Eigenschaften:
`Count` 651
`Item` 651

© 2016-2022 Altova GmbH

Beschreibung:
Sammlung von Command[648]-Objekten für den Zugriff auf Befehlsbezeichnungen und IDs des AuthenticDesktopControl. Diese Befehle können mit der Exec[655] Methode ausgeführt werden und ihr Status kann mittels QueryStatus[656] abgefragt werden.

14.4.7.2.1 Count

Eigenschaft: `Count` als long

Beschreibung:
Anzahl der Command[648] Objekte auf dieser Ebene der Collection

14.4.7.2.2 Item

Eigenschaft: `Item` (`n` als long) als Command[648]

Beschreibung:
Ruft den Befehl mit den Index `n` in dieser Sammlung auf. Der Index basiert auf 1.

14.4.7.3 AuthenticDesktopControl

Eigenschaften:
IntegrationLevel[653]
Appearance[652]
Application[652]
BorderStyle[652]
CommandsList[653]
EnableUserPrompts[653]
MainMenu[654]
Toolbars[654]

Methoden:
Open[656]
Exec[655]
QueryStatus[656]

Events:
OnUpdateCmdUI[659]
OnOpenedOrFocused[658]
OnCloseEditingWindow[657]
OnFileChangedAlert[658]
OnContextChanged[657]
OnDocumentOpened[657]
OnValidationWindowUpdated[659]

Dieses Objekt ist ein vollständiges ActiveX Control und sollte nur sichtbar sein, wenn die Authentic Desktop Bibliothek im Applikationsebenenmodus verwendet wird.

14.4.7.3.1 Eigenschaften

Es sind die folgenden Eigenschaften definiert:

IntegrationLevel 653
EnableUserPrompts 653
Appearance 652
BorderStyle 652

Befehlsbezogene Eigenschaften:
CommandsList 653
MainMenu 654
Toolbars 654

Zugriff auf die AuthenticDesktopAPI:
Application 652

14.4.7.3.1.1 Appearance

Eigenschaft: `Appearance` als short

Dispatch Id: *-520*

Beschreibung:
Bei einem Wert, der nicht gleich `0` ist, wird ein Client-Rand rund um das Control angezeigt. Der Standardwert ist `0`.

14.4.7.3.1.2 Application

Eigenschaft: `Application` als Application

Dispatch Id: *1*

Beschreibung:
Mit der Eigenschaft `Application` erhalten Sie Zugriff auf das `Application` Objekt der vollständigen Authentic Desktop Automation Server API. Die Eigenschaft ist schreibgeschützt.

14.4.7.3.1.3 BorderStyle

Eigenschaft: `BorderStyle` als short

Dispatch Id: *-504*

Beschreibung:
Bei einem Wert `1` wird das Control mit einer dünnen Umrandung angezeigt. Der Standardwert ist `0`.

 © 2016-2022 Altova GmbH

14.4.7.3.1.4 CommandsList

Eigenschaft: `CommandList` als `Commands` 650 (schreibgeschützt)

Dispatch Id: *1004*

Beschreibung:
Diese Eigenschaft gibt eine flache Liste aller Befehle zurück, die mit AuthenticDesktopControl verfügbar sind. Um Befehle, geordnet nach Menüstruktur, abzurufen, verwenden Sie `MainMenu` 654. Um Symbolleistenbefehle abzurufen, verwenden Sie `Toolbars` 654.

```
public void GetAllAuthenticCommands()
{
    // Get all commands from the Authentic ActiveX control assigned to the current form
    AuthenticControlLib.XMLSpyCommands commands =
this.axAuthenticDesktopControl1.CommandList;
    // Loop through all commands
    for (int i = 0; i < commands.Count; i++)
    {
      // Get each command by index and output it to the console
      AuthenticControlLib.XMLSpyCommand cmd = axAuthenticDesktopControl1.CommandList[i];
      Console.WriteLine("{0} {1} {2}", cmd.ID, cmd.Name, cmd.Label.Replace("&", ""));
    }
}
```

C#-Beispiel

14.4.7.3.1.5 EnableUserPrompts

Eigenschaft: `EnableUserPrompts` als boolean

Dispatch Id: *1006*

Beschreibung:
Wenn Sie diese Eigenschaft auf *false* setzen, wird die Eingabeaufforderung im Control deaktiviert. Der Standardwert ist *true*.

14.4.7.3.1.6 IntegrationLevel

Eigenschaft: `IntegrationLevel` als `ICActiveXIntegrationLevel` 669

Dispatch Id: *1000*

Beschreibung:
Die Eigenschaft `IntegrationLevel` bestimmt den Operationsmodus des Control. Nähere Informationen dazu siehe auch Integration auf Applikationsebene 622 und Integration auf Dokumentebene 624.

Anmerkung: Diese Eigenschaft muss unbedingt sofort nach Erstellung des `AuthenticDesktopControl` Objekts definiert werden.

14.4.7.3.1.7 MainMenu

Eigenschaft: `MainMenu` als Command 648 (schreibgeschützt)

Dispatch Id: *1003*

Beschreibung:
Diese Eigenschaft enthält Informationen über die Struktur und die Befehle im AuthenticDesktopControl-Hauptmenü als `Command`-Objekt. Das `Command`-Objekt enthält alle verfügbaren Untermenüs von Authentic Desktop (z.B. Datei, Bearbeiten, Ansicht, usw.). Verwenden Sie die Eigenschaft `SubCommands` der Eigenschaft `MainMenu`, um die Untermenüobjekte abzurufen. Jedes Untermenü ist ebenfalls ein `Command`-Objekt. Sie können bei jedem Untermenü weiter durch dessen `SubCommands`-Eigenschaft iterieren, um die jeweiligen Child-Befehle und Trennzeichen dieser Untermenüs abzurufen (Auf diese Art können Sie z.B. das Applikationsmenü programmatisch erstellen). Beachten Sie, dass einige Menübefehle als Container ("Parents") für andere Menübefehle dienen. In diesen Fällen haben diese ebenfalls eine Eigenschaft `SubCommands`. Um die Struktur aller Menübefehle programmatisch abzurufen, müssen Sie wahrscheinlich eine rekursive Funktion erstellen.

```
public void GetAuthenticMenus()
{
    // Get the main menu from the Authentic ActiveX control assigned to the current form
    AuthenticControlLib.XMLSpyCommand mainMenu =
this.axAuthenticDesktopControl1.MainMenu;

    // Loop through entries of the main menu (e.g. File, Edit, etc.)
    for (int i = 0; i < mainMenu.SubCommands.Count; i++)
    {
      AuthenticControlLib.XMLSpyCommand menu = mainMenu.SubCommands[i];
      Console.WriteLine("{0} menu has {1} children items (including separators)",
menu.Label.Replace("&", ""), menu.SubCommands.Count);
    }
}
```

C# example

14.4.7.3.1.8 Toolbars

Eigenschaft: `Toolbars` als Commands 650 (schreibgeschützt)

Dispatch Id: *1005*

 © 2016-2022 Altova GmbH

Beschreibung:
Diese Eigenschaft enthält Informationen über die Struktur von AuthenticDesktopControl-Symbolleisten, als `Command`-Objekt. Das `Command`-Objekt enthält alle verfügbaren Symbolleisten von Authentic Desktop. Verwenden Sie die Eigenschaft `SubCommands` der Eigenschaft `Toolbars`, um die Symbolleisten abzurufen. Jede Symbolleiste ist ebenfalls ein `Command`-Objekt. Sie können bei jeder Symbolleiste weiter durch deren `SubCommands`-Eigenschaft iterieren, um deren Befehle abzurufen (Auf diese Art können Sie z.B. die Symbolleisten der Applikation programmatisch erstellen).

```
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++)
    {
      AuthenticControlLib.XMLSpyCommand toolbar = toolbars[i];
      Console.WriteLine();
      Console.WriteLine("The toolbar \"{0}\" has the following commands:",
toolbar.Label);

      // Iterate through all commands of this toolbar
      for (int j = 0; j < toolbar.SubCommands.Count; j++)
      {
         AuthenticControlLib.XMLSpyCommand cmd = toolbar.SubCommands[j];
         // Output only command objects that are not separators
         if ( ! cmd.IsSeparator)
         {
            Console.WriteLine("{0}, {1}, {2}", cmd.ID, cmd.Name, cmd.Label.Replace("&",
""));
         }
      }
    }
}
```

C# example

14.4.7.3.2 Methoden

Es sind die folgenden Methoden definiert:

`Open` 656
`Exec` 655
`QueryStatus` 656

14.4.7.3.2.1 Exec

Methode: `Exec` (`nCmdID` als long) als boolean

Dispatch Id: *6*

Beschreibung:
`Exec` ruft den Authentic Desktop Befehl mit der ID `nCmdID` auf. Wenn der Befehl ausgeführt werden kann, gibt die Methode `true` zurück. Eine Liste aller verfügbaren Befehle finden Sie unter `CommandsList` 653. Um den Status eines Befehls abzurufen, verwenden Sie `QueryStatus` 656.

14.4.7.3.2.2 Open

Methode: `Open` (`strFilePath` als string) als boolean

Dispatch Id: *5*

Beschreibung:
Das Ergebnis der Methode ist von der Erweiterung abhängig, die im Argument `strFilePath` übergeben wird. Ist die Dateierweiterung `.sps`, wird ein neues Dokument geöffnet. Ist die Dateierweiterung `.svp`, wird das entsprechende Projekt geöffnet. Wird eine andere Dateierweiterung in die Methode übergeben, versucht das Control, die Datei als neue Komponente in das aktive Dokument zu laden.

Verwenden Sie diese Methode nicht, um Dokumente oder Projekte bei Verwendung des Control auf Dokumentebene zu laden. Verwenden Sie statt dessen `AuthenticDesktopControlDocument.Open` 663 und `AuthenticDesktopControlPlaceHolder.OpenProject` 668.

14.4.7.3.2.3 QueryStatus

Methode: `QueryStatus` (`nCmdID` als long) als long

Dispatch Id: *7*

Beschreibung:
`QueryStatus` gibt den Status "enabled/disabled" und "checked/unchecked" des von `nCmdID`. definierten Befehls zurück. Der Status wird als Bitmaske zurückgegeben.

Bit	Wert	Name	Bedeutung
0	1	Supported	Setzen, wenn der Befehl unterstützt wird.
1	2	Enabled	Setzen, wenn der Befehl aktiviert ist (ausgeführt werden kann).
2	4	Checked	Setzen, wenn der Befehl angehakt ist.

