Altova StyleVision 2025 Basic Edition

Benutzer- und Referenzhandbuch

Altova StyleVision 2025 Basic Edition
Benutzer- und Referenzhandbuch

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

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

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

Published: 2025

© 2019-2025 Altova GmbH

Inhaltsverzeichnis

1 Einführung 13

1.1 Produktfeatures ... 14
1.2 Authentic-Ansicht in Altova-Produkten ... 18
1.3 Was ist ein SPS? ... 19
1.4 Einrichten von StyleVision ... 20
1.5 Terminologie ... 21
1.6 Informationen zu dieser Dokumentation ... 25

2 Benutzeroberfläche 27

2.1 Hauptfenster ... 28
2.1.1 Design-Ansicht ... 29
2.1.2 Ausgabeansichten ... 31
2.2 Seitenleisten ... 33
2.2.1 Design-Übersicht ... 36
2.2.2 Schema-Struktur ... 38
2.2.3 Design-Struktur ... 41
2.2.4 Globale Stile ... 45
2.2.5 Stil ... 47
2.2.6 Eigenschaften ... 49
2.2.7 Meldungen ... 53
2.2.8 Suchen und Ersetzen ... 53

3 Schnellstart-Tutorial 55

3.1 Erstellen und Einrichten eines neuen SPS ... 56
3.2 Einfügen von dynamischen Inhalten (aus der XML-Quelle) ... 61
3.3 Einfügen von statischen Inhalten ... 68
3.4 Formatieren des Inhalts ... 74
3.5 Verwenden von automatischen Berechnungen ... 80

3.6 Verwendung von Bedingungen ... 84
3.7 Verwendung von globalen Vorlagen und "Restlicher Inhalt" ... 92
3.8 Fertig! ... 96

4 Übersicht über die Verwendung 97

4.1 SPS und Quellen ... 98
4.2 Erstellen des Designs ... 99
4.3 XSLT- und XPath-Versionen ... 100
4.4 Kompatibilität mit Internet Explorer ... 101
4.5 Generierte Dateien ... 103
4.6 Kataloge in StyleVision ... 104
4.6.1 Funktionsweise von Katalogen ... 104
4.6.2 Katalogstruktur in StyleVision ... 105
4.6.3 Anpassen von Katalogen ... 107
4.6.4 Variablen für Windows-Systempfade ... 108

5 SPS-Inhalt 110

5.1 Einfügen von XML-Inhalt als Text ... 111
5.1.1 Einfügen von Inhalt mit einem vordefinierten Format ... 113
5.1.2 Restlicher Inhalt ... 113
5.2 Einfügen von MS Word-Inhalt ... 115
5.3 Einfügen von MS Excel-Inhalt ... 118
5.4 Benutzerdefinierte Vorlagen ... 120
5.5 Benutzerdefinierte Elemente, XML-Textblöcke ... 123
5.5.1 Benutzerdefinierte Elemente ... 123
5.5.2 Benutzerdefinierte XML-Textblöcke ... 124
5.6 Tabellen ... 126
5.6.1 Statische Tabellen ... 128
5.6.2 Dynamische Tabellen ... 129
5.6.3 Bedingte Verarbeitung in Tabellen ... 134
5.6.4 Tabellen in der Design-Ansicht ... 135
5.6.5 Formatieren von Tabellen ... 136
5.6.6 Anzeigen von Zeilen und Spalten ... 140

5.6.7 CALS/HTML-Tabellen ... 142
5.7 Listen ... 147
5.7.1 Statische Listen ... 147
5.7.2 Dynamische Listen ... 149
5.8 Grafiken ... 152
5.8.1 Bilder: URIs und Inline-Daten ... 152
5.8.2 Bildtypen und Ausgabe ... 154
5.8.3 Beispiel: Eine Vorlage für Grafiken ... 157
5.9 Formularsteuerelemente ... 158
5.9.1 Eingabefelder, mehrzeilige Eingabefelder ... 159
5.9.2 Kontrollkästchen ... 160
5.9.3 Auswahllisten ... 161
5.9.4 Optionsfelder, Schaltflächen ... 164
5.10 Links ... 165
5.11 Barcodes ... 166
5.12 Layout-Module ... 171
5.12.1 Layout Container ... 171
5.12.2 Layout-Felder ... 174
5.12.3 Linien ... 178
5.13 Die Funktion "Ändern in" ... 181

6 SPS-Struktur 184

6.1 Schemaquellen ... 186
6.1.1 DTDs und XML-Schemas ... 187
6.1.2 Benutzerdefinierte Schemas ... 193
6.2 Schema-Manager ... 197
6.2.1 Ausführen des Schema-Managers ... 200
6.2.2 Statuskategorien ... 202
6.2.3 Anwenden eines Patch oder Installation eines Schemas ... 203
6.2.4 Deinstallieren eines Schemas, Zurücksetzen ... 204
6.2.5 Befehlszeilenschnittstelle (CLI) ... 205
6.3 Zusammenführen von XML-Daten aus mehreren Quellen ... 213
6.4 Modulare SPSs ... 216
6.4.1 Verfügbare Modulobjekte ... 217

6.4.2 Erstellen eines modularen SPS ... 221
6.4.3 Beispiel: Ein Adressbuch ... 225
6.5 Vorlagen und Design-Fragmente ... 230
6.5.1 Hauptvorlage ... 230
6.5.2 Globale Vorlagen ... 231
6.5.3 Benutzerdefinierte Vorlagen ... 235
6.5.4 Variablenvorlagen ... 238
6.5.5 Node-Vorlagenoperationen ... 239
6.5.6 Design-Fragmente ... 242
6.6 XSLT-Vorlagen ... 246
6.7 Mehrfach-Dokumentausgabe ... 248
6.7.1 Einfügen einer Vorlage für ein neues Dokument ... 249
6.7.2 Vorlagen für neue Dokumente und Design-Struktur ... 250
6.7.3 URLs von Vorlagen für neue Dokumente ... 250
6.7.4 Vorschaudateien und Ausgabedokumentdateien ... 253
6.7.5 Dokumenteigenschaften und Stile ... 255

7 Komplexe Funktionen 256

7.1 Automatische Berechnungen ... 257
7.1.1 Bearbeiten und Verschieben von automatischen Berechnungen ... 257
7.1.2 Beispiel: Eine Rechnung ... 259
7.2 Bedingungen ... 262
7.2.1 Definieren der Bedingungen ... 262
7.2.2 Bearbeiten von Bedingungen ... 265
7.2.3 Bedingungen und automatische Berechnungen ... 266
7.3 Gruppieren ... 267
7.3.1 Beispiel: Group-By (Persons.sps) ... 270
7.3.2 Beispiel: Group-By (Scores.sps) ... 271
7.4 Sortieren ... 276
7.4.1 Funktionsweise ... 276
7.4.2 Beispiel: Sortieren nach mehreren Sortierschlüsseln ... 278
7.5 Parameter und Variablen ... 282
7.5.1 Vom Benutzer deklarierte Parameter ... 282
7.5.2 Parameter für Design-Fragmente ... 284

7.5.3 SPS Parameter für Quelldokumente ... 286
7.5.4 Variablen ... 287
7.6 Inhaltsverzeichnis, Referenzierung, Lesezeichen ... 290
7.6.1 Markieren von Elementen für das Inhaltsverzeichnis ... 293
7.6.2 Erstellen der Inhaltsverzeichnis-Vorlage ... 301
7.6.3 Beispiel: Einfaches Inhaltsverzeichnis ... 307
7.6.4 Beispiel: Hierarchische und sequenzielle Inhaltsverzeichnisse ... 310
7.6.5 Automatische Nummerierung im Hauptteil der Dokuments ... 314
7.6.6 Querverweise ... 318
7.6.7 Lesezeichen und Hyperlinks ... 320

8 Darstellung und Ausgabe 327

8.1 Vordefinierte Formate ... 328
8.2 Output Escaping ... 330
8.3 Werteformatierung (Formatieren von numerischen Datentypen) ... 333
8.3.1 Funktionsweise der Werteformatierung ... 333
8.3.2 Syntax der Werteformatierung ... 336
8.4 Arbeiten mit CSS-Stilen ... 342
8.4.1 Externe Stylesheets ... 343
8.4.2 Globale Stile ... 346
8.4.3 Lokale Stile ... 348
8.4.4 Definieren von Stilwerten ... 351
8.4.5 Stileigenschaften über XPath ... 353
8.4.6 Kombistile ... 356
8.5 HTML-Dokumenteigenschaften ... 359

9 Zusätzliche Funktionen 361

9.1 Globale Altova-Ressourcen ... 362
9.1.1 Definieren globaler Ressourcen ... 362
9.1.2 Verwendung globaler Ressourcen ... 372
9.2 URIs von ungeparsten Entities ... 377
9.3 Neu von XSLT-, XSL-FO- oder FO-Datei ... 379
9.4 Benutzerdefinierte XPath-Funktionen ... 383

9.4.1 Definieren einer XPath-Funktion ... 386
9.4.2 Funktionen zum Auffinden von Nodes wiederverwenden ... 388
9.4.3 Parameter in XPath-Funktionen ... 389
9.5 Arbeiten mit Datumswerten ... 399
9.5.1 Formatieren von Datumswerten ... 399
9.6 Verwenden von Skripts ... 403
9.6.1 Definieren von JavaScript-Funktionen ... 404
9.6.2 Zuweisen von Funktionen als Event Handler ... 405
9.6.3 Externe JavaScript-Dateien ... 406
9.7 HTML-Import ... 408
9.7.1 Erstellen eines neuen SPS anhand einer HTML-Datei ... 408
9.7.2 Erstellen des Schemas und SPS Designs ... 410
9.7.3 Erstellen von Tabellen und Listen als Elemente/Attribute ... 412
9.7.4 Generieren der Ausgabedateien ... 414
9.8 ASPX-Schnittstelle für Web-Applikationen ... 415
9.8.1 Beispiel: Lokaler Host auf Windows 7 ... 416
9.9 PXF-Datei: Behälter für SPS und dazugehörige Dateien ... 418
9.9.1 Erstellen einer PXF-Datei ... 418
9.9.2 Bearbeiten einer PXF-Datei ... 421
9.9.3 Bereitstellen einer PXF-Datei ... 422

10 Automatisierte Verarbeitung 424

10.1 Befehlszeilenschnittstelle ... 425
10.1.1 StyleVision ... 425
10.1.2 StyleVision Server ... 427
10.2 Verwendung von RaptorXML ... 428
10.2.1 PDF-Ausgabe ... 428
10.3 Automatisierung mit FlowForce Server ... 431
10.4 Anleitung zum Automatisieren der Verarbeitung ... 433

11 Menübefehle und Referenz 434

11.1 Symbole in der Design-Ansicht ... 435
11.2 Dialogfeld "'XPath-Ausdruck bearbeiten" ... 440

11.2.1 Auswertung ... 441
11.2.2 Debugger ... 445
11.2.3 Ausdruckserstellung ... 452
11.3 Symbolleisten ... 457
11.3.1 Format ... 459
11.3.2 Tabelle ... 460
11.3.3 Design-Elemente einfügen ... 462
11.3.4 Design-Filter ... 464
11.3.5 Globale Ressourcen ... 465
11.3.6 Standard ... 466
11.4 Menü "Datei" ... 468
11.4.1 Neu ... 468
11.4.2 Öffnen, Neu laden, Schließen, Alle schließen ... 475
11.4.3 Design speichern, Alles speichern ... 480
11.4.4 Speichern unter ... 485
11.4.5 Als MobileTogether Design-Datei exportieren ... 486
11.4.6 Generierte Dateien speichern ... 486
11.4.7 In FlowForce bereitstellen ... 487
11.4.8 Web Design ... 489
11.4.9 Eigenschaften ... 489
11.4.10 Druckvorschau, Drucken ... 491
11.4.11 Zuletzt verwendete Dateien, Beenden ... 492
11.5 Menü "Bearbeiten" ... 493
11.5.1 Rückgängig, Wiederherstellen, Alles markieren ... 493
11.5.2 Suchen, Weitersuchen, Ersetzen ... 493
11.5.3 Stylesheet-Parameter ... 498
11.5.4 Markup reduzieren/erweitern ... 499
11.6 Menü "Ansicht" ... 501
11.6.1 Symbolleisten und Statusleiste ... 501
11.6.2 Design-Seitenleisten ... 502
11.6.3 Design-Filter, Vergrößern/Verkleinern ... 503
11.6.4 Ausgabevorschauen ... 503
11.7 Menü "Einfügen" ... 505
11.7.1 Inhalt ... 505
11.7.2 Restlicher Inhalt ... 506

11.7.3 Formular-Steuerelemente ... 506
11.7.4 Automatische Berechnung ... 507
11.7.5 Absatz, Spezieller Absatz ... 508
11.7.6 Bild ... 509
11.7.7 Horizontale Linie ... 511
11.7.8 Tabelle ... 512
11.7.9 Nummerierung und Aufzählungszeichen ... 513
11.7.10 Lesezeichen ... 515
11.7.11 Hyperlink ... 517
11.7.12 Bedingung, Ausgabe-basierte Bedingung ... 518
11.7.13 Deaktiviert ... 520
11.7.14 Vorlage ... 520
11.7.15 Benutzerdefinierte Vorlage ... 522
11.7.16 Variablenvorlage ... 522
11.7.17 Design Fragment ... 523
11.7.18 Layout Container, Layout-Feld, Linie ... 523
11.7.19 Inhaltsverzeichnis ... 523
11.7.20 Neues Dokument ... 524
11.7.21 Benutzerdefiniertes Element ... 524
11.8 Menü "Einschließen in" ... 525
11.8.1 Vorlage ... 525
11.8.2 Benutzerdefinierte Vorlage ... 526
11.8.3 Variablenvorlagen ... 526
11.8.4 Absatz, Spezieller Absatz ... 527
11.8.5 Nummerierung und Aufzählungszeichen ... 527
11.8.6 Lesezeichen und Hyperlinks ... 528
11.8.7 Bedingung, ausgabebasierte Bedingung ... 528
11.8.8 Deaktiviert ... 530
11.8.9 Inhaltsverz.-Markierungen und Inhaltsverz.-Ebenen ... 530
11.8.10 Neues Dokument ... 531
11.8.11 Benutzerdefiniertes Element ... 531
11.9 Menü "Tabelle" ... 532
11.9.1 Tabelle einfügen, Tabelle löschen ... 532
11.9.2 Kopf-/Fußzeilen für Tabelle einfügen ... 533
11.9.3 Zeile/Spalten anhängen, einfügen ... 533

11.9.4 Zeile, Spalte löschen ... 534
11.9.5 Zelle links, rechts, unten, oben verbinden ... 534
11.9.6 Zelle horizontal oder vertikal teilen ... 535
11.9.7 Zellumrandung / Tabellen-Markup-Code anzeigen ... 535
11.9.8 Tabelleneigenschaften ... 535
11.9.9 CALS/HTML-Tabellen bearbeiten ... 536
11.9.10 Vertikale Ausrichtung des Zelleninhalts ... 537
11.10 Menü "Eigenschaften" ... 538
11.10.1 Nummerierung und Aufzählungszeichen bearbeiten ... 538
11.10.2 Vordefinierte Wertformatierungsstrings ... 538
11.11 Menü "Extras" ... 541
11.11.1 Rechtschreibung ... 541
11.11.2 Rechtschreiboptionen ... 542
11.11.3 Globale Ressourcen ... 545
11.11.4 Aktive Konfiguration ... 546
11.11.5 XML-Schema-Manager ... 547
11.11.6 Anpassen ... 547
11.11.7 Symbolleisten und Fenster wiederherstellen ... 552
11.11.8 Optionen ... 552
11.12 Menü "Fenster" ... 560
11.13 Menü "Hilfe" ... 561
11.13.1 Hilfe ... 561
11.13.2 Aktivierung, Bestellformular, Registrieren, Updates ... 561
11.13.3 Weitere Befehle ... 565

12 Anhänge 566

12.1 Informationen zum XSLT- und XQuery-Prozessor ... 567
12.1.1 XSLT 1.0 ... 567
12.1.2 XSLT 2.0 ... 567
12.1.3 XSLT 3.0 ... 569
12.1.4 XQuery 1.0 ... 571
12.1.5 XQuery 3.1 ... 574
12.2 XSLT- und XPath/XQuery-Funktionen ... 576
12.2.1 Altova-Erweiterungsfunktionen ... 577

12.2.2 Diverse Erweiterungsfunktionen ... 657
12.3 Datentypen in XML-Schemas, die von DB generiert wurden ... 676
12.3.1 ADO ... 676
12.3.2 MS Access ... 677
12.3.3 MS SQL Server ... 678
12.3.4 MySQL ... 679
12.3.5 ODBC ... 679
12.3.6 Oracle ... 680
12.3.7 Sybase ... 681
12.4 Technische Daten ... 683
12.4.1 OS- und Arbeitsspeicheranforderungen ... 683
12.4.2 Altova-Prozessoren ... 683
12.4.3 Unicode-Unterstützung ... 684
12.4.4 Internet-Verwendung ... 684
12.5 Lizenzinformationen ... 686
12.5.1 Electronic Software Distribution ... 686
12.5.2 Software-Aktivierung und Lizenzüberwachung ... 687
12.5.3 Altova Endbenutzer-Lizenzvereinbarung ... 688
12.5.4 Verpacken von Lizenzdateien in den StyleVision Installer ... 688

Index **689**

1 Einführung

Die Altova StyleVision 2025 Basic Edition ist eine Applikation zum grafischen Erstellen und Bearbeiten von StyleVision Power Stylesheets. StyleVision® läuft auf Windows 10, Windows 11 und Windows Server 2016 oder höher. Einige Funktionalitäten von StyleVision und Altova MissionKit können mit Applikationen des Microsoft Office-Pakets (MS Access, MS Excel, MS Word) ab Version 2007 integriert werden.

Ein StyleVision Power Stylesheet (SPS) kann für die folgenden Zwecke verwendet werden:

- für den Aufbau einer grafischen WYSIWYG-Ansicht von **XML-Dokumenten** in der **Authentic**-Ansicht, einem XML-Dokument-Editor, der in den folgenden Altova-Produkten zur Verfügung steht: Altova XMLSpy, Altova StyleVision, Altova Authentic Desktop und Altova Authentic Browser. Mit Hilfe der Authentic-Ansicht können Sie elektronische Formulare erstellen, die auf XML-Dokumenten basieren.
- zum Generieren von **XSLT Stylesheets** auf Basis des SPS Designs (Unterstützt werden XSLT 1.0, XSLT 2.0 und XSLT 3.0). Mit Hilfe der XSLT Stylesheets können Sie XML-Dokumente außerhalb von StyleVision in Ausgabedokumente wie z.B. HTML transformieren.
- zum Generieren von **HTML--Dokumenten** anhand eines XML-Dokuments - und zwar direkt von StyleVision aus.

Letzte Aktualisierung: 17.03.2025

1.1 Produktfeatures

Im Folgenden sind die wichtigsten Produktfeatures von StyleVision aufgelistet:

Allgemeine Produktfeatures

Im Folgenden finden Sie eine Liste der wichtigsten allgemeinen Funktionen von StyleVision.

- Sowohl die Enterprise als auch die Professional Edition ist jetzt als separate 64-Bit und 32-Bit-Applikation verfügbar.

Quellen

Einen SPS-Design kann ein XML-Schema oder einer DTD zugrunde gelegt werden. Weitere in einem Design verwendete Quelldateien sind XML- und CSS-Dateien. Im Zusammenhang mit Quelldateien werden die folgenden zusätzlichen Funktionen unterstützt:

- Globale Altova Ressourcen 362 können verwendet werden, um einen Pfad zu Quelldateien wie z.B. Schemas, XML- und CSS-Dateien anzugeben. Globale Ressourcen ermöglichen schnellere Entwicklung und einfacheres Testen, da Quelldaten schnell geändert werden können und die Funktionalitäten anderer Altova Applikationen von StyleVision aus verwendet werden können.
- HTML-Dokumente können in XML konvertiert werden 408.

Benutzeroberfläche

Die folgenden allgemeinen GUI-Funktionen stehen zur Verfügung:

- Es können gleichzeitig mehrere SPS Designs 27 geöffnet sein, wobei immer nur eines aktiv ist. Jedes SPS Design wird auf einem eigenen Register angezeigt.
- Mit Hilfe von Vorlagenfiltern 464 können Sie die Anzeige des Design-Dokuments anpassen. Mit dieser Funktion können Sie zur effizienteren Bearbeitung Vorlagen ausblenden, die gerade nicht bearbeitet werden.
- Markup-Symbole in der Design-Ansicht ausblenden 29: Die Markup Tags können in der Design-Ansicht ausgeblendet und reduziert werden, um mehr Platz in der Design-Ansicht zu schaffen.
- Während der Erstellung des SPS können Sie jederzeit durch Klicken auf das entsprechende Register die Ausgabeansichten 31 und Stylesheets anzeigen und auf diese Weise schnell eine Vorschau des Resultats und des XSLT-Codes sehen.

Ausgabeformate

Je nach verwendeter Edition werden unterschiedliche Ausgabeformate unterstützt. Um Zusammenhang mit der Dokumentausgabe werden die folgenden Funktionen unterstützt:

- Es werden die XSLT-Versionen (1.0, 2.0 und 3.0) 100 unterstützt.
- In der Enterprise und Professional Edition werden anhand eines einzigen SPS Designs mehrere Ausgabeformate 103 (HTML generiert.
- Sowohl XSLT-Dateien als auch Ausgabedateien 103 können generiert und gespeichert werden 103 - entweder direkt von der Benutzeroberfläche aus oder über StyleVision Server.
- Altova hat ein spezielles PXF-Dateiformat 418 entwickelt, über das eine SPS-Datei zusammen mit den dazugehörigen Quell- und Datendateien gespeichert werden kann. Auf diese Weise kann anstatt nur der SPS-Datei ein gesamtes SPS-Projekt transferiert werden.

© 2019-2025 Altova GmbH

- ASPX-Schnittstelle für Web-Applikationen[415]: Mit dieser Funktion können HTML-Webseiten rasch aktualisiert werden. StyleVision generiert aus einem SPS alle Dateien, die für eine ASPX-Anwendung nötig sind. Wenn die Webseite (eine `.aspx` Datei) aktualisiert wird, werden die Quelldaten (inklusive allfälliger Updates) via XSLT dynamisch zur Webseite transformiert.

SPS Designfeatures

Im Folgenden finden Sie eine Liste der wichtigsten StyleVision-Funktionen zum Erstellen des SPS.

- Das SPS kann statischen Text[111] enthalten, der in das SPS eingegeben wird, und dynamischen Text[111], der aus dem Quelldokument[186] geholt wird.
- Dynamischer Inhalt[111] wird durch Ziehen von Nodes inklusiver bestimmter Datentypen aus der Schemaquelle[186] in das Design eingefügt. Auch Design-Elemente (Absätze, Listen, Bilder, usw.) können zuerst eingefügt werden, um erst dann einen XML-Node aus der Schemastruktur zugewiesen zu erhalten.
- Dynamischer Inhalt[111] kann als Text oder in Form eines Steuerelements zur Dateneingabe[158] (z.B. als Eingabefeld[159] oder Auswahlliste[161]) eingefügt werden.
- Die Struktur des Designs[184] wird durch eine einzige Hauptvorlage[23] definiert und gesteuert. Diese Struktur kann durch optionale Vorlagen für einzelne Elemente, so genannte globale Vorlagen[230] global, weil sie global für dieses Element angewendet werden können - geändert werden.
- Globale Vorlagen[231] können auch für einzelne Datentypen erstellt werden, sodass die Verarbeitung auch auf Basis von Typen behandelt werden kann.
- Mehrfach-Dokumentausgabe[248]: Die vom SPS erzeugte Ausgabe kann in mehrere Dokumente aufgeteilt werden. Im Design werden Vorlagen für ein neues Dokument erstellt und Inhalt in sie eingefügt. Jede Vorlage für ein neues Dokument generiert ein eigenes Dokument in der Ausgabe.
- Benutzerdefinierte Vorlagen[120]: Für eine Sequenz von Elementen kann mit Hilfe eines XPath-Ausdrucks eine Vorlage generiert werden. Bei diesen Elementen kann es sich um atomare Werte oder Nodes handeln. Mit Hilfe eines XPath-Ausdrucks können Sie eine spezifischere Auswahl der Nodes treffen und für die Auswahl Bedingungen und Filter verwenden.
- Benutzerdefinierte Elemente[123]: Diese Funktion soll das Einfügen von Elementen von Auszeichnungssprachen (wie HTML, XSLT und XSL-FO) an jeder Stelle im Design ermöglichen.
- Benutzerdefinierte XML-Textblöcke[123]: XML-Textblöcke können an jeder beliebigen Stelle im Design eingefügt werden und werden dann im generierten XSLT Stylesheet an dieser Stelle erstellt.
- Design-Fragmente[242] ermöglichen eine Modularisierung und Wiederverwendung von Vorlagen in einem oder mehreren SPSs (*siehe modulare SPSs*[216]), ähnlich wie bei der Verwendung von Funktionen.
- SPS Module[216] können zu anderen SPS Modulen hinzugefügt werden, sodass Objekte, die in einem SPS Modul definiert sind, auch anderen Modulen zur Verfügung stehen. Auf diese Art können Modulobjekte in mehreren SPSs wiederverwendet werden, was die Verwaltung erleichtert.
- XSLT-Vorlagen[246]: XSLT-Dateien können in die generierten Stylesheets importiert werden. Wenn ein Node im XML-Instanzdokument einer Vorlage in der importierten XSLT-Datei zugewiesen wird und keine andere Vorlage Priorität hat, so wird die importierte Vorlage verwendet. Außerdem können benannte Vorlagen in der importierten XSLT-Datei vom Design aus aufgerufen werden.
- Neu von XSLT-Datei[379]: Ein SPS kann aus einem XSLT-forHTML oder einem XSLT-for-FO erstellt werden. Vorlagenstruktur und Styling im XSLT wird im SPS erstellt. Sie können die SPS-Komponenten dann modifizieren und Inhalt und Formatierung zum SPS hinzufügen.
- Benutzerdefinierte XPath-Funktionen[383]: Benutzer können XPath-Funktionen definieren, die an jeder Stelle im Dokument, an der XPath-Funktionen erlaubt sind, verwendet werden können.
- Layout Container[171]: Ein Layout Container ist ein Block, in den Design-Elemente gelegt und darin absolut positioniert werden können.
- Grafikvorlagen[174]: In einem Layout Container kann dem Design ein Formular als Bild hinterlegt werden. Anhand dieser Vorlage kann das bestehende Design genau nachgeahmt werden.
- Häufig weisen XML-Dokumente eine Struktur sich wiederholender Daten auf. So hat z.B. eine Abteilung in einem Unternehmen mehrere Angestellte. Die Daten für jeden Angestellten würden in einer

Datenstruktur gespeichert, die sich bei jedem Angestellten wiederholt. Im SPS wird die Verarbeitung solcher Datenstrukturen[111] einmal definiert und auf jeden einzelnen Node (jeden Angestellten-Node in diesem Fall) angewendet.

- In XSLT 2.0 und 3.0 SPSs können mehrere Inhaltsverzeichnisse[290] eingefügt werden.
- Sich wiederholende Datenstrukturen können auch als dynamische Tabellen[126] eingefügt werden. Dabei werden die Tabellendaten in Form einer Schleife abgearbeitet, wobei bei jeder Schleife durch die Datenstruktur eine Zeile (oder bei Bedarf eine Spalte) in einer Tabelle erstellt wird.
- Ein sich wiederholendes Element kann nach einem oder mehreren von Ihnen ausgewählten Sortierschlüsseln sortiert werden[276] und das sortierte Element wird an die Ausgabe (HTML)gesendet.
- Variablen[287]: Eine Variable kann nun in einer Vorlage deklariert werden und einen mittels eines XPath-Ausdrucks definierten Wert erhalten. Früher war der Wert einer Variablen auf die Auswahl des Node beschränkt, in der die Variable erstellt wurde. In Version 2010 kann nun jeder beliebige XPath-Ausdruck als Wert einer Variablen definiert werden.
- Nodes können auf Basis gemeinsamen Dateninhalts (z.B. dem gemeinsamen Wert eines Attributs) und ihrer Positionen gruppiert[267] werden.
- Mit Hilfe der Bedingungsfunktion[262] kann eine Vorlage aus einer Gruppe von Vorlagen auf Basis der Bedingungen verarbeitet werden, die im XML-Dokument oder der Systemumgebung erfüllt werden. Die Verarbeitung eines Dokument ist dadurch abhängig davon, welche Informationen im Quelldokument vorhanden sind und von Informationen, die der Ersteller des SPS zum Zeitpunkt der Erstellung nicht kennt (z.B. dem Zeitpunkt der Verarbeitung). Die verfügbaren Bedingungen sind jene, die mit Hilfe von XPath-Ausdrücken getestet werden können.
- Mit Hilfe von automatischen Berechnungen[257] können Daten aus dem/den Quelldokument/en bearbeitet und das Ergebnis angezeigt werden. Dies ist nützlich, wenn Sie Berechnungen an Zahlen durchführen (z.B. die Preise in einer Rechnung summieren), Strings bearbeiten (z.B. Bindestriche in Schrägstriche ändern), Inhalt generieren wollen usw. Die verfügbaren Bearbeitungen sind jene, die mit Hilfe von XPath-Ausdrücken durchgeführt werden können. In den XPath-Ausdrücken von automatischen Berechnungen können native Java- und .NET-Funktionen verwendet werden.
- Sie können in das Design Bilder[152] einfügen. Dabei kann es sich bei der URI für das Bild um eine statische handeln (die in das SPS eingegeben wird), eine dynamische (die aus dem Node im Quelldokument geholt wird) oder um eine Kombination aus statischen und dynamischen Komponenten.
- Bilder aus Inline-Daten[152]: Bilder können aus Base 16- und Base 64-kodiertem Text im XML-Dokument generiert und daher direkt im XML-Quelldokument als Text gespeichert werden. Ein SPS kann einen solchen Text nun dekodieren und das Bild darstellen.
- Es können zwei Arten von Listen[147] erstellt werden: statische und dynamische. In einer statischen Liste[147] wird jedes Element im SPS definiert. In einer dynamischen Liste[149] wird ein Node als Listenelement erstellt; die Werte aller Instanzen dieses Nodes werden als Elemente der Liste erstellt.
- Sie können in das Design statische und dynamische Links[320] einfügen. Die Ziel-URI kann statisch (ins SPS eingegeben) oder dynamisch sein (aus einem Node im Quelldokument geholt) oder es kann sich um eine Kombination aus statischen und dynamischen Komponenten handeln.
- Es können statische Lesezeichen[320] eingefügt werden. Sie dienen als Ankerpunkte, auf die ein Hyperlink verweist.
- Sie können Parameter[282] global für das gesamte SPS deklarieren. Ein Parameter wird mit einem Namen und einem String-Wert deklariert und kann in XPath-Ausdrücken und im SPS verwendet werden. Der deklarierte Parameterwert ist der Standardwert. Dieser Wert kann durch einen über StyleVision Server übergebenen Wert außer Kraft gesetzt werden.
- Mit der Eingabeformatierungsfunktion[333] kann der Inhalt numerischer XML-Schema-Datentypen-Nodes für die Ausgabe formatiert werden. Mit der Eingabeformatierung kann auch das Ergebnis einer automatischen Berechnung[257] formatiert werden.
- Mit Hilfe von JavaScript Funktionen[403] können Sie im SPS benutzerdefinierte Funktionalitäten für die Authentic-Ansicht und die HTML-Ausgabe bereitstellen.
- Über die Benutzeroberfläche steht eine Reihe von vordefinierten HTML-Formaten[328] zur Verfügung, die auf einzelne SPS Komponenten angewendet werden können.

© 2019-2025 Altova GmbH

- Über die Seitenleiste "Stil" [348] kann eine Vielzahl von CSS-Textformatierungs- und Layout-Eigenschaften auf einzelne SPS Komponenten angewendet werden.
- Zusätzlich dazu können auf der globalen Ebene [346] eines SPS und in externen CSS Stylesheets CSS-Stile für HTML-Selektoren definiert werden. Diese Stilregeln werden anschließend auf die HTML-Ausgabe angewendet und bieten dadurch flexible Formatierungs- und Layoutmöglichkeiten.
- Stile können auch mittels XPath-Ausdrücken zugewiesen werden [353]. Auf diese Art können Sie aus XML-Dokumenten Stileigenschaftswerte auswählen und Eigenschaftswerte auf Basis von Bedingungen setzen

1.2 Authentic-Ansicht in Altova-Produkten

Die Authentic-Ansicht ist ein grafischer XML-Editor, der in den folgenden Altova-Produkten zur Verfügung steht:

* Altova XMLSpy
* Altova Authentic Desktop
* Altova Authentic Browser
* Altova StyleVision

© 2019-2025 Altova GmbH

1.3 Was ist ein SPS?

Ein StyleVision Power Stylesheet (oder SPS) ist ein erweitertes XSLT Stylesheet, welches zum grafischen Erstellen eines Designs für ein HTML-Ausgabedokument verwendet wird.