Das bedeutet, dass die Befehls-ID bei Rückgabe von 0 durch `QueryStatus` nicht als gültiger Authentic Desktop Befehl erkannt wird. Wenn `QueryStatus` einen Wert `1` oder `5` zurückgibt, wird der Befehl deaktiviert.

© 2016-2022 Altova GmbH

14.4.7.3.3 Events

Das AuthenticDesktopControl ActiveX Control stellt die folgenden Verbindungspunkt-Events bereit:

OnUpdateCmdUI 659
OnOpenedOrFocused 658
OnCloseEditingWindow 657
OnFileChangedAlert 658
OnContextChanged 657

OnDocumentOpened 657
OnValidationWindowUpdated 659

14.4.7.3.3.1 OnCloseEditingWindow

Event: `OnCloseEditingWindow` (i_strFilePath als String) als boolean

Dispatch Id: *1002*

Beschreibung:
Dieses Event wird ausgelöst, wenn Authentic Desktop ein bereits geöffnetes Dokument schließen muss. Als Antwort auf dieses Event müssen Clients das mit *i_strFilePath* verknüpfte Bearbeitungsfenster schließen. Bei Rückgabe von *true* von diesem Event, wird angezeigt, dass der Client das Dokument geschlossen hat. Clients können *false* zurückgeben, wenn keine bestimmte Behandlung erforderlich ist und AuthenticDesktopControl versuchen soll, das Bearbeitungsfenster zu schließen und das damit verknüpfte Dokument-Control zu zerstören.

14.4.7.3.3.2 OnContextChanged

Event: `OnContextChanged` (`i_strContextName` als String, `i_bActive` als bool) als bool

Dispatch Id: *1004*

Beschreibung:
Dieses Event wird in Authentic Desktop nicht verwendet

14.4.7.3.3.3 OnDocumentOpened

Event: `OnDocumentOpened` (`objDocument` als Document)

Dispatch Id: *1*

Beschreibung:
Dieses Event wird immer, wenn ein Dokument geöffnet wird, ausgelöst. Das Argument `objDocument` ist ein `Document` Objekt aus dem Authentic Desktop Automation Interface und kann dazu verwendet werden, weitere Details zum Dokument abzurufen oder weitere Operationen durchzuführen. Bei Integration auf Dokumentebene

ist es oft besser, stattdessen das Event `AuthenticDesktopControlDocument.OnDocumentOpened` 665 zu verwenden.

14.4.7.3.3.4 OnFileChangedAlert

Event: `OnFileChangedAlert` (`i_strFilePath` als String) als bool

Dispatch Id: *1001*

Beschreibung:
Dieses Event wird ausgelöst, wenn eine mit AuthenticDesktopControl geladene Datei auf der Festplatte von einer anderen Applikation geändert wurde. Clients sollten true zurückgeben, wenn Sie das Event behandelt haben oder false, wenn Authentic Desktop es auf die übliche Art behandeln soll, also fragen soll, ob das Dokument neu geladen werden soll.

14.4.7.3.3.5 OnLicenseProblem

Event: `OnLicenseProblem` (`i_strLicenseProblemText` als String)

Dispatch Id: *1005*

Beschreibung:
Dieses Event wird ausgelöst, wenn AuthenticDesktopControl feststellt, dass für dieses Control keine gültige Lizenz vorhanden ist. Wenn die Lizenz auf eine bestimmte Anzahl an Benutzern beschränkt ist, kann dies auch einige Zeit nach Initialisierung des Control geschehen. Dieses Event sollte dazu verwendet werden, um den Zugriff auf die Funktionalität dieses Control zu deaktivieren. Nach diesem Event blockiert dieses Control den Zugriff auf seine Funktionalitäten (und zeigt z.B. leere Fenster in seinen Controls an und gibt bei Anfragen Fehlermeldungen zurück).

14.4.7.3.3.6 OnOpenedOrFocused

Event: `OnOpenedOrFocused` (`i_strFilePath` als String, `i_bOpenWithThisControl` als bool)

Dispatch Id: *1000*

Beschreibung:
Bei Integration auf Applikationsebene informiert dieses Event Clients, dass ein Dokument von Authentic Desktop geöffnet oder aktiv gemacht wurde.

Bei Integration auf Dokumentebene gibt dieses Event dem Client die Anweisung, die Datei `i_strFilePath` in einem Dokumentfenster zu öffnen. Wenn die Datei bereits offen ist, sollte das entsprechende Dokumentfenster zum aktiven Fenster gemacht werden.

Wenn `i_bOpenWithThisControl` `true` ist, muss das Dokument mit AuthenticDesktopControl geöffnet werden, da ein interner Zugriff erforderlich ist. Andernfalls kann die Datei mit anderen Editoren geöffnet werden.

 © 2016-2022 Altova GmbH

14.4.7.3.3.7 OnToolWindowUpdated

Event: `OnToolWindowUpdated(pToolWnd` als long `)`

Dispatch Id: *1006*

Beschreibung:
Dieses Event wird ausgelöst, wenn das Fenster "Extras" aktualisiert wird.

14.4.7.3.3.8 OnUpdateCmdUI

Event: `OnUpdateCmdUI ()`

Dispatch Id: *1003*

Beschreibung:
Wird häufig aufgerufen, um dem Programmierer die Möglichkeit zu geben, den Status von Authentic Desktop Befehls mittels `AuthenticDesktopControl.QueryStatus` 656 zu überprüfen. Führen Sie an diesem Callback keine langen Operationen durch.

14.4.7.3.3.9 OnValidationWindowUpdated

Event: `OnValidationWindowUpdated` ()

Dispatch Id: *3*

Beschreibung:
Dieses Event wird ausgelöst, wenn das Fenster "Validierungsausgabe" mit neuen Informationen aktualisiert wird.

14.4.7.4 AuthenticDesktopControlDocument

Eigenschaften:
`Appearance` 660
`BorderStyle` 660
`Document` 661
`IsModified` 661
`Path` 661
`ReadOnly` 661

Methoden:
`Exec` 662
`New` 662
`Open` 663
`QueryStatus` 663

Reload 663
Save 663
SaveAs 664

Events:
OnDocumentOpened 665
OnDocumentClosed 665
OnModifiedFlagChanged 666
OnContextChanged 664
OnFileChangedAlert 665
OnActivate 664

Wenn das AuthenticDesktopControl im Modus auf Dokumentebene integriert ist, wird jedes Dokument in einem eigenen Objekt vom Typ `AuthenticDesktopControlDocument` angezeigt. Das `AuthenticDesktopControlDocument` enthält immer nur ein Dokument, kann aber wiederverwendet werden, um hintereinander mehrere Dateien anzuzeigen.

Dieses Objekt ist ein komplettes ActiveX Control.

14.4.7.4.1 Eigenschaften

Es sind die folgenden Eigenschaften definiert:

ReadOnly 661
IsModified 661
Path 661
Appearance 660
BorderStyle 660

Zugriff auf die AuthenticDesktopAPI:
Document 661

14.4.7.4.1.1 Appearance

Eigenschaft: `Appearance` als short

Dispatch Id: *-520*

Beschreibung:
Bei einem Wert, der nicht gleich `0` ist, wird ein Client-Rand rund um das Control angezeigt. Der Standardwert ist `0`.

14.4.7.4.1.2 BorderStyle

Eigenschaft: `BorderStyle` als short

© 2016-2022 Altova GmbH

Dispatch Id: *-504*

Beschreibung:
Bei einem Wert `1` wird das Control mit einer dünnen Umrandung angezeigt. Der Standardwert ist `0`.

14.4.7.4.1.3 Document

Eigenschaft: `Document` als Document

Dispatch Id: *3*

Beschreibung:
Mit der Eigenschaft `Document` erhalten Sie Zugriff auf das `Document` Objekt der Authentic Desktop Automation Server API. Diese Schnittstelle bietet zusätzliche Funktionalitäten, die mit dem im Control geladenen Dokument verwendet werden können. Die Eigenschaft ist schreibgeschützt.

14.4.7.4.1.4 IsModified

Eigenschaft: `IsModified` als boolean (schreibgeschützt)

Dispatch Id: *1006*

Beschreibung:
`IsModified` ist *true,* wenn der Dokumentinhalt seit dem letzten Öffnen, Neuladen oder Speichern geändert wurde. Andernfalls ist es *false*.

14.4.7.4.1.5 Path

Eigenschaft: `Path` als String

Dispatch Id: *1005*

Beschreibung:
Definiert den vollständigen Pfadnamen des im Control geladenen Dokuments bzw. ruft diesen ab.

14.4.7.4.1.6 ReadOnly

Eigenschaft: `ReadOnly` als boolean

Dispatch Id: *1007*

Beschreibung:

Mit Hilfe dieser Eigenschaft können Sie den schreibgeschützten Status des Dokuments aktivieren und deaktivieren. Wenn `ReadOnly` `true` ist, können keine Änderungen vorgenommen werden.

14.4.7.4.2 Methoden

Es sind die folgenden Methoden definiert:

Behandlung von Dokumenten:
New 662
Open 663
Reload 663
Save 663
SaveAs 664

Behandlung von Befehlen:
Exec 662
QueryStatus 663

14.4.7.4.2.1 Exec

Methode: `Exec` (`nCmdID` as long) als boolean

Dispatch Id: *8*

Beschreibung:
`Exec` ruft den Authentic Desktop Befehl mit der ID `nCmdID` auf. Wenn der Befehl ausgeführt werden kann, gibt die Methode `true` zurück. Diese Methode sollte nur dann aufgerufen werden, wenn in der Applikation gerade ein aktives Dokument verfügbar ist.

Um Befehle, geordnet nach Menüstruktur, abzurufen, verwenden Sie die Eigenschaft `MainMenu` 654 von AuthenticDesktopControl. Um Symbolleistenbefehle abzurufen, verwenden Sie die Eigenschaft `Toolbars` 654 von AuthenticDesktopControl.

14.4.7.4.2.2 New

Methode: `New` () als boolean

Dispatch Id: *1000*

Beschreibung:
Diese Methode initialisiert ein neues Mapping innerhalb des Control.

© 2016-2022 Altova GmbH

14.4.7.4.2.3 Open

Methode: `Open` (`strFilePath` als string) als boolean

Dispatch Id: *1001*

Beschreibung:
`Open` lädt die Datei `strFileName` als das neue Dokument in das Control.