Ein SPS hat die Dateierweiterung `.sps`.

Design des SPS

Ein SPS wird in StyleVision grafisch erstellt und basiert auf einen Schema (DTD oder XML-Schema). Das Design des SPS ist flexibel. Es kann dynamische und statische Inhalte enthalten. Bei dynamischem Inhalt[22] handelt es sich um Daten aus einem XML-Dokument. Bei statischem Inhalt[23] handelt es sich um Inhalt, der direkt in das SPS eingegeben wird. Dynamische Inhalte können entweder als einfacher Text oder in Form von Komponenten wie Eingabefelder, Auswahllisten und Tabellen eingefügt werden. Außerdem können dynamische Inhalte (mit Hilfe automatischer Berechnungen) bearbeitet und angezeigt werden, wenn bestimmte Bedingungen im Quelldokument erfüllt werden. Sie können die verschiedensten Inhalte an mehreren unterschiedlichen Stellen im SPS einfügen. Des Weiteren kann ein SPS verschiedene andere Komponenten enthalten, z.B. Bilder, Hyperlinks und JavaScript-Funktionen. Die einzelnen Komponenten des SPS können anschließend je nach Bedarf für die Darstellung formatiert werden.

Das SPS und XSLT Stylesheets

Nachdem Sie das SPS fertig gestellt haben, können Sie auf Basis dieses Designs XSLT Stylesheets generieren. StyleVision unterstützt sowohl XSLT 1.0 als auch XSLT 2.0 und XSLT 3.0. Sie können anhand eines einzigen SPS XSLT Stylesheets für die Ausgabe im HTML-, RTF-, XSL-FO-. Text- und Word 2007 (und höher)-Format (*XSL-FO, Text und Word 2007 (und höher) nur in der Enterprise Edition; RTF und Text in der Enterprise und der Professional Edition; in der Basic Edition wird nur die HTML-Ausgabe unterstützt*) erstellen. Die generierten XSLT Stylesheets können in externen Transformationen verwendet werden, um XML-Dokumente zu transformieren, die auf demselben Schema basieren, wie das SPS, von dem das XSLT Stylesheet generiert wurde. Nähere Informationen zur Arbeit mit XSLT Stylesheets finden Sie im Abschnitt Generierte Dateien[103].

Das SPS und die Ausgabe

Sie können mit Hilfe von StyleVision Ausgabedokumente auch direkt generieren (*HTML, RTF, Text, XSL-FO und PDF in der Enterprise Edition; HTML und RTF in der Professional Edition und und RTF in der Basic Edition*). Auf den Registern für die Ausgabeansichten[31] wird die Ausgabe für das aktive SPS Dokument direkt auf der grafischen Benutzeroberfläche von StyleVision angezeigt. Die gewünschte Ausgabe kann direkt über die Benutzeroberfläche über den Befehl **Datei | Generierte Dateien speichern**[486] oder über StyleVision Server Form einer Datei generiert werden.

Authentic-Ansicht in Altova-Produkten

Die Authentic-Ansicht ist ein grafischer XML-Editor, der in den folgenden Altova-Produkten zur Verfügung steht:

* Altova XMLSpy
* Altova Authentic Desktop
* Altova Authentic Browser
* Altova StyleVision

1.4 Einrichten von StyleVision

Altova StyleVision läuft auf Windows 10, Windows 11. Nachdem Sie StyleVision von der Altova Website heruntergeladen haben, doppelklicken Sie auf die ausführbare Datei (`.exe`), um das Installationsprogramm zu starten. Das Installationsprogramm installiert StyleVision im gewünschten Ordner. Die Altova XSLT-Prozessoren (1.0 und 2.0) sind in StyleVision integriert und werden für alle internen Transformationen verwendet. Sie müssen daher zusätzlich zu Ihrer StyleVision Installation keinen zusätzlichen XSLT-Prozessor installieren.

Es müssen jedoch die folgenden Komponenten installiert sein:

- Internet Explorer 5.5 oder höher für die HTML-Vorschau und die Design-Ansicht. Internet Explorer 6.0 und höher bietet eine bessere XML-Unterstützung. Für das Kopieren und Einfügen von Inhalt aus Word-Dokumenten 115 (und von Inhalt wie Excel-Tabellen und HTML-Seiteninhalt, der in Word-Dokumente eingefügt werden kann) ist Word 2007+ erforderlich.

© 2019-2025 Altova GmbH

1.5 Terminologie

In diesem Abschnitt werden die Begriffe erklärt, die auf der Benutzeroberfläche von StyleVision und in dieser Dokumentation verwendet werden. Die Begriffe sind in die unten aufgelisteten Gruppen gegliedert und innerhalb der einzelnen Gruppen alphabetisch geordnet.

Begriffe im Zusammenhang mit Altova-Produkten

Eine Liste von Begriffen, die sich auf Altova-Produkte beziehen.

***Authentic*-Ansicht**	Eine XML-Dokument Editor-Ansicht, die in folgenden Produkten zur Verfügung steht: Altova XMLSpy; Altova StyleVision; Altova Authentic Desktop; Altova Authentic Browser. Nähere Informationen zur Authentic-Ansicht und zu Altova-Produkten finden Sie auf der Altova Website.
SPS	Abkürzung für StyleVision Power Stylesheet. Wird in dieser Dokumentation verwendet und bezeichnet das Design-Dokument, das in StyleVision erstellt und als Datei mit der Erweiterung `.sps` gespeichert wird. Eine ausführliche Beschreibung finden Sie unter Was ist ein SPS? 19.
Globale Ressource	Ein Alias für eine Gruppe von Dateien, Ordnern oder Datenbanken. Jeder Alias hat eine Gruppe von Konfigurationen und jede Konfiguration ist einer Ressource zugeordnet. Wenn eine globale Ressource In StyleVision verwendet wird, kann die Ressource durch Ändern der aktiven Konfiguration in StyleVision gewechselt werden.

Allgemeine XML-Termini

Definitionen bestimmter XML-Termini, die in dieser Dokumentation verwendet werden.

Schema	Als Schema wird jede Art von Schema bezeichnet. Bei Schemas, die von StyleVision unterstützt werden, handelt es sich um ein XML-Schema 21 bzw. eine DTD.
XML-Schema	In dieser Dokumentation werden als XML-Schemas Schemas bezeichnet, die der XML-Schema-Spezifikation des W3C entsprechen. Bei XML-Schemas handelt es sich um eine Untergruppe von Schemas 21.
URI und URL	In dieser Dokumentation wird ausschließlich der allgemeinere Begriff "URI" verwenden - selbst wenn es sich nur um eine "Adresse" handelt und selbst bei `http`-Adressen.

XSLT und XPath-Begriffe

Von XSLT 1.0 und XPath 1.0 zu XSLT 2.0 und XPath 2.0 gab es Änderungen in der Terminologie. So wird z.B. der Root Node in XPath 1.0 als Dokument-Node 22 in XPath 2.0 bezeichnet. In dieser Dokumentation wird die neueste XSLT 3.0- und XPath 3.0-Terminologie verwendet.

Absoluter XPath	Ein Pfadausdruck, der am Root Node der Baumstruktur beginnt, die den Kontext-Node 22 enthält. In StyleVision kann ein Pfadausdruck bei Eingabe in ein Dialogfeld als absoluter Pfad eingegeben werden, wenn Sie das Kontrollkästchen "Absoluter Pfad" im Dialogfeld aktivieren. Bei deaktiviertem Kontrollkästchen ist der Pfad relativ zum Kontext-Node 22.

Kontextobjekt / Kontext-Node	Das Kontextobjekt ist das Objekt (Node oder String-Wert), relativ zu dem ein Ausdruck ausgewertet wird. Ein Kontext-Node ist ein Kontextobjekt, bei dem es sich um einen Node handelt. Das Kontextobjekt kann sich innerhalb eines Ausdrucks ändern, z.B. bei jedem Schritt innerhalb einer Pfadangabe oder innerhalb eines Filterausdrucks (Prädikat).
Aktueller Node	Der aktuelle Node ist jener Node, der gerade verarbeitet wird. Der aktuelle Node ist in Ausdrücken, die keine Unterausdrücke haben, identisch mit dem Kontext-Node[22]. Bei Unterausdrücken kann sich der Kontext-Node[22] allerdings ändern. Beachten Sie, dass es sich bei der `current()` Funktion um eine XSLT-Funktion und nicht um eine XPath-Funktion.
Dokument-Element	In einem wohlgeformten XML-Dokument wird das äußerste Element als Dokumentelement bezeichnet. Es ist ein Child des Dokument-Node[22]. In einem wohlgeformten XML-Dokument gibt es nur ein Dokumentelement. Auf der Benutzeroberfläche wird das Dokumentelement als Root Element bezeichnet.
Dokument-Node	Der Dokument-Node steht für das gesamte Dokument, d.h. er enthält es. Er ist der Root-Node der Baumstrukturdarstellung des Dokuments und wird in einem XPath-Ausdruck als: `'/'`dargestellt. Im Schema-Struktur-Fenster von StyleVision wird er folgendermaßen dargestellt: `'/ Root elements'`.

StyleVision-spezifische Begriffe

Termini zur Bezeichnung von StyleVision-Mechanismen, Begriffen und Komponenten.

Grafikvorlage	Eine Grafikvorlage ist eine Vorlage, die als Hintergrundbild in einem Layout Container[21] verwendet wird und normalerweise ein eingescanntes Formular ist. Das SPS-Design kann anhand dieser Grafikvorlage modelliert werden, sodass das Formular so aussieht wie die Vorlage.
dynamische Objekte	Objekte, die aus XML-Datenquellen stammen. Dabei kann es sich um Text, Tabellen und Listen handeln sowie um Bilder und Hyperlinks (wenn die URIs dynamisch sind).
globales Element	Ein Element in der Liste der globalen Elemente im Fenster "Schema-Struktur". In einem XML-Schema werden alle Elemente, die als globale Elemente definiert sind, in der Liste der globalen Elemente aufgelistet. In einer DTD sind alle Elemente globale Elemente und werden in der Liste der globalen Elemente aufgelistet. Globale Vorlagen[22] können nur für globale Elemente definiert werden.
globale Vorlage	Eine globale Vorlage kann für ein globales Element[22] definiert werden. Sie können die globale Vorlage anschließend überall dort für dieses Element verwenden, wo es im Dokument vorkommt. Alternativ zur globalen Vorlage kann die Verarbeitung für ein globales Element in einer lokalen Vorlage[22] definiert werden.
Layout Container	Ein Layout Container ist ein Design-Block, in dem Design-Elemente gestaltet und absolut positioniert werden können. Wenn ein Design anhand eines Formulars erstellt werden soll, kann es als Layout Container erstellt werden, sodass die Design-Elemente des Formulars absolut positioniert werden können. Alternativ dazu kann im Design ein frei fließendes Dokument verwendet werden, in das Layout Container platziert werden.
lokale Vorlage	Eine lokale Vorlage ist die Vorlage, die definiert, wie ein (globales oder nicht globales) Element in der Hauptvorlage[23] verarbeitet wird. Die lokale Vorlage wird auf dieses eine Element an dieser bestimmten Stelle angewendet, an der das Element in der

© 2019-2025 Altova GmbH

	Hauptvorlage[23] vorkommt. Anstelle der lokalen Vorlage kann eine globale Vorlage[22] auf eine bestimmte Instanz eines Elements in der Hauptvorlage[23] angewendet werden.
Hauptschema	Eine der zugewiesenen Schemaquellen wird als das Hauptschema definiert; der Dokument-Node der XML-Arbeitsdatei[23], die mit dem Hauptschema verknüpft ist, wird als Ausgangspunkt für die Hauptvorlage[23] verwendet.
Hauptvorlage	Die Hauptvorlage dient als Ausgangspunkt. In StyleVision entspricht diese Vorlage dem Dokumentelement[22]. Sie ist das erste Objekt, das vom XSLT-Prozessor verarbeitet wird. Im Fenster "Schema-Struktur" wird sie als Child des Dokument-Node[22] aufgelistet. In der Hauptvorlage[23] ist die grundlegende Struktur des Ausgabedokuments definiert und wie ein oder mehrere Eingabedokumente verarbeitet werden sollen. Sie kann lokale Vorlagen[22] enthalten und globale Vorlagen[22] referenzieren.
Ausgabe	Das Ausgabedokument, das durch die Verarbeitung eines XML-Dokuments mit einem XSLT Stylesheet erzeugt wird. Sie können mit StyleVision Dokumente im HTML-Format generieren. Auch XSLT Stylesheets, die mit StyleVision generiert werden, werden nicht als Ausgabe bezeichnet, sondern separat als XSLT Stylesheets behandelt.
statische Objekte	Objekte, die aus dem SPS und nicht aus XML-Datenquellen stammen. Dabei kann es sich um Text, Tabellen und Listen sowie Bilder, Hyperlinks und Lesezeichen handeln (wenn die URI statisch ist).
SPS Komponente	Bei einer SPS-Komponente kann es sich um Folgendes handeln: (i) einen Schema-Node (z.B. einen Element-Node); (ii) eine statische SPS-Komponente wie z.B. eine automatische Berechnung[257] oder einen Text-String; oder (iii) ein vordefiniertes Format[328] (das im SPS durch Start- und End-Tags dargestellt wird).
Vorlage	Wird grob als eine Gruppe von Anweisungen zur Verarbeitung eines Node oder eines Gruppe von Nodes definiert.
XML-Vorlagen -datei	Eine XML-Vorlagendatei wird einem SPS in StyleVision (Enterprise und Professional Edition) zugewiesen. Es handelt sich dabei um eine Datei, die die Ausgangsdaten für ein neues XML-Dokument liefert, das mit einem bestimmten SPS erstellt wird, wenn dieses SPS in der Authentic-Ansicht geöffnet wird. Die XML-Vorlagendatei muss dem Schema entsprechen, auf der das SPS basiert.
Benutzerdefiniertes Element	Ein Element, das weder ein Node in der Schemastruktur noch ein vordefiniertes Element oder ein Design-Element ist, sondern ein vom Benutzer definiertes Element. Ein Element kann mit Attributen definiert werden.
Benutzerdefinierte Vorlage	Eine Vorlage, die für eine in einem XPath-Ausdruck definierte Sequenz erstellt wird.
Benutzerdefinierte XML-Textblöcke	XML-Textblöcke können nach Belieben an jeder Stelle im Design eingefügt werden.
XML/XBRL-Arbeits -datei	Eine XML/XBRL-Arbeitsdatei ist eine XML-Datei, die einem SPS in StyleVision zugewiesen wird, um eine Vorschau der Ausgabe des XML-Dokuments in StyleVision zu sehen. Ohne eine XML/XBRL-Arbeitsdatei hat das SPS in StyleVision keine dynamischen XML-Daten zu verarbeiten. Wenn das SPS auf einem Schema basiert, das mehr als ein globales Element hat, ist manchmal nicht eindeutig, welches globale Element das Dokumentelement ist. Durch Zuweisung einer XML/XBRL-Arbeitsdatei wird dieses Problem gelöst (da ein gültiges XML-Dokument definitionsgemäß nur ein

Dokumentelement[22] hat). Beachten Sie, dass die XBRL-Funktionen nur in der Enterprise Edition zur Verfügung stehen.

XML-Dokument Die Bezeichnung XML-Dokument wird in zwei Bedeutungen verwendet: (i) um ein bestimmtes XML-Dokument zu bezeichnen; (ii) zur Bezeichnung einer beliebigen XML-Datenquelle. Die Bedeutung ergibt sich jeweils aus dem Zusammenhang.

© 2019-2025 Altova GmbH

1.6 Informationen zu dieser Dokumentation

Bei dieser Dokumentation handelt es sich um das Benutzerhandbuch zu StyleVision. Sie steht über die integrierte Hilfe von StyleVision zur Verfügung, kann online über die Altova Website aufgerufen und von dort auch als PDF zum Ausdruck heruntergeladen werden.

Das Benutzerhandbuch ist in die folgenden Abschnitte gegliedert:

- eine Einleitung 13, in der erklärt wird, was ein SPS ist und in der die wichtigsten Funktionen und Begriffe in StyleVision vorgestellt werden.
- eine Beschreibung der Benutzeroberfläche 27 von StyleVision.
- ein Tutorial 55, in dem die Funktionen von StyleVision anhand von praktischen Beispielen erklärt werden.
- Übersicht über die Verwendung 97. Hier wird die Verwendung der Applikation im Allgemeinen beschrieben z.B: die zur Erstellung eines SPS verwendeten Schemaquellen, der Design-Vorgang im Überblick, die Bereitstellung mit Authentic und Projekte.
- Inhalt von SPS-Dateien 110. Hier wird beschrieben, wie statische (Stylesheet-basierte) und dynamische (XML-Dokument-basierte) Komponenten im SPS erstellt und bearbeitet werden.
- Struktur der SPS-Datei 184. Hier wird beschrieben, wie ein SPS strukturiert und modularisiert werden kann. Außerdem wird die Behandlung von StyleVision-Vorlagen beschrieben.
- SPS-Datei - Komplexe Funktionen 256. Hier werden komplexe Design-Funktionen beschrieben wie z.B. die automatische Generierung von Berechnungen, das Einrichten von Bedingungen, das Gruppieren und Sortieren von benutzerdefinierten Kriterien und das Erstellen von Inhaltsverzeichnissen und Querverweisen im Ausgabedokument.
- SPS-Datei - Darstellung 327. Hier wird erklärt, wie Sie SPS Komponenten formatieren und ihnen ein Layout zuweisen.
- Zusätzliche Funktionen für die SPS-Datei 361. Hier wird eine Reihe von zusätzliche Funktionen beschrieben, mit denen Sie Ihr SPS verbessern können. Dazu gehören: Globale Ressourcen zum Nutzen von Funktionalitäten aus anderen Altova-Produkten, zusätzliche Validierung, Skripts und Variablen und Parameter.
- einen Abschnitt Referenz 434 mit einer Beschreibung aller in StyleVision verwendeter Symbole und Befehle.
- Anhänge 566, die Informationen über den Altova XSLT-Prozessor enthalten; technische Daten zu StyleVision; und Lizenzinformationen.

Anleitung zur Verwendung

Lesen Sie am besten zuerst die Abschnitte Einführung 13, Benutzeroberfläche 27 und Übersicht über die Verwendung 97 um einen ersten Eindruck von den Funktionen und der Verwendung von StyleVision zu gewinnen. Als nächstes finden Sie im Tutorial 55 praktische Beispiele zur Erstellung eines SPS. In den folgenden Abschnitten zur SPS-Datei (Inhalt von SPS-Dateien 110, Struktur der SPS-Datei 184, SPS-Datei - Komplexe Funktionen 256, SPS-Datei - Darstellung 327, Zusätzliche Funktionen für die SPS-Datei 361) wird ausführlich beschrieben, wie die verschiedenen Funktionen von StyleVision verwendet werden. Im Anschluss daran finden Sie im Abschnitt Referenz 434 eine genaue Beschreibung aller Symbolleistenschaltflächen, Design-Symbole und Menübefehle, geordnet nach Symbolleisten und Menü.

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: StyleVision läuft auch unter Windows Server 2016 oder höher.

Support-Optionen

Bei Fragen oder Problemen im Zusammenhang mit StyleVision stehen Ihnen die folgenden Support-Optionen zur Verfügung:

1. Lesen Sie in der Hilfe 13 (also in dieser Dokumentation) nach. Die Hilfe-Datei enthält eine Funktion zur Volltextsuche sowie einen ausführlichen Index.
2. Werfen Sie auf der Altova Website einen Blick in den Abschnitt Fragen und Antworten und das Diskussionsforum.
3. Kontaktieren Sie den Altova Support Center.

Gebräuchliche Abkürzungen

Die folgenden Abkürzungen werden häufig in dieser Dokumentation verwendet:

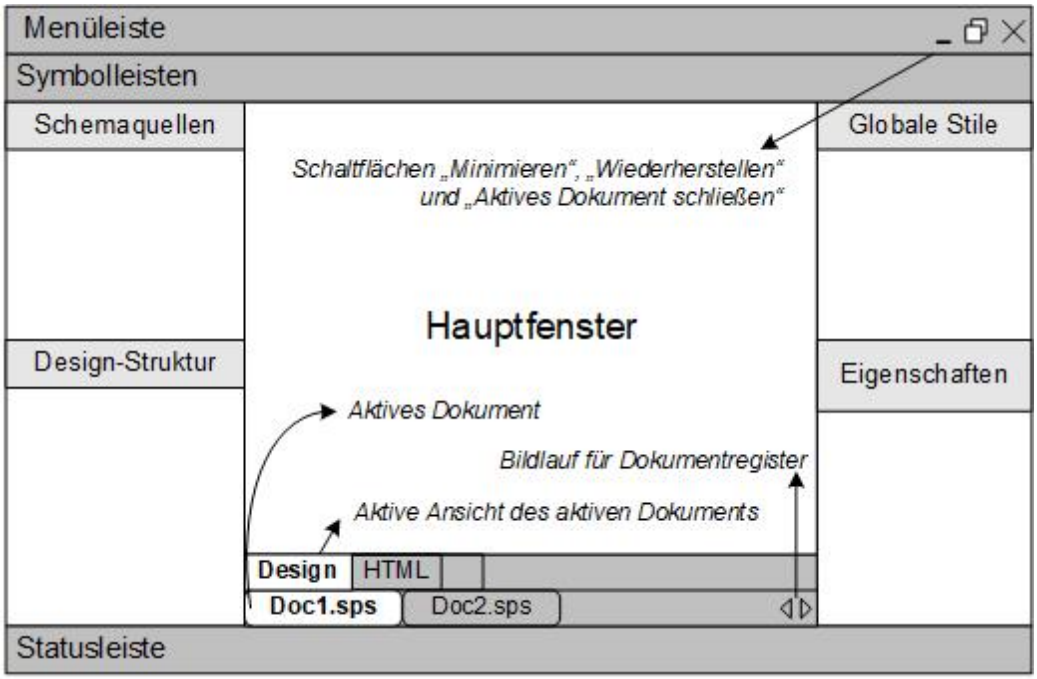
- **SPS**: StyleVision Power Stylesheet
- **CSS**: Cascading Style Sheets
- **FAQ**: (Frequently Asked Questions) häufig gestellte Fragen

© 2019-2025 Altova GmbH

2 Benutzeroberfläche

Die Benutzeroberfläche von StyleVision (*siehe Abbildung unten, in der nicht alle Seitenleisten zu sehen sind*) besteht aus den folgenden Bereichen:

- einer **Menüleiste**. Klicken Sie auf ein Menü, um die Einträge in diesem Menü zu sehen. Alle Menüs und Menüeinträge sind im Abschnitt Benutzerreferenz[434] beschrieben. Die Menüleiste enthält außerdem die Schaltflächen zum Minimieren, Wiederherstellen und Schließen des aktiven Dokuments.
- einem **Symbolleistenbereich**. Die verschiedenen Symbolleisten[457] und die Schaltflächen zum Aufrufen der Befehle in den einzelnen Symbolleisten sind im Abschnitt Benutzerreferenz[434] beschrieben.
- einem aus mehreren Registern bestehenden **Hauptfenster**[28], in dem ein oder mehrere geöffnete SPS angezeigt werden. In diesem Fenster können Sie das Design des SPS[29] bearbeiten und eine Vorschau der XSLT Stylesheets und Ausgabedokumente[31] anzeigen.
- die **Design-Seitenleisten**[33] - die Fenster Design-Übersicht[36], Schema-Struktur[38], Design-Struktur[41], Gobale Stile[45], Stile[47], Eigenschaften[49] - welche angedockt oder freischwebend angezeigt werden können.
- eine **Statusleiste**, in der Informationen zum Status der Applikation angezeigt werden.

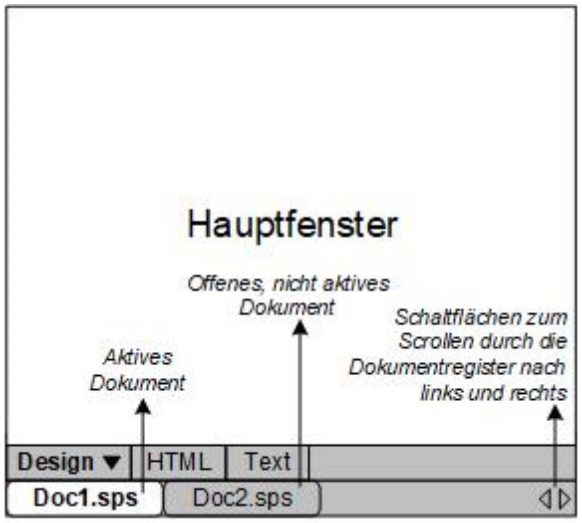


Das Hauptfenster[28] und die Design-Seitenleisten[33] werden in den Unterabschnitten dieses Abschnitts näher beschrieben.

Anmerkung: Sie können die Menüleiste und die Symbolleisten durch Ziehen an ihren Ziehpunkten an die gewünschte Position verschieben.

2.1 Hauptfenster

Im **Hauptfenster** (*siehe Abbildung unten*) werden das SPS Design, XSLT Stylesheets und Ausgabevorschauen angezeigt.



SPS Dokumente im Hauptfenster

- In StyleVision können mehrere SPS geöffnet sein. Es kann jedoch immer nur eines aktiv sein. Die Namen aller offenen Dokumente werden auf Registern am unteren Rand des Hauptfensters angezeigt, wobei das Register des aktiven Dokuments markiert ist.
- Um ein offenes Dokument zum aktiven zu machen, klicken Sie auf dessen Register. Alternativ dazu verwenden Sie die Optionen im Menü "Fenster".
- Wenn so viele Dokumente geöffnet sind, dass nicht alle Register in der Registerleiste zu sehen sind, klicken Sie auf die entsprechende Bildlaufschaltfläche (rechts von der Registerleiste siehe *Abbildung oben*) um die Register zu sehen.
- Um das aktive Dokument zu schließen, klicken Sie in der Menüleiste rechts oben im Applikationsfenster auf die Schaltfläche **Dokument schließen** (oder wählen Sie **Datei | Schließen** 475).

Dokumentansichten

Ein Dokument wird in den folgenden Ansichten angezeigt, wobei immer nur eine davon aktiv sein kann:

- Design-Ansicht 29. In dieser Ansicht wird das SPS erstellt und JavaScript-Funktionen für die Verwendung in diesem SPS editiert. Sie können zwischen dem Design-Dokument und dem JavaScript Editor hin und herwechseln, indem Sie auf den Pfeil für das Dropdown-Menü klicken und je nach Wunsch Design oder JavaScript auswählen.
- Ausgabeansichten 31 (HTML-, Text-Ausgabe). Diese Ansichten bieten eine Vorschau auf das tatsächliche Ausgabeformat und auf das XSLT Stylesheet, anhand dessen das Ausgabedokument generiert wird. Durch Klicken auf den Pfeil für das Dropdown-Menü und Auswahl der entsprechenden Option können Sie in dieser Ansicht zwischen der Ausgabevorschau und dem XSLT Stylesheet wechseln.

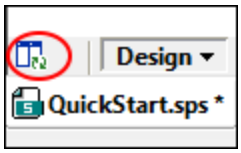
© 2019-2025 Altova GmbH

Jede der oben angeführten Ansichten steht in Form eines Registers in der Ansichtsleiste am unteren Rand des Hauptfensters zur Verfügung. Zur Auswahl einer Ansicht klicken Sie auf das entsprechende Register. Das Register der ausgewählten Ansicht erscheint markiert.

Ausgabevorschau

Mit Hilfe der Funktion "Ausgabevorschau" können Sie das Design auf der linken Seite und eine Vorschau auf die Ausgabe im rechten Bereich anzeigen. Dadurch können Sie noch, während Sie das Design erstellen, eine Vorschau auf die Ausgabe anzeigen und Ihr Design anschließend nicht nur im Hinblick auf die Darstellung, sondern auch auf seinen Inhalt entsprechend anpassen.

Durch Klick auf das Symbol **Ausgabevorschau** (*in der Abbildung unten rot umrandet*) können Sie das Hauptfenster in zwei vertikale Fenster teilen: (i) die Design-Ansicht: 29 im linken Bereich und (ii) Ausgabevorschauen im rechten Bereich. Das Symbol **Ausgabevorschau** (*siehe Abbildung unten*) befindet sich am linken unteren Rand des Hauptfensters. Um die Ausgabevorschau zu schließen, klicken Sie erneut auf das **Ausgabevorschau** Symbol.

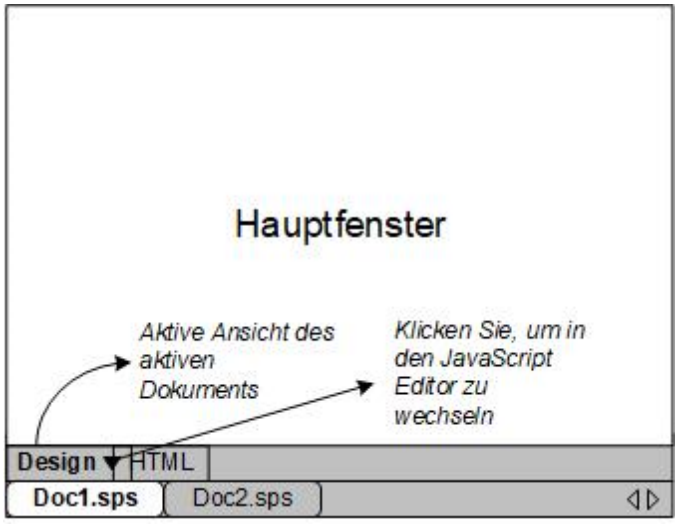


Im Fenster "Ausgabevorschauen":

- Klicken Sie auf das Symbol ganz links, um die aktuell ausgewählte Ausgabevorschau neu zu generieren.
- Klicken Sie auf das Symbol ganz rechts, um zwischen der vertikalen und der horizontalen Ausgabevorschau zu wechseln.

2.1.1 Design-Ansicht

Die **Design-Ansicht** (*siehe Abbildung unten*) ist die Ansicht, in der das SPS erstellt wird. In der Design-Ansicht wird das Aussehen des Ausgabedokuments festgelegt: Durch (i) Einfügen von Inhalt (über die Seitenleisten, die Tastatur und verschiedene Menü- und Symbolleistenfunktionen zur Erstellung und Bearbeitung von Inhalt); und (ii) durch Formatierung des Inhalts mittels verschiedener Formatierungsfunktionen, die über die Seitenleisten und Menüs zur Verfügung stehen. Diese Aspekte der Design-Ansicht werden weiter unten näher beschrieben.



Sie können in der Design-Ansicht auch zu einem JavaScript Editor[404] wechseln. Im JavaScript-Editor können JavaScript-Funktionen[403] erstellt und bearbeitet werden können. Diese Funktionen stehen dann auf der Benutzeroberfläche für die Verwendung im SPS zur Verfügung. . Um zum JavaScript Editor[404] zu wechseln, klicken Sie auf dem Register "Design" auf die Dropdown-Schaltfläche (*siehe Abbildung oben*) und wählen Sie im Dropdown-Menü die Option JavaScript. Um zurück zur Design-Ansicht zu wechseln, klicken Sie auf dem Register "JavaScript" auf die Dropdown-Schaltfläche und wählen Sie im Menü, das daraufhin erscheint, die Option "Design".

In der Design-Ansicht kann das SPS mehrere Vorlagen haben: die Hauptvorlage, globale Vorlagen und Design-Fragmente. Mit Hilfe von Vorlagenanzeigefiltern[464], die in Form von Symbolleisten-Schaltflächen[464] zur Verfügung stehen, können Sie festlegen, welche dieser Vorlagenfilter in der Design-Ansicht angezeigt werden sollen. Mit Hilfe der Anzeigenfilter können Sie die Anzeige Ihres SPS optimieren bzw. zwischen mehreren Anzeigen wechseln.

Anzeige von Markup-Tags

Die Anzeige von Markup-Tags in der Design-Ansicht kann über die Markup-Schaltflächen gesteuert werden (*siehe Abbildung unten*).