14.4.7.4.2.4 QueryStatus

Methode: `QueryStatus` (`nCmdID` als long) als long

Dispatch Id: *9*

Beschreibung:
`QueryStatus` gibt den Status "enabled/disabled" und "checked/unchecked" des von `nCmdID`. definierten Befehls zurück. Der Status wird als Bitmaske zurückgegeben.

Bit	Wert	Name	Bedeutung
0	1	Supported	Setzen, wenn der Befehl unterstützt wird.
1	2	Enabled	Setzen, wenn der Befehl aktiviert ist (ausgeführt werden kann).
2	4	Checked	Setzen, wenn der Befehl angehakt ist.

Das bedeutet, dass die Befehls-ID bei Rückgabe von 0 durch `QueryStatus` nicht als gültiger Authentic Desktop Befehl erkannt wird. Wenn `QueryStatus` einen Wert `1` oder `5` zurückgibt, ist der Befehl deaktiviert. Der Client sollte die `QueryStatus` Methode des Dokument-Control aufrufen, wenn in der Applikation gerade ein aktives Dokument vorhanden ist.

14.4.7.4.2.5 Reload

Methode: `Reload` () als boolean

Dispatch Id: *1002*

Beschreibung:
`Reload` aktualisiert den Dokumentinhalt aus dem Dateisystem.

14.4.7.4.2.6 Save

Methode: `Save` () als boolean

Dispatch Id: *1003*

Beschreibung:
`Save` speichert das aktuelle Dokument unter dem Pfad `Path` 661.

14.4.7.4.2.7 SaveAs

Methode: `SaveAs` (`strFileName` als string) als boolean

Dispatch Id: *1004*

Beschreibung:
`SaveAs` setzt `Path` 661 auf *strFileName* und speichert das Dokument anschließend unter diesem Pfad.

14.4.7.4.3 Events

Das AuthenticDesktopControlDocument ActiveX Control stellt die folgenden Verbindungspunkt-Events zur Verfügung:

`OnDocumentOpened` 665
`OnDocumentClosed` 665
`OnModifiedFlagChanged` 666
`OnContextChanged` 664
`OnFileChangedAlert` 665
`OnActivate` 664
`OnSetEditorTitle` 666

14.4.7.4.3.1 OnActivate

Event: `OnActivate` ()

Dispatch Id: *1005*

Beschreibung:
Dieses Event wird ausgelöst, wenn das Dokument-Control aktiviert ist, den Fokus hat und bereit für die Benutzereingabe ist.

14.4.7.4.3.2 OnContextChanged

Event: `OnContextChanged` (`i_strContextName` als String, `i_bActive` als bool) als bool

Dispatch Id: *1004*

Beschreibung: Keine

© 2016-2022 Altova GmbH

14.4.7.4.3.3 OnDocumentClosed

Event: `OnClosed` ()

Dispatch Id: *1001*

Beschreibung:
Dieses Event wird immer dann ausgelöst, wenn das in dieses Control geladene Event geschlossen wird. Das Argument `objDocument` ist ein `Document` Objekt aus dem Authentic Desktop Automation Interface und sollte mit Vorsicht verwendet werden.

14.4.7.4.3.4 OnDocumentOpened

Event: `OnDocumentOpened` (`objDocument` als Document)

Dispatch Id: *1000*

Beschreibung:
Dieses Event wird immer dann ausgelöst, wenn ein Dokument in diesem Control geöffnet wird. Das Argument `objDocument` ist ein `Document` Objekt aus dem Authentic Desktop Automation Interface und dient dazu, nähere Informationen über das Dokument abzurufen oder weitere Operationen auszuführen.

14.4.7.4.3.5 OnDocumentSaveAs

Event: `OnContextDocumentSaveAs` (`i_strFileName` als String)

Dispatch Id: *1007*

Beschreibung:
Dieses Event wird ausgelöst, wenn dieses Dokument intern unter einem neuen Namen gespeichert wird.

14.4.7.4.3.6 OnFileChangedAlert

Event: `OnFileChangedAlert` () als bool

Dispatch Id: *1003*

Beschreibung:
Dieses Event wird ausgelöst, wenn eine in dieses Control geladene Datei auf der Festplatte von einer anderen Applikation geändert wurde. Clients sollten true zurückgeben, wenn Sie das Event behandelt haben oder false, wenn Authentic Desktop es auf die übliche Art behandeln soll, also fragen soll, ob das Dokument neu geladen werden soll.

14.4.7.4.3.7 OnModifiedFlagChanged

Event: `OnModifiedFlagChanged` (`i_bIsModified` als boolean)

Dispatch Id: *1002*

Beschreibung:
Dieses Event wird immer dann ausgelöst, wenn das Dokument zwischen dem Status "geändert" und "nicht geändert" wechselt. Der Parameter *i_bIsModifed* ist *true,* wenn sich der Inhalt des Dokuments vom ursprünglichen Inhalt unterscheidet und ist andernfalls *false*.

14.4.7.4.3.8 OnSetEditorTitle

Event: OnSetEditorTitle ()

Dispatch Id: *1006*

Beschreibung:
Dieses Event wird ausgelöst, wenn das enthaltene Dokument intern umbenannt wird.

14.4.7.5 AuthenticDesktopControlPlaceHolder

Eigenschaften für alle Arten von Platzhalterfenstern:
`PlaceholderWindowID` 667

Eigenschaften für das Projekt-Platzhalterfenster:
`Project` 667

Methoden für das Projekt-Platzhalterfenster:
`OpenProject` 668
`CloseProject` 668

Das `AuthenticDesktopControlPlaceHolder` Control dient dazu, die zusätzlichen Authentic Desktop Fenster, wie Übersicht, Bibliothek oder das Projektfenster anzuzeigen. Es wird wie jedes andere ActiveX Control verwendet und kann überall in die Client-Applikation platziert werden.

14.4.7.5.1 Eigenschaften

Es sind die folgenden Eigenschaften definiert:

`PlaceholderWindowID` 667

Zugriff auf die AuthenticDesktopAPI:
`Project` 667

 © 2016-2022 Altova GmbH

14.4.7.5.1.1 Label

Eigenschaft: `Label` als String (schreibgeschützt)

Dispatch Id: *1001*

Beschreibung:
Mit dieser Eigenschaft erhalten Sie Zugriff auf den Titel des Platzhalters. Diese Eigenschaft ist schreibgeschützt.

14.4.7.5.1.2 PlaceholderWindowID

Eigenschaft: `PlaceholderWindowID` als `AuthenticDesktopControlPlaceholderWindow` 669

Dispatch Id: *1*

Beschreibung:
Mit Hilfe dieser Eigenschaft weiß das Objekt, welches Authentic Desktop Fenster im Client-Bereich des Control angezeigt werden soll. Die `PlaceholderWindowID` kann jederzeit auf jeden gültigen Wert der `AuthenticDesktopControlPlaceholderWindow` 669 Enumeration gesetzt werden. Das Control ändert seinen Status sofort und zeigt das neue Authentic Desktop Fenster an.

14.4.7.5.1.3 Project

Eigenschaft: `Project` als Project (schreibgeschützt)

Dispatch Id: *2*

Beschreibung:
Mit der Eigenschaft `Project` erhalten Sie Zugriff auf das `Project` Objekt der Authentic Desktop Automation Server API. Diese Schnittstelle bietet zusätzliche Funktionalitäten, die mit dem in das Control geladenen Projekt verwendet werden können. Die Eigenschaft gibt nur dann eine gültige Projektschnittstelle zurück, wenn das Platzhalterfenster `PlaceholderWindowID` 667 mit einem Wert von `Authentic DesktopXProjectWindow (=3)` hat. Die Eigenschaft ist schreibgeschützt.

14.4.7.5.2 Methoden

Es sind die folgenden Methoden definiert:

`OpenProject` 668
`CloseProject` 668

14.4.7.5.2.1 OpenProject

Methode: `OpenProject` (`strFileName` als string) als boolean

Dispatch Id: *3*

Beschreibung:
`OpenProject` lädt die Datei `strFileName` als das neue Projekt in das Control. Wenn das Platzhalterfenster eine `PlaceholderWindowID` 667 hat, die nicht mit `XMLSpyXProjectWindow (=3)` übereinstimmt, schlägt die Methode fehl.

14.4.7.5.2.2 CloseProject

Methode: `CloseProject` ()

Dispatch Id: *4*

Beschreibung:
`CloseProject` schließt das in das Control geladene Projekt. Wenn das Platzhalterfenster eine `PlaceholderWindowID` 667 hat, die nicht mit `Authentic DesktopXProjectWindow (=3)` übereinstimmt, schlägt die Methode fehl.

14.4.7.5.3 Events

Das AuthenticDesktopControlPlaceholder ActiveX Control stellt die folgenden Verbindungspunkt-Events zur Verfügung:

`OnModifiedFlagChanged` 668

14.4.7.5.3.1 OnModifiedFlagChanged

Event: `OnModifiedFlagChanged` (`i_bIsModified` als boolean)

Dispatch Id: *1*

Beschreibung:
Dieses Event wird nur bei Platzhalter-Controls mit einer `PlaceholderWindowID` 667 von `Authentic DesktopXProjectWindow (=3)` ausgelöst. Das Event wird immer dann ausgelöst, wenn der Projektinhalt sich zwischen dem Status "geändert" und "nicht geändert" ändert. Der Parameter *i_bIsModifed* ist *true,* wenn sich der Projektinhalt vom ursprünglichen Inhalt unterscheidet und *false*, wenn dies nicht der Fall ist.

© 2016-2022 Altova GmbH

14.4.7.5.3.2 OnSetLabel

Event: `OnSetLabel` (i_strLabel als String)

Dispatch Id: *1000*

Beschreibung:
Dieses Event wird immer dann ausgelöst, wenn die Bezeichnung eines Platzhalter-Control-Fensters geändert wird.

14.4.7.6 Enumerationen

Es sind die folgenden Enumerationen definiert:

`ICActiveXIntegrationLevel` 669
`AuthenticDesktopControlPlaceholderWindow`

14.4.7.6.1 ICActiveXIntegrationLevel

Mögliche Werte für die Eigenschaft IntegrationLevel 653 für das AuthenticDesktopControl.

```
ICActiveXIntegrationOnApplicationLevel  = 0
ICActiveXIntegrationOnDocumentLevel     = 1
```

14.4.7.6.2 AuthenticDesktopControlPlaceholderWindow

Diese Enumeration enthält die Liste der unterstützten zusätzlichen Authentic Desktop Fenster.

```
AuthenticDesktopControlNoToolWnd                    =  -1
AuthenticDesktopControlEntryHelperTopToolWnd        =   0
AuthenticDesktopControlEntryHelperMiddleToolWnd     =   1
AuthenticDesktopControlEntryHelperBottomToolWnd     =   2
AuthenticDesktopControlValidatorOutputToolWnd       =   3
AuthenticDesktopControlProjectWindowToolWnd         =   4
AuthenticDesktopControlInfoToolWnd                  =  18
```

15 Anhänge

Diese Anhänge enthalten technische Informationen über Authentic Desktop und wichtige Lizenzinformationen. Die einzelnen Anhänge sind in die folgenden Unterabschnitte gegliedert:

Technische Daten 671

- Betriebssystem und erforderlicher Arbeitsspeicher
- Altova XML Parser
- Altova XSLT- und XQuery-Prozessor
- Unicode-Unterstützung
- Internet-Verwendung

Lizenzinfomationen 674

- Electronic Software Distribution (EDI)
- Copyright
- Endbenutzer-Lizenzvereinbarung (EULA)

 © 2016-2022 Altova GmbH

15.1 Technische Daten

Dieses Kapitel enthält Informationen zu einigen technischen Aspekten Ihrer Software. Es ist in die folgenden Abschnitte gegliedert:

- OS- und Arbeitsspeicheranforderungen 671
- Altova-Prozessoren 671
- Unicode-Unterstützung 672
- Internet-Verwendung 672

15.1.1 OS- und Arbeitsspeicheranforderungen

Betriebssystem

Die Altova-Software-Applikationen stehen für die folgenden Plattformen zur Verfügung:

- Windows 7 SP1 mit Plattformupdate, Windows 8, Windows 10, Windows 11
- Windows Server 2008 R2 SP1 mit Plattformupdate oder höher

Arbeitsspeicher

Da die Software in C++ geschrieben wurde, wird dafür nicht so viel Platz wie in einer Java Runtime Umgebung benötigt und normalerweise wird dafür weniger Arbeitsspeicher als bei einer vergleichbaren Java-basierten Applikation benötigt. Es ist notwendig, dass die einzelnen Dokumente in den Hauptarbeitsspeicher geladen werden, damit jedes Dokument zur Gänze geparst und analysiert werden kann und die Anzeige- und Bearbeitungsgeschwindigkeit während der normalen Arbeit verbessert wird. Daher steigt mit der Größe des Dokuments auch der Arbeitsspeicherbedarf.

Auch die unbegrenzte Rückgängig-Funktion kann einiges an Arbeitsspeicher in Anspruch nehmen. Wenn Sie große Abschnitte in großen Dokumenten immer wieder ausschneiden und einfügen, kann dies enorm viel Speicherplatz verbrauchen.

15.1.2 Altova-Prozessoren

XML Validator

Wenn Sie ein XML-Dokument öffnen, verwendet die Applikation den integrierten XML Validator, um das Dokument auf Wohlgeformtheit zu prüfen, es anhand eines Schemas zu validieren (falls eines angegeben wurde) und Baumstrukturen und Infosets zu erstellen. Der Altova XML Validator dient auch dazu, beim Editieren von Dokumenten intelligente Eingabehilfen zur Verfügung zu stellen und etwaige Validierungsfehler dynamisch anzuzeigen.

Im integrierten XML Validator ist die Final Recommendation der W3C XML Schema Spezifikationen 1.0 und 1.1 implementiert. Neue Entwicklungen, die von der XML Schema-Arbeitsgruppe empfohlen werden, werden ständig in den XML Validator integriert, sodass Ihnen mit Altova-Produkten immer eine Entwicklungsumgebung auf dem neuesten Stand der Technik zur Verfügung steht.

XSLT- und XQuery-Prozessor

Die Altova Produkte arbeiten mit dem Altova XSLT 1.0, 2.0 und 3.0-Prozessor und dem Altova XQuery 1.0 und 3.1-Prozessor. Falls einer dieser Prozessoren im Produkt enthalten ist, finden Sie die Dokumentation zurm implementierungsspezifischen Verhalten der einzelnen Prozessoren in den Anhängen zu dieser Dokumentation.

Anmerkung: Altova MapForce verwendet den XSLT 1.0-, 2.0- und XQuery 1.0-Prozessor zur Codegenerierung.

15.1.3 Unicode-Unterstützung

Die XML-Produkte von Altova bieten vollständige Unicode-Unterstützung. Um ein XML-Dokument benötigen Sie eine Schriftart, die die von diesem Dokument verwendeten Unicode-Zeichen unterstützt.

Beachten Sie bitte, dass die meisten Schriftarten nur eine bestimmte Untergruppe des gesamten Unicode-Bereichs enthalten und normalerweise für das entsprechende Schriftsystem ausgelegt sind. Wenn Zeichen falsch dargestellt werden, könnte der Grund darin liegen, dass die gewählte Schriftart die erforderlichen Glyphen nicht enthält. Es ist daher nützlich, eine Schriftart zu verwenden, die den gesamten Unicode-Bereich abdeckt - v.a. wenn Sie XML-Dokumente in unterschiedlichen Sprachen oder Schriftsystemen editieren. Ein typische auf Windows PCs installierte Unicode-Schriftart ist Arial Unicode MS.

Im Ordner`/Examples` Ihres Applikationsordners finden Sie eine XHTML-Datei mit dem Namen `UnicodeUTF-8.html`, die den folgenden Satz in verschiedenen Sprachen und Schriftsystemen enthält:

- *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。)

Wenn Sie diese XHTML-Datei öffnen, erhalten Sie einen kurzen Eindruck davon, was mit Unicode möglich ist und welche Schriftsysteme von den auf Ihrem PC verfügbaren Schriftarten unterstützt werden.

15.1.4 Internet-Verwendung

Altova-Applikationen können für Sie auch eine Verbindung mit dem Internet herstellen. Dies geschieht in den folgenden Fällen:

- Wenn Sie im Registrierungsdialogfeld (**Hilfe | Software-Aktivierung**) auf "Kostenlosen Evaluierungs-Key anfordern" klicken, werden die drei Felder im Registrierungsdialogfeld über eine normale HTTP-Verbindung (Port 80) an unseren Webserver übertragen. Sie erhalten dann per E-Mail (normales SMTP) den kostenlosen Evaluierungs-Keycode zugesandt.
- In einige Altova-Produkten können Sie eine Datei über das Internet öffnen (**Datei | Öffnen | Zu URL wechseln**). In diesem Fall wird das Dokument mittels einer der folgenden Protokollmethoden und Verbindungen aufgerufen: HTTP (normalerweise Port 80), FTP (nomalerweise Port 20/21), HTTPS (normalerweise Port 443). Sie können auch einen HTTP-Server auf Port 8080 verwendden. (Definieren Sie den Port im Dialogfeld "URL", indem Sie ihn durch ein Komma getrennt nach dem Servernamen angeben.*)*

 © 2016-2022 Altova GmbH

- Wenn Sie ein XML-Dokument öffnen, das sich auf ein XML-Schema oder eine DTD bezieht und das Dokument durch eine URL definiert wird, wird das referenzierte Schema-Dokument ebenfalls über eine HTTP-Verbindung (Port 80) oder ein anderes in der URL definiertes Protokoll (siehe Punkt 2 oben) aufgerufen. Ebenso wird ein Schema-Dokument aufgerufen, wenn eine XML-Datei validiert wird. Beachten Sie, dass die Validierung automatisch beim Öffnen des Dokuments erfolgen kann, wenn Sie dies in der Applikation so konfiguriert haben (Register "Datei" des Dialogfelds "Optionen" **Extras | Optionen**).
- Webservice-Verbindungen werden in Altova Applikationen, die WSDL und SOAP verwenden, mittels WSDL-Dokumenten definiert.
- Wenn Sie den Befehl "**Als Mail senden**..." (**Datei | Als Mail senden**) in <%SPY-GEN%> verwenden, wird die aktuelle Auswahl bzw. Datei über ein MAPI-kompatibles Mail-Programm, das auf dem PC des Benutzers installiert ist, versendet
- Im Rahmen der Altova-Software-Lizenzvereinbarung beschriebenen Software-Aktivierung und beim Live-Update.

15.2 Lizenzinformationen

Dieser Anhang enthält die folgenden Informationen:

- Informationen über den Vertrieb dieses Software-Produkts
- die Lizenzvereinbarung zu diesem Software-Produkt

Lesen Sie die Informationen bitte sorgfältig - sie sind rechtlich bindend, da Sie sich bei der Installation dieses Software-Produkts damit einverstanden erklärt haben.

Den Inhalt aller Altova-Lizenzenvereinbarungen finden Sie auf der Altova Website unter Rechtliches.

15.2.1 Electronic Software Distribution

Dieses Produkt ist über EDS (Electronic Software Distribution), also auf elektronischem Weg erhältlich, eine Methode, die die folgenden einzigartigen Vorteile bietet:

- Sie können die Software kostenlos 30 Tage lang testen, bevor Sie sich zu einem Kauf entscheiden. (*Anmerkung: Die Lizenz für MobileTogether Designer ist kostenlos*.)
- Wenn Sie sich entschieden haben, die Software zu kaufen, können Sie Ihre Bestellung online auf der Altova Website tätigen. Sie erhalten dann innerhalb weniger Minuten ein vollständig lizenziertes Produkt.
- Sie erhalten immer die neueste Version unserer Software
- Die Software enthält ein umfassendes Hilfesystem, das Sie von der Benutzeroberfläche der Appikation aus aufrufen können. Die neueste Version des Benutzerhandbuchs steht auf unserer Website www.altova.com (i) im HTML-Format zum Aufrufen online und (ii) im PDF-Format zum Download und Ausdrucken zur Verfügung.