Die oben gezeigten Schaltflächen dienen zum Ein- und Ausschalten von Tags. Die Schaltflächen von links nach rechts sind: (i) Kleine Design Markup-Symbole einblenden (Tags ohne Namen); (ii) Große Design Markup-Symbole einblenden (Tags mit Namen). Wenn kleine Markup-Symbole eingeblendet werden, wird der Pfad zu einem Node angezeigt, wenn Sie die Maus über den Node platzieren.

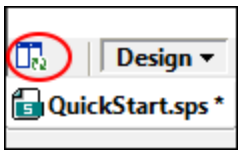
Ausgabevorschau

Mit Hilfe der Funktion "Ausgabevorschau" können Sie das Design auf der linken Seite und eine Vorschau auf die Ausgabe im rechten Bereich anzeigen. Dadurch können Sie noch, während Sie das Design erstellen, eine Vorschau auf die Ausgabe anzeigen und Ihr Design anschließend nicht nur im Hinblick auf die Darstellung, sondern auch auf seinen Inhalt entsprechend anpassen.

Durch Klick auf das Symbol **Ausgabevorschau** (*in der Abbildung unten rot umrandet*) können Sie das

© 2019-2025 Altova GmbH

Hauptfenster in zwei vertikale Fenster teilen: (i) die Design-Ansicht: [29] im linken Bereich und (ii) Ausgabevorschauen im rechten Bereich. Das Symbol **Ausgabevorschau** (*siehe Abbildung unten*) befindet sich am linken unteren Rand des Hauptfensters. Um die Ausgabevorschau zu schließen, klicken Sie erneut auf das **Ausgabevorschau** Symbol.



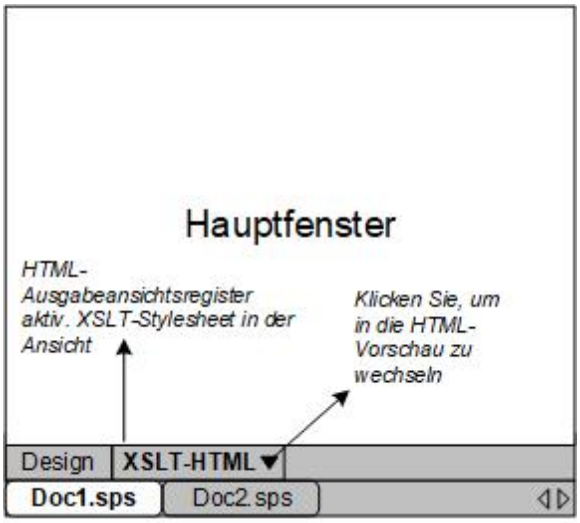
Im Fenster "Ausgabevorschauen":

- Klicken Sie auf das Symbol ganz links, um die aktuell ausgewählte Ausgabevorschau neu zu generieren.
- Klicken Sie auf das Symbol ganz rechts, um zwischen der vertikalen und der horizontalen Ausgabevorschau zu wechseln.

2.1.2 Ausgabeansichten

Auf dem **Ausgabeansichtsregister** (*siehe Abbildung unten*) sehen Sie: (i) das XSLT-Stylesheet für die HTML-Ausgabe, das anhand des SPS Designs generiert wird; und (ii) eine Vorschau der HTML-Ausgabe, die durch Transformation der XML-Arbeitsdatei [23] mit dem generierte XSLT Stylesheet erzeugt wird.

Um auf dem HTML-Ausgabeansichtsregister zwischen dem XSLT Stylesheet für die HTML-Ausgabe und der HTML-Ausgabevorschau zu wechseln, klicken Sie auf dem HTML-Ausgabeansichtsregister auf die Dropdown-Schaltfläche und wählen Sie je nach Bedarf die XSLT-Option oder die Ausgabevorschau.



XSLT-Ansicht

In der XSLT-Ansicht sehen Sie das XSLT Stylesheet für die HTML-Ausgabe, das anhand des gerade aktiven SPS erzeugt wurde. Das Stylesheet wird jedesmal neu generiert, wenn Sie die XSLT-Ansicht auswählen.

Ein Stylesheet wird auf dem Ausgabevorschauregister mit Zeilennummerierung und aus-/einklappbaren Elementen angezeigt. Klicken Sie auf das + und - Symbol am linken Rand, um Elemente zu erweitern bzw. zu reduzieren. Sie können das Stylesheet in der XSLT-Ansicht nicht bearbeiten, aber durchsuchen (wählen Sie den Befehl **Bearbeiten | Suchen** [493]) und Sie können daraus Text in die Zwischenablage kopieren (mit Hilfe von **Bearbeiten | Kopieren** [493]).

Anmerkung: Die anhand des SPS generierten Stylesheets können mit Hilfe des Befehls **Datei | Generierte Dateien speichern** [486] separat generiert und gespeichert werden.

HTML-Vorschau

In der HTML-Vorschau wird die durch Transformation der XML-Arbeitsdatei [23] mit dem XSLT Stylesheet für die HTML-Ausgabegenerierte Ausgabe angezeigt. Die Ausgabe wird jedesmal neu generiert, wenn Sie auf das Register HTML-Vorschau klicken. Beachten Sie, dass die gespeicherte Version der XML-Arbeitsdatei transformiert wird - nicht die temporäre Version, die in der Authentic-Ansicht bearbeitet wird.

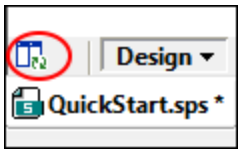
Wenn bei Auswahl der HTML-Vorschau auf dem HTML-Ansichtsregister keine XML-Arbeitsdatei [23] zugewiesen ist, werden Sie aufgefordert, eine XML-Arbeitsdatei zuzuweisen. Bei Datenbank-basierten SPSs ist dies nicht notwendig, da beim Laden der Datenbank automatisch eine nicht editierbare XML-Datei generiert wird, die als XML-Arbeitsdatei [23] verwendet wird.

Anmerkung: Die anhand des SPS generierten Ausgabedateien können mit Hilfe des Befehls **Datei | Generierte Dateien speichern** [486] separat generiert und gespeichert werden.

Ausgabevorschau

Mit Hilfe der Funktion "Ausgabevorschau" können Sie das Design auf der linken Seite und eine Vorschau auf die Ausgabe im rechten Bereich anzeigen. Dadurch können Sie noch, während Sie das Design erstellen, eine Vorschau auf die Ausgabe anzeigen und Ihr Design anschließend nicht nur im Hinblick auf die Darstellung, sondern auch auf seinen Inhalt entsprechend anpassen.

Durch Klick auf das Symbol **Ausgabevorschau** (*in der Abbildung unten rot umrandet*) können Sie das Hauptfenster in zwei vertikale Fenster teilen: (i) die Design-Ansicht: [29] im linken Bereich und (ii) Ausgabevorschauen im rechten Bereich. Das Symbol **Ausgabevorschau** (*siehe Abbildung unten*) befindet sich am linken unteren Rand des Hauptfensters. Um die Ausgabevorschau zu schließen, klicken Sie erneut auf das **Ausgabevorschau** Symbol.



Im Fenster "Ausgabevorschauen":

- Klicken Sie auf das Symbol ganz links, um die aktuell ausgewählte Ausgabevorschau neu zu generieren.
- Klicken Sie auf das Symbol ganz rechts, um zwischen der vertikalen und der horizontalen Ausgabevorschau zu wechseln.

© 2019-2025 Altova GmbH

2.2 Seitenleisten

Die **Seitenleisten** (auch als Seitenleistenfenster oder Fenster bezeichnet) sind Komponenten der Benutzeroberfläche, die Ihnen bei der Erstellung des SPS helfen und Informationen über die aktive Ansicht bereitstellen. Die einzelnen (*unten aufgelisteten*) Seitenleisten werden in einem Unterabschnitt dieses Abschnitts beschrieben.

- Design-Übersicht 36
- Schema-Struktur 38
- Design-Struktur 41
- Gobale Stile 45
- Stile 47
- Eigenschaften 49
- Meldungen 53
- Suchen und ersetzen 53

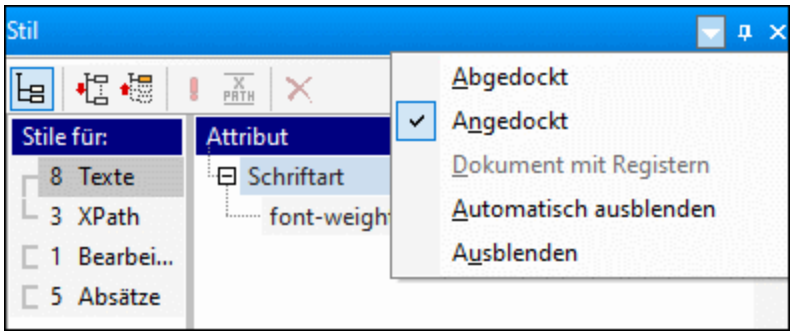
Layout der Ansichten

Das Layout einer Ansicht bezieht sich darauf, welche Seitenleisten in der Ansicht zur Verfügung stehen und wie dieses Seitenleisten auf der Benutzeroberfläche positioniert sind. Die Layouts können für einzelne Ansichtskategorien angepasst werden. Dabei werden (i) einzelne Seitenleisten in einer Ansicht ein- oder ausgeblendet (über das Menü **Ansicht** oder durch Rechtsklick auf die Titelleiste der Seitenleiste und Auswahl des Befehls **Ausblenden**); (ii) durch bedarfsgemäße Positionierung der Seitenleisten auf der Benutzeroberfläche. Das Layout, das auf diese Art für eine Ansichtskategorie definiert wurde, wird für diese Ansichtskategorie so lange beibehalten, bis es geändert wird. Wenn z.B. in der Design-Ansicht alle Seitenleisten mit Ausnahme der Seitenleiste "Stile" eingeblendet sind, wird dieses Layout in der Design-Ansicht auch bei mehrmaligem Wechseln der Ansicht beibehalten, bis Sie das Layout der Design-Ansicht ändern. Die Ansichtskategorien sind: (i) kein offenes Dokument; (ii) Design-Ansicht; (iii) Ausgabevorschau.

An- und Abdocken der Seitenleistenfenster

Die Seitenleistenfenster können auf der StyleVision Benutzeroberfläche angedockt werden oder freischwebend platziert werden. Um ein Fenster anzudocken, ziehen Sie es an der Titelleiste auf eine der vier inneren oder äußeren Pfeilspitzen, die beim Ziehen eingeblendet werden. Die inneren Pfeilspitzen dienen zum Andocken des Fensters relativ zum Fenster, in dem die inneren Pfeilspitzen angezeigt werden. Die vier äußeren Pfeilspitzen dienen zum Andocken des Fensters an einem der vier Ränder des Anwendungsfensters. Um ein Fenster freischwebend anzuzeigen, (i) doppelklicken Sie auf die Titelleiste oder (ii) ziehen Sie die Titelleiste an eine beliebige Stelle auf dem Bildschirm, jedoch nicht auf eine der Pfeilspitzen, die bei Ziehen eingeblendet werden.

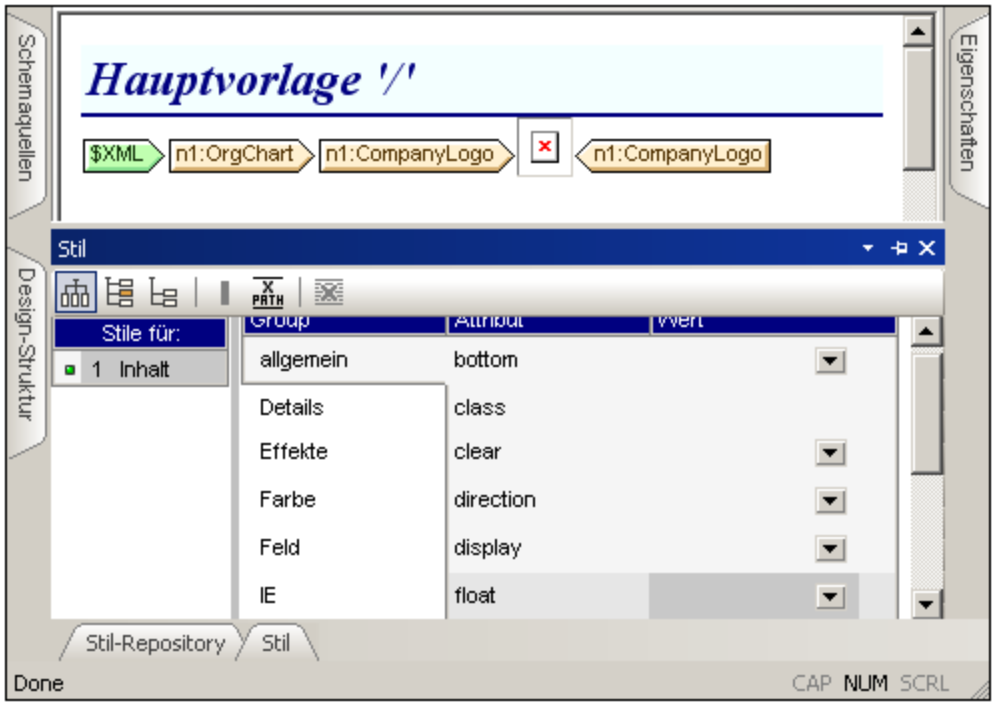
Alternativ dazu können Sie auch die folgenden Methoden verwenden. Um ein angedocktes Fenster freischwebend anzuzeigen, klicken Sie auf die Schaltfläche "**Menü**" in der rechten oberen Ecke eines angedockten Fensters (*siehe Abbildung unten*) und wählen Sie die Option "Abgedockt". Sie können dieses Menü auch durch Rechtsklick auf die Titelleiste des angedockten Fensters aufrufen.



Um ein freischwebendes Fenster anzudocken, rechtsklicken Sie auf die Titelleiste des freischwebenden Fensters und wählen Sie im Menü das daraufhin angezeigt wird, die Option "Angedockt"; das Fenster wird in der Position angedockt, an der es sich befand, als es zuletzt angedockt war.

Automatisches Ausblenden der Seitenleistenfenster

Ein angedocktes Fenster kann automatisch ausgeblendet werden. In diesem Fall wird es in Form eines minimierten Registers am Rand der Benutzeroberfläche angezeigt.
Wenn Sie den Cursor über das Register platzieren, wird dieses in die Benutzeroberfläche und über das Hauptfenster ausgerollt. Wenn Sie den Cursor im Screenshot unten über das Register "Stil" platzieren, wird die Seitenleiste "Stil" in das Hauptfenster ausgerollt.



Wenn Sie den Cursor aus dem ausgerollten Fenster herausbewegen und vom Register wegziehen, wird das Fenster wieder in das Register am Rand der Benutzeroberfläche eingerollt.

Die Funktion zum automatischen Ausblenden ist nützlich, wenn Sie selten gebrauchte Seitenleisten aus der Benutzeroberfläche entfernen und jederzeit wieder darauf Zugriff haben möchten, sollten Sie diese doch

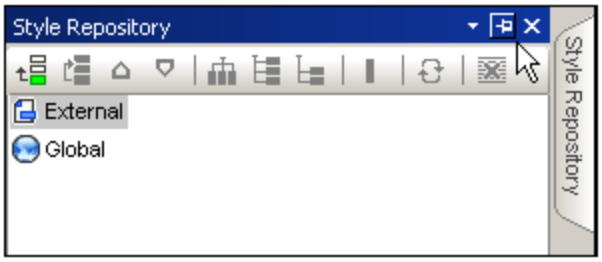
© 2019-2025 Altova GmbH

benötigen. Auf diese Art haben Sie im Hauptfenster mehr Platz zur Verfügung und trotzdem jederzeit Zugriff auf die Seitenleistenfenster.

Um ein Fenster in einem angedockten Fenster automatisch auszublenden, klicken Sie auf die Schaltfläche "Automatisch ausblenden" (das Pin-Symbol) rechts oben im Fenster (*Abbildung unten*). Alternativ dazu können Sie auch im Menü[33] die Option "Automatisch ausblenden" auswählen; (zum Anzeigen des Menüs[33] klicken Sie mit der rechten Maustaste auf die Titelleiste des Fensters oder klicken Sie in der Titelleiste des angedockten Fensters auf die Schaltfläche Menü[33]).

Das Fenster wird automatisch ausgeblendet.

Um die Funktion zum automatischen Ausblenden für ein bestimmtes Fenster zu deaktivieren, platzieren Sie den Cursor über das Register, so dass das Fenster ausgerollt wird und klicken Sie auf die Schaltfläche "Automatisch ausblenden" (*Screenshot unten*). Alternativ dazu können Sie auch im Menü[33] die Option "Automatisch ausblenden" deaktivieren; (zum Anzeigen des Menüs[33] klicken Sie mit der rechten Maustaste auf die Titelleiste des Fensters oder klicken Sie in der Titelleiste des angedockten Fensters auf die Schaltfläche Menü[33]).



Anmerkung: Wenn die Funktion zum automatischen Ausblenden eines Seitenleistenfensters deaktiviert ist, zeigt das Pin-Symbol dieses Fensters nach unten; wenn die Funktion aktiv ist, zeigt das Pin-Symbol nach links.

Ausblenden (Schließen) von Seitenleistenfenstern

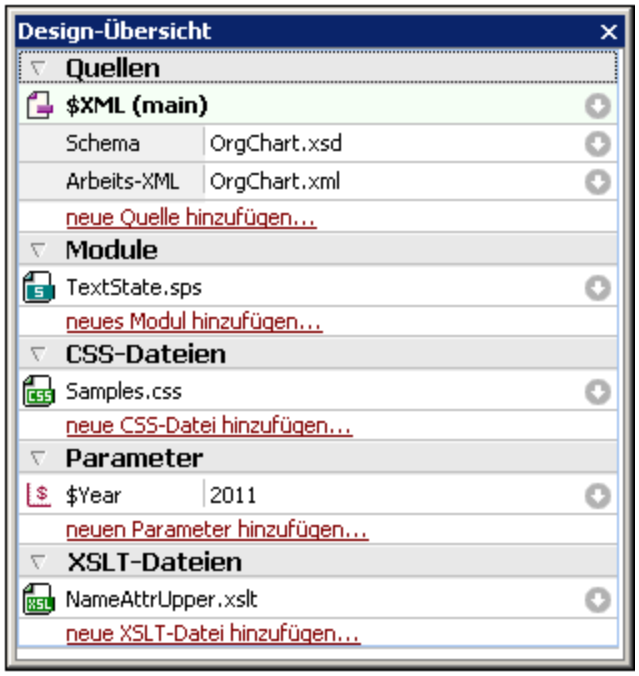
Wenn ein Seitenleistenfenster ausgeblendet ist, ist es auf der Benutzeroberfläche nicht mehr zu sehen, weder in seiner maximierten Form (angedockt oder freischwebend) noch in seiner minimierten Form (als Register am Rand der Benutzeroberfläche).

Um ein Fenster auszublenden, klicken Sie rechts oben in einem angedockten oder freischwebenden Fenster auf die Schließen-Schaltfläche. Alternativ dazu wählen Sie im Menü[33] die Option "Ausblenden"; (Klicken Sie zum Anzeigen des Menüs[33] mit der rechten Maustaste auf die Titelleiste des Fensters oder klicken Sie in der Titelleiste des Fensters auf die Schaltfläche Menü[33]).

Um ein ausgeblendetes (oder geschlossenes) Fenster wieder sichtbar zu machen, wählen Sie im Menü **Ansicht**[502] den Namen der jeweiligen Seitenleiste aus. Das Seitenleistenfenster wird an der Stelle eingeblendet, an der es sich befand, als es ausgeblendet wurde.

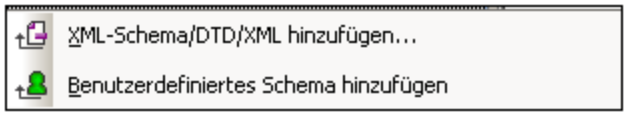
2.2.1 Design-Übersicht

Über die Seitenleiste **Design-Übersicht** (*Abbildung unten*) können Sie Schemaquellen, globale Parameter, SPS-Module und CSS-Dateien zum aktiven SPS hinzufügen. Sie haben hier eine Übersicht über diese Komponenten und können sie ganz einfach verwalten.



Hinzufügen von Schemaquellen

Schemaquellen können zu einem leeren SPS hinzugefügt werden. Eine Schemaquelle wird hinzugefügt, indem man unter der Überschrift "Quellen" auf den Befehl **Neue Quelle hinzufügen** klickt. Darauf erscheint ein Menü (*Abbildung unten*), über das Sie ein XML-Schema, eine DTD, ein automatisch anhand einer XML-Datei generiertes Schema oder ein benutzerdefiniertes Schema hinzufügen können.



Die XML-Arbeitsdatei

Wenn ein Schema hinzugefügt wird, wird es unter dem Eintrag "Schemaquellen" aufgelistet. Jedes Schema hat unterhalb des Eintrags für die XML-Datei 23 einen Eintrag für die XML-Arbeitsdatei 23.

Hinzufügen von Modulen, CSS-Dateien, Parametern und XSLT-Dateien

Klicken Sie auf den entsprechenden Befehl (**Neue(s)...hinzufügen)** am unteren Rand der Abschnitte "Parameter", "Module" und "CSS-Dateien" und "XSLT-Dateien", um ein neues Element zum entsprechenden Abschnitt hinzuzufügen.

© 2019-2025 Altova GmbH

Funktionen der Design-Übersicht

Die folgenden Funktionen stehen in allen Abschnitten (Quellen, Parameter, usw.) der Seitenleiste "Design-Übersicht" zur Verfügung:

- Jeder Abschnitt kann durch Klicken auf die dreieckige Pfeilspitze links vom Abschnittsnamen erweitert oder reduziert werden.
- Dateien in den Abschnitten "Quellen", "Module" und "CSS-Dateien" werden nur mit ihren Dateinamen aufgelistet. Wenn Sie die Maus über einen Dateinamen platzieren, wird der vollständige Pfadname in einem Popup-Fenster angezeigt.
- Grau angezeigte Einträge sind in einem importierten Modul, nicht aber in der derzeit aktiven SPS Datei aufgelistet.
- Jeder Abschnitt hat am unteren Rand des Abschnitts auch einen Befehl **Neue(n) <Eintrag> hinzufügen**, wodurch Sie einen neuen Eintrag zu diesem Abschnitt hinzufügen können. Wenn Sie z.B. auf den Befehl **Neuen Parameter hinzufügen** klicken, wird ein neuer Parameter zum SPS und zur Liste der Parameter in der Design-Übersicht hinzugefügt.
- Jeder Eintrag in einem Abschnitt hat ein Kontextmenü, das entweder durch Rechtsklick auf einen Eintrag oder durch Klicken auf die Schaltfläche **Kontextmenü** (den nach-unten-Pfeil rechts vom Eintrag) aufgerufen wird.
- Über die Option **Entfernen** im Kontextmenü wird der ausgewählte Eintrag entfernt. Dieser Befehl steht auch in Kontextmenüs zur Verfügung, wenn der Befehl angewendet werden kann.
- Über die Option **Datei in XMLSpy bearbeiten** im Kontextmenü wird die ausgewählte Datei in der Altova-Applikation XMLSpy geöffnet.
- Die Befehle **Nach oben** und **Nach unten** im Kontextmenü können nur verwendet werden, wenn im Abschnitt "Module" eines von mehreren Modulen[221] ausgewählt ist. Mit diesen Befehlen verschieben Sie das ausgewählte Modul jeweils um ein Modul nach oben bzw. nach unten.

Quellen

Im Abschnitt "Quellen" werden das Schema, auf dem das SPS basiert, und die XML-Arbeitsdatei die dem SPS zugewiesen wurde aufgelistet. Sie können jede dieser ausgewählten Dateien durch Aufrufen des Kontextmenüs (Rechtsklick oder Klick auf die Schaltfläche "Kontextmenü") und Auswahl der entsprechenden **...zuweisen** Option ändern.

Module

Im Abschnitt "Module" werden die vom aktiven SPS verwendeten SPS-Module[216] aufgelistet. Neue Module werden durch Klick auf den Befehl **Neues Modul hinzufügen** und Suche nach der gewünschten SPS-Datei angehängt. Da die Reihenfolge, in der die Module aufgelistet sind,[221] von Bedeutung ist, können die Befehle **Nach oben / Nach unten** (im Kontextmenü eines Modul) verwendet werden um die Reihenfolge zu ändern. Das Kontextmenü bietet außerdem einen Befehl zum Öffnen des ausgewählten Moduls in StyleVision.

Anmerkung: Die Seitenleiste "Design-Übersicht" enthält eine Übersicht über die Module, sodass Sie Module auf Dateiebene verwalten können. Die verschiedenen Modulobjekte[217] (Objekte innerhalb von Modulen) werden jedoch in der Seitenleiste "Design-Struktur"[41] aufgelistet.

CSS-Dateien

Im Abschnitt "CSS-Dateien" werden die vom aktiven SPS verwendeten CSS-Dateien aufgelistet. Neue CSS-Dateien werden durch Klick auf den Befehl **Neue CSS-Datei hinzufügen** und Suche nach der gewünschten CSS-Datei an die Liste angehängt. Da die Reihenfolge, in der die CSS-Dateien aufgelistet sind,[343] von

Bedeutung ist, werden die Befehle **Nach oben / Nach unten** (im Kontextmenü) aktiv, wenn eine CSS-Datei ausgewählt ist. Durch Klicken auf den entsprechenden Befehl kann die ausgewählte CSS-Datei nach oben bzw. unten verschoben werden. Das Kontextmenü bietet außerdem einen Befehl zum Öffnen der ausgewählten CSS-Datei in XMLSpy.

Anmerkung: Die Seitenleiste "Design-Übersicht" enthält eine Übersicht über die CSS-Dateien, sodass Sie CSS-Dateien auf Dateiebene verwalten können. Die verschiedenen CSS-Regeln[343] werden jedoch in der Seitenleiste "Globale Stile"[45] aufgelistet.

Parameter

Im Abschnitt "Parameter" werden die globalen Parameter im SPS aufgelistet. Mit dem Befehl **Neuen Parameter hinzufügen** am unteren Rand des Abschnitts können Sie neue Parameter hinzufügen. Bei Doppelklick auf den Parameternamen oder -wert können Sie den Namen bzw. Wert bearbeiten. Um einen Parameter zu entfernen, wählen Sie ihn aus und klicken Sie im Kontextmenü auf den Befehl **Entfernen**.

XSLT-Dateien

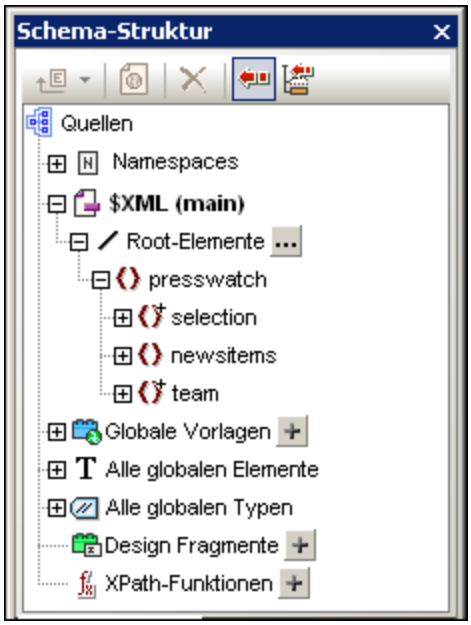
Im Abschnitt "XSLT-Dateien" sind die XSLT-Dateien aufgelistet, die in das SPS importiert wurden. XSLT-Vorlagen in diesen XSLT-Dateien stehen im Stylesheet als globale Vorlagen zur Verfügung. Eine vollständige Beschreibung dazu finden Sie unter XSLT-Vorlagen[246].

2.2.2 Schema-Struktur

Im Fenster **Schema-Struktur** (*Abbildung unten*) haben Sie folgende Möglichkeiten:

- Auswählen mehrerer Root-Elemente (Dokumentelemente) für ein Schema.
- Ziehen von Nodes aus einer Schema-Struktur in das Design. Diese Nodes stellen den XML-Inhalt dar, der in die Ausgabe inkludiert werden soll.
- Auflisten aller globalen Elemente und Typen in der Schemaquelle. Ein globales Element oder ein globaler Typ kann als globale Vorlage erstellt werden.
- Anzeigen einer Liste aller im SPS verwendeten Namespaces.
- Design-Fragmente[242] einfügen und bearbeiten.
- Benutzerdefinierte XPath-Funktionen[383] für das SPS einfügen und bearbeiten.

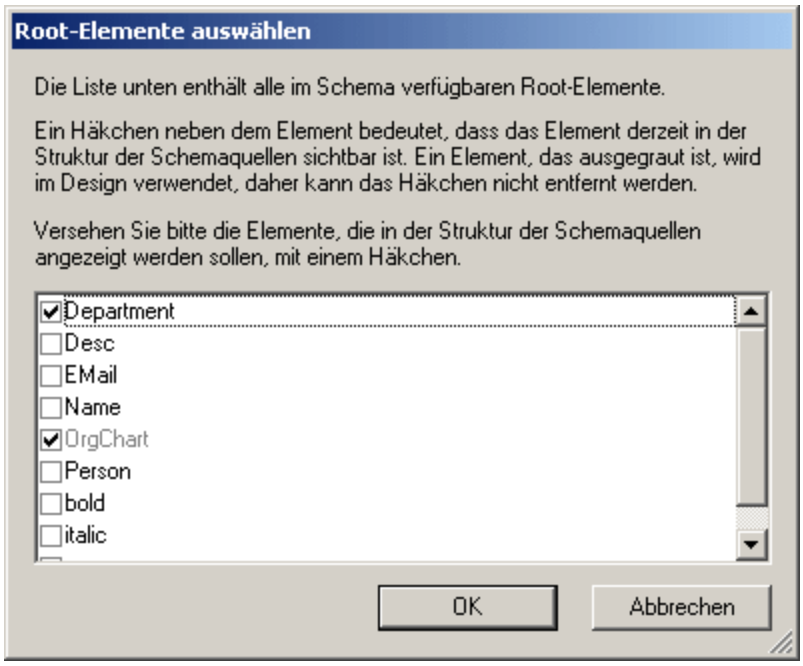
© 2019-2025 Altova GmbH



Root-Elemente

Für jedes Schema werden unter der Überschrift `$XML` die ausgewählten Root-Elemente[22] (auch Dokumentelemente[22] genannt) aufgelistet. Diese Liste enthält alle Root-Elemente, die Sie für das Schema auswählen (Beschreibung siehe unten). Jedes Root-Element kann erweitert werden, um die Struktur des Inhaltsmodells anzuzeigen. Der Inhalt der Hauptvorlage wird anhand der Nodes dieser Root-Element-Strukturen erstellt. Beachten Sie, dass der Ausgangspunkt für die Hauptvorlage der Dokument-Node des Hauptschemas ist, der jederzeit ausgewählt oder geändert werden kann (Beschreibung siehe unten).

Um Root-Elemente für ein Schema auszuwählen, gehen Sie folgendermaßen vor: Klicken Sie rechts vom Eintrag `Root Elemente` auf die Auswählen-Schaltfläche [...]. Daraufhin wird das Dialogfeld "Root-Elemente auswählen" (*siehe Abbildung unten*) angezeigt, in dem Sie auswählen können, welche/s der globalen Elemente im Schema als Root-Element/e fungieren soll(en). Eine Beschreibung der Möglichkeiten, die die Auswahl mehrerer Root-Elemente bietet, finden Sie unter SPS Struktur | Schemaquellen[186].



Zusätzlich dazu werden alle globalen Elemente im Schema unter dem Eintrag "Alle globalen Elemente" aufgelistet. Für jedes globale Element kann eine globale Vorlage[231] erstellt werden.

Globale Elemente und globale Typen

Mit Hilfe von globalen Elementen und globalen Typen können globale Vorlagen[231] erstellt werden, die in anderen Vorlagen wiederverwendet werden können. Außerdem können globale Typen auch direkt in Vorlagen verwendet werden.

Design-Fragmente

Alle Design-Fragmente[242] im Dokument sind unter einer Überschrift aufgelistet und können angezeigt werden, wenn das Design-Fragment erweitert wird. Ihnen steht die folgende Design-Fragment-Funktionalität zur Verfügung:

- Erstellen eines Design-Fragments durch Klicken auf das Symbol **Hinzufügen** des Elements "Design-Fragmente".
- Durch Doppelklick auf den Namen eines Design-Fragments in der Schema-Struktur kann der Name dieses Design-Fragments bearbeitet werden.
- Ein Design-Fragment kann durch Aktivieren bzw. Deaktivieren des Kontrollkästchens neben dem Design-Fragment aktiviert bzw. deaktiviert werden.
- Ein Design-Fragment kann aus der Schema-Struktur in das Design gezogen werden.