30-Tage-Evaluierungszeitraum

Nachdem Sie dieses Software-Produkt heruntergeladen haben, können Sie es 30 Tage lang kostenlos testen. Während dieses Zeitraums werden Sie nach etwa 20 Tagen in regelmäßigen Abständen daran erinnert, dass die Software noch nicht lizenziert wurde. Diese Erinnerungsmeldung wird allerdings nur einmal, nämlich bei jedem Start des Programms, angezeigt. Wenn Sie das Programm nach Ablauf des 30-tägigen Evaluierungszeitraums weiterhin verwenden möchten, müssen Sie eine Produktlizenz erwerben, die Sie in Form einer Lizenzdatei mit einem Keycode erhalten. Laden Sie die Lizenzdatei über das Dialogfeld "Software-Aktivierung" Ihres Produkts hoch, um das Produkt freizuschalten.

Sie können Ihre Produktlizenz über https://shop.altova.com/ erwerben.

 © 2016-2022 Altova GmbH

Weitergabe der Software an andere Mitarbeiter in Ihrem Unternehmen zu Testzwecken

Wenn Sie die Evaluierungsversion der Software auch anderen Personen in Ihrem Unternehmen über das Netzwerk zur Verfügung stellen möchten oder wenn Sie sie auf einem PC installieren möchten, der nicht mit dem Internet verbunden ist, dürfen Sie nur das Installationsprogramm weitergeben, vorausgesetzt es wurde nicht modifiziert. Jeder, der das von Ihnen zur Verfügung gestellte Installationsprogramm aufruft, muss einen eigenen Evaluierungs-Keycode für 30 Tage anfordern. Nach Ablauf des Testzeitraums, muss eine Lizenz erworben werden, damit das Produkt weiter verwendet werden kann.

15.2.2 Software-Aktivierung und Lizenzüberwachung

Im Rahmen der Aktivierung der Software durch Altova, verwendet die Software unter Umständen Ihr internes Netzwerk und Ihre Internetverbindung, um die Lizenzdaten während der Installation, Registrierung, der Verwendung oder der Aktualisierung an einen von Altova betriebenen Lizenzserver zu übertragen und die Authentizität der Lizenzdaten zu überprüfen, damit Altova-Software nicht ohne Lizenz oder auf unzulässige Art und Weise verwendet werden kann und um den Kundenservice gleichzeitig zu verbessern. Bei der Aktivierung werden zwischen Ihrem Computer und dem Altova-Lizenzserver für die Lizenzierung erforderliche Daten wie Informationen über Betriebssystem, IP-Adresse, Datum/Uhrzeit, Software-Version und Computername sowie andere Informationen ausgetauscht.

Ihr Altova-Produkt verfügt über ein integriertes Lizenzüberwachungsmodul, das ebenfalls dazu beiträgt, unbeabsichtigte Verletzungen der Lizenzvereinbarung zu vermeiden. Ihr Produkt kann entweder mit einer Einzelplatzlizenz oder einer Mehrfachlizenz erworben werden. Je nach Lizenz stellt das Lizenzüberwachungsmodul sicher, dass nicht mehr als die lizenzierte Anzahl an Benutzern die Applikation gleichzeitig verwendet.

Bei dieser Lizenzüberwachungsmethode wird Ihr LAN-Netzwerk verwendet, um die Kommunikation zwischen Instanzen der Applikation, die auf verschiedenen Computern laufen, zu überwachen.

Einzelplatzlizenz

Beim Start der Applikation wird im Rahmen der Lizenzüberprüfung ein kurzes Broadcast-Datagramm abgesendet, um andere Instanzen des Produkts, die auf anderen Computern im selben Netzwerk laufen, zu finden. Wenn keine Antwort einlangt, wird ein Port geöffnet, der Informationen von anderen Instanzen der Applikation empfangen kann.

Mehrplatzlizenz

Wenn Sie im selben LAN mehrere Instanzen der Applikation verwenden, kommunizieren diese beim Start kurz miteinander, um Keycode-Informationen auszutauschen, damit Sie sicher sein können, dass nicht mehr als die lizenzierte Anzahl an Lizenzen gleichzeitig in Verwendung ist. Dieselbe Lizenzüberwachungstechnologie wird auch bei Unix und vielen anderen Datenbankentwicklungstools verwendet. Sie gestattet Benutzern den Erwerb von Parallellizenzen für mehrere Benutzer zu vernünftigen Preisen.

Wir sind außerdem bestrebt, nur wenige, kleine Netzwerkpakete zu versenden, um Ihr Netzwerk nicht zu überlasten. Die von Ihrem Altova Produkt verwendeten TCP/IP Ports (2799) sind offiziell bei IANA registriert, (*nähere Informationen siehe IANA Service Name Registry*) und unser Lizenzüberwachungsmodul basiert auf einer bewährten und erprobten Technologie.

Wenn Sie eine Firewall verwenden, werden Sie unter Umständen feststellen, dass die Computer, auf denen Altova-Produkte laufen, über Port 2799 miteinander kommunizieren. Sie können diesen Netzwerkverkehr

zwischen verschiedenen Gruppen in Ihrem Unternehmen natürlich blockieren, solange Sie mit anderen Mitteln sicherstellen können, dass Ihre Lizenzvereinbarung eingehalten wird.

Anmerkung zu Zertifikaten

Ihre Altova Applikation kontaktiert den Altova Lizenzierungsserver über HTTPS (`link.altova.com`). Für diese Kommunikation verwendet Altova ein registriertes SSL-Zertifikat. Wenn dieses Zertifikat ersetzt wird (z.B. von Ihrer IT-Abteilung oder einer externen Agentur), werden Sie von Ihrer Altova Applikation gewarnt, dass die Verbindung nicht sicher ist. Sie könnten Ihre Altova Applikation mit dem Ersetzungszertifikat starten. Dies würde jedoch auf Ihr eigenes Risiko geschehen. Wenn Sie eine Warnung sehen, dass die *Verbindung nicht sicher* ist, überprüfen Sie den Ursprung des Zertifikats und wenden Sie sich an Ihr IT-Team (die in der Lage sein sollten, zu entscheiden, ob das Abfangen und die Ersetzung des Altova-Zertifkats fortgesetzt werden soll).

Wenn Ihr Unternehmen sein eigenes Zertifikat verwenden muss (z.B. um die Kommunikation zu und von Client-Rechnern zu überwachen), empfehlen wir Ihnen, Altova LicenseServer, die kostenlose Lizenzverwaltungssoftware von Altova in Ihrem Netzwerk zu installieren. Client-Rechner verwenden mit dieser Konfiguration weiterhin die Zertifikate Ihres Unternehmens, während der Altova LicenseServer für die Kommunikation mit Altova das Altova-Zertifikat das Altova-Zertifikat verwenden kann.

15.2.3 Altova Endbenutzer-Lizenzvereinbarung zu Authentic

- Die Altova-Endbenutzer-Lizenzvereinbarung für Authentic kann unter https://www.altova.com/de/legal/authentic-eula eingesehen werden.
- Die Altova-Datenschutzbestimmungen finden Sie unter https://www.altova.com/de/privacy.

© 2016-2022 Altova GmbH

Index

.

.NET,
Integration von Authentic Desktop, 157
Unterschiede zu Authentic Desktop Standalone, 159

A

Abfragen,
für DB-Anzeige in der Authentic View-Ansicht, 75
ActiveX,
Integration auf Applikationsebene, 622
Integration auf Dokumentebene, 624
Voraussetzungen für die Integration, 619
ActiveX Controls,
Unterstützung, 326
zur Visual Studio Toolbox hinzufügen, 621
Aktive Konfiguration,
für globale Ressourcen, 249
Aktivierung der Software, 288
Alias,
siehe Globale Ressourcen, 96
Alles markieren,
Befehl, 185
Altova Globale Ressourcen,
siehe Globale Ressourcen, 96
Altova Support, 23
Altova XML Parser,
Info, 671
Altova-Produkte, 23
Änderungen verfolgen, 268
Angedockte Fenster, 14
Anhängen,
Zeile (in Authentic View-Ansicht), 236
Anpassen, 258
Kontextmenü, 258
Kontextmenü 'Anpassen', 263
Makros, 259
Menü, 258
Symbolleiste/Menü Befehle, 250
Anpassung, 22
Ansicht,
Browser-Ansicht, 239
Menü, 239
Ansicht wechseln,
zur Authentic View-Ansicht, 44
API,
Dokumentation, 341
Application,
ActiveDocument, 376
AddMacroMenuItem, 376
AddXSLT_XQParameter, 377
Application, 377
ClearMacroMenu, 377
CurrentProject, 378
Dialogs, 378
Documents, 379
GetDatabaseImportElementList, 379
GetDatabaseSettings, 380
GetDatabaseTables, 380
GetExportSettings, 381
GetTextImportElementList, 381
GetTextImportExportSettings, 382
GetXSLT_XQParameterCount, 383
GetXSLT_XQParameterName, 383
GetXSLT_XQParameterXPath, 383
ImportFromDatabase, 383
ImportFromSchema, 384
ImportFromText, 385
ImportFromWord, 386
NewProject, 387
OnBeforeOpenDocument, 374
OnBeforeOpenProject, 375
OnDocumentOpened, 375
OnProjectOpened, 376
OpenProject, 387
Parent, 388
Quit, 388
ReloadSettings, 388
RemoveXSLT_XQParameter, 389
RunMacro, 389
ScriptingEnvironment, 389
ShowApplication, 390
ShowForm, 390
URLDelete, 391
URLMakeDirectory, 391
WarningNumber, 392
WarningText, 392