Nähere Informationen zum Arbeiten mit Design-Fragmenten finden Sie im Abschnitt Design-Fragmente[242].

Benutzerdefinierte XPath-Funktionen

Eine benutzerdefinierte XPath-Funktion kann hinzugefügt werden, indem Sie auf das Symbol **Hinzufügen** des Elements "XPath-Funktioen" klicken. Nachdem eine XPath-Funktion erstellt worden ist, wird sie in der Schema-Struktur angeführt. Durch Doppelklicken auf eine XPath-Funktion wird die Funktion in ihrem Dialogfeld zum Bearbeiten geöffnet. Folgende XPath-Funktionalität steht zur Verfügung:

© 2019-2025 Altova GmbH

- Eine XPath-Funktion kann durch Aktivieren bzw. Deaktivieren des Kontrollkästchens neben dem Element "XPath-Funktionen" aktiviert oder deaktiviert werden.
- Mit einem Klick der rechten Maustaste auf die XPath-Funktion wird auch ein Kontextmenü geöffnet, das Befehle zum Umbenennen und Entfernen einer XPath-Funktion enthält.

Nähere Informationen zum Arbeiten mit benutzerdefinierten XPath-Funktionen finden Sie im Abschnitt Benutzerdefinierte XPath-Funktionen 383.

Namespaces

Die im SPS verwendeten Namespaces werden zusammen mit ihren Präfixen unter der Überschrift "Namespaces" aufgelistet. Die Namespaces in dieser Liste stammen aus zwei Quellen: (i) Namespaces, die im referenzierten Schema bzw. den referenzierten Schemas definiert sind (*siehe Anmerkung unten*); und (ii) Namespaces, die standardmäßig zu jedem neu erstellten SPS hinzugefügt werden. Die Zuhilfenahme einer solchen Liste kann beim Schreiben von XPath-Ausdrücken sehr hilfreich sein. Außerdem können Sie eine XPath-Standard-Namespace für das gesamte SPS definieren, indem Sie in das Wertefeld des Eintrags `xpath-default-ns` doppelklicken und anschließend den Namespace eingeben.

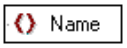
Anmerkung: Wenn Sie einen Namespace zu einem SPS oder einem anhand eines SPS generierten XSLT-Stylesheet hinzufügen möchten, muss der Namespace zum `schema`-Element der obersten Ebene des XML-Schemas, auf dem das SPS basiert, hinzugefügt werden.

Symbolleiste und Schemastruktursymbole

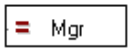
Die folgenden Symbolleisten-Schaltflächen sind Shortcuts für häufig gebrauchte Seitenleistenbefehle für die Schema-Struktur.

Symbole in Schemastrukturen

Im Folgenden finden Sie eine Liste der Symbole in Schemastrukturen.

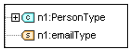


Element.



Attribut.

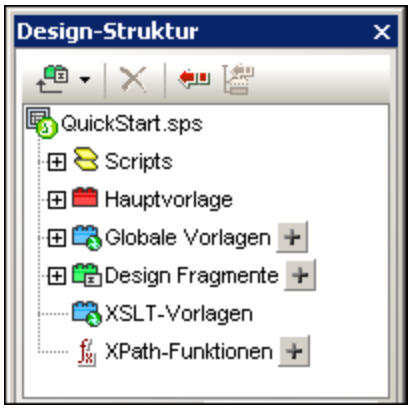
Element mit Child-Elementen. Doppelklicken Sie auf das Element oder das +/- Symbol links davon, um das Element zu erweitern/zu reduzieren.



Bei globalen Typen kann es sich entweder um complexTypes oder simpleTypes handeln. Complex Types werden durch ein zyanfarbenes Symbol, simple Types durch ein braunes Symbol gekennzeichnet.

2.2.3 Design-Struktur

Das Fenster **Design-Struktur** (*Abbildung unten*) bietet einen Überblick über das SPS Design.



An der Root der Design-Struktur befindet sich der Pfad für die SPS-Datei; der Dateipfad wird in einem Popup-Fenster angezeigt, wenn Sie die Maus über die Datei platzieren. Die nächste Ebene der Design-Struktur ist in die folgenden Kategorien gegliedert:

- Skripts 403: Zeigt alle JavaScript-Funktionen an, die mit dem JavaScript Editor von StyleVision für das SPS definiert wurden.
- Hauptvorlage 230: Zeigt die detaillierte Struktur der Hauptvorlage an.
- Globale Vorlagen 43: Listet die globalen Vorlagen im SPS sowie die globalen Vorlagen in allen inkludierten SPS Modulen auf.
- Design-Fragmente 43: Zeigt alle Design-Fragmente im Design und ermöglicht Ihnen sie zu erstellen, bearbeiten, umzubenennen und zu löschen.
- XSLT-Vorlagen 44: Geben Ihnen die Möglichkeit, XSLT-Vorlagen in importierten XSLT-Dateien anzuzeigen.
- Benutzerdefinierte XPath-Funktionen 383: Geben Ihnen die Möglichkeit, Ihre eigenen XPath-Funktionen zu erstellen, bearbeiten, umzubenennen und zu entfernen.

Symbolleistenschaltflächen

Die folgenden Symbolleistenschaltflächen sind Shortcuts für gebräuchliche Seitenleistenbefehle für die Schema-Struktur.

Fügt ein Design-Fragment, eine Hauptvorlage oder ein Layout-Element zum Design hinzu. Wenn Sie auf den linken Bereich des Symbols klicken, wird ein Design-Fragment hinzugefügt. Wenn Sie auf den Nach unten-Pfeil klicken, wird eine Liste mit Befehlen ausgeklappt, mit denen ein Design-Fragment oder ein Element aus der Reihe der Layout-Elemente hinzugefügt wird.

Ausgewähltes Objekt löschen; das Symbol ist aktiv, wenn das Objekt in den globalen Vorlagen oder Layout-Substrukturen ausgewählt ist.

Baumstruktur synchronisieren ein/aus. Wenn dieser Befehl aktiv ist (das Symbol wird mit einer Umrandung angezeigt), werden bei Auswahl eines Node in der Baumstruktur (i) der entsprechende Node im Design sowie (ii) der entsprechende Node in der Schema-Struktur ausgewählt, wenn das Symbol in der Schema-Struktur aktiviert ist. Ist dieser Befehl deaktiviert, werden die entsprechenden Nodes im Design und der Schema-Struktur nicht ausgewählt.

Wenn die Schaltfläche "Baumstruktur synchronisieren" aktiviert und ein Element im Design ausgewählt wird, werden die anderen erweiterten Objekte in der Schema-Struktur reduziert. Beachten Sie, dass diese Schaltfläche nur dann zur Verfügung steht, wenn die Schaltfläche "Baumstruktur synchronisieren" deaktiviert ist

© 2019-2025 Altova GmbH

Ändern der Anzeige der Design-Struktur

Sie können die Anzeige der Design-Struktur über das Kontextmenü (*Screenshot unten*), das Sie durch Rechtsklick auf ein Objekt in der Design-Struktur aufrufen können, ändern.

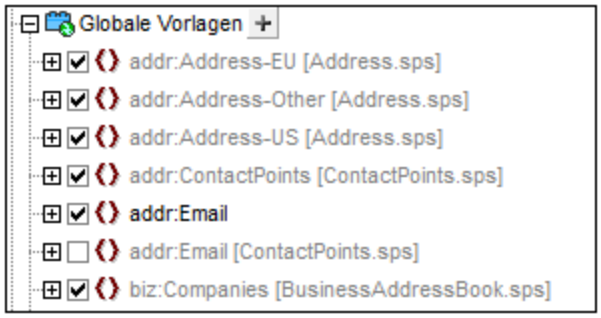
Mit dem Befehl **Entfernen** entfernen Sie den ausgewählten Eintrag aus der Design-Struktur. Mit dem Befehl **Als Design-Fragment definieren** erstellen Sie ein Design-Fragment[43] im Design und fügen dafür einen Eintrag in der Design-Struktur hinzu. Mit dem Befehl **Alle erweitern** werden alle Einträge der Design-Struktur erweitert.

Skripts und Hauptvorlagen

Unter der Überschrift "Skripts" werden alle Skripts im Design einschließlich jeder in importierten Modulen angezeigt. In der Liste "Hauptvorlage" wird eine Struktur der Hauptvorlage angezeigt. Elemente in der Struktur und im Design können durch Rechtsklick auf das Element und Auswahl von **Entfernen** entfernt werden.

Globale Vorlagen

Unter dem Eintrag "Globale Vorlagen[231]" werden alle globalen Vorlagen im aktuellen SPS und in allen hinzugefügten SPS Modulen aufgelistet. Im aktuellen SPS definierte globale Vorlagen werden schwarz angezeigt, globale Vorlagen, die in hinzugefügten Modulen definiert wurden, werden grau angezeigt (*siehe Abbildung unten*). Jede globale Vorlage hat links davon ein Kontrollkästchen, über das sie aktiviert oder deaktiviert werden kann. Wenn eine globale Vorlage deaktiviert wird, wird sie aus dem Design entfernt.

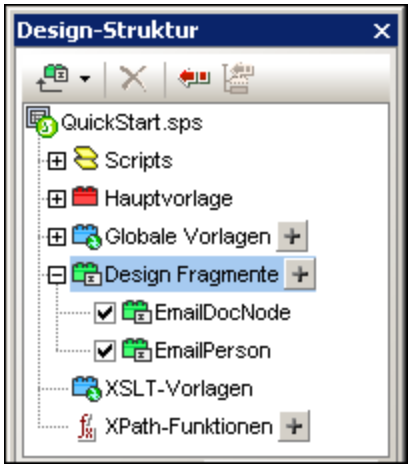


Eine globale Vorlagen im aktuellen SPS (nicht eine im hinzugefügten Modul) kann durch Auswahl der Komponente und Klicken auf die Schaltfläche **Entfernen** in der Symbolleiste oder durch Auswahl des Befehls **Entfernen** im Kontextmenü gelöscht werden. Die Komponente wird aus dem Design und aus der Baumstruktur entfernt.

Design-Fragmente

Unter dem Eintrag Design-Fragmente[242] werden alle Design-Fragmente im aktuellen SPS und in allen hinzugefügten SPS Modulen aufgelistet. Im aktuellen SPS definierte Design Fragmente werden schwarz angezeigt, Design-Fragmente, die in hinzugefügten Modulen definiert wurden, werden grau angezeigt (*siehe*

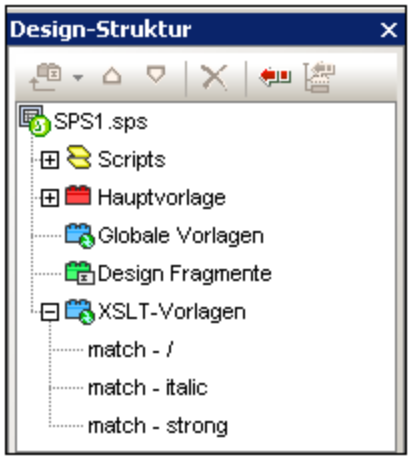
Abbildung unten). Jedes Design-Fragment hat links davon ein Kontrollkästchen, über das es aktiviert oder deaktiviert werden kann. Ein Design-Fragment im aktuellen SPS (nicht eines im hinzugefügten Modul) kann durch Auswahl und Klicken auf den Befehl **Entfernen** im Kontextmenü gelöscht werden. Die Komponente wird aus dem Design und aus der Design-Struktur entfernt.



Ein Design-Fragment kann durch Klicken auf das Symbol **Hinzufügen** rechts neben dem Eintrag "Design-Fragmente" hinzugefügt werden. Jedes Design-Fragment ist in einer Baumstruktur mit erweiterbaren/reduzierbaren Nodes angelegt. Jede Komponente in einer Design-Fragment-Struktur (die im aktuellen SPS definiert ist) kann durch Auswählen und Klicken auf die Schaltfläche **Entfernen** in der Symbolleiste oder durch Auswahl des Befehls **Entfernen** im Kontextmenü entfernt werden. Die Komponente wird aus dem Design und der Struktur entfernt.

XSLT-Vorlagen

In der Seitenleiste "Design-Struktur" (*Abbildung unten*) werden die in der importierten XSLT-Datei enthaltenen XSLT-Vorlagen unter der Überschrift "XSLT-Vorlagen" angezeigt.

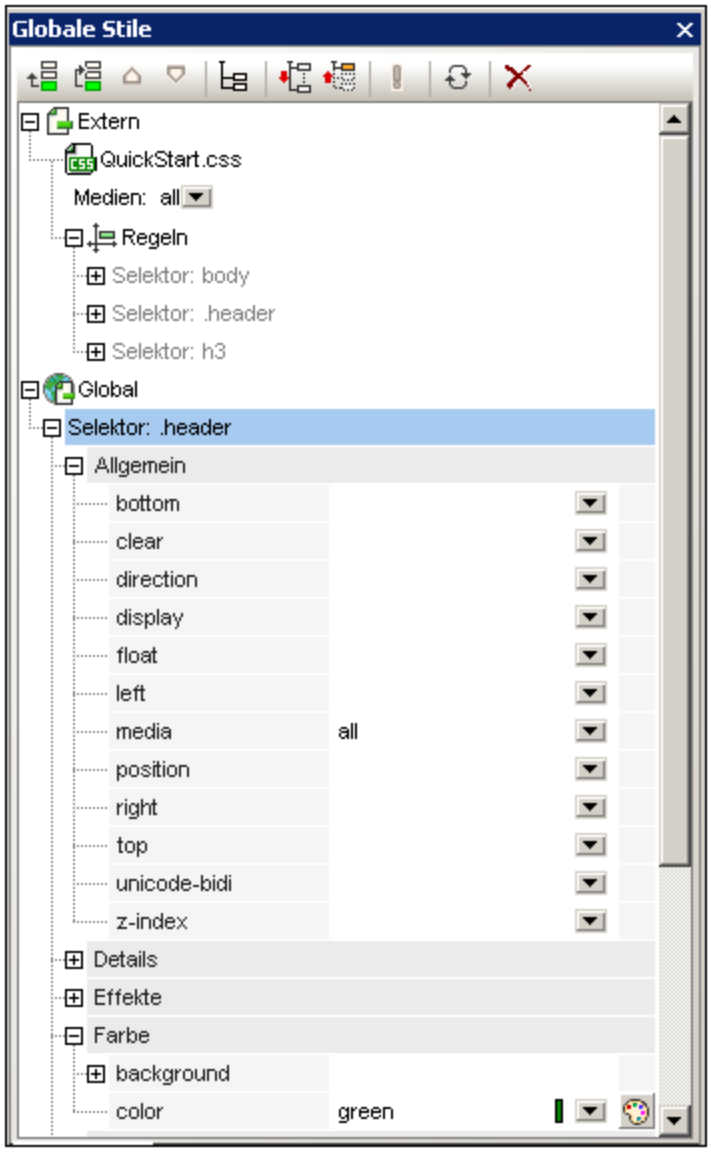


Es gibt zwei Arten von importierten XSLT-Vorlagen: (i) Match-Vorlagen (die durch *Match* gekennzeichnet sind), und (ii) benannte Vorlagen (die durch *Name* gekennzeichnet sind). In der Design-Struktur sind diese beiden Typen mit (i) dem Wert des `select` Attributs von Match-Vorlagen bzw. (ii) dem Wert des `name` Attributs von benannten Vorlagen aufgelistet. Eine ausführliche Beschreibung zur Funktionsweise von XSLT-Vorlagen finden Sie unter XSLT-Vorlagen 246.

 © 2019-2025 Altova GmbH

2.2.4 Globale Stile

Im Fenster **Gobale Stile** (*Screenshot unten*) können Sie externe CSS Stylesheets zuweisen und globale CSS-Stile für das SPS definieren. Stilregeln in externen CSS Stylesheets und global definierten CSS Stilen werden auf das HTML-Ausgabedokument angewendet.



Die Seitenleiste "Gobale Stile" enthält zwei Listen: **Extern** und **Global**, jede in Form einer Baumstruktur. Unter "Extern" befindet sich eine Liste externer CSS Stylesheets, die mit dem SPS verknüpft sind. Unter "Global" finden Sie eine Liste aller globalen Stile, die mit dem SPS verknüpft sind.

Die Listenstruktur unter Gobale Stile sieht folgendermaßen aus:

```
    External
- CSS-1.css (Location given in popup that appears on mouseover)
  - Media (can be defined in Style Repository window)
  - Rules (non-editable; must be edited in CSS file)
    - Selector-1
      - Property-1
      - ...
      - Property-N
    - ...
    - Selector-N
+ ...
+ CSS-N.css
    Global
- Selector-1
  + Selector-1 Properties
- ...
+ Selector-N
```

Vorrang von Stilregeln

Wenn eine globale Stilregel und eine Stilregel in einem externen CSS Stylesheet Selektoren haben, die sich auf dieselbe Dokumentkomponente beziehen, hat die globale Stilregel Vorrang vor der Stilregel im externen Stylesheet und wird angewendet. Wenn sich zwei oder mehrere globale Stilregeln auf dieselbe Dokumentkomponente beziehen, so wird die Regel angewendet, die als letzte in diesen Regeln aufgelistet ist. Wenn sich analog dazu zwei oder mehr Stilregeln im externen Stylesheet auf dieselbe Dokumentkomponente beziehen, so wird die letzte dieser Regeln im letzten der Stylesheets, das sie enthält, angewendet.

Verwalten von Stilen im Fenster Gobale Stile

Im Fenster "Gobale Stile" stehen über die Symbolleiste und/oder Kontextmenüeinträge die folgenden Möglichkeiten zur Verfügung:

Hinzufügen: Über das Symbol **Hinzufügen** können Sie, je nachdem, ob die Struktur "Extern" oder die Struktur "Global" ausgewählt ist, einen neuen externen Stylesheet-Eintrag zur Baumstruktur "Extern" bzw. einen neuen globalen Stileintrag zur Baumstruktur "Global" hinzufügen. Der neue Eintrag wird an die Liste der bereits vorhandenen Einträge in der Baumstruktur angehängt. Der Befehl **Hinzufügen** steht auch über das Kontextmenü zur Verfügung. Nähere Informationen zur Verwendung externer Stylesheets finden Sie unter Arbeiten mit CSS-Stilen 342. Beachten Sie, dass auch ein externes CSS-Stylesheet oder ein über die Seitenleiste "Design-Übersicht" 36 entferntes Stylesheet hinzugefügt werden kann. .

Einfügen: Über das Symbol **Einfügen** wird oberhalb des ausgewählten externen Stylesheet (in der Struktur "Extern") bzw. oberhalb des ausgewählten globalen Stils (in der Struktur "Global") ein neues externes Stylesheet bzw. ein neuer Eintrag für einen globalen Stil eingefügt. Der Befehl **Einfügen** steht auch über das Kontextmenü zur Verfügung. Nähere Informationen zur Verwendung externer Stylesheets und globaler Stile finden Sie unter Arbeiten mit CSS-Stilen 342.

Nach oben/Nach unten: Über das Symbol **Nach oben** bzw. **Nach unten** wird das ausgewählte externe Stylesheet bzw. der ausgewählte globale Stil im Vergleich zu den anderen Einträgen in der Baumstruktur nach oben bzw. nach unten verschoben. Diese Befehle eignen sich, um die Priorität externer Stylesheets relativ zueinander und die globaler Stilregeln relativ zueinander zu ändern. Diese beiden Befehle stehen auch über das

 © 2019-2025 Altova GmbH

Kontextmenü zur Verfügung. Nähere Informationen zum Ändern der Vorrangigkeit von Stilen finden Sie unter Arbeiten mit CSS-Stilen[342].

Ansichten der Stile eines Selektors: Jeder Selektor, ob in einem externen Stylesheet oder global definiert, kann in einer Ansicht angezeigt werden, die mit Hilfe von drei Ansichtseinstellungen konfiguriert werden kann. Diese Einstellungen sind: **Nicht leere auflisten** , **Alle erweitern** und **Alle reduzieren** , die alle in Form von Symbolleisten-Schaltflächen und Kontextmenübefehlen zur Verfügung stehen: Wenn Sie die Einstellung *Nicht leere auflisten* einschalten, werden nur die Stileigenschaften aufgelistet, für die ein Wert definiert ist. Andernfalls werden alle verfügbaren Stileigenschaften angezeigt (wodurch die Ansicht sehr unübersichtlich werden kann). Die Einstellungen *Alle erweitern* und *Alle reduzieren* lassen sich mit der Einstellung *Nicht leere auflisten* kombinieren und klappen alle Stildefinitionen des ausgewählten Selektors aus bzw. ein.

Flag 'Wichtig' ein/aus: Über das Symbol 'Wichtig' ein/aus können Sie den CSS-Wert `!important` für die ausgewählte CSS-Regel ein- oder ausschalten.

Alle neu laden: Über das Symbol **Alle neu laden** werden alle externen CSS Stylesheets neu geladen.

Zurücksetzen: Über das Symbol **Zurücksetzen** wird das ausgewählte externe Stylesheet bzw. der globale Stil gelöscht.

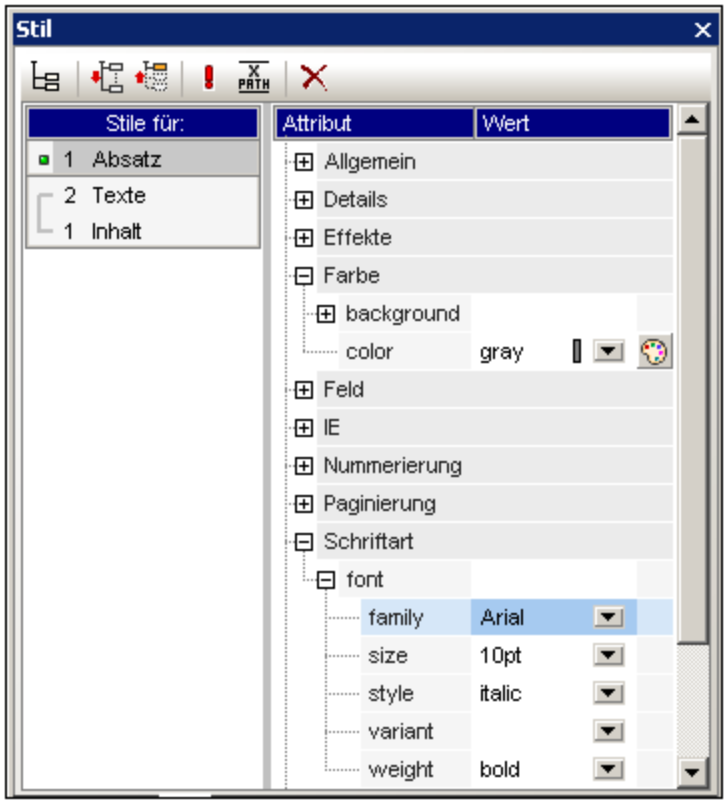
Bearbeiten von CSS-Stilen im Fenster Gobale Stile

Im Fenster Gobale Stile stehen die folgenden Bearbeitungsmethoden zur Verfügung:

- Sie können ein CSS Stylesheet hinzufügen und entfernen und das Medium definieren, auf das das externe CSS Stylesheet angewendet werden soll. Eine Erläuterung dazu finden Sie im Abschnitt Externe CSS Stylesheets[343].
- Die Selektoren und Eigenschaften globaler Stile können direkt im Fenster Gobale Stile bearbeitet werden. Eine Beschreibung dazu finden Sie im Abschnitt Globale Definition von CSS-Stilen[346].

2.2.5 Stil

Über das Fenster **Stil** (*Abbildung unten*) können Sie CSS-Stile **lokal** für in der Design-Ansicht ausgewählte SPS Komponenten definieren. Dies steht im Gegensatz zu Stilen, die in der Seitenleiste "Globale Stile" global definiert werden.



Die Seitenleiste "Stil" ist in zwei größere Bereiche gegliedert:

- die Spalte auf der linken Seite mit der Überschrift **Stile für**, in der die ausgewählten Komponententypen aufgelistet werden. Beachten Sie, dass eine in der Design-Ansicht markierte Auswahl mehrere Komponenten enthalten könnte, die nach Komponententyp geordnet sind. Die ausgewählten Komponenten werden in der Spalte "Stile für" nach Komponententyp geordnet aufgelistet. Für die Stilzuweisung kann immer nur jeweils ein Komponententyp ausgewählt werden. Wenn nur eine Instanz des Komponententyps vorhanden ist, so wird diese eine Instanz für die Stilzuweisung ausgewählt. Wenn mehrere Instanzen des Komponententyps vorhanden sind, so kann die Stilzuweisung für alle Instanzen des ausgewählten Komponententyps erfolgen. Die definierten Stile werden lokal auf die einzelnen Instanzen angewendet. Wenn Sie einen Stil nur einer bestimmten Instanz zuweisen möchten, so wählen Sie diese Komponenteninstanz in der Design-Ansicht aus und weisen Sie ihr in der Seitenleiste "Stil" lokal einen Stil zu. Sie können auch einen Komponentenbereich auswählen, indem Sie zuerst auf die erste Komponente des Bereichs und anschließend bei gedrückter Umschalttaste auf die zweite Komponente am Ende des Bereichs klicken. Nähere Informationen zur Auswahl von Komponententypen finden Sie unter: Lokale Definition von CSS-Stilen 348.
- den Bereich auf der rechten Seite für die **Stildefinitionen:** Hier werden die CSS-Stile für den/die in der Spalte "Stile für" ausgewählten Komponententyp/en definiert. Die Stildefinitionsspalte kann in drei Ansichten angezeigt werden (*Beschreibung siehe unten*). Nähere Informationen zum Festlegen der Stildefinitionen finden Sie im Abschnitt Definieren von CSS-Eigenschaftswerten 351. Mit der Schaltfläche "XPath" wird die Anwendung von XPath-Ausdrücken als Quelle von Stilwerten ein- und ausgeschaltet. Wenn eine Stileigenschaft ausgewählt ist und die Schaltfläche "XPath" aktiviert ist, so kann ein XPath-Ausdruck für diese Eigenschaft eingegeben werden. Der Rückgabewert des XPath-Ausdrucks wird als Wert für diese Stileigenschaft verwendet. Auf diese Art kann der Wert eines Node in einem XML-Dokument zur Laufzeit als Wert einer Eigenschaft zurückgegeben werden. Wenn die

 © 2019-2025 Altova GmbH

Schaltfläche XPath ausgeschaltet ist, kann ein statischer Wert als Wert der Eigenschaft eingegeben werden.

Einstellungen für die Definitionsansicht

Sie können die Ansicht der Definitionen Ihren Wünschen entsprechend anpassen. Es stehen drei (*unten aufgelistete)* Ansichtseinstellungen in Form von Symbolleisten-Schaltflächen und als Kontextmenübefehle zur Verfügung.

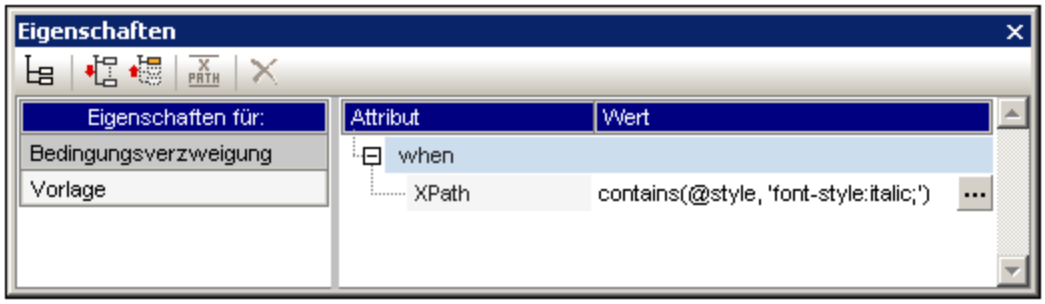
- **Nicht leere auflisten** : Wenn Sie diese Einstellung einschalten, werden für die in der Spalte auf der linken Seite ausgewählten Komponententypen nur die Stileigenschaften (in alphabetischer Reihenfolge) aufgelistet, für die ein Wert definiert ist. Andernfalls werden alle verfügbaren Stileigenschaften angezeigt. Diese Einstellung ist hilfreich, wenn Sie sehen möchten, welche Eigenschaften für einen bestimmten Komponententyp definiert sind. Wenn Sie neue Eigenschaften für den ausgewählten Komponententyp definieren möchten, muss diese Einstellung ausgeschaltet werden, damit Sie auf die benötigte Eigenschaft Zugriff haben.
- **Alle erweitern** : Für den in der linken Spalte ausgewählten Komponententyp werden alle im rechten Fenster angezeigten Eigenschaften ausgeklappt. Diese Einstellung kann mit der Einstellung **Nicht leere auflisten** kombiniert werden.
- **Alle reduzieren** : Für den in der linken Spalte ausgewählten Komponententyp werden alle im rechten Fenster angezeigten Eigenschaften eingeklappt. Diese Einstellung kann mit der Einstellung **Nicht leere auflisten** kombiniert werden.

Die Symbolleistenschaltflächen "Wichtige ein/aus" und "Zurücksetzen"

Wenn Sie auf das Symbol "Wichtige ein/aus" klicken, wird der CSS-Wert `!important` für die ausgewählte CSS-Regel ein- bzw. ausgeschaltet. Wenn Sie auf das Symbol "Zurücksetzen" klicken, wird der Wert der ausgewählten Eigenschaft zurückgesetzt.

2.2.6 Eigenschaften

Im Fenster **Eigenschaften** (*Screenshot unten*) können Sie Eigenschaften für in der Design-Ansicht ausgewählte SPS Komponenten definieren.



Das Fenster "Eigenschaften" besteht aus zwei Bereichen:

- dem Bereich **Eigenschaften für:** Hier werden die ausgewählten Komponententypen aufgelistet. Sie können immer nur jeweils einen Komponententyp auswählen und Eigenschaften dafür auswählen. (In der obigen Abbildung ist die Komponente *Vorlage* ausgewählt.) Nähere Informationen zum Gruppieren von Komponenten mit Eigenschaften finden Sie im Abschnitt unten: Komponenten und ihre Eigenschaftsgruppen 50.
- dem Bereich für die **Eigenschaftsdefinitionen:** Hier werden die Komponenteneigenschaften für den Komponententyp definiert, der im Bereich "Eigenschaften für" ausgewählt ist. Der Bereich für die Eigenschaftsdefinitionen kann in drei Ansichten angezeigt werden (*siehe unten*). Nähere Informationen dazu, welche Eigenschaften die einzelnen Eigenschaftsgruppen enthalten, finden Sie weiter unten im Abschnitt Eigenschaftsgruppen 52.

Einstellungen für die Definitionsansicht

Sie können die Ansicht der Definitionen Ihren Wünschen entsprechend anpassen. Es stehen drei (*unten aufgelistete)* Ansichtseinstellungen in Form von Symbolleisten-Schaltflächen und als Kontextmenübefehle zur Verfügung.

- **Nicht leere auflisten** : Wenn Sie diese Einstellung einschalten, werden für die in der Spalte auf der linken Seite ausgewählten Komponententypen nur die Stileigenschaften (in alphabetischer Reihenfolge) aufgelistet, für die ein Wert definiert ist. Andernfalls werden alle verfügbaren Stileigenschaften angezeigt. Diese Einstellung ist hilfreich, wenn Sie sehen möchten, welche Eigenschaften für einen bestimmten Komponententyp definiert sind. Wenn Sie neue Eigenschaften für den ausgewählten Komponententyp definieren möchten, muss diese Einstellung ausgeschaltet werden, damit Sie auf die benötigte Eigenschaft Zugriff haben.
- **Alle erweitern** : Für den in der linken Spalte ausgewählten Komponententyp werden alle im rechten Fenster angezeigten Eigenschaften ausgeklappt. Diese Einstellung kann mit der Einstellung **Nicht leere auflisten** kombiniert werden.
- **Alle reduzieren** : Für den in der linken Spalte ausgewählten Komponententyp werden alle im rechten Fenster angezeigten Eigenschaften eingeklappt. Diese Einstellung kann mit der Einstellung **Nicht leere auflisten** kombiniert werden.

Symbolleistenschaltfläche "Zurücksetzen"

Wenn Sie auf das Symbol "Zurücksetzen" klicken, wird der Wert der ausgewählten Eigenschaft auf den Standardwert zurückgesetzt.

Komponenten und ihre Eigenschaftsgruppen

Die Verfügbarkeit von Eigenschaften ist kontextsensitiv. Welche Eigenschaften zur Verfügung stehen, hängt davon ab, welche Design-Komponente ausgewählt ist. In der Tabelle unten sind SPS Komponenten und ihre Eigenschaftsgruppen aufgelistet.

Komponente	Eigenschaftsgruppe
Inhalt	Inhalt; Allgemein; Ereignis
Text	Text; Allgemein; Ereignis

© 2019-2025 Altova GmbH

Automatische Berechnung	XPath; Allgemein; Ereignis
Bedingungsverzweigung	When
Dateneingabe -element	Allgemein; [Dateneingabeelement]; Ereignis; HTML
Bild	Bild; Allgemein; Ereignis; HTML
Link	Link; Allgemein; Ereignis; HTML
Tabelle	Tabelle; Allgemein; Ereignis; HTML
Absatz	Absatz; Allgemein; Ereignis; HTML

Beachten Sie bei Komponententypen bitte folgende Dinge:

- Inhaltskomponenten sind die Platzhalter für `Inhalt` und `Restlicher Inhalt`. Sie stehen für den Textinhalt eines oder mehrerer Knoten aus dem XML-Dokument.
- Eine Textkomponente ist ein einzelner String mit statischem Text. Ein einzelner String steht zwischen zwei beliebigen Komponenten mit Ausnahme von Textkomponenten und beinhaltet auch Whitespaces, falls vorhanden.
- Dateneingabelelemente sind Eingabefelder, mehrzeilige Eingabefelder, Auswahllisten, Kontrollkästchen, Optionsfelder und Schaltflächen. Ihre Eigenschaften beziehen sich sowohl auf das Dateneingabeelement als auch auf dessen Inhalt, falls vorhanden.
- Eine Tabellenkomponente bezieht sich auf die Tabellenstruktur im Design. Beachten Sie, dass sie Unterkomponenten enthält, die als eigenständige Komponenten betrachtet werden. Die Unterkomponenten sind: Zeile, Spalte, Kopfzeile und Fußzeile.
- Bei einer Absatzkomponente handelt es sich um jedes vordefinierte Format.

Die folgende Tabelle enthält Beschreibungen der einzelnen Eigenschaftsgruppen.

Eigenschafts-gruppe	Beschreibung
XPath	Diese Eigenschaften sind aktiviert, wenn eine automatische Berechnung ausgewählt wurde. Mit der Eigenschaft `Werteformatierung` wird die Formatierung[333] einer automatischen Berechnung, bei der es sch um einen numerischen oder einen Datums-Datentyp handelt, festgelegt. Mit der Eigenschaft `XPath` wird der XPath-Ausdruck festgelegt, der für die automatische Berechnung[257] verwendet wird.
Allgemein	Die Eigenschaftsgruppe *Allgemein* steht für alle Komponententypen mit Ausnahme der Komponententypen "Vorlage" und "XPath" zur Verfügung. Sie enthält die folgenden Eigenschaften: `class` (einen Klassennamem), `dir` (Schreibrichtung), `id` (eine eindeutige ID), `lang` (die Sprache) und `title` (einen Namen). Werte für diese Eigenschaften können mittels XPath-Ausdrücken ausgewählt werden.
Dateneingabe-element	Definiert den Wertebereich von Auswahllisten, Kontrollkästchen und Optionsfeldern. Beachten Sie, dass diese Eigenschaftsgruppe nicht für Eingabefelder und Schaltflächen gilt.
Ereignis	Enthält Eigenschaften, über die Sie für die folgenden clientseitigen HTML-Ereignisse JavaScript Funktionen[403] definieren können: `onclick`, `ondblclick`, `onkeydown`, `onkeypressed`, `onkeyup`, `onmousedown`, `onmousemove`, `onmouseout`, `onmouseover`, `onmouseup`.
HTML	Steht für die folgenden Komponententypen zur Verfügung: Dateneingabeelemente[158]; Bild[152]; Link[320]; Tabelle[126]; Absätze[328]. Beachten Sie, dass es verschiedene Arten von Dateneingabeelementen[158] und Absätzen[328] gibt und dass Tabellen[126] Unterkomponenten haben. Diese Eigenschaften sind HTML-Eigenschaften, die bei den entsprechenden HTML-Elementen (`img`, `table`, `p`, `div`, usw.) eingestellt werden können. Die verfügbaren Eigenschaften variieren daher je nach ausgewählter Komponente. Werte für diese Eigenschaften können mittels XPath-Ausdrücken ausgewählt werden.

Des Weiteren gibt es komponentenspezifische Eigenschaften für Bilder[152], Links[322], Absätze und andere vordefinierte Formate[113] und Bedingungsverzweigungen[265]. Diese Eigenschaften werden in den entsprechenden Abschnitten beschrieben.

Einstellen von Eigenschaftswerten

Eigenschaftswerte können je nach Eigenschaft auf eine, zwei oder drei Arten eingegeben werden.

- durch Eingabe direkt in die Spalte "Wert". Wählen Sie dazu eine Eigenschaft aus, doppelklicken Sie in ihre Wertespalte. Geben Sie den Wert über die Tastatur ein und drücken Sie die **Eingabetaste** oder klicken Sie an eine beliebige Stelle auf der Benutzeroberfläche.
- durch Auswahl eines Werts aus der Dropdown-Liste im Auswahllistenfeld für diese Eigenschaft. Klicken Sie auf den Abwärtspfeil der Auswahlliste, um die Liste der Optionen mit den Eigenschaftswerten auszuklappen.
- über die Bearbeiten Schaltfläche rechts von der Wertespalte für diese Eigenschaft. Wenn Sie auf diese Schaltfläche klicken, erscheint ein Dialogfeld zu dieser Eigenschaft.

© 2019-2025 Altova GmbH

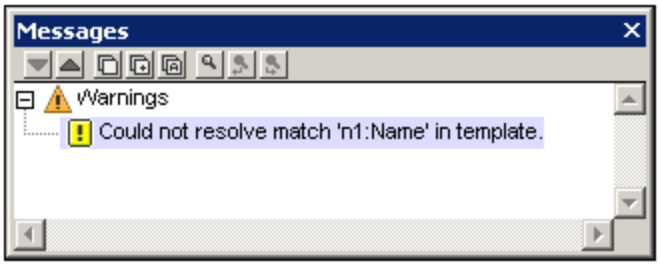
Bei einigen Eigenschaften in den Eigenschaftsgruppen "Allgemein" und "HTML" können die Werte einer Eigenschaft mittels XPath-Ausdrücken bereitgestellt werden. Das XPath-Symbol dient zum Aktivieren und Deaktivieren des XPath-Ausdrucks als Quelle für die Eigenschaftswerte. Wenn eine Eigenschaft ausgewählt ist, kann ein XPath-Ausdruck für diese Eigenschaft eingegeben werden und der Rückgabewert des XPath-Ausdrucks wird als Wert für diese Eigenschaft verwendet, wenn das XPath-Symbol aktiviert ist. Auf diese Art kann der Wert eines Node in einem XML-Dokument in Laufzeit als Wert einer Eigenschaft zurückgegeben werden. Wenn das XPath-Symbol deaktiviert ist, kann ein statischer Wert als Wert der Eigenschaft eingegeben werden. *Siehe auch* *Stileigenschaften über XPath*[353].

Ändern oder Löschen eines Eigenschaftswerts

Zum Ändern eines Eigenschaftswerts verwenden Sie eine der im vorigen Absatz Einstellen von Eigenschaftswerten[52] beschriebenen verfügbaren Methoden. Um einen Eigenschaftswert zu löschen, wählen Sie die Eigenschaft aus und klicken Sie in der Symbolleiste der Seitenleiste "Eigenschaften" auf das Symbol "Zurücksetzen" .

2.2.7 Meldungen

Wenn Sie StyleVision zum ersten Mal starten, wird unterhalb des Hauptfensters die Seitenleiste "Meldungen" (*Abbildung unten*) angezeigt. Um die Seitenleiste "Meldungen" ein- und auszublenden, klicken Sie auf **Ansicht | Meldungen**.

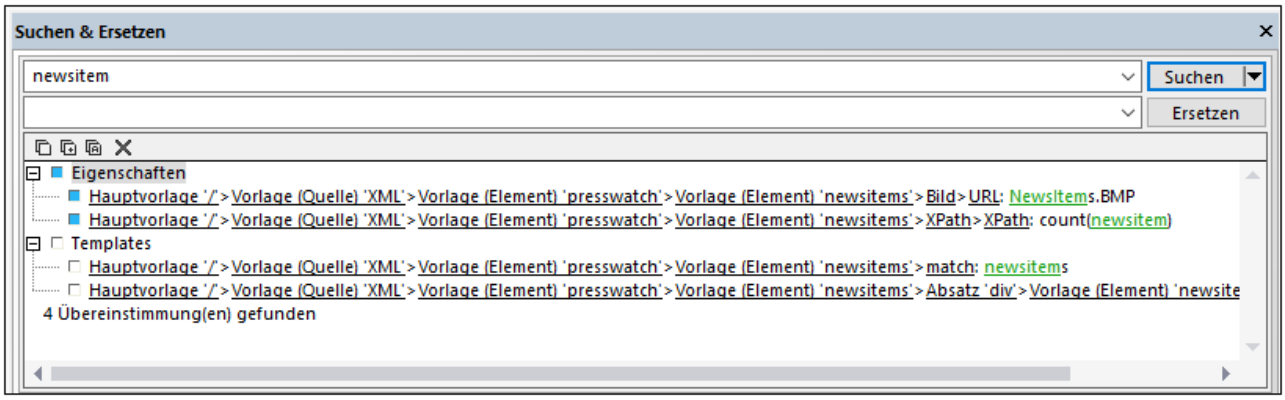


In der Seitenleiste "Meldungen" werden Warnmeldungen in der Design-Ansicht und der Authentic-Ansicht angezeigt. Die Warnmeldungen in der Design-Ansicht beziehen sich auf verschiedene Aspekte im Zusammenhang mit dem SPS-Dokument, von einer fehlenden XML-Arbeitsdatei bis zu Fehlern in der Design-Struktur. In der Authentic-Ansicht beziehen sich die Warnmeldungen auf die Gültigkeit der eingegebenen XML-Daten, d.h. ob diese gemäß dem zugrunde liegenden Schema oder anderen zusätzlichen Validierungskriterien gültig sind.

2.2.8 Suchen und Ersetzen

Über die Seitenleiste "Suchen und Ersetzen" (*Abbildung unten*) können Sie in der Design-Ansicht Text suchen und ersetzen. Klicken Sie auf den Nach unten-Pfeil der Schaltfläche **Suchen** (*in der Abbildung unten blau markiert*), um die gewünschten Suchoptionen auszuwählen. Sie können in Text, Stilen, Eigenschaften, Variablen, Vorlagen und XPath-Ausdrücken nach Strings suchen, die Sie entweder direkt in das Feld *Suchen* eingeben, oder mit Hilfe von Regular Expressions definieren. Alle in der Seitenleiste durchgeführten Suchvorgänge werden auf die Design-Ansicht angewendet. Mit den Menübefehlen **Bearbeiten | Suchen**[493]

und **Bearbeiten | Ersetzen**[493] setzen Sie den Fokus auf diese Seitenleiste und platzieren den Cursor in das Feld *Suchen*, sodass Sie mit einer Suche in der Design-Ansicht fortfahren können. Die Ergebnisse der Suche werden in der Seitenleiste angezeigt. Wenn Sie auf ein Ergebnis klicken, gelangen Sie zur entsprechenden Stelle im Design. Um die Seitenleiste "Suchen & Ersetzen" ein- und auszublenden, klicken Sie auf **Ansicht | Suchen & Ersetzen**.



Informationen zum Suchen in anderen Ansichten (JavaScript Editor und XSLT Stylesheets) finden Sie in der Menüreferenz im Kapitel Suchen, Weitersuchen, Ersetzen[493].

Suchen

Geben Sie den Suchbegriff in das Feld *Suchen* ein. Klicken Sie anschließend auf den Nach unten-Pfeil der Schaltfläche **Suchen** (*in der Abbildung oben blau markiert*), um die gewünschten Suchoptionen auszuwählen. Es stehen die folgenden Optionen zur Verfügung:

- *Wo soll gesucht werden?* Der entsprechende *<Komponente> inkludieren*-Eintrag sollte aktiviert werden, damit die jeweilige Komponente in die Suche miteinbezogen wird.
- *Groß-/Kleinschreibung berücksichtigen, Suche nach ganzem Wort:* Dies sind Ein/Aus-Optionen.
- *Regular Expressions:* Ihre Eingabe wird als Regular Expression behandelt. Eine Beschreibung zur Verwendung von Regular Expressions finden Sie in der Menüreferenz im Kapitel Suchen, Weitersuchen, Ersetzen[495].

Ergebnisse

Die Ergebnisse werden geordnet nach der Komponente, in der der gesuchte String vorkommt, in Gruppen organisiert (*siehe Abbildung oben*). Jedes Ergebnis wird in Form eines hierarchischen Pfades angezeigt. Sie können auf jeden Link in der Hierarchie klicken, um in der Design-Ansicht zu diesem Treffer zu gelangen.

Der Ergebnisbereich hat eine Symbolleiste mit Schaltflächen für die folgenden Befehle (von links nach rechts): Eintrag oder Gruppe von Einträgen in die Zwischenablage kopieren, Ergebnisbereich leeren.

Ersetzen

Wenn die Ergebnisse angezeigt werden, können Sie einen oder mehrere dieser Einträge für die Ersetzung auswählen. Der/die ausgewählte(n) Eintrag/Einträge wird durch ein blaues Kästchen markiert (*siehe Abbildung oben*), woraufhin die **Ersetzen**-Schaltfläche aktiv wird. Geben Sie den Ersetzungsstring in das Textfeld *Ersetzen* ein und klicken Sie auf **Ersetzen**. Daraufhin wird die Ersetzung durchgeführt und das blaue Kästchen wird zu einem grünen.

© 2019-2025 Altova GmbH

3 Schnellstart-Tutorial

Dieses Tutorial ist eine kurze Anleitung zu den wichtigsten Schritten beim Erstellen eines effizienten SPS. Zu Beginn finden Sie einen Abschnitt zum Erstellen und Einrichten des SPS. Sie lernen Inhalt in das SPS einzufügen, die Komponenten des SPS zu formatieren und die Verwendung von zwei wichtigen SPS Features: automatischen Berechnungen und Bedingungen. Im Zuge des Tutorials erfahren Sie, wie Sie Ihre Ausgabedokumente strukturieren und die verschiedensten Gliederungs- und Darstellungsfunktionen effizient anwenden können.

Benötigte Dateien

Die für dieses Schnellstart-Tutorial benötigten Dateien befinden sich im Ordner (Eigene) Dokumente[25] `\C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2025\StyleVisionExamples\Tutorial\QuickStart`:

- `QuickStart.xsd`, die XML-Schemadatei, auf der das SPS basiert.
- `QuickStart.xml`, die XML-Arbeitsdatei, aus der die Daten stammen, die in den Ausgabevorschauen angezeigt werden.
- `QuickStart.sps`, die fertige SPS Datei; Sie können die von Ihnen erstellte SPS Datei mit dieser Datei vergleichen.
- `QuickStart.css`, das externe CSS Stylesheet, das in diesem Tutorial verwendet wird.
- `NewsItems.BMP`, eine Bilddatei, die im SPS verwendet wird.

Durcharbeiten des Tutorials

Am besten fangen Sie mit dem Beginn des Tutorials an und arbeiten die einzelnen Abschnitte der Reihe nach durch. Sie sollten außerdem vor Beginn des Tutorials die XSD- und die XML-Datei öffnen und sich deren Struktur und Inhalt ansehen. Lassen Sie die XSL- und die XML-Datei geöffnet, während Sie das Tutorial durchgehen, um nachsehen zu können. Speichern Sie Ihr SPS-Dokument unter einem anderen Namen als `QuickStart.sps`, (z.B. unter `MyQuickStart.sps`) um die mitgelieferte SPS Datei nicht zu überschreiben. Und vergessen Sie nicht, die Datei nach jedem Abschnitt zu speichern.

3.1 Erstellen und Einrichten eines neuen SPS

In diesem Abschnitt erfahren Sie Folgendes:

- Wie man ein neues SPS Dokument erstellt 56
- Wie man eine Schemaquelle für das SPS hinzufügt 58
- Wie man die XSLT-Version des SPS auswählt 59
- Wie man die XML-Arbeitsdatei zuweist 59
- Wie man die Ausgabekodierung definiert 59
- Wie man das SPS Dokument speichert 59

Dateien in diesem Abschnitt

Die in diesem Abschnitt verwendeten Dateien befinden sich im Ordner (Eigene) Dokumente 25, `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2025\StyleVisionExamples\Tutorial\QuickStart`:

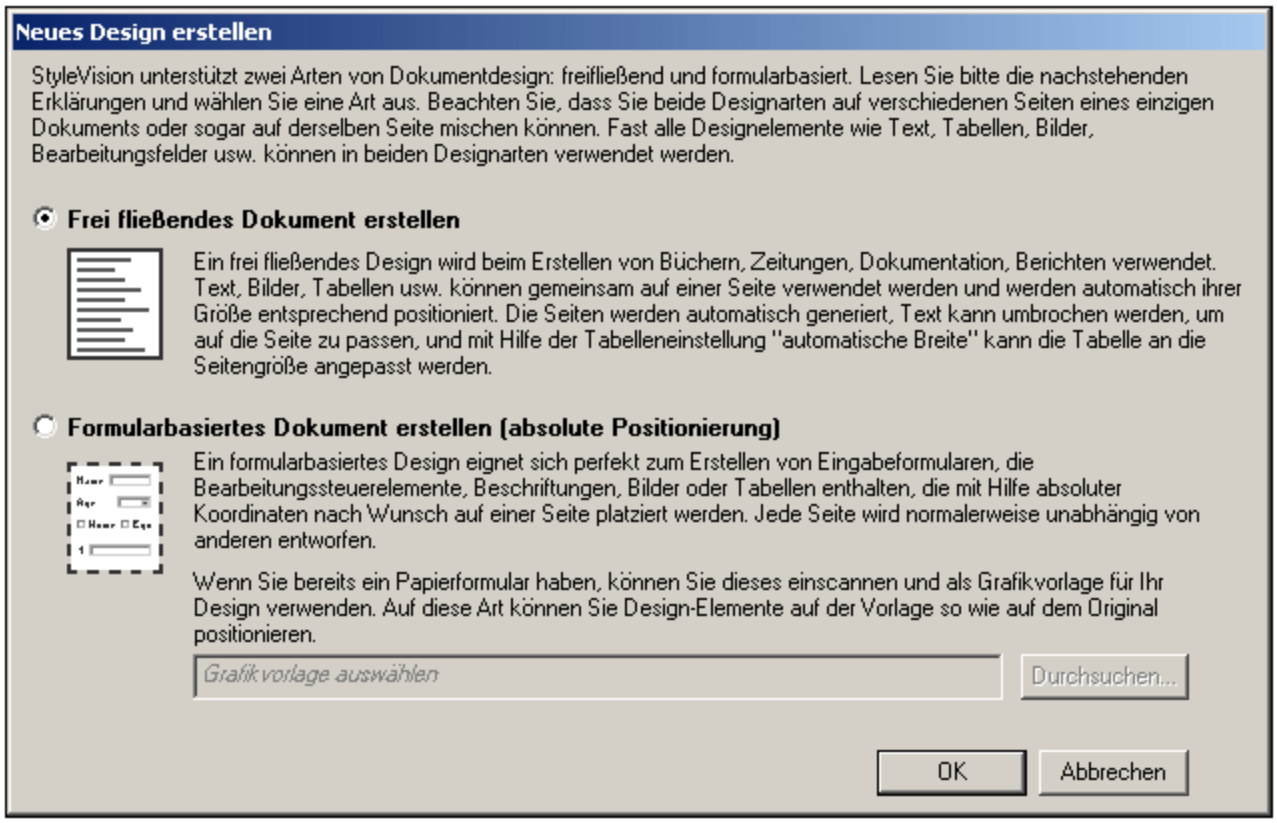
- `QuickStart.xsd`: die XML-Schema-Datei, auf der das SPS basiert.
- `QuickStart.xml`, die XML-Arbeitsdatei, aus der die Daten stammen, die in den Ausgabevorschauen angezeigt werden.
- `QuickStart.sps`, die fertige SPS Datei; Sie können die von Ihnen erstellte SPS Datei mit dieser Datei vergleichen.

Erstellen eines neuen SPS Dokuments

Erstellen Sie ein neues SPS-Dokument. Klicken Sie dazu auf **Datei | Neu | Neu (leer)** 468 oder wählen Sie in der Applikationssymbolleiste über die Dropdown-Liste des Symbols "Neu" 466 den Befehl **Neu (Leer)** aus. Daraufhin wird das Dialogfeld "Neues Design erstellen" angezeigt.

Im Dialogfeld "Neues Design erstellen" (*Abbildung unten*) werden Sie aufgefordert entweder: (i) ein frei fließendes Dokument oder (ii) ein formularbasiertes Dokument zu erstellen (in dem die Komponenten wie in einem Layout-Programm absolut positioniert werden).

© 2019-2025 Altova GmbH

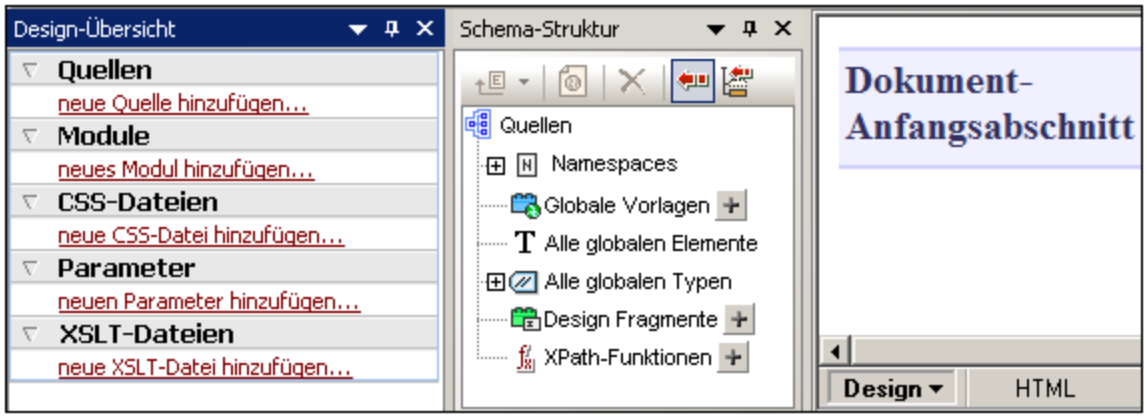


In einem frei fließenden Dokument ist das Dokument vom Layout her so angelegt, dass es den Maßen des Ausgabemediums entspricht (Papier oder Bildschirm). Elemente im Dokumentinhalt können nur relativ zueinander platziert werden, nicht aber an einer absoluten Position. Diese Art von Design eignet sich für Dokumente wie z.B. Berichte, Artikel und Bücher.

Ein einem formularbasierten Dokument wird ein einziger Layout Container[171] erstellt, in den Design-Komponenten absolut positioniert werden können. Die Dimensionen des Layout Containers sind benutzerdefiniert und Layout-Felder können innerhalb des Layout Containers absolut positioniert werden. Dokumentinhalt kann in einzelne Layout-Felder platziert werden. Wenn sich das Design Ihres SPS an einem spezifischen formularbasierten Design orientierten soll, können Sie ein Originalformular als Grafikvorlage[171] hinterlegen. Die Grafikvorlage kann dann als Hintergrundbild in den Layout Container inkludiert werden und dient als Hilfe für das Design Ihres Formulars; sie ist im Ausgabedokument nicht enthalten.

Sie erstellen hier ein frei fließendes Dokument, wählen Sie daher diese Option durch Klicken auf das Optionsfeld *Frei fließendes Dokument erstellen* aus und klicken Sie anschließend auf **OK**.

Es wird ein neues Dokument mit dem Namen `SPS1.sps` erstellt und in der Design-Ansicht[29] angezeigt (*Abbildung unten*).

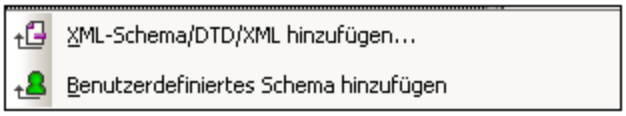


In der Design-Ansicht[29] wird eine leere Vorlage angezeigt. Die Seitenleisten Design-Übersicht[36] und Schema-Struktur[38] enthalten keine Schemaeinträge.

Hinzufügen einer Schemaquelle

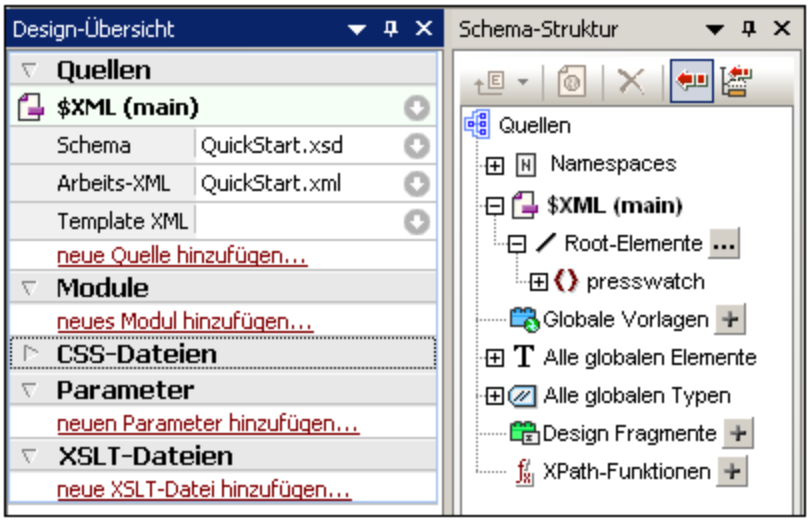
Für dieses SPS wird das Schema `QuickStart.xsd` verwendet. Um dieses Schema als Schemaquelle hinzuzufügen, gehen Sie folgendermaßen vor:

1. Klicken Sie in der Seitenleiste "Design-Übersicht" unter der Überschrift "Quellen" auf den Befehl **Neue Quelle hinzufügen** (*siehe Abbildung unten*) und wählen Sie den Befehl **XML-Schema/DTD/XML hinzufügen**.



2. Suchen Sie im Dialogfeld "Öffnen", das daraufhin angezeigt wird, im Ordner (Eigene) Dokumente[56] nach der Datei `QuickStart.xsd` und klicken Sie auf **Öffnen**.
3. Sie werden aufgefordert, eine XML-Arbeitsdatei auszuwählen. Aktivieren Sie die Option, um eine Datei aus dem Dateisystem auszuwählen, suchen Sie anschließend im Verzeichnis (Eigene) Dokumente[56] nach der Datei `QuickStart.xml` (siehe oben) und klicken Sie auf **Öffnen**. Das Schema wird als Schemaquelle zu den Seitenleisten "Design-Übersicht" und "Schema-Struktur" hinzugefügt (*Abbildung unten*). Außerdem wird dem Schema die von Ihnen ausgewählte XML-Arbeitsdatei zugewiesen.

© 2019-2025 Altova GmbH



Beachten Sie die folgenden Punkte: (i) Unter dem Schemaquellen-Eintrag `$XML` werden das Schema und die XML-Arbeitsdatei[23] aufgelistet; (ii) In der Baumstruktur des Root-Elements werden die Root-Elemente (Dokumentelemente)[22] angezeigt, die Sie aus den im Schema definierten globalen Elementen[22] auswählen. Das Element `presswatch` ist standardmäßig ausgewählt, das es das einzige globale Element[22] im Schema ist, das sich eindeutig auf der obersten Ebene der im Schema definierten Hierarchie befindet. (iii) Alle globalen Elemente[22] im Schema sind unter dem Eintrag Alle globalen Elemente[38] aufgelistet.

Auswahl der XSLT-Version

Für dieses SPS wird Version XSLT 2.0 verwendet. Um die XSLT Version über die Symbolleiste festzulegen, klicken Sie auf das Symbol XSLT 20.

Zuweisen oder Ändern der XML-Arbeitsdatei

Als vorher ein XML-Schema zum SPS hinzugefügt wurde, wurde dem Schema auch eine XML-Arbeitsdatei[23] zugewiesen. Eine XML-Arbeitsdatei liefert dem SPS eine XML-Datenquelle für die Verarbeitung. Um einem Schema eine XML-Arbeitsdatei[23] zuzuweisen, die Zuweisung aufzuheben oder zu ändern, klicken sie in der Seitenleiste "Design-Übersicht" mit der rechten Maustaste auf eine beliebige Stelle in der Zeile der gewünschten XML-Arbeitsdatei (oder klicken Sie auf das Kontextmenüsymbol rechts davon) und wählen Sie im Kontextmenü den gewünschten Befehl aus. Die XML-Arbeitsdatei[23] wurde nun zugewiesen und der Dateiname wird in die Zeile für die XML-Arbeitsdatei eingetragen. Stellen Sie, bevor Sie fortfahren sicher, dass Sie die Datei `QuickStart.xml`, die sich im Verzeichnis (Eigene) Dokumente[56] befindet, korrekt als XML-Arbeitsdatei zugewiesen haben.

Definieren der Kodierung für die Ausgabe

Wählen Sie im Dialogfeld "Optionen"(**Extras | Optionen**[552]) auf dem Register "Standardkodierung" für die HTML-Kodierung die Option Unicode UTF-8.

Speichern des SPS Dokuments

Nachdem Sie das SPS wie oben beschrieben eingerichtet haben, speichern Sie es als `MyQuickStart.sps` im Ordner (Eigene) Dokumente[56]. Klicken Sie dazu auf den Menübefehl **Datei | Design speichern**[480] oder

drücken Sie **Strg+S** und wählen Sie daraufhin im Dialogfeld "Design speichern" die Option *Als SPS-Datei speichern* und geben Sie den Namen der SPS-Datei ein.

© 2019-2025 Altova GmbH

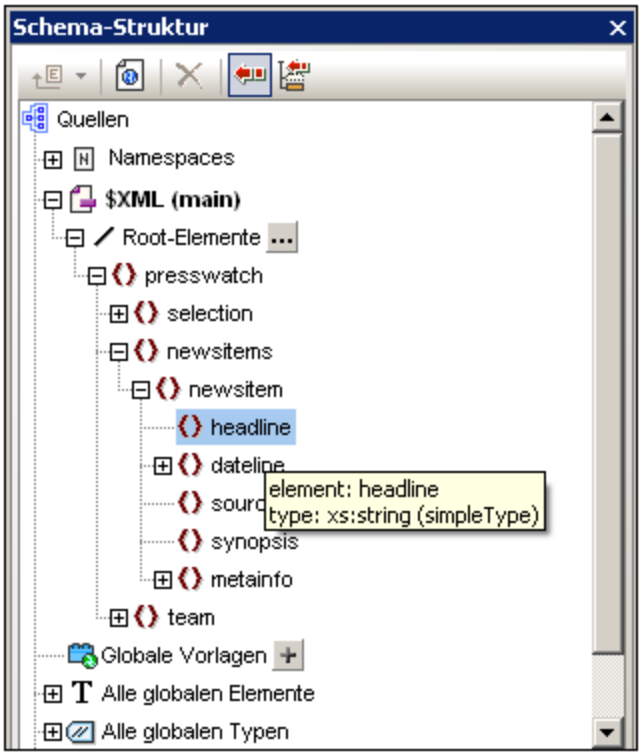
3.2 Einfügen von dynamischen Inhalten (aus der XML-Quelle)

In diesem Abschnitt wird eine Methode vorgestellt, mit der Sie Daten aus Nodes im XML-Dokument einfügen können. Sie erfahren hier, wie man Element- und Attribut-Nodes aus der Schema-Struktur in das Design zieht und diese Nodes als Inhalt definiert. Wenn ein Node als Inhalt definiert wird, werden die Daten darin als String ausgegeben, wobei die Child-Text-Nodes des Elements und die Text-Nodes aller untergeordneten Elemente miteinander verkettet werden.

Einfügen von Elementinhalt

Gehen Sie in Ihrem SPS folgendermaßen vor:

1. Erweitern Sie die Schemastruktur in der Seitenleiste "Schema-Struktur" 38 bis zu den Child-Elementen des Elements `newsitem` (*Abbildung unten*).