Arbeitsspeicher-Anforderungen, 671
ATL,
Plug-In Beispieldateien, 330
Attribut-Eingabehilfe,
in der Authentic View-Ansicht, 50
Attribut-Vorschau, 275
Attributwerte,
in der Authentic View-Ansicht eingeben, 37
Ausgabefenster,
Anzeige, 285
Ausgabeformatierung, 268
Ausschneiden (Befehl), 184
Authentic,
Menü, 228
Authentic (Menü),
Bearbeitung dynamischer Tabellen, 44
Markup-Anzeige, 44
Authentic Desktop,
Features, 23
Hilfe, 23
Integration, 619
user manual, 10
Authentic DesktopCommand,
in AuthenticDesktopControl, 648
Authentic DesktopCommands,
in AuthenticDesktopControl, 650
Authentic Desktop-Perspektive in Eclipse, 163
Authentic Integrationspaket, 158
Authentic Plugin für Eclipse,
installieren, 161
Authentic Plugin für VS .NET,
installieren, 158
Authentic Skripterstellung,
Sicherheitseinstellungen, 237
vertrauenswürdige Pfade, 237
Authentic View,
CDATA-Abschnitte, 34
Daten eingeben, 34
Daten in einer XML-Datenbank bearbeiten, 230
Dateneingabe-Steuerelemente, 34
Eingabehilfen, 28
Elemente anwenden, 31
Elemente löschen, 31
Entities, 34
Kontextmenü, 28
Markup-Tags anzeigen, 28
Nodes einfügen, 31
Nodes entfernen, 31
Nodes hinzufügen, 31
Sonderzeichen, 34
Tabellen, 31
Authentic View-Ansicht, 67
als XML/Text einfügen, 55
Attributwerte eingeben, 37
Ausgabedokumente anhand der PXF-Datei generieren, 237
Benutzeroberfläche, 41
DB-Daten bearbeiten, 229
Dokumentanzeige, 47
Eingabehilfen, 50
Entities einfügen, 38
Hauptfenster, 47
Kontextmenüs, 55
Markup-Anzeige, 47
Markup-Anzeige in, 44
neue XML-Datei öffnen, 229
SPS-Tabellen, 66
Symbolleisten-Schaltflächen, 44
Tabellen (SPS und XML), 66
Text formatieren in, 44
Übersicht über die GUI, 42
Verwendung von XML-Tabellen, 67
Verwendung wichtiger Features, 58
wechseln zur, 239
XML-Dokument drucken, 40
XML-Dokument öffnen in, 26
XML-Tabellen, 67
XML-Tabellensymbole, 71
Authentic View-Ansicht-Vorlage, 26
AuthenticDataTransfer,
dropEffect, 395
getData, 395
ownDrag, 395
type, 395
AuthenticDesktopControl, 651
Beispiele für die Integration auf Dokumentebene, 627
Dokumentation, 619
Integration mittels C#, 628
Objektreferenz, 648
AuthenticDesktopControlDocument, 659
AuthenticDesktopControlPlaceHolder, 666
Authentic-Integrationspaket, 161
AuthenticRange,
AppendRow, 401
Application, 402
CanPerformAction, 402
CanPerformActionWith, 402

AuthenticRange,
Close, 403
CollapsToBegin, 403
CollapsToEnd, 403
Copy, 404
Cut, 404
Delete, 404
DeleteRow, 404
DuplicateRow, 405
ExpandTo, 406
FirstTextPosition, 406
FirstXMLData, 407
FirstXMLDataOffset, 408
GetElementAttributeNames, 409
GetElementAttributeValue, 410
GetElementHierarchy, 410
GetEntityNames, 411
Goto, 411
GotoNext, 412
GotoNextCursorPosition, 413
GotoPrevious, 413
GotoPreviousCursorPosition, 414
HasElementAttribute, 414
InsertEntity, 414
InsertRow, 415
IsCopyEnabled, 416
IsCutEnabled, 416
IsDeleteEnabled, 416
IsEmpty, 416
IsEqual, 417
IsFirstRow, 417
IsInDynamicTable, 417
IsLastRow, 417
IsPasteEnabled, 418
IsTextStateApplied, 418
LastTextPosition, 418
LastXMLData, 419
LastXMLDataOffset, 420
MoveBegin, 421
MoveEnd, 422
MoveRowDown, 422
MoveRowUp, 422
Parent, 423
Paste, 423
PerformAction, 423
Select, 424
SelectNext, 425
SelectPrevious, 425
SetElementAttributeValue, 426
SetFromRange, 427
Text, 428
AuthenticView, 445
Application, 437
AsXMLString, 437
DocumentBegin, 439
DocumentEnd, 439
Event, 440
Goto, 441
IsRedoEnabled, 442
IsUndoEnabled, 442
MarkupVisibility, 442
OnBeforeCopy, 429
OnBeforeCut, 430
OnBeforeDelete, 430
OnBeforeDrop, 431
OnBeforePaste, 431
OnDragOver, 432
OnKeyboardEvent, 433
OnMouseEvent, 434
OnSelectionChanged, 434
Parent, 443
Print, 443
Redo, 443
Selection, 443
Undo, 444
WholeDocument, 445
XMLDataRoot, 446
Automatische Validierung, 271
Automatische Validierung ausschalten, 271
Automatisches Backup (Einstellungen), 268

B

Bearbeiten,
Makro Schaltfläche, 263
Menü, 184
Bearbeitet mit XMLSpy, 268
Beendungsmodus, 268
Befehl, 258
hinzufügen zu Symbolleiste/Menü, 250
Kontextmenü, 258
löschen aus Menü, 258
Menü zurücksetzen, 258
Befehle,

Befehle,
Auflistung in Tastaturzuordnung, 288
Befehlszeilenaktionen, 294
Bei Bearbeitung validieren (Befehl), 220
Benutzerdefiniertes Wörterbuch, 241
Benutzerhandbuch Authentic Desktop, 12
Benutzeroberfläche,
Beschreibung, 14
mittels Plug-in konfigurieren, 326
Beschreibung der Benutzeroberfläche, 14
Bestellen von Altova-Software, 288
Betriebssysteme für Altova-Produkte, 671
Big-endian, 272
Browser, 275
Ansicht, 239
Browser-Ansicht,
Inhalt aktualisieren, 240
Laden einer Seite abbrechen, 240
Schriftgröße, 240
separate Fenster, 240
zurück und vorwärts bewegen, 240

C

C#,
Integration von Authentic Desktop, 628
CDATA-Abschnitte,
in der Authentic View-Ansicht einfügen, 61
CodeGeneratorDlg,
Application, 447
CPPSettings_DOMType, 447
CPPSettings_LibraryType, 448
CPPSettings_UseMFC, 449
CSharpSettings_ProjectType, 449
OutputPath, 449
OutputPathDialogAction, 450
OutputResultDialogAction, 450
Parent, 450
ProgrammingLanguage, 451
PropertySheetDialogAction, 451
TemplateFileName, 451
COM API,
im Skript-Editor, 308
COM-API,
Dokumentation, 341
Copyright-Informationen, 674
CR&LF, 268

D

Darstellung, 275
DatabaseConnection,
ADOConnection, 452
AsAttributes, 453
CreateMissingTables, 453
CreateNew, 454
DatabaseKind, 454
ExcludeKeys, 454
File, 455
IncludeEmptyElements, 455
NumberDateTimeFormat, 456
ODBCConnection, 456
SQLSelect, 457
TextFieldLen, 457
Datei, 268
als E-Mail senden, 181
Druckoptionen, 182
Kodierung, 174
neue erstellen, 168
Öffnen, 169
Optionen beim Öffnen, 268
Register, 268
schließen, 175
speichern, 175
Standard-Encoding, 272
Dateiarten, 271
Dateien,
Zu Versionskontrolle hinzufügen, 198
zuletzt verwendete, 183
Dateien per E-Mail senden, 181
Dateien speichern,
Kodierung, 174
Datenbanken,
in Authentic View-Ansicht bearbeiten, 229
siehe auch DB, 74
Datumswahl,
in der Authentic View-Ansicht verwenden, 81
Datumswerte,
manuell ändern, 82
DB, 74, 75
Abfragen erstellen, 75
Abfragen in der Authentic View-Ansicht, 74

© 2016-2022 Altova GmbH

DB, 74, 75
Anzeige in der Authentic View-Ansicht filtern, 75
Bearbeiten in der Authentic View-Ansicht, 74
in der Authentic View-Ansicht bearbeiten, 79
Navigieren in Tabellen in der Authentic View-Ansicht, 74
Parameters in DB-Abfragen, 75
Delete,
Application.URLDelete, 391
Designs, 285
Dialogs,
Application, 458
CodeGeneratorDlg, 459
DTDSchemaGeneratorDlg, 460
FileSelectionDlg, 459
GenerateSampleXMLDlg, 460
Parent, 459
SchemaDocumentationDlg, 460
Document, 474
Application, 466
AssignDTD, 467
AssignSchema, 467
AssignXSL, 467
AssignXSLFO, 467
AuthenticView, 468
Close, 469
ConvertDTDOrSchema, 469
CreateChild, 471
CreateSchemaDiagram, 471
CurrentViewMode, 472
DataRoot, 472
DocEditView, 472
Encoding, 473
EndChanges, 473
ExecuteXQuery, 474
ExportToDatabase, 474
ExportToText, 475
FullName, 477
GenerateDTDOrSchema, 477, 478
GenerateProgramCode, 478
GenerateSampleXML, 478
GenerateSchemaDocumentation, 479
GetExportElementList, 481
GetPathName, 481
GridView, 482
IsModified, 482
IsValid, 482
IsWellFormed, 485
Name, 485
OnBeforeCloseDocument, 464
OnBeforeSaveDocument, 464
OnBeforeValidate, 465
OnCloseDocument, 465
OnViewActivation, 466
Path, 486
RootElement, 486
Save, 486
SaveAs, 487
Saved, 487
SaveInString, 487
SaveToURL, 488
SetActiveDocument, 488
SetEncoding, 488
SetExternalIsValid, 489
SetPathName, 489
StartChanges, 490
SwitchViewMode, 490
Title, 491
TransformXSL, 491
TransformXSLFO, 491
UpdateViews, 492
UpdateXMLData, 492
XQuery, 474
Documents,
Count, 494
Item, 494
NewAuthenticFile, 494
NewFile, 494
NewFileFromText, 495
OpenAuthenticFile, 495
OpenFile, 495
OpenURL, 496
OpenURLDialog, 496
Dokument,
Rechtschreibprüfung, 241
Dokumentation, 287
Dokumente im Hauptfenster, 15
Dokumentebene,
Beispiele für die Integration von <%SPY-GEN%>, 627
Drucken,
von der Authentic View-Ansicht, 40
Druckoptionen, 182
Druckvorschau und Druckereinrichtung, 182
DTDs, 268, 271
DTDSchemaGeneratorDlg,
Application, 497
AttributeTypeDefinition, 498

DTDSchemaGeneratorDlg,
DTDSchemaFormat, 498
FrequentElements, 498
GlobalAttributes, 499
MaxEnumLength, 499
MergeAllEqualNamed, 499
OnlyStringEnums, 499
OutputPath, 500
OutputPathDialogAction, 500
Parent, 500
ResolveEntities, 500
TypeDetection, 501
ValueList, 501
Duplizieren,
Zeile (in Authentic View-Ansicht), 236
Dynamische (SPS) Tabellen in der Authentic View-Ansicht,
Verwendung, 66
Dynamische Tabellen,
bearbeiten, 44