2. Wählen Sie das Element `headline`, (Beachten Sie, dass der Datentyp des Elements in einem Popup-Fenster angezeigt wird, wenn Sie die Maus darüber platzieren) und ziehen Sie es in die Design-Ansicht 33. Wenn sich der Pfeilcursor in einen Einfügepunkt verwandelt, lassen Sie die Maustaste los und legen Sie das Element in der Hauptvorlage ab.
3. Wählen Sie im Kontextmenü den Befehl **Inhalt erstellen**. Die Start- und Endtags des Elements `headline` werden an der Stelle eingefügt, an die Sie das Element `headline` gezogen haben. Sie enthalten den Platzhalter `Inhalt`. Die `headline` Tags befinden sich innerhalb der Start- und End-Tags der übergeordneten Elemente von `headline` (*Abbildung unten*).
4. Platzieren Sie die Elemente wie in der Abbildung unten gezeigt auf unterschiedlichen Zeilen, indem Sie die **Eingabetaste** drücken.

Klicken Sie auf das Register "HTML", um eine Vorschau der HTML-Ausgabe[31] zu sehen (*Abbildung unten*). In der HTML-Vorschau sehen Sie den Inhalt der `headline` Child-Elemente von `newsitem`, jedes als Textstring.

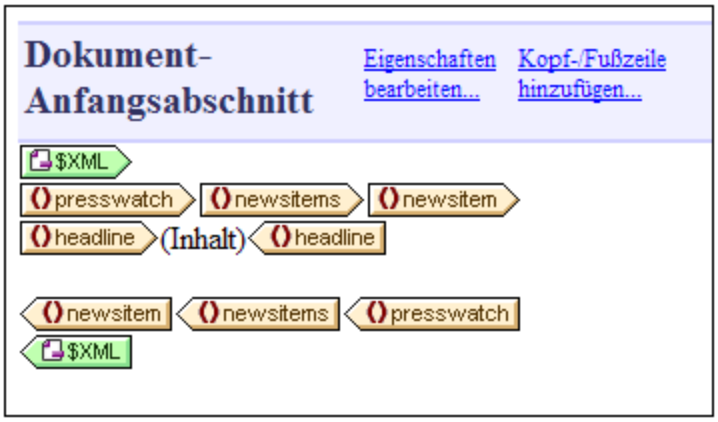
Anmerkung: Sie können den Inhalt eines Node auch folgendermaßen erstellen: (i) durch Klicken auf die Schaltfläche "Inhalt einfügen" in der Symbolleiste "Design-Elemente einfügen"[462], (ii) durch Klicken auf die gewünschte Stelle im Design, (iii) durch Auswahl des gewünschten Node in der Schema-Selektorstruktur, die angezeigt wird.

Einfügen von Attributinhalt

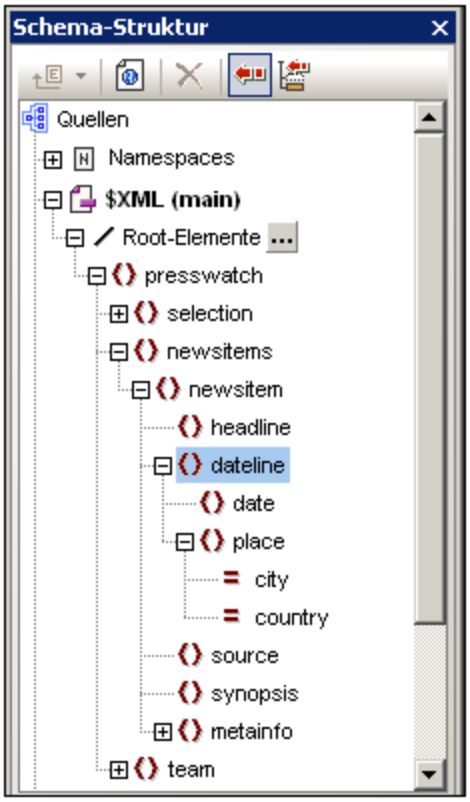
Beim Einfügen eines Elements in das Design in Form von Inhalt wird der Inhalt seiner Attribute nicht automatisch eingefügt. Sie müssen den Attribut-Node explizit in das Design ziehen, damit der Attributwert ausgegeben wird. Gehen Sie ihn Ihrem SPS folgendermaßen vor:

1. Platzieren Sie den Cursor hinter den End-Tag des Elements `headline` und drücken Sie die **Eingabetaste**. Dadurch wird eine leere Zeile erstellt (A*bbildung unten*).

 © 2019-2025 Altova GmbH



2. Erweitern Sie in der Seitenleiste "Schema-Struktur" das Element `dateline` (*Abbildung unten*).



Beachten Sie, dass das Element `dateline` zwei Child-Elemente hat, nämlich `date` und `place` und dass das Element `place` zwei Attribute hat: `city` und `country`.

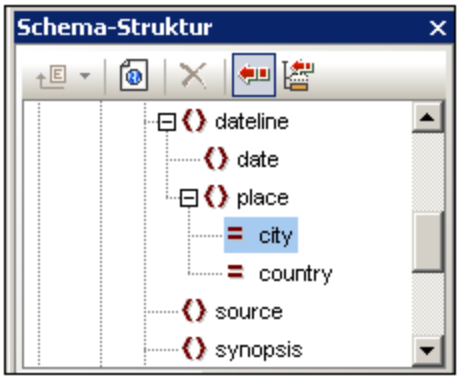
3. Ziehen Sie das Element `dateline` in das Design an den Beginn der neu erstellten leeren Zeile (*Abbildung unten*) und wählen Sie im Kontextmenü den Eintrag "Inhalt erstellen".

4. Wechseln Sie in die HTML-Vorschau[31] und sehen Sie sich die Ausgabe von `dateline` genau an (*Abbildung unten*).

Beachten Sie: Während der Inhalt der `date` Child-Elemente von `dateline` Elementen ausgegeben wurde, wurde kein Inhalt für die `place` Children von `dateline` ausgegeben. Der Grund dafür ist, dass die `place`-Daten in den Attributen des Elements `place`, nämlich `city` und `country`, enthalten sind und dass Attributinhalt nicht ausgegeben wird, wenn das übergeordnete Element des Attributs verarbeitet wird.

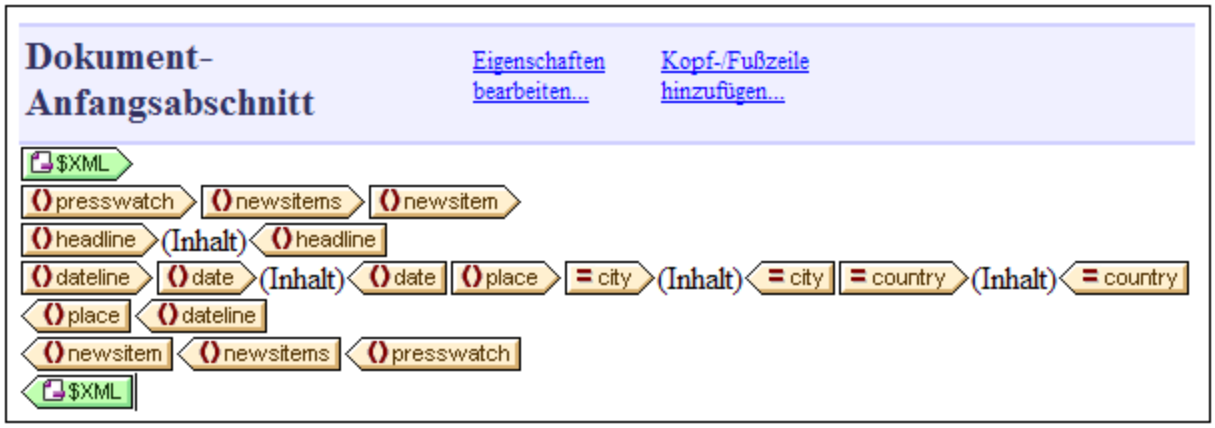
5. . Ziehen Sie das `date` Element aus der Seitenleiste "Schema-Struktur"[38] zwischen den Start- und End-Tag des `dateline` Elements (und erstellen Sie es als Inhalt).
6. Wählen Sie in der Seitenleiste "Schema-Struktur"[38] das `city` Attribut des `dateline/place` Elements (*Abbildung unten*) aus.

© 2019-2025 Altova GmbH



8. Ziehen Sie den `@city` Attribut-Node in die Design-Ansicht[29] und platzieren Sie ihn (Als Inhalt erstellen) genau hinter den End-Tag des `date` Elements.
9. Ziehen Sie den `@country` Attribut-Node in die Design-Ansicht[29] und platzieren Sie ihn (Als Inhalt erstellen) genau hinter den End-Tag des `@city` Attributs.

Das SPS Design sollte anschließend etwa folgendermaßen aussehen:

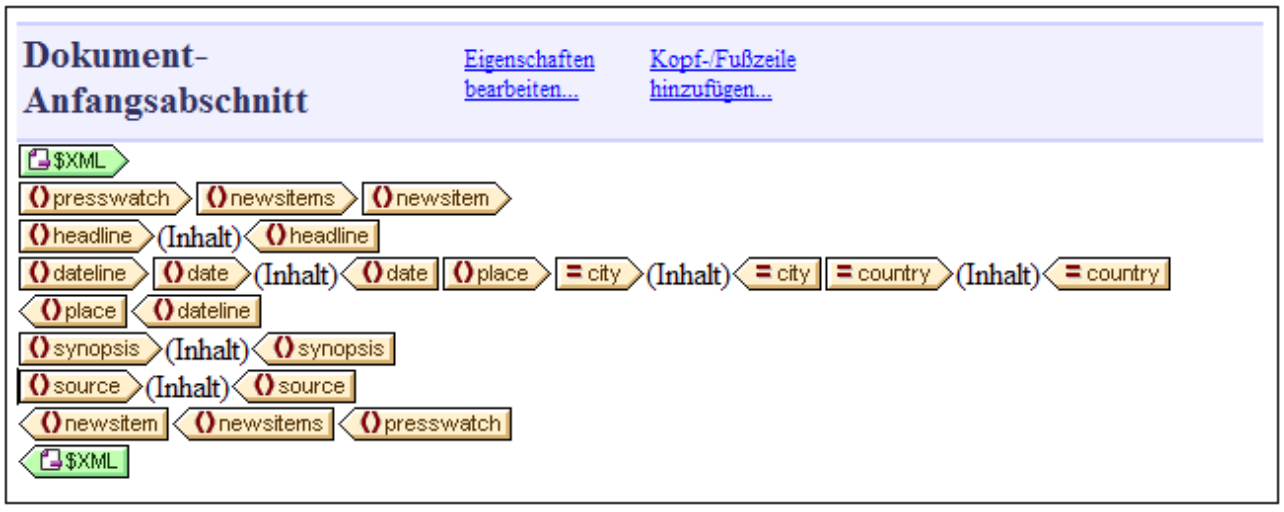


Die HTML-Vorschau[31] sollte folgendermaßen aussehen:

Beachten Sie, das die Werte der Attribute `@city` und `@country` nun in der Ausgabe enthalten sind.

Hinzufügen von weiterem dynamischem Inhalt

Mit der oben beschriebenen Methode können Sie überall im Design den Inhalt von Elementen und Attributen aus der XML-Datenquelle einfügen. Fügen Sie zum Abschluss die Elemente `synopsis` und `source` zum Design hinzu, sodass das Design folgendermaßen aussieht:



Beachten Sie: Das Element `synopsis` wurde vor das Element `source` platziert. Dies entspricht nicht der Reihenfolge der Elemente im Schema. Überprüfen Sie nach dem Hinzufügen der Elemente `synopsis` und `source` zum Design die HTML-Vorschau[31], um die Ausgabe zu sehen. Dies ist ein wichtiger Punkt: Die Reihenfolge, in der Nodes in der Hauptvorlage[23] angeordnet werden, gibt die Struktur des Ausgabedokuments[230] vor (Nähere Informationen zum Strukturieren des Ausgabedokuments finden Sie im Abschnitt Vorlagen und Design-Fragmente[230]).

 © 2019-2025 Altova GmbH

Beachten Sie bitte auch, in welcher Form ein Node in das Design eingefügt wird. In der HTML-Vorschau 31 sehen Sie, dass alle bisher in das Design eingefügten Nodes als Textstrings an die Ausgabe gesendet wurden. Alternativ dazu kann ein Node auch in einer anderen Form ausgegeben werden, z.B. in Form einer Tabelle oder als Auswahlliste. In diesem Abschnitt haben Sie festgelegt, dass alle Nodes als Textstrings ausgegeben werden sollen, indem Sie alle Nodes als `(Inhalt)` erstellt haben. Im Abschnitt Verwendung von Bedingungen 84 erfahren Sie, wie man einen Node als Auswahlliste erstellt und im Abschnitt Verwendung von globalen Vorlagen und "Restlicher Inhalt" 92 wird erklärt, wie ein Node als (dynamische) Tabelle erstellt wird.

Stellen Sie sicher, dass Sie die Datei gespeichert haben, bevor Sie mit dem nächsten Abschnitt fortfahren.

3.3 Einfügen von statischen Inhalten

Statischer Inhalt ist Inhalt, den Sie direkt in das Design eingeben oder einfügen - im Gegensatz zu Inhalt, der aus der XML-Quelle stammt. Sie können in das SPS Design eine ganze Reihe von statischen Komponenten einfügen. In diesem Teil des Tutorials erfahren Sie, wie man die folgenden statischen Komponenten einfügt:

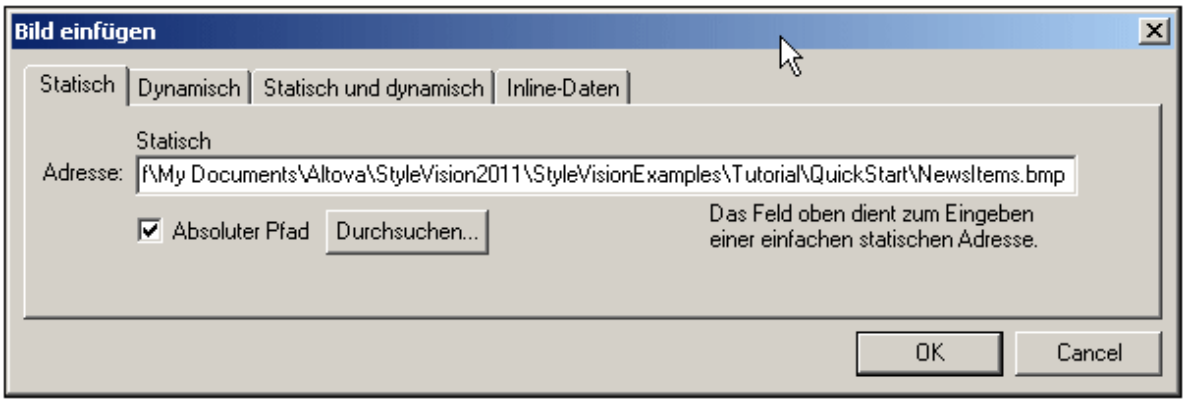
- ein Bild
- eine horizontale Linie
- Text

Einfügen eines statischen Bilds

Das statische Bild, das wir einfügen werden, befindet sich im Verzeichnis (Eigene) Dateien: `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2025\StyleVisionExamples\Tutorial\QuickStart\NewsItems.BMP`. Es wird als Dokumentüberschrift verwendet. Um dieses Bild ganz oben im Dokument einzufügen, gehen Sie folgendermaßen vor:

1. Platzieren Sie den Cursor zwischen die Start-Tags von `newsitems` und `newsitem` (*siehe Screenshot unten*).

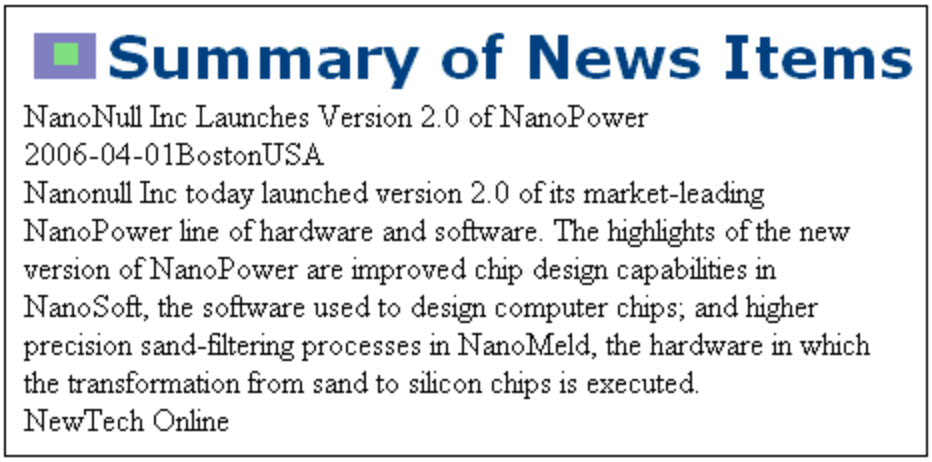
   Beachten Sie, dass sich der Cursor innerhalb des Elements `newsitems` jedoch außerhalb des Elements `newsitem` befindet. Das Bild wird daher einmal in die Ausgabe eingefügt, und zwar am Beginn der Verarbeitung des `newsitems` Elements (da im Schema nur ein `newsitems` Element definiert ist).
2. Rechtsklicken Sie und wählen Sie den Befehl **Einfügen | Bild**. Daraufhin wird das Dialogfeld "Bild einfügen" eingefügt (*Screenshot unten*).



 © 2019-2025 Altova GmbH

3. Aktivieren Sie auf dem Register "Statisch" das Kontrollkästchen "Absoluter Pfad", suchen Sie die Datei `NewsItems.BMP` und wählen Sie sie aus.
4. Bestätigen Sie den Vorgang durch Klicken auf **OK**.

Die HTML-Vorschau sollte in etwa so aussehen:



Einfügen von horizontalen Linien

Die erste horizontale Linie soll nun zwischen der Dokumentüberschrift und dem Textkörper eingefügt werden. Gehen Sie dazu folgendermaßen vor:

1. Platzieren Sie den Cursor direkt hinter das gerade eingefügte statische Bild.
2. Klicken Sie auf die rechte Maustaste und wählen Sie den Befehl **Einfügen | Horizontale Linie** 511. Es wird eine horizontale Linie eingefügt.

Stellen Sie für die Linie die folgenden Eigenschaften ein:

1. Wählen Sie, während die Linie in der Design-Ansicht 29 ausgewählt ist, in der Seitenleiste "Eigenschaften" 49 (in der Spalte "Eigenschaften für) die Komponente *Linie* und dann die HTML-Eigenschaftsgruppe.
2. Weisen Sie der Linie die Eigenschaften `color` und `size` zu.
3. Wählen Sie, während die Linie in der Design-Ansicht 29 ausgewählt ist, in der Seitenleiste "Stil" 47 die Komponente *Linie* und anschließend die Eigenschaftsgruppe *box* aus. Definieren Sie eine `margin-bottom` Eigenschaft mit `12pt`.
4. Überprüfen Sie die Ausgabe in der HTML-Vorschau 31.

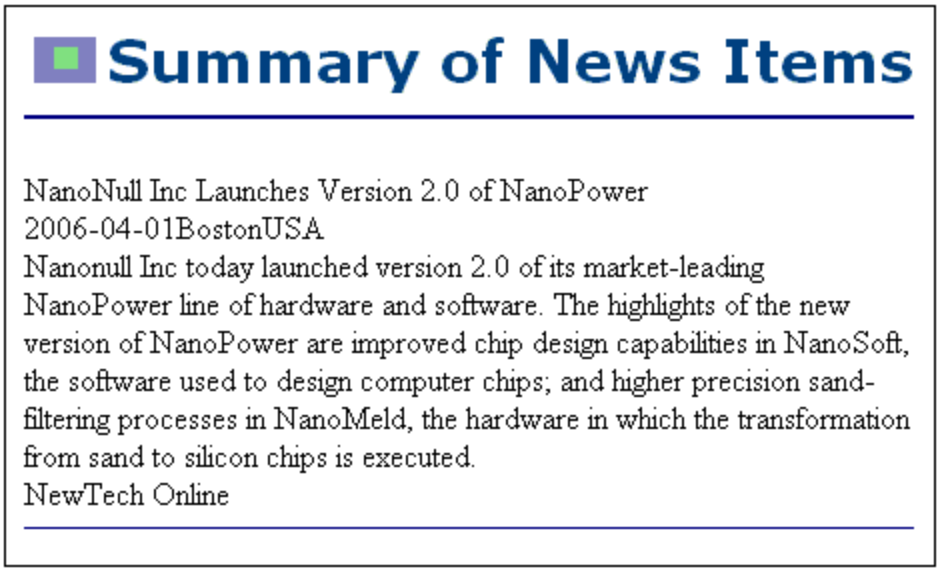
Fügen Sie nun hinter jedem "Newsitem" eine horizontale Linie ein. Damit am Ende jedes `newsitem` Elements eine Linie ausgegeben wird, muss der Cursor genau vor den End-Tag des Elements `newsitem` platziert werden. Sie können die Dicke der Linie ändern. Wählen Sie dazu für die Linieneigenschaft `Größe` eine Zahl ohne Einheit (wählen Sie in der Seitenleiste "Eigenschaften" den Eintrag *line* und definieren Sie z.B. den Wert `3`).

Einfügen von statischem Text

Sie haben zu Ihrem Design bereits statischen Text hinzugefügt (im Abschnitt Einfügen von dynamischem Inhalt (aus der XML-Quelle) 61. Bei Drücken der **Eingabetaste** zum Einfügen neuer Zeilen wurden Leerzeichen (statischer Text) hinzugefügt. In diesem Abschnitt werden Sie nun einige Zeichen statischen Textes in Ihr Design einfügen.

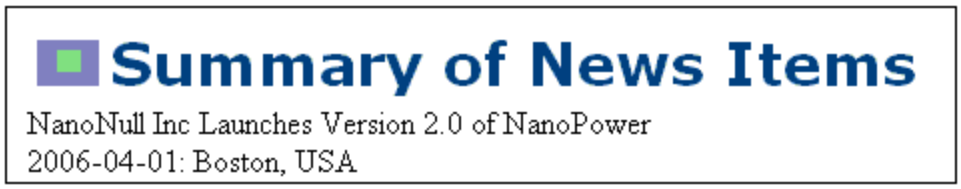
Das SPS, das Sie bisher erstellt haben, erzeugt eine Ausgabe, die ungefähr folgendermaßen aussieht:

© 2019-2025 Altova GmbH



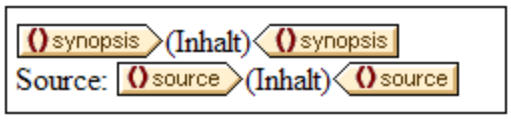
Beachten Sie: In der Ausgabe des Elements `dateline` wird der Inhalt des Elements `date` und der Attribute `place/@city` und `place/@country` zusammen, ohne Leerzeichen dazwischen angezeigt. Sie können die Leerzeichen in Form von statischem Text hinzufügen. Setzen Sie den Cursor im Design hinter das Element `date` und geben Sie einen Doppelpunkt und ein Leerzeichen ein. Geben Sie anschließend ein Komma und ein Leerzeichen hinter dem Attribut `@city` ein (*Abbildung unten*)

Dieser Teil der Ausgabe sieht nun folgendermaßen aus:



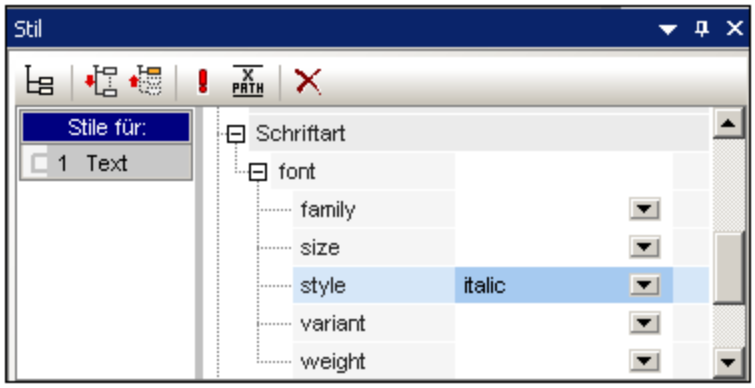
Beachten Sie in der `dateline` Ausgabe den Doppelpunkt, die Leerzeichen und das Komma. Alle diese Textelemente sind statische Textelemente, die direkt in das Design eingefügt wurden.

Nun wollen wir ein weiteres statisches Textelement hinzufügen. Geben Sie im Design genau vor dem Start-Tag des Elements `source` den String "`Source: `" ein (*siehe Abbildung unten*).

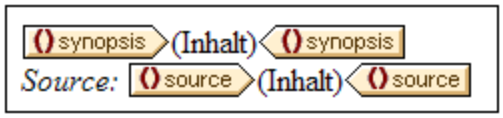


Formatieren von statischem Text

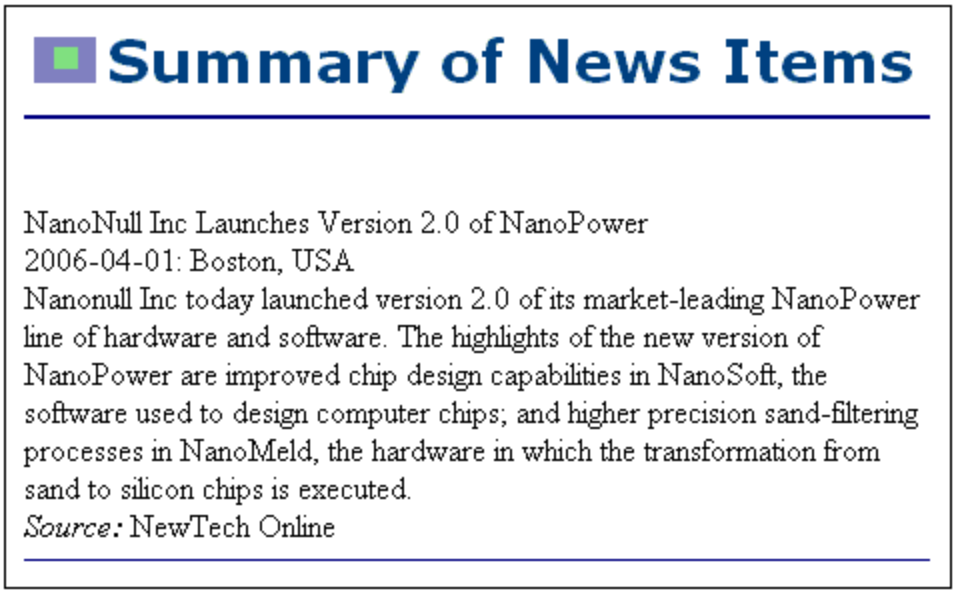
Um statischen Text zu formatieren, markieren Sie den gewünschten Text und definieren Sie lokale Stileigenschaften. Markieren Sie im Design den gerade eingegebenen Text "`Source:`". Beachten Sie, dass in der Seitenleiste "Stil" 47 (*Abbildung unten*) die Komponente *1 Text* ausgewählt ist. Erweitern Sie nun die Eigenschaftsgruppe *Schriftart* wie in der Abbildung unten gezeigt und wählen Sie für die Eigenschaft `font-style` die Option `italic` aus dem Dropdown-Menü aus.



Der statische Text (d.h. der String "`Source:`") erhält daraufhin im Design einen kursiven Stil. Dies sieht folgendermaßen aus:



Die Ausgabe in der HTML-Vorschau sieht folgendermaßen aus:



Summary of News Items

NanoNull Inc Launches Version 2.0 of NanoPower
2006-04-01: Boston, USA
Nanonull Inc today launched version 2.0 of its market-leading NanoPower line of hardware and software. The highlights of the new version of NanoPower are improved chip design capabilities in NanoSoft, the software used to design computer chips; and higher precision sand-filtering processes in NanoMeld, the hardware in which the transformation from sand to silicon chips is executed.
Source: NewTech Online

 © 2019-2025 Altova GmbH

Wenn Sie der Meinung sind, der Abstand zwischen dem Eintrag "Source" und der horizontalen Linie zwischen zwei `newsitem` Elementen sei zu gering, dann fügen Sie im Design zwischen "Source" und der horizontalen Linie durch Drücken der **Eingabetaste** eine Leerzeile ein.

Speichern Sie die Datei anschließend.

In diesem Abschnitt haben Sie gelernt, wie man statischen Inhalt einfügt und formatiert. Im nächsten Abschnitt erfahren Sie mehr darüber, wie man Design-Komponenten anhand von CSS-Grundregeln und Eigenschaften formatieren kann.

3.4 Formatieren des Inhalts

StyleVision bietet eine flexible Methode zum Zuweisen von Stilen[342] zu Formatierungskomponenten im Design auf Basis von CSS. Nachfolgend sind die wichtigsten Aspekte der Stildefinition und -zuweisung in StyleVision aufgelistet:

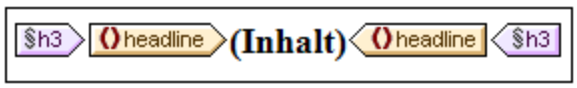
- CSS-Stilregeln können sowohl für Blockkomponenten als auch für Komponenten innerhalb von Zeilen definiert werden.
- Vordefinierte Formate[328] sind Blockkomponenten, für die bereits Stile definiert sind und die als "Hülle" für eine Gruppe von Komponenten verwendet werden können, die als Block behandelt werden sollen. Die Stile dieser vordefinierten Formate können durch Stile außer Kraft gesetzt werden, die Sie lokal für jede einzelne Komponente definieren. Dies entspricht dem CSS-Prinzip kaskadierender Stile.
- Klassenattribute können für Komponenten im Design deklariert werden und die Klasse kann als Selektor für externe[343] oder globale[346] Stilregeln verwendet werden.
- Sie können auf drei Ebenen Stile definieren und zwar, nach Priorität gereiht: (i) Stilregeln in externen Stylesheets[343], (ii) globale Stilregeln[346] und (iii) lokale Stilregeln[348].

In diesem Abschnitt lernen Sie Folgendes:

- Zuweisen vordefinierter Formate[74]
- Zuweisen eines Klassenattributs zu einer Komponente[75]
- Definieren von Stilen in einem externen CSS Stylesheet[76] und Hinzufügen dieses Stylesheets zu den Globalen Stilen des SPS
- Definieren globaler Stilregeln[76]
- Definieren lokaler Stile für eine Auswahl mehrerer Designkomponenten[78]
- Definieren lokaler Stile für eine einzelne Komponente[78]

Zuweisen vordefinierter Formate

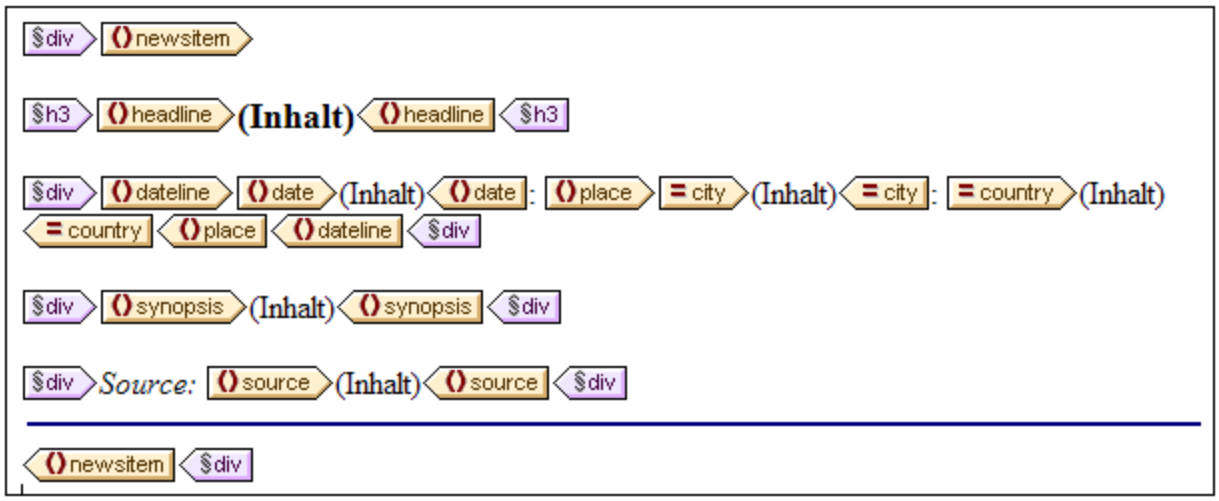
Ein Grund für die Zuweisung eines vordefinierten Formats[328] ist, einer Komponente die Stileigenschaften dieses vordefinierten Formats[328] zu geben. Wählen Sie im Design das Element `headline` aus und wählen Sie anschließend den Befehl **Einschließen in | Spezieller Absatz | Überschrift 3 (Ü3)** (oder aber Sie verwenden die Auswahlliste der vordefinierten Formate in der Symbolleiste). Das Element `headline` wird innerhalb der Tags für das vordefinierte Format gesetzt (*Abbildung unten*).