E

Eclipse-Plattform,
Authentic Desktop-Einstiegspunkte, 166
Authentic Desktop-Perspektive in, 163
und Authentic Desktop, 160
und Authentic-Integrationspaket, 161
Einfügen,
als Text, 61
als XML, 61
Zeile (in Authentic View-Ansicht), 236
Einfügen (Befehl), 184
Einfügen als,
Text, 55
XML, 55
Eingabehilfen, 19
Anzeige, 285
Eingabetaste,
drücken, 90
Einstellungen, 22, 267
Skripting, 280
Element-Eingabehilfe,
in der Authentic View-Ansicht, 50
ElementList,
Count, 502
Item, 502
RemoveElement, 502
ElementListItem,
ElementKind, 502
FieldCount, 503
Name, 503
RecordCount, 503
E-Mail,
Dateien senden mit, 181
Encoding,
Standard, 272
Endbenutzer-Lizenzvereinbarung, 674
Entities,
in der Authentic View-Ansicht definieren, 61, 84
in der Authentic View-Ansicht einfügen, 61
in der Authentic View-Ansicht Entities einfügen, 38
Entities-Eingabehilfe,
in der Authentic View-Ansicht, 50
Enumerationen,
in AuthenticDesktopControl, 669
Enumerations,
SPYAttributeTypeDefinition, 605
SPYAuthenticActions, 605
SPYAuthenticDocumentPosition, 605
SpyAuthenticElementActions, 605
SPYAuthenticElementKind, 606
SPYAuthenticMarkupVisibility, 606
SPYDatabaseKind, 607
SPYDialogAction, 607
SPYDOMType, 607
SPYDTDSchemaFormat, 608
SPYEncodingByteOrder, 608
SPYExportNamespace, 608
SPYFrequentElements, 608
SPYKeyEvent, 609
SPYLibType, 610
SPYLoading, 610
SPYMouseEvent, 610
SPYNumberDateTimeFormat, 611
SPYProgrammingLanguage, 611
SPYProjectItemTypes, 611
SPYProjectType, 612
SPYSampleXMLGenerationOptimization, 612
SPYSampleXMLGenerationSchemaOrDTDAssignment, 613
SPYSchemaDefKind, 613
SPYSchemaDocumentationFormat, 614
SPYTextDelimiters, 615

© 2016-2022 Altova GmbH

Enumerations,
SPYTextEnclosing, 615
SPYTypeDetection, 615
SPYURLTapes, 615
SPYViewModes, 616
SPYVirtualKeyMask, 617
SPYXMLDataKind, 618
Ersetzen,
Text in Dokument, 186
Evaluierungs-Keycode für Altova-Software, 288
Evaluierungszeitraum,
für Altova-Software-Produkte, 674
von Altova Software-Produkten, 674
Event, 374, 375, 376, 429, 430, 431, 432, 433, 434, 464, 465, 466, 525, 526, 527
Explorer, 271
ExportSettings,
CreateKeys, 504
ElementList, 504
EntitiesToText, 504
ExportAllElements, 504
FromAttributes, 505
FromSingleSubElements, 505
FromTextValues, 505
IndependentPrimaryKey, 505
Namespace, 505
SubLevelLimit, 506
Extern geparste Entitäten, 271
Externe Applikationen,
Dateien öffnen in, 254
Externer XSL Prozessor, 276
Extras,
siehe auch externe Applikationen, 254

F

Fenster,
anordnen, 285
Anzeige ein/auschalten, 285
Anzeige verwalten, 14
automatisch ausblenden, 14
frei schwebend, andocken, als Register, 14
Kacheln anzeigen, 285
kaskadierende, 285
Fenster "Meldungen",
Anzeige, 285
Fenster automatisch ausblenden, 14
FileSelectionDlg,
Application, 506
DialogAction, 507
FullName, 507
Parent, 507
Finden,
siehe Suchen, 186
Frei schwebende Fenster, 14
Freigeben,
aus Versionskontrolle, 199
Funktionstasten, 255

G

Gemischte Markup-Symbole (in der Authentic View-Ansicht), 235
Generate Sample XML, 605, 612, 613
GenerateSampleXMLDlg,
Application, 520
FillWithSampleData, 522
NonMandatoryAttributes, 522
NonMandatoryElements, 523
Parent, 523
RepeatCount, 524
TakeFirstChoice, 524
Global, 672
Globale,
Einstellungen, 267
Globale Ressourcen, 96
definieren, 97, 248
Konfigurationen ändern, 112
Symbolleiste aktivieren, 252
Typ Datei definieren, 99
Typ Datenbank definieren, 106
Typ Ordner definieren, 104
verwenden, 109, 112
vom Typ Datei und Ordner verwenden, 109
Grafikformate,
in der Authentic View-Ansicht, 89
Grammatik, 271
GridView,
CurrentFocus, 528
Deselect, 528
IsVisible, 528
OnBeforeDrag, 525

GridView,
OnBeforeDrop, 526
OnBeforeStartEditing, 526
OnEditingFinished, 527
OnFocusChanged, 527
Select, 528
SetFocus, 528
Große Markup-Symbole anzeigen, 44, 47
Große Markup-Symbole in (in Authentic View-Ansicht), 235

H

Häkchen,
Rechtschreibprüfung, 241
Hauptfenster, 15
Hilfe,
Menü, 287
Hilfe und Dokumentation, 287
Hintergrundinformationen, 671
Hinzufügen,
Dateien zu Projekt, 206
Ordner zu Projekt, 208
HTML-Ausgabe,
in der Authentic View-Ansicht anhand der PXF-Datei generieren, 237

I

Info-Fenster, 19
Anzeige, 285
Inhaltsverzeichnis, 287
Integrating,
Authentic Desktop in Applikationen, 619
Internet, 292
Internet-Verwendung,
in Altova-Produkten, 672
ISO/IEC 10646, 672

J

Java, 631
Java settings, 279
Java virtual machine,
path setting, 279
JScript,
Skripterstellung mit Authentic Desktop, 297

K

Keine Markup-Symbole (in der Authentic View-Ansicht), 235
Keycodes für Altova-Software, 288
Kleine Markup-Symbole anzeigen, 44, 47
Kodierung,
von Dateien, 174
Konfigurationen,
einer globalen Ressource, 97, 249
Konfigurationen in globalen Ressourcen, 112
Konfigurieren,
XMLSPY-Benutzeroberfläche, 326
Kontextmenü,
anpassen, 263
Befehle, 258
Kontextmenüs,
in der Authentic View-Ansicht, 55
Kopieren (Befehl), 184
Kürzel, 255
anzeigen in Tooltip, 263
zuweisen/löschen, 255

L

Leere Elemente, 271
Links,
in der Authentic View-Ansicht verwenden, 61
Little-endian, 272
Lizenz,
Informationen, 674
Lizenzen für Altova-Software, 288
Lizenzüberwachung,
in Altova-Produkten, 675
loading, 496
Löschen, 250
Befehl aus Kontextmenü, 258
Befehl aus Symbolleiste, 250
Kürzel zu Befehl, 255
Symbol aus Symbolleiste, 250

© 2016-2022 Altova GmbH

Löschen, 250
Symbolleiste, 252
Zeile (in Authentic View-Ansicht), 236
Löschen (Befehl), 184

M

Makro,
Hinzufügen zu Menü/Symbolleiste, 259
Schaltfläche bearbeiten, 263
Makros,
aktivieren, 310, 322
Applikationsmakros ausführen, 248
ausführen, 323
entwickeln, 297, 303
Markup,
in Authentic View-Ansicht, 44
Markup ausblenden, 44
Markup-Code (in der Authentic View-Ansicht) ausblenden, 235
Markup-Symbole,
in der Authentic View-Ansicht, 47
Markup-Symbole ausblenden, 47
Markup-Symbole in (in der Authentic View-Ansicht),
ausblenden, 235
kleine/große/gemischte einblenden, 235
Maximale Zellenbreite, 275
Meldungen (Fenster), 19
Menü, 258
anpassen, 258
Ansicht, 239
Bearbeiten, 184
Befehl hinzufügen/löschen, 250
Befehle löschen, 258
Extras, 241
Hilfe, 287
Makro hinzufügen, 259
Projekt, 187
Standard/XMLSPY, 258
Menü "Datei", 168
Menü "Extras", 241
Menü "Fenster", 285
Menübefehle, 167
Menüleiste, 20
Microsoft® SharePoint® Server, 211
MIME, 271
Mixed Markup-Symbole anzeigen, 44, 47
MSXML, 276
Multi-user, 268

N

Nebeneinander, 275
Neu laden, 268
geänderte Dateien, 174
Neue Datei,
erstellen, 168
Nicht gespeicherte Änderungen, 268
Nicht-XML Dateien, 271

O

Öffnen,
Datei, 169
Optimale Breite, 275
Optionen beim Öffnen,
Datei, 268

P

Parameter,
in DB-Abfragen, 75
Übergabe an Stylesheet über Schnittstelle, 223
Parent, 485
Parser,
in Altova-Produkte integrierter, 671
XSLT, 276
PDF-Ausgabe,
in der Authentic View-Ansicht anhand der PXF-Datei generieren, 237
Plattformen für Altova-Produkte, 671
Plug-In, 262
ATL-Beispieldateien, 330
Konfiguration der Benutzeroberfläche, 326
Registrierung, 325
XMLSPY, 325
Programmeinstellungen, 267
Programmers' Reference, 295
Projekt,

Projekt,
aktive Dateien hinzufügen, 207
Dateien zu Projekt hinzufügen, 206
Eigenschaften, 215
externen Webordner hinzufügen, 211
Hinzufügen aktiver und verwandter Dateien, 207
Hinzufügen zu Versionskontrolle, 198
Menü, 187
Neu laden, 190
öffnen, 190
Ordner hinzufügen, 208
schließen, 190
Speichern, 191
URL zu Projekt hinzufügen, 207
zuletzt verwendete, 218
Projekte,
externe Ordner hinzufügen, 208
globale Ressourcen hinzufügen, 207
neue erstellen, 190
Übersicht, 187
Projektfenster, 17
Projekt-Fenster,
Anzeige, 285
PXF-Datei,
Ausgabedokumente von der Authentic View-Ansicht aus generieren, 237