Beachten Sie, dass sich die Schriftarteigenschaften des Inhalts ändern und dass oberhalb und unterhalb des vordefinierten Formats vertikale Abstände hinzugefügt werden. Diese Eigenschaftswerte sind in dem vordefinierten Format `h3` bereits enthalten.

Eine weitere Verwendung vordefinierter Formate ist die Gruppierung von Design-Komponenten in Blöcken, sodass sie als Block formatiert werden können oder dass ihnen als Gruppe Eigenschaften innerhalb einer Zeile zugewiesen werden können. Die praktischste vordefinierte Eigenschaft für diesen Zweck ist das vordefinierte Format `div`, welches einen Block ohne Zeilenabstände oberhalb oder unterhalb davon erstellt. Weisen Sie in Ihrem Design den Nodes `newsitem`, `dateline`, `synopsis` und `source` separate `div`-Komponenten zu. Ihr Design sollte ungefähr wie in der unten gezeigten Abbildung aussehen. Beachten Sie, dass in der `div`-Komponente, die das Element `source` enthält, auch der statische Text "`Source: `" enthalten ist, und dass sich das gesamte `newsitem`-Element innerhalb einer `div`-Komponente befindet.

© 2019-2025 Altova GmbH

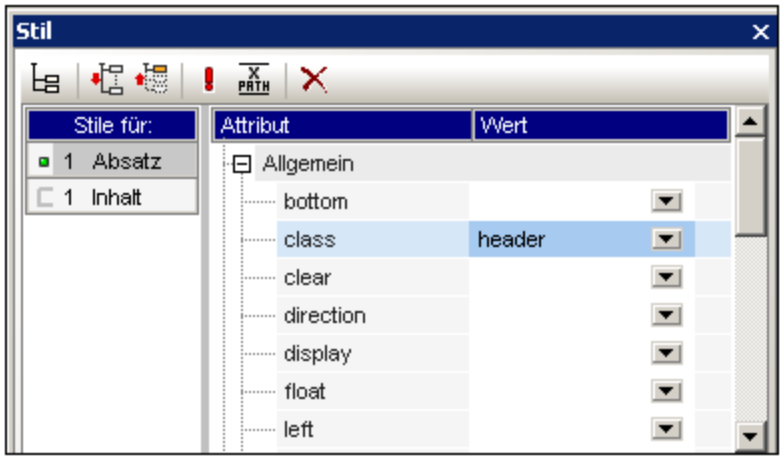


Sie haben Komponenten nun in verschiedenen `div` Blöcken gruppiert. Weiter hinten in diesem Abschnitt 78 erfahren Sie, wie man solchen Blöcken gruppierter Komponenten Stile zuweist.

Zuweisen von Komponenten zu Klassenattributen

Sie können für eine Klasse von Komponenten eine Stilregel definieren. So können Sie allen Überschriften die gemeinsame Eigenschaften zuweisen (z.B. eine bestimmte Schriftartfamilie, Linienstärke und Farbe). Dazu sind zwei Schritte erforderlich: (i) Sie müssen die Komponenten, die gemeinsame Eigenschaften haben sollen, einer einzige Klasse zuweisen; (ii) Sie müssen die Stileigenschaften für diese Klasse definieren.

Wählen Sie in Ihrem Design den Tag `h3` aus und wählen Sie in der Seitenleiste "Stil" *1 Absatz* aus (um das vordefinierte Format auszuwählen) sowie die Eigenschaftsgruppe *Allgemein.* Erweitern Sie die Eigenschaftsgruppe *Allgemein* und doppelklicken Sie anschließend auf das Wertefeld der Eigenschaft `class` und geben Sie ein `header`.



Diese bestimmte Instanz des Formats `h3` ist nun einer Klasse mit dem Namen `header` zugewiesen. Wenn Sie Stileigenschaften für die Klasse `header` definieren (Stile aus einem externen Stylesheet oder globale SPS Stile), werden diese Eigenschaften auf alle Komponenten im SPS angewendet, die die Klasse `header` haben.

Hinzufügen eines externen CSS Stylesheet zum Fenster Gobale Stile

Stilregeln in einem externen CSS Stylesheet können Komponenten im SPS Design zugewiesen werden. Zuerst müssen die externen Stylesheets jedoch zu den globalen Stilen hinzugefügt werden, damit die Regeln darin auf Komponenten angewendet werden können. Gehen Sie in der Seitenleiste "Gobale Stile" (45) (in der Design-Ansicht) folgendermaßen vor:

1. Wählen Sie den Eintrag `Extern`.
2. Klicken Sie in der Symbolleiste der Seitenleiste "Globale Stile" (45) auf die Schaltfläche **Hinzufügen**. Daraufhin wird das Dialogfeld "Öffnen" geöffnet..
3. Suchen Sie nach der Datei `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2025\StyleVisionExamples\Tutorial\QuickStart\QuickStart.css`, die sich im Verzeichnis (Eigene) Dateien (25) befindet, und klicken Sie auf **Öffnen**.

Das Stylesheet wird zu den globalen Stilen hinzugefügt. Es enthält die folgenden Regeln, die zu diesem Zeitpunkt relevant sind:

```
.header {
         font-family: "Arial", sans-serif;
         font-weight: bold;
         color: red;
        }

h3      {
         font-size: 12pt;
        }
```

Die Stilregeln für die Klasse `header` und das Element `h3` werden kombiniert und erzeugen für das Element `headline` die folgende HTML-Ausgabe.

NanoNull Inc Launches Version 2.0 of NanoPower

2006-04-01: Boston, USA
Nanonull Inc today launched version 2.0 of its market-leading NanoPower line of hardware and software. The highlights of the new version of NanoPower are improved chip design capabilities in NanoSoft, the software used to design computer chips; and higher precision sand-filtering processes in NanoMeld, the hardware in which the transformation from sand to silicon chips is executed.
Source: NewTech Online

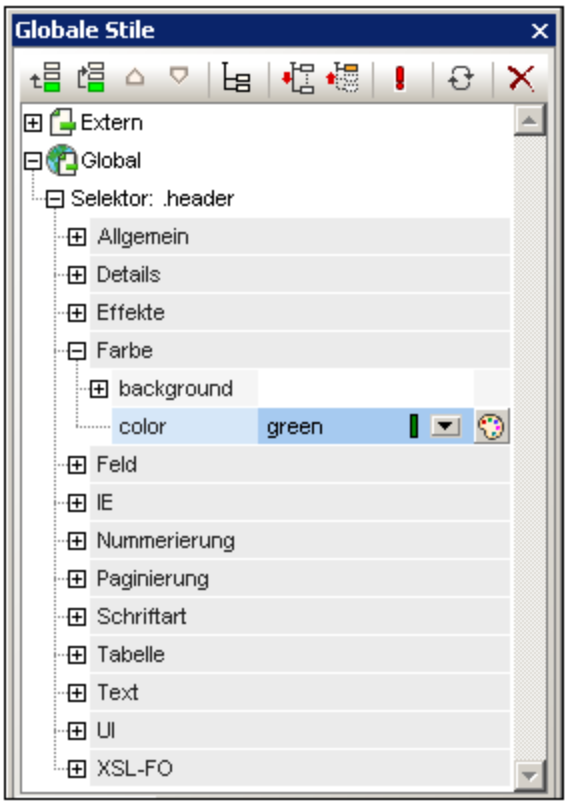
Definieren globaler Stilregeln

Globale Stilregeln (346) können mittels CSS-Selektoren für das gesamte SPS definiert werden. Die Regeln werden direkt in der Seitenleiste "Gobale Stile (45) definiert. Erstellen Sie nun für die Klasse `header` eine globale Stilregel:

1. Wählen Sie bei aktiver Design-Ansicht (29) in der Seitenleiste "Gobale Stile" (29) den Eintrag "Global".

© 2019-2025 Altova GmbH

2. Klicken Sie in der Symbolleiste auf die Schaltfläche **Hinzufügen**. Daraufhin wird ein leerer Eintrag für den Platzhalter-Selektor (*) erstellt. Dieser Selektor wird markiert angezeigt.
3. Geben Sie den Text `.header` ein, um den Platzhalter-Selektor zu ersetzen.
4. Erweitern Sie die Eigenschaftsgruppe *Farbe* und wählen Sie aus der Dropdown-Liste der Eigenschaftswerte `color` den Wert `green` aus (*Abbildung unten*).



In Fällen, in denen die globale Stilregel eine Eigenschaft definiert, die auch im externen Stilesheet definiert ist (die Eigenschaft `color`), hat der Eigenschaftswert in der globalen Stilregel Vorrang. Daher wird der Inhalt der Überschrift in der HTML-Vorschau grün angezeigt. Andere Eigenschaftsdefinitionen aus dem externen Stylesheet (die nicht durch eine Eigenschaft in einer globalen Stilregel außer Kraft gesetzt werden) werden beibehalten (in diesem Fall, `font-family` und `font-weight`).

Definieren von lokalen Stilen für mehrere Komponenten auf einmal

Lokale Stile können gleichzeitig für mehrere Komponenten definiert werden. Um in Ihrem Design festzulegen, dass der gesamte Textinhalt einer Nachricht (newsitem) die Schriftart "Arial" haben soll, klicken Sie auf die `div`-Komponente, innerhalb der sich das Element `newsitem` befindet und wählen Sie in der Seitenleiste "Stil" [47] in der Spalte "Stile für" 1 `Absatz`. Wählen Sie anschließend in der Eigenschaftsgruppe *Schriftart* als `font-family` die Schriftart `Arial` aus. Diese Eigenschaft wird auch für alle 5 untergeordneten vordefinierten Formate übernommen.

Wählen Sie nun im Design die drei `div`-Komponenten rund um die Nodes `dateline`, `synopsis` und `source` aus (Halten Sie dazu die **Umschalttaste** gedrückt, während Sie auf die einzelnen `div`-Komponenten klicken). Wählen Sie in der Seitenleiste "Stil" [47] `3 Absätze` aus, anschließend die Eigenschaftsgruppe *Schriftart* und setzen Sie `font-size` auf `10pt`. (Die `h3`-Komponente wurde nicht ausgewählt, da sie bereits die gewünschte Schriftgröße von `12pt` hat)

Wählen Sie nun zum Abschluss die `div` -Komponente rund um das Element `dateline` aus. Wählen Sie in der Seitenleiste "Stil" [47] in der Spalte "Stile für" den Eintrag `1 Absatz` aus. Setzen Sie in der Eigenschaftsgruppe *Schriftart* `font-weight` auf `bold` und `font-style` auf `italic`. Wählen Sie in der Eigenschaftsgruppe *Farbe* für `color` den Eintrag `gray` aus. Die Ausgabe für das Element "Dateline" sollte nun folgendermaßen aussehen:

Beachten Sie, dass die für die `div`-Komponente definierten Stileigenschaften auch auf den statischen Text innerhalb der `div`-Komponente (d.h. auf den Doppelpunkt und das Komma) angewendet wurden.

Definieren von lokalen Stilen für eine einzelne Komponente

Ein lokaler Stil, der für eine einzelne Komponente definiert ist, setzt alle anderen Stile, die für diese Komponente auf höheren Ebenen des SPS definiert wurden, außer Kraft. Wählen Sie im Design das Element `headline` aus und weisen Sie ihm die Farbe "navy" zu (Eigenschaft `color` in der Eigenschaftsgruppe *Farbe* der Stileigenschaften). Die lokal definierte Eigenschaft (`color:navy`) setzt den globalen Stil für die Klasse `.header` (`color:green`) außer Kraft.

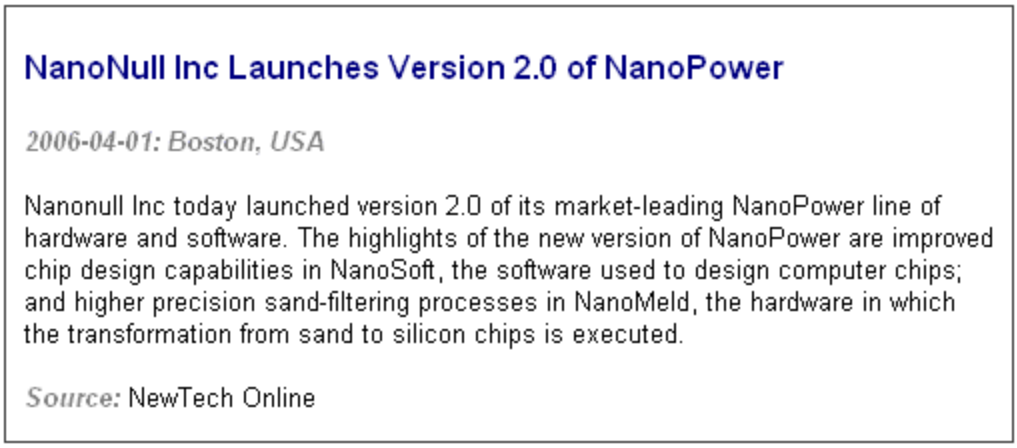
Wählen Sie die div Komponente, innerhalb der sich das source Element befindet. Setzen Sie in der Seitenleiste "Stil" [47], während in der Spalte "Stile für" der Eintrag 1 Absatz ausgewählt ist, die Eigenschaft `color` in der Eigenschaftsgruppe *color* der Stileigenschaften auf `gray` (*Abbildung unten*). Setzen Sie in der Eigenschaftsgruppe *Schriftart* der Stileigenschaften die Eigenschaft `font-weight` auf `bold`. Diese Werte werden auf den statischen Text angewendet. Erinnern Sie sich, dass der statische Text, dass wir im letzten Abschnitt dem statischen Text "`Source: `" einen `font-style`-Wert `italic` zugewiesen haben. Die neuen Eigenschaften (`font-weight:bold` und `color:gray`) kommen zusätzlich zur Eigenschaft `font-style:italic` hinzu.

Wählen Sie nun in der Design-Ansicht den Platzhalter `(Inhalt)` des Elements "Source" aus. Setzen Sie in der Spalte "Stile für", während *1 Inhalt* ausgewählt ist, die Eigenschaft `color` (in der Eigenschaftsgruppe *Farbe* der Stileigenschaften) auf `black`. Wählen Sie in der Eigenschaftsgruppe *Schriftart* für `font-weight` den Wert `normal` aus. Die neuen Eigenschaften werden für den Platzhalter-Node `Inhalt` des Elements `source` gesetzt und setzen die für die `div`-Komponente definierten Eigenschaften außer Kraft (*siehe Abbildung unten*).

© 2019-2025 Altova GmbH

Fertigstellen der Formatierung

Um die Formatierung in diesem Abschnitt fertigzustellen, wählen Sie die `div`-Komponente des Elements `synopsis` aus und in der Symbolleiste in der Auswahlliste der Vordefinierten Formate[328] den Eintrag Absatz (p). Daraufhin werden dem Block die vordefinierten Stile des HTML-Elements `p` zugewiesen. Die HTML-Vorschau sollte nun etwa folgendermaßen aussehen:



Wenn Sie damit fertig sind, speichern Sie die Datei.

3.5 Verwenden von automatischen Berechnungen

Automatische Berechnungen 257 sind eine effiziente Methode, um anhand der verfügbaren XML-Daten zusätzliche Informationen bereitzustellen. In diesem Abschnitt werden zwei Informationen zum Design hinzugefügt: die Gesamtzahl der Nachrichten (news items) und der Zeitraum, den diese Nachrichtenmeldungen im XML-Dokument abdecken. Keine dieser Informationen ist im XML-Dokument direkt verfügbar, sondern beide Informationen müssen anhand der verfügbaren Daten errechnet werden.

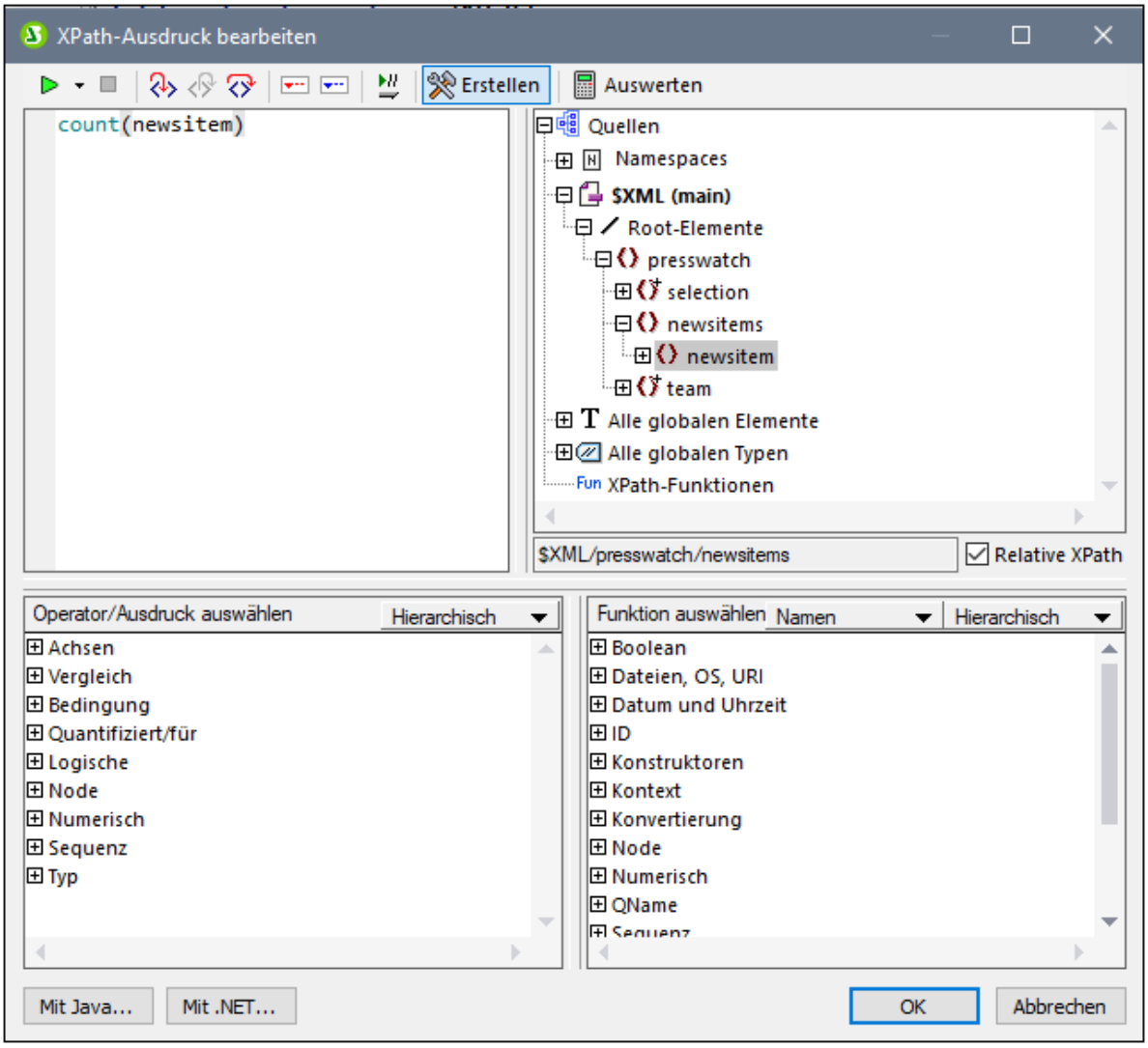
Zählen der "newsitem" Nodes

Gehen Sie im Design folgendermaßen vor:

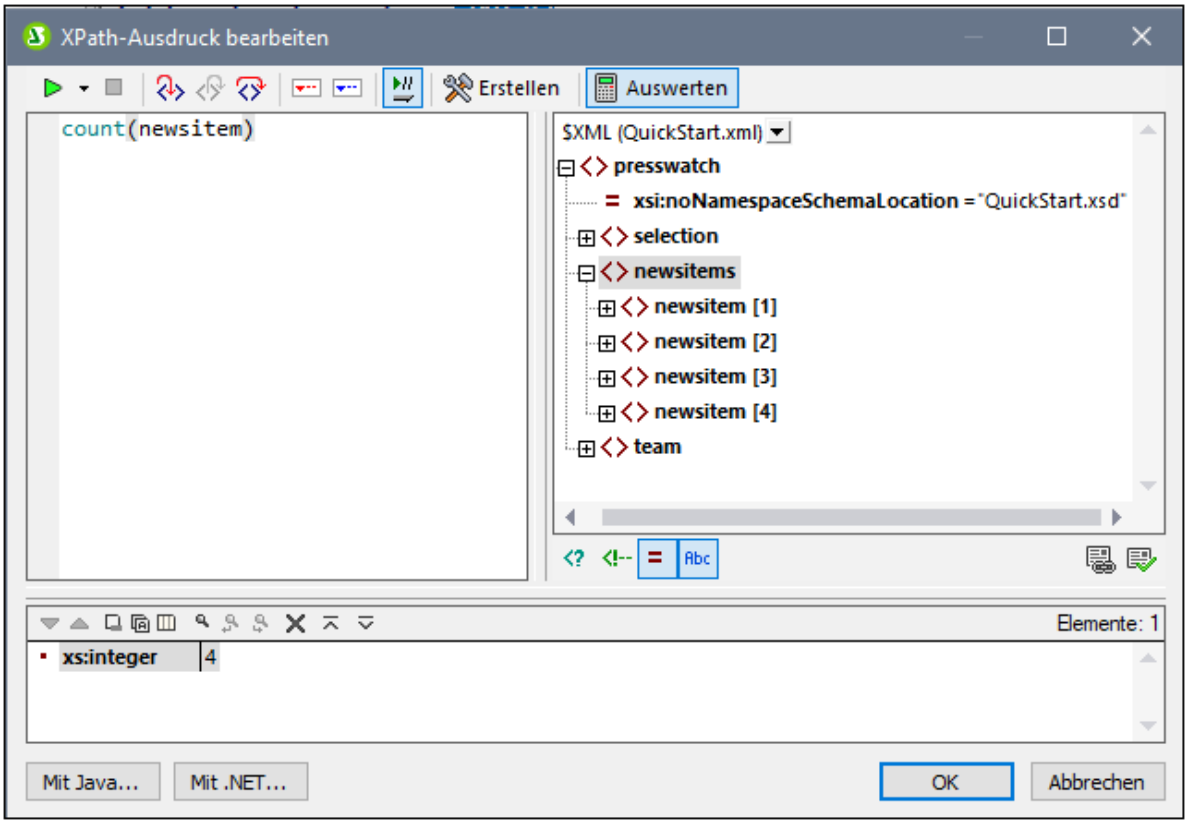
1. Schaffen Sie, wie in der Abbildung unten gezeigt, Platz für eine Zeile statischen Texts (in die die automatische Berechnung platziert wird). Drücken Sie die **Eingabetaste,** um neue Zeilen hinzuzufügen und fügen Sie unterhalb der Leerzeile eine horizontale Linie ein (*siehe Abbildung*).

2. Geben Sie, wie in der Abbildung oben gezeigt, den statischen Text "`Total number of news items:` " ein.
3. Wenden Sie auf den statischen Text lokale Stileigenschaften Ihrer Wahl an. Ein Anleitung dazu finden Sie im Abschnitt Formatieren des Inhalts 78.
4. Platzieren Sie den Cursor hinter den Doppelpunkt und wählen Sie **Einfügen | Automatische Berechnung | Wert**. Daraufhin wird das Dialogfeld "XPath-Ausdruck bearbeiten" eingefügt (*Abbildung unten*). (Alternativ dazu können Sie auch auf die rechte Maustaste klicken und den Befehl im Kontextmenü auswählen.)

© 2019-2025 Altova GmbH



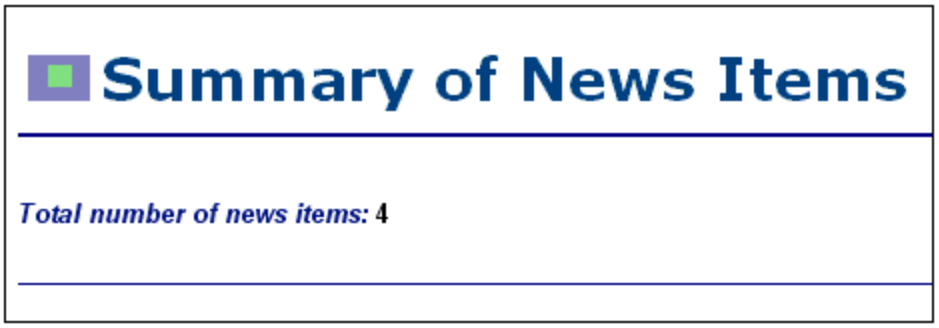
5. Beachten Sie, dass der Kontext-Node in der Schemastruktur der Node `newsitems` ist. Dieser Node erscheint markiert. Geben Sie nun im Textfeld "Ausdruck" entweder den Ausdruck `count(newsitem)` ein oder erstellen Sie den Ausdruck mit Hilfe der Eingabehilfefenster unterhalb des Textfelds für den Ausdruck. (Doppelklicken Sie in der Funktionsgruppe *Sequenz* auf die Funktion `count`, um Sie zu übernehmen, setzen Sie anschließend im Textfeld "Ausdruck" den Cursor zwischen die Klammern der Funktion und doppelklicken Sie in der Schemastruktur auf den `newsitem` Node. Um zu sehen, welches Ergebnis vom XPath-Ausdruck zurückgegeben wird, klicken Sie auf die Schaltfläche **Auswerten**. Das Ergebnis der Auswertung wird im Bereich *Ergebnisse* angezeigt (*siehe Abbildung unten*). Eine ausführliche Beschreibung des Dialogfelds "XPath-Ausdruck bearbeiten" finden Sie im Abschnitt XPath-Ausdruck bearbeiten[440].



6. Klicken Sie abschließend auf **OK**. Die automatische Berechnung wird an der Cursor-Position in das Design eingefügt (*Abbildung unten*). Formatieren Sie die automatische Berechnung mit lokalen Stilen[78].

 © 2019-2025 Altova GmbH

Ihre HTML-Ausgabe sieht folgendermaßen aus:

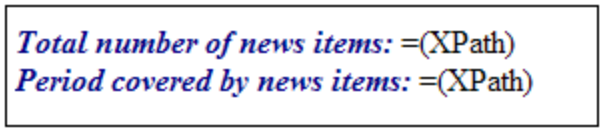


Anzeigen des Zeitraums der Nachrichtenmeldungen

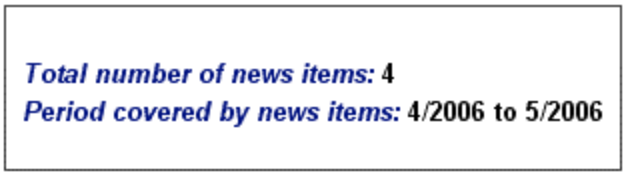
Der Zeitraum, in dem alle Nachrichtenmeldungen insgesamt veröffentlicht wurden, wird anhand des Datums der ersten und dem der letzten Nachrichtenmeldung ermittelt. Dies lässt sich anhand von XPath-Ausdrücken wie den unten angeführten eruieren. Das Resultat des ersten Ausdrucks ist der Inhalt des `date` Node. Im zweiten wurde der erste Ausdruck eingeschränkt, sodass nur die Werte Monat und Datum ausgegeben werden. Sie können beide Ausdrücke verwenden.

- `concat(min(//date), ' to ', max(//date)).`
- `concat(month-from-date(min(//date)), '/', year-from-date(min(//date)), ' to ', month-from-date(max(//date)), '/', year-from-date(max(//date)))`

Fügen Sie im Design den statischen Text und die automatische Berechnung ein, wie in der Abbildung unten gezeigt. Wenden Sie einen beliebigen lokalen Stil darauf an.



Die HTML-Vorschau sieht etwa folgendermaßen aus:



Wenn Sie damit fertig sind, speichern Sie die Datei.

3.6 Verwendung von Bedingungen

Wenn Sie einen Blick in die Datei `QuickStart.xml` machen, werden Sie feststellen, dass jedes `newsitem`-Element ein `metainfo` Child-Element hat, welches wiederum ein oder mehrere `relevance` Child-Elemente enthalten kann. Jeder relevance Node enthält eine Überschrift, unter der die Relevanz der Nachrichtenmeldungen inzidiert ist. Des Weiteren gibt es einen Node `/presswatch/selection/byrelevance`. Der Inhalt dieses Node enthält eine oder mehrere Relevanzüberschriften und gibt an, welche Nachrichtenmeldungen angezeigt werden sollen. Wenn der Inhalt des Node `byrelevance` z.B. `NanoPower` ist, werden alle Nachrichtenmeldungen, die einen `relevance` Node mit dem Inhalt `NanoPower` haben, angezeigt. Mit Hilfe einer Bedingung kann überprüft werden, was der Inhalt des `byrelevance` Node ist (indem der Node mit lookup überprüft wird) und mittels einer Bedingungsvorlage die entsprechende Verarbeitung zur Verfügung stellen. In diesem Abschnitt wird eine Bedingungsvorlage erstellt, die jene Nachrichtenmeldungen anzeigt, die ein `relevance` Element haben, das mit dem Inhalt von `byrelevance` übereinstimmt.

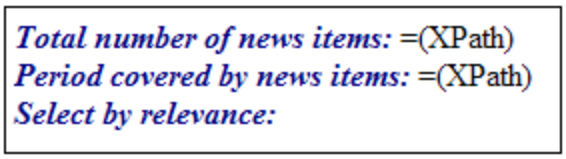
Dabei gehen wir folgendermaßen vor:

1. Erstellung einer Auswahlliste, in der der Wert des `byrelevance` Node angezeigt wird. Die Werte in der Dropdown-Liste der Auswahlliste werden mit Hilfe eines XPath-Ausdrucks ermittelt, der eine Liste aller eindeutigen `relevance` Nodewerte dynamisch kompiliert.
2. Einfügen einer Bedingung rund um das Element `newsitem`. Mit dieser Bedingung werden alle `newsitem` Elemente, die ein `relevance` Element haben, dessen Inhalt mit dem Inhalt des `byrelevance` Node übereinstimmt, ausgewählt. Der Inhalt, der in eine Verzweigung einer Bedingung eingeschlossen ist, wird als Bedingungsvorlage bezeichnet.
3. Innerhalb der Bedingungsvorlage werden die einzelnen `relevance` Nodes dieser Nachrichtenmeldung aufgelistet.
4. Markieren des `relevance` Elements (in der Liste der `relevance` Elemente), das mit dem `byrelevance` Element übereinstimmt. Dies erfolgt durch Erstellung einer Bedingung zur Auswahl solcher `relevance` Elemente und durch Anwendung einer speziellen Formatierung auf diese Bedingungsvorlage.
5. In die Bedingung für das `newsitem` Element wird eine Verzweigung eingefügt, die alle Nachrichtenmeldungen auswählt.

Erstellen der Auswahlliste zur Auswahl eindeutiger Node-Werte

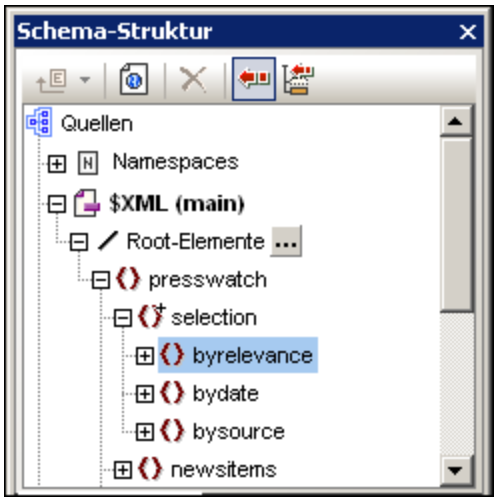
Der Node im XML-Dokument, der die Benutzerselektion enthält, ist `/presswatch/selection/byrelevance`.Erstellen Sie diesen Node als Auswahlliste. Gehen Sie dazu folgendermaßen vor:

1. Fügen Sie den statischen Text "`Select by relevance: `" oben im Dokument knapp unterhalb der zweiten automatischen Berechnung 80 ein (*Abbildung unten*).