R

Rechtliches, 674
Rechtschreiboptionen, 244
Rechtschreibprüfung,
benutzerdefiniertes Wörterbuch, 241
Registrieren Ihrer Altova-Software, 288
Registrierung,
Plug-In, 325
Registry,
Einstellungen, 267
RichEdit, 235
RTF-Ausgabe,
in der Authentic View-Ansicht anhand der PXF-Datei generieren, 237
Rückgängig, 184

S

save, 488
schema, 384
Einstellungen, 268
SchemaDocumentationDlg,
AllDetails, 530
Application, 530
IncludeAll, 532
IncludeAttributeGroups, 532
IncludeComplexTypes, 532
IncludeGlobalElements, 533
IncludeGroups, 533
IncludeIndex, 534
IncludeLocalElements, 534
IncludeRedefines, 535
IncludeSimpleTypes, 535
OptionsDialogAction, 536
OutputFile, 536
OutputFileDialogAction, 536
OutputFormat, 537
Parent, 537
ShowAnnotations, 537
ShowAttributes, 538
ShowChildren, 538
ShowConstraints, 539
ShowDiagram, 538
ShowEnumerations, 539
ShowNamespace, 539
ShowPatterns, 540
ShowProgressBar, 540
ShowProperties, 540
ShowResult, 540
ShowSingleFacets, 541
ShowSourceCode, 541
ShowType, 541
ShowUsedBy, 542
Schema-Manager,
CLI-Befehl Help, 150
CLI-Befehl Info, 151
CLI-Befehl Initialize, 151
CLI-Befehl Install, 152
CLI-Befehl List, 152
CLI-Befehl Reset, 153
CLI-Befehl Uninstall, 154

© 2016-2022 Altova GmbH

Schema-Manager,
CLI-Befehl Update, 155
CLI-Befehl Upgrade, 155
CLI-Übersicht, 150
Patch für Schema installieren, 147
Schema deinstallieren, 149
Schema installieren, 147
Schemas nach Status auflisten in, 145
starten, 142
Status von Schemas, 145
Übersicht, 139
Upgrade für Schema installieren, 147
zurücksetzen, 149
Schließen,
Projekt, 190
SharePoint® Server, 211
Skript-Editor,
Übersicht, 297, 299
Skripting,
Zu Skripting-Umgebung wechseln, 247
Skripts, 280
Skriptsprache, 280
Speichern,
Projekt, 191
Splashscreen, 275
SPP Datei-Speicherpfade, 187
SPS,
neuer XML-Datei zuweisen, 168
SPS-Tabellen,
dynamischen Tabellen bearbeiten, 44
SPS-Tabellen in der Authentic View-Ansicht,
Verwendung, 66
SpyProject,
CloseProject, 543
ProjectFile, 543
RootItems, 543
SaveProject, 544
SaveProjectAs, 544
SpyProjectItem,
ChildItems, 544
FileExtensions, 545
ItemType, 545
Name, 545
Open, 545
ParentItem, 545
Path, 546
ValidateWith, 546
XMLForXSLTransformation, 546
XSLForXMLTransformation, 546
XSLTransformationFileExtension, 546
XSLTransformationFolder, 546
SpyProjectItems,
AddFile, 547
AddFolder, 547
AddURL, 547
Count, 548
Item, 548
RemoveItem, 548
Standard,
Encoding, 272
Standardansicht,
Einstellungen im Hauptfenster, 271
Standard-Editor, 271
Standardwert,
Menü, 258
Startgruppe,
hinzufügen (Kontextmenü), 263
Statische (SPS) Tabellen in der Authentic View-Ansicht,
Verwendung, 66
Statusleiste, 20
Suchen,
und Text in Dokument ersetzen, 186
Suchen von Text in Dokument, 185
Support Center, 292
Support-Optionen, 23
Symbol,
Hinzufügen zu Symbolleiste/Menü, 250
Symbole,
große anzeigen, 263
Symbolleiste, 20, 252
Befehle anpassen, 250
Ein/Aus, 252
große Symbole anzeigen, 263
Makro hinzufügen, 259
Neue erstellen, 252
Symbolleiste & Menübefehle zurücksetzen, 252
Syntaxfärbung, 271, 275

T

Tabelle,
automatisch erstellen, 271
Tabellen,
dynamische (SPS)-Tabellen (SPS) bearbeiten, 44

Tabellen,
in der Authentic View-Ansicht, 31
Tabellen in der Authentic View-Ansicht,
Arbeiten mit (statischen und dynamischen) SPS Tabellen, 66
Arbeiten mit XML-Tabellen, 67
Symbole zum Bearbeiten von XML-Tabellen, 71
Verwendung, 66
Tabulatorzeichen, 268
Tastaturzuordnungen, 288
Tatstaturkürzel, 255
Technische Informationen, 671
Technischer Support, 292
terminate, 388
Text,
in der Authentic View-Ansicht bearbeiten, 61
in der Authentic View-Ansicht formatieren, 61
in Dokument suchen, 185
suchen und ersetzen, 186
TextImportExportSettings,
DestinationFolder, 549
EnclosingCharacter, 549
Encoding, 549
EncodingByteOrder, 550
FieldDelimiter, 550
FileExtension, 550
HeaderRow, 550
ImportFile, 550
Tooltip, 263
anzeigen, 263
Kürzel anzeigen, 263
Transformation,
siehe XSLT-Transformation, 222

U

Übernehmen, 267
UCS-2, 272, 672
UCS-4, 672
Unicode,
Konsortium, 672
Standard, 672
URL, 391, 488, 496
als E-Mail senden, 181
User manual, 10
UTF-16, 272
UTF-8, 672

V

Validator,
in Altova-Produkten, 671
Validieren von XML-Dokumenten, 219
Validierung, 22
Validierungseinstellungen, 268
Validierungsmeldungen, 19
VBScript,
Skripterstellung mit Authentic Desktop, 297
Versionskontrolle, 280
aktivieren /deaktivieren, 193
aktuellste Version holen, 193
auschecken, 195
Auschecken rückgängig, 198
ausgliedern, 199
Dateien abrufen, 194
Eigenschaften, 204
Hinzufügen in Versionskontrolle, 198
installieren eines Versionskontroll-Plug-in, 113
Projekt öffnen, 192
Status aktualisieren, 205
Unterschiede anzeigen, 202
unterstützte Anbieter, 191
Verlauf anzeigen, 201
Versionskontrollsystem wechseln, 205
Versionskontrollmanager, 205
Vertrauenswürdige Pfade für Authentic Skripts, 237
Vertrieb,
von Altova Software-Produkten, 674
von Altova-Software-Produkten, 674
Verzeichnisse,
mit Application.URLMakeDirectory erstellen, 391
Visual Studio,
die Authentic Desktop ActiveX Controls zur Toolbox hinzufügen, 621
Visual Studio .Net,
und Authentic Desktop, 157
und Authentic Desktop Unterschiede, 159
Vorlagen,
von XML-Dokumenten in Authentic View-Ansicht, 229
Vorlagendateien,
für neue Dokumente, 168
VS .NET,
und Authentic Integrationspaket, 158

© 2016-2022 Altova GmbH

W

Wechseln,
Zu Skript-Umgebung, 247
Windows,
Unterstützung für Altova-Produkte, 671
Wohlgeformtheitsprüfung, 219
Word 2007+-Ausgabe,
in der Authentic View-Ansicht anhand der PXF-Datei generieren, 237
Wörterbuch,
benutzerdefiniertes hinzufügen, 241
bestehendes ändern, 241
Rechschreibprüfung, 241

X

XML, 672
Menü, 219
Rechtschreibprüfung, 241
XML Parser,
Info, 671
XMLData,
AppendChild, 594
EraseAllChildren, 595
EraseCurrentChild, 596
GetChild, 597
GetChildKind, 598
GetCurrentChild, 598
GetFirstChild, 598
GetNextChild, 599
HasChildren, 601
HasChildrenKind, 601
InsertChild, 601
IsSameNode, 602
Kind, 603
MayHaveChildren, 603
Name, 603
Parent, 603
TextValue, 604
XML-Datei Global Resources, 97
XML-Datenbank,
neue Datenzeile in die Authentic View-Ansicht laden, 230
XML-DB,
neue XML-Datenzeile laden, 74
XML-Dokument,
in der Authentic View-Ansicht öffnen, 26
XML-Konformität, 271
XML-Signatur, 231
XML-Signaturen, 87
XMLSPY,
Plug-In Registrierung, 325
XMLSpy API,
Dokumentation, 341
XMLSPY Plug-In, 325
XMLSpyLib, 341
Application, 372
AuthenticDataTransfer, 394
AuthenticRange, 399
AuthenticView, 428
CodeGeneratorDlg, 446
DatabaseConnection, 452
Dialogs, 458
Document, 462
Documents, 493
DTDSchemaGeneratorDlg, 497
ElementList, 501
ElementListItem, 502
ExportSettings, 503
FileSelectionDlg, 506
GenerateSampleXMLDlg, 519
GridView, 525
ProjectItem, 544
SchemaDocumentationDlg, 529
SpyProject, 543
SpyProjectItems, 547
TextImportExportSettings, 548
XMLData, 593
XML-Tabellen in der Authentic View-Ansicht,
Editiersymbole, 71
Verwendung, 67
XML-Vorlagendatei,
in Authentic View-Ansicht, 26
XPath zum ausgewählten Node, 42
XQuery,
Variablen an XQuery-Dokument übergeben, 223
XSL/Query Menü, 221
XSL:FO,
Transformation, 222
XSLT,
Prozessor, 276
XSLT-Parameter,

XSLT-Parameter,
über Schnittstelle an Stylesheet übergeben, 223
XSLT-Transformation, 221, 222
in FO, 222
in PDF, 222

Z

Zeichensatz,
Kodierung, 272
Zeile,
anhängen (in Authentic View-Ansicht), 236
einfügen (in Authentic View-Ansicht), 236
kopieren (in Authentic View-Ansicht), 236
löschen, 236
löschen (in Authentic View-Ansicht), 236
nach oben/nach unten verschieben, 236
Zeile in Authentic View-Ansicht verschieben, 236
Zeilenschaltung,
siehe Eingabetaste, 90
Zeilenumbrüche, 268
Zuletzt verwendete Dateien,
Liste, 183
Zurücksetzen,
Kürzel, 255
Menübefehle, 258
Symbolleiste & Menübefehle, 252
Zuweisen,
Kürzel zu Befehl, 255

© 2016-2022 Altova GmbH