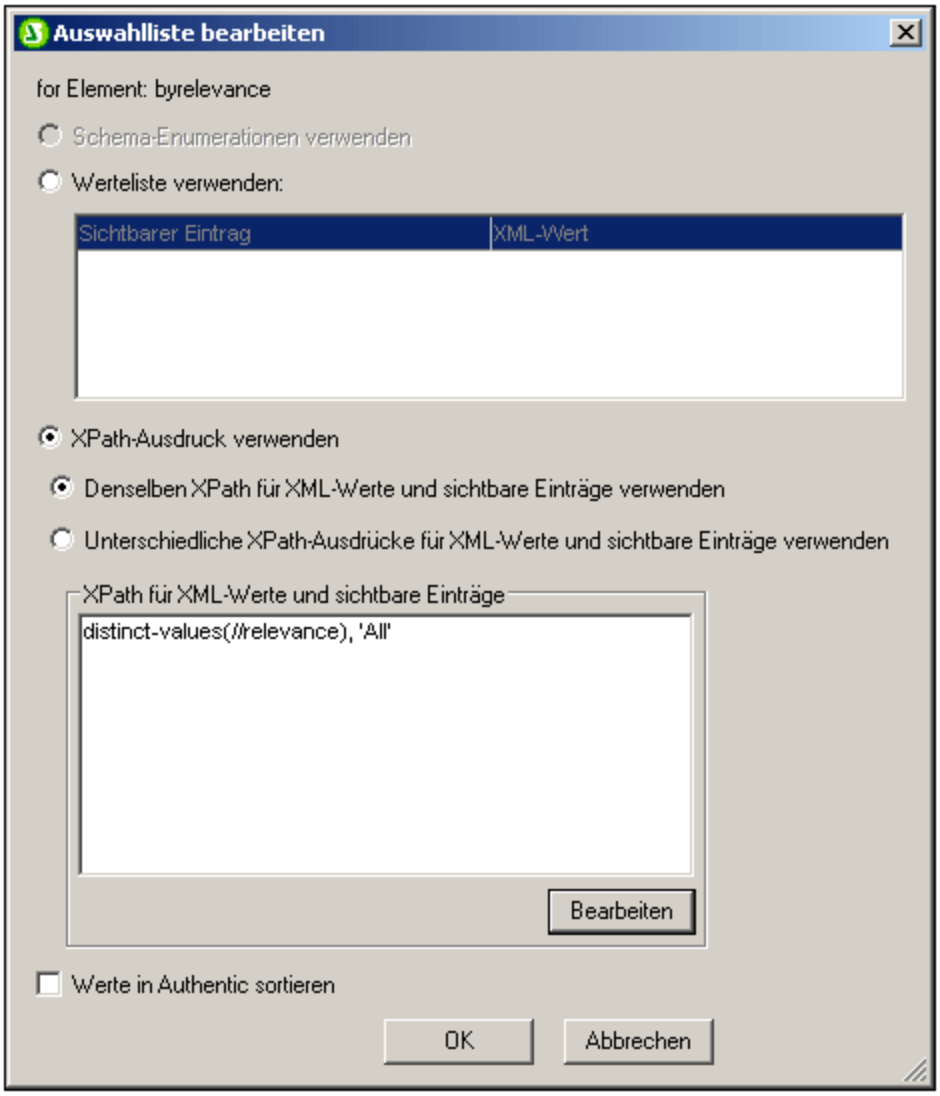


2. Ziehen Sie den `byrelevance`-Node aus der Seitenleiste "Schema-Struktur" 38 (*Abbildung unten*) hinter den neu eingegebenen statischen Text.

 © 2019-2025 Altova GmbH

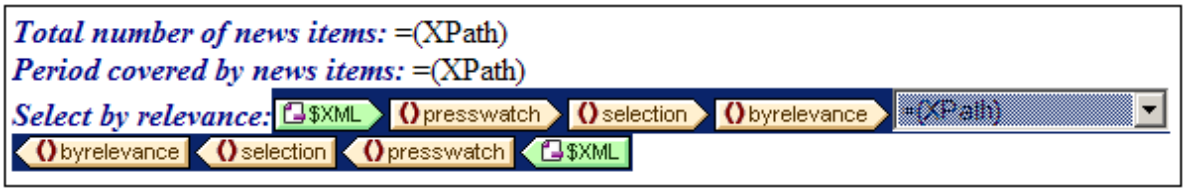


3. Wählen Sie im Kontextmenü, das daraufhin angezeigt wird, den Befehl "Auswahlliste erstellen". Das folgende Dialogfeld wird angezeigt.

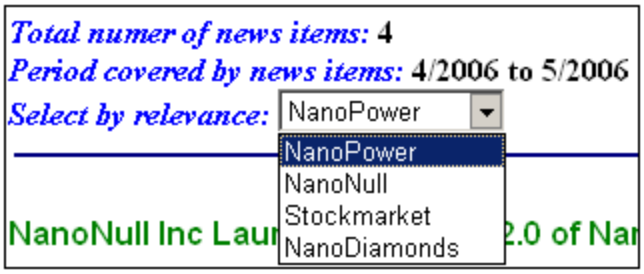


4. Wählen Sie im Dialogfeld "Auswahlliste bearbeiten" (*Abbildung oben*) die Option "XPath-Ausdruck verwenden" und dann "Denselben XPath für XML-Werte und sichtbare Einträge verwenden". Geben Sie im Bereich "XPath für XML-Werte und sichtbare Einträge" den XPath-Ausdruck: `distinct-values(//relevance)` ein. Damit werden eindeutige Werte aller `relevance`-Elemente im XML-Dokument ausgewählt. Beachten Sie: Obwohl die Werte aller relevance Nodes in der HTML-Auswahlliste angezeigt werden, hat die Auswahl eines dieser Werte in der HTML-Vorschau keinen Einfluss auf den Inhalt des Node im XML-Dokument (auf Basis dessen das SPS%> arbeitet). Das HTML-Dokument ist ein Ausgabedokument, das durch Transformation des XML-Dokuments erzeugt wird; es ist darin keine Eingabe zulässig. Die Auswahlliste dient hier dazu, um verschiedene Arten zur Darstellung von Inhalt zu demonstrieren.
- Klicken Sie zum Abschluss auf **OK**. Die Auswahlliste wird eingefügt und das Design sieht daraufhin etwa folgendermaßen aus:

© 2019-2025 Altova GmbH



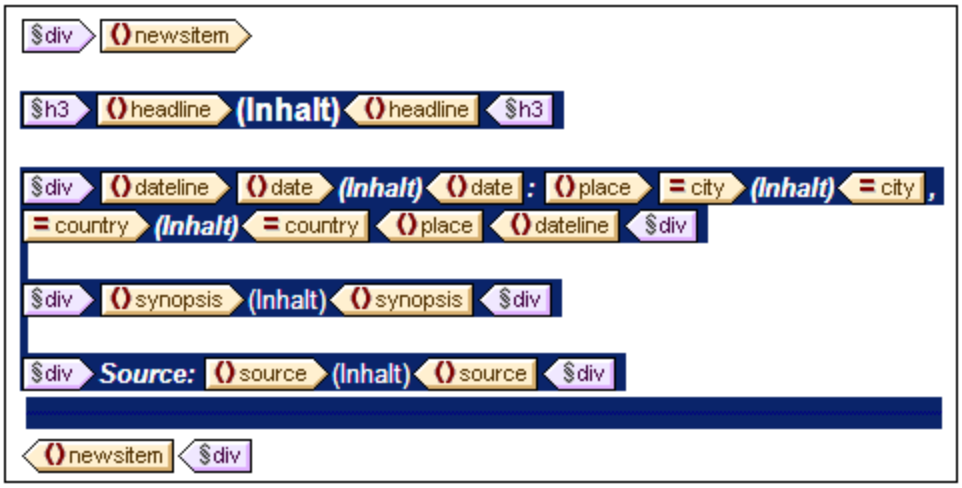
6. Wechseln Sie in die HTML-Vorschau[31]. Wenn Sie im Auswahllistenfeld auf den Abwärtspfeil klicken, beachten Sie, dass die Liste die eindeutigen Werte aller `relevance` Nodes enthält (*Abbildung unten*). Vergleichen Sie dies mit dem XML-Dokument. Dies ist eine dynamische Liste, die jedes Mal, wenn ein neuer `relevance` Wert zum XML-Dokument hinzugefügt wird, erweitert wird.



Einfügen einer Bedingung, um Nachrichtenmeldungen anzuzeigen, die die ausgewählte **relevance** haben

Die Bedingung wählt `newsitem`-Elemente aus, die ein `metainfo/relevance`-Element mit einem Wert haben, der mit mit dem Wert im Element `/presswatch/selection/byrelevance` übereinstimmt. Gehen Sie zum Einfügen der Bedingung folgendermaßen vor:

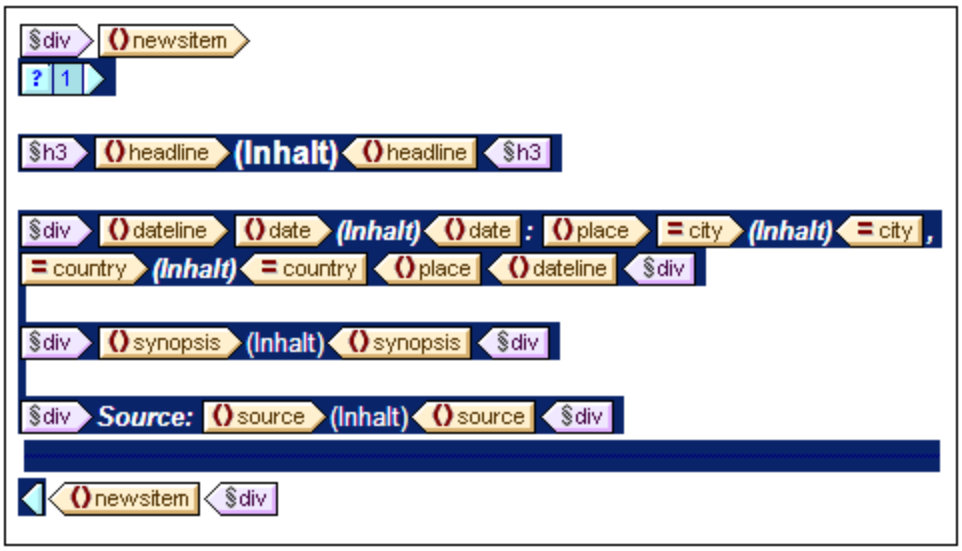
1. Wählen Sie den Inhalt des `newsitem` Teils des Designs aus, welcher in der Bedingung enthalten sein soll (in der Abbildung unten markiert).



2. Wählen Sie den Menübefehl (oder den Kontextmenübefehl) **Einschließen in | Bedingung**[528]. Daraufhin wird das Dialogfeld "XPath-Ausdruck bearbeiten[440]" geöffnet.
3. Geben Sie den Ausdruck `metainfo/relevance=/presswatch/selection/byrelevance` ein. Dieser Ausdruck ergibt das Resultat "true", wenn der Wert des `metainfo/relevance` Child-Elements des

aktuellen `newsitem`-Elements mit dem Wert des `/presswatch/selection/byrelevance`-Elements (der Auswahl des Benutzers) übereinstimmt.

4. Klicken Sie auf **OK**. Die Bedingung wird rund um den Inhalt des `newsitem`-Elements erstellt (*Abbildung unten*).



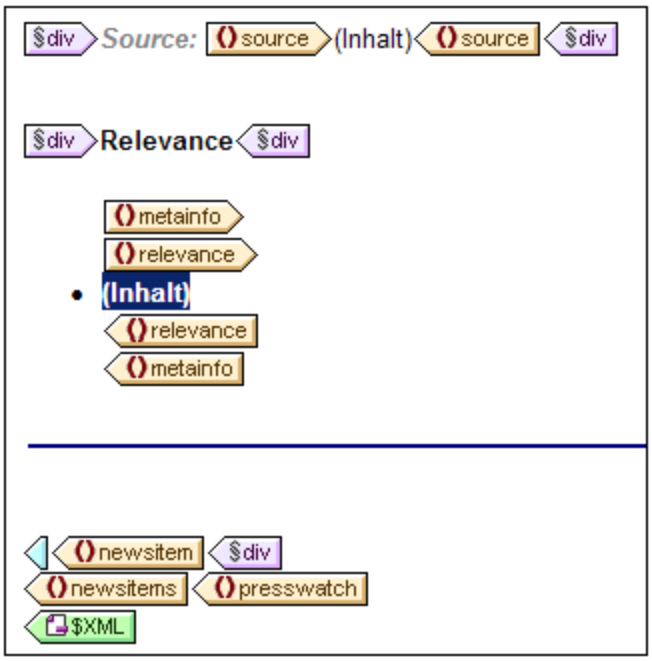
Beachten Sie, dass es in dieser Bedingung nur einen Branch gibt. Nachrichtenmeldungen, für die die Auswertung der Bedingung "true" ergibt, werden angezeigt, jene, für die die Auswertung nicht "true" ergibt, werden nicht angezeigt. Die Bedingung funktioniert also in diesem Fall als Filter. Weiter unten in diesem Abschnitt werden Sie zu dieser Bedingung einen zweiten Branch hinzufügen.

Einfügen des **relevance** Node als Liste

Um die `relevance` Nodes der einzelnen `newsitem` Elemente anzuzeigen, fügen Sie sie folgendermaßen in das Design ein *(siehe Abbildung unten)*:

1. Erstellen Sie unterhalb der `div`-Komponente für das `source`-Element und innerhalb des End-Tags der bedingten Vorlage einen vertikalen Abstand.
2. Geben Sie den statischen Text "`Relevance:`" ein und setzen Sie den Text innerhalb des vordefinierten Formats `div` (Markieren Sie den statischen Text und fügen Sie das vordefinierte Format ein).
3. Ziehen Sie das `relevance`-Element aus der Root-Elemente-Struktur in der Seitenleiste "Schema-Struktur" 38 in das Design unterhalb des statischen Texts `Relevance:`.
4. Erstellen Sie das Element im Listenformat. (Wählen Sie im Kontextmenü, das angezeigt wird, wenn Sie den Node in das Design ziehen, den Eintrag "Nummerierung und Aufzählungszeichen" und wählen Sie anschließend das gewünschte Listenformat.)
5. Wenden Sie auf den Inhalt der Liste eine Textformatierung an. Wenn Sie fertig sind, sollte das Design ungefähr folgendermaßen aussehen:

© 2019-2025 Altova GmbH

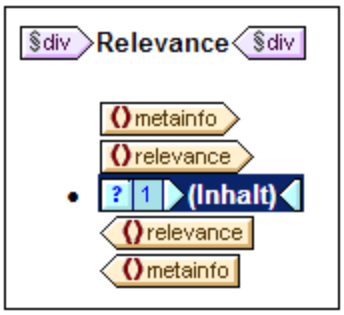


Überprüfen Sie nun in HTML-Vorschau, die Ergebnisse der unterschiedlichen Auswahlen von `relevance`; Dazu müssen Sie: (i) den Wert des byrelevance Node im XML-Dokument ändern; (ii) das XML-Dokument speichern; (iii) und anschließend die SPS-Datei in StyleVision öffnen.

Fettformatierung des ausgewählten **relevance**-Elements

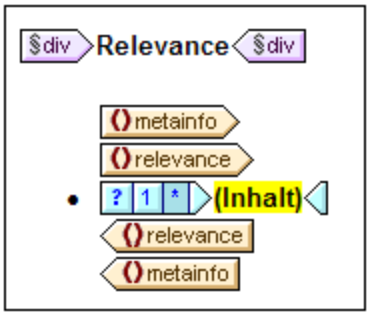
Einige Nachrichtenmeldungen haben mehr als ein `relevance`-Element. In solchen Fällen wäre es übersichtlicher, wenn das relevance-Element, das mit der Auswahl des Benutzers übereinstimmt, markiert erscheint und die anderen nicht. Dies lässt sich folgendermaßen bewerkstelligen:

1. Wählen Sie das `relevance`-Element im Design aus.
2. Fügen Sie eine Bedingung ein und zwar mit dem XPath-Ausdruck: `.=/presswatch/selection/byrelevance`. Dadurch wird eine Bedingung mit einer einzigen Verzweigung erstellt (*Abbildung unten*), die `relevance` Elemente auswählt, die mit dem `byrelevance` Element übereinstimmen.



3. Wählen Sie den Platzhalter `Inhalt` aus und weisen Sie ihm (über die Seitenleiste "Stil") eine lokale Formatierung "bold" (Gruppe *Schriftart*) und einen gelben (yellow) Hintergrund (Gruppe *Farbe*) zu.
4. Rechtsklicken Sie auf die Bedingung und wählen Sie aus dem Kontextmenü den Befehl **Verzweigung kopieren**.

5. Aktivieren Sie im Dialogfeld "XPath-Ausdruck bearbeiten[440]" das Kontrollkästchen "Andernfalls" (rechts oben).
6. Klicken Sie zum Fertigstellen auf **OK**. Eine neue Verzweigung (`Otherwise`) wird erstellt (*Abbildung unten*). Diese Bedingungsverzweigung wählt alle `relevance` Elemente aus, die nicht mit dem `byrelevance` Element übereinstimmen.



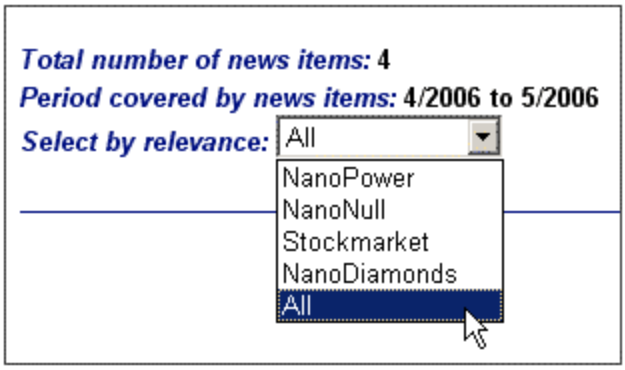
7. Beachten Sie, dass der Inhalt des `Otherwise`-Branch eine Kopie des ersten Branch ist; der Platzhalter `Inhalt` erscheint fett und mit gelbem Hintergrund. Entfernen Sie diese Formatierung (fett und Hintergrundfarbe Schwarz) aus dem Platzhalter `Inhalt`.

Sie haben eine Bedingung mit zwei Verzweigungen (jede mit ihrer Bedingungsvorlage) gesetzt, die an jedem `relevance`-Element die folgende Überprüfung durchführt: Wenn der Inhalt von `relevance` mit dem von `/presswatch/selection/byrelevance` übereinstimmt, wird der Inhalt von `relevance` fett und mit einem gelben Hintergrund angezeigt. Andernfalls (zweite Verzweigung) wird er normal formatiert angezeigt. Überprüfen Sie dies in der HTML-Vorschau

Ändern der Auswahlliste und Einfügen einer zweiten Bedingungsverzweigung

In der Auswahlliste gibt es keine Option zur Auswahl aller Nachrichtenmeldungen. Um diese Option einzufügen, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Design-Ansicht die Auswahlliste aus.
2. Klicken Sie in der Seitenleiste "Eigenschaften", während in der Spalte "Eigenschaften für" die Option "Auswahlliste" ausgewählt ist, (in der Eigenschaftsgruppe *Auswahlliste*) auf die Schaltfläche **Bearbeiten** der Eigenschaft `Eingabewerte der Auswahlliste`.
3. Ändern Sie im nun angezeigten Dialogfeld "XPath-Ausdruck bearbeiten[440]" den XPath-Ausdruck von `distinct-values(//relevance)` in `distinct-values(//relevance), 'All'`. Daraufhin wird der String `All` zur Liste der Einträge hinzugefügt, die vom XPath-Ausdruck zurückgegeben werden.
4. Überprüfen Sie die Dropdown-Auswahlliste in der HTML-Vorschau(*Abbildung unten*).



 © 2019-2025 Altova GmbH

Der Wert `All` kann nun in den `byrelevance` Node eingegeben werden. Damit soll bezweckt werden, dass alle Nachrichtenmeldungen angezeigt werden, wenn der `byrelevance` Node den Wert all enthält

Die Bedingung, die die Vorlage für die Nachrichtenmeldungen anzeigt, hat eine einzige Verzweigung mit dem Ausdruck `metainfo/relevance=/presswatch/selection/byrelevance`. Da kein `metainfo/relevance` Node den Wert `All` hat, wird keine Nachrichtenmeldung angezeigt, wenn `All` ausgewählt ist. Sie müssen daher nun eine zweite Verzweigung für die Bedingung erstellen, der eine Überprüfung auf den Wert `All` durchführt. Indem Sie die Vorlage für die Nachrichtenmeldungen in dieser Verzweigung erstellen, wird die Nachrichtenmeldung ausgegeben, wenn das Ergebnis der Auswertung des Ausdrucks "true" ist. Gehen Sie dazu folgendermaßen vor:

1. Wählen Sie in der Design-Ansicht die Bedingung für die Nachrichtenmeldungen aus.
2. Rechtsklicken Sie auf die Bedingung und wählen Sie im Kontextmenü **Verzweigung kopieren**.
3. Geben Sie im Dialogfeld "XPath-Ausdruck bearbeiten[440]", das nun angezeigt wird, den Ausdruck: `/presswatch/selection/byrelevance='All'` ein.
4. Klicken Sie zum Abschluss auf **OK**. Es wird nun eine zweite Verzweigung erstellt.

Die zweite Verzweigung hat als Inhalt dieselbe Vorlage, wie die erstel Verzweigung. Sie gibt die Vorlage für die Nachrichtenmeldungen aus, wenn der Inhalt des `byrelevance` Node `All` ist.

Speichern Sie Ihr Design, nachdem Sie diesen Abschnitt abgeschlossen haben.

3.7 Verwendung von globalen Vorlagen und "Restlicher Inhalt"

Globale Vorlagen[22] eignen sich, um die Verarbeitung eines Elements global zu definieren, sodass die (an einer Stelle definierten) Regeln der globalen Vorlage an mehreren Stellen im Stylesheet verwendet werden können. Eine globale Vorlage kann auf zwei Arten verwendet werden:

- Die Regeln der globale Vorlage können in die lokale Vorlage kopiert werden.
- Eine lokale Vorlage (in der Hauptvorlage) kann die Verarbeitung dieses Node an die globale Vorlage übergeben. Anschließend wird wieder die Verarbeitung in der Hauptvorlage aufgenommen. In diesem Fall heißt es, dass die globale Vorlage von der Hauptvorlage aus aufgerufen oder verwendet wird.

Es gibt zwei Methoden, um eine globale Vorlage von der Hauptvorlage aus aufzurufen.

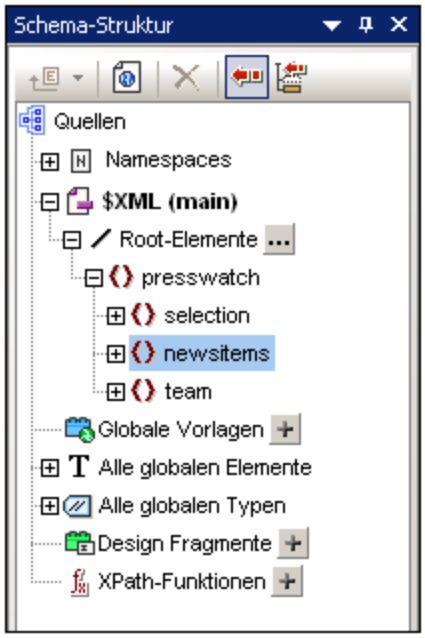
- Wenn eine lokale Vorlage eine globale Vorlage referenziert.

Wenn eine `(Restlicher Inhalt)` Anweisung in der Hauptvorlage Vorlagen auf die untergeordneten Elemente (den restlichen Inhalt) des aktuellen Elements anwendet. Wenn für eines der untergeordneten Elemente eine globale Vorlage existiert, wird diese auf dieses Element angewendet. Andernfalls wird die vordefinierte Vorlage für Elemente angewendet. (Die vordefinierte Vorlage für Elemente verarbeitet Child-Elemente und die Ausgabe von Textinhalt von Elementen. Als Folge davon wird der Textinhalt aller untergeordneten Elemente ausgegeben. Beachten Sie, dass die Werte von Attributen **nicht** ausgegeben werden).

In diesem Abschnitt werden wir nun mit Hilfe der Anweisung "restlicher Inhalt" und einer globalen Vorlage für das globale Element[22] `member` ein Design für die Vorlage der Mitarbeiter des Presseteams erstellen.

Einfügen der Anweisung "restlicher Inhalt"

In der Abbildung unten sehen Sie die allgemeine Struktur des Schemas.



 © 2019-2025 Altova GmbH

Das Dokumentelement `presswatch` enthält drei Child-Elemente: (i) `selection`; (ii) `newsitems`; und (iii) `team`. Die bisher erstellte Hauptvorlage verarbeitet das Element `/presswatch`. Innerhalb des Elements `presswatch` wird nur das Element `newsitems` verarbeitet. Die Elemente `selection` und `team` innerhalb des Elements `presswatch` werden nicht verarbeitet (obwohl `selection` im Element `newsitems` verarbeitet wurde). Wenn Sie daher die Anweisung `restlicher Inhalt` in `presswatch` einfügen, werden auch die Elemente `selection` und `team` verarbeitet.

Fügen Sie die Anweisung `restlicher Inhalt` in das Design ein, indem Sie den Cursor zwischen die End-Tags von `newsitems` und `presswatch` platzieren und den Menü- oder Kontextmenübefehl **Einfügen | Restlicher Inhalt** 506 auswählen. Der Platzhalter `restlicher Inhalt` wird eingefügt (*Abbildung unten*).

In der HTML-Vorschau sehen Sie einen Textstring (*Abbildung unten*):

Dieser String ist das Ergebnis der Anwendung der vordefinierten Vorlagen auf die Elemente `selection` und `team`. Die vordefinierte Vorlage für Elemente verarbeitet Child-Elemente. Die vordefinierte Vorlage für Text-Nodes gibt den Text im Text-Node aus. In Kombination geben diese beiden vordefinierten Vorlagen den Textinhalt aller untergeordneten Nodes der Elemente `selection` und `team` aus. Der Text `All` kommt von `selection/byrelevance`, gefolgt von der Textausgabe der untergeordneten Nodes von `team/member` nämlich `first`, `last`, `email` in der Dokumentenreihenfolge. Beachten Sie, dass das `id`-Attribut von `member` nicht ausgegeben wird (da es als Attribut nicht als Child von `member` betrachtet wird).

Erstellen einer globalen Vorlage für "selection"

Da der Inhalt von `selection` in der Ausgabe nicht erforderlich ist, sollten Sie für `selection` eine leere globale Vorlage erstellen, damit der Inhalt nicht verarbeitet wird. Gehen Sie dazu folgendermaßen vor:

1. Klicken Sie in der Design-Ansicht in der Seitenleiste "Schema-Struktur" 38 mit der rechten Maustaste in der Baumstruktur "Alle globalen Elemente" auf `selection`.
2. Wählen Sie im Kontextmenü den Befehl **Als global definieren / Globale Vorlage löschen**. Es wird nun eine globale Vorlage für `selection` erstellt (*Abbildung unten*).

3. Klicken Sie in der globalen Vorlage auf den Platzhalter `Inhalt` und drücken Sie die **Entf**-Taste auf Ihrer Tastatur. Der Platzhalter "Inhalt" wird gelöscht.
4. Überprüfen Sie die HTML-Vorschau. Der Text `All` ist in der Textausgabe, die von den vordefinierten Vorlagen ausgegeben wird, nicht mehr vorhanden (*Abbildung unten*).

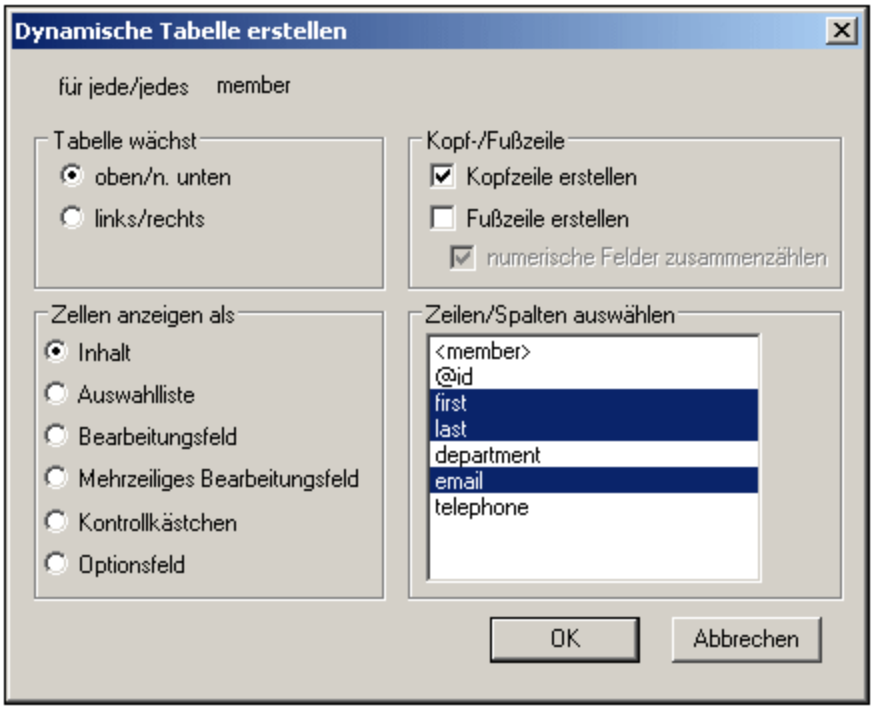
AndrewBentincka.bentinck@nanonull.comNadiaEdwardsn.e

Da die globale Vorlage für `selection` leer ist, werden die Child-Elemente von `selection` nicht verarbeitet.

Erstellen einer globalen Vorlage für "team/member"

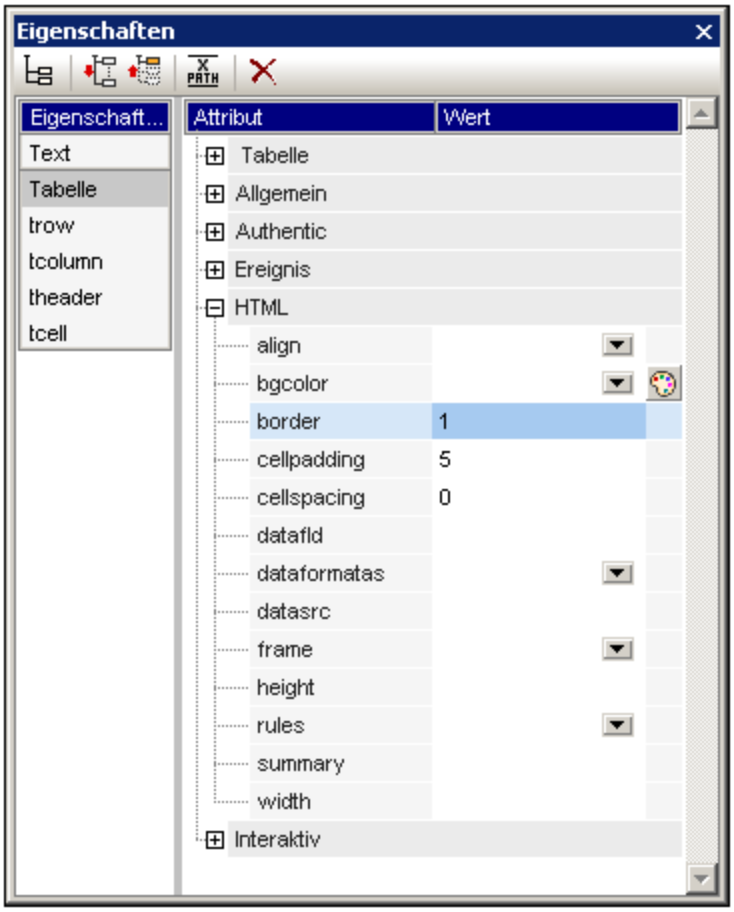
Ziel ist es, eine Tabelle zu erstellen, in der die Informationen über die Mitarbeiter des Presseteams angezeigt werden. Diese Tabelle wird in einer globalen Vorlage für das Element `team` erstellt. Gehen Sie dazu folgendermaßen vor:

1. Erstellen Sie eine globale Vorlage für das Element `team` (rechtsklicken Sie in der Seitenleiste "Schema-Struktur" in der Liste "Alle globalen Elemente" auf `team` und wählen Sie den Befehl **Als global definieren / Globale Vorlage löschen**.
2. Erweitern Sie in der Liste "Alle globalen Elemente" das Element `team` und ziehen Sie dessen Child-Element `member` in die globale Vorlage von `team` (im Design).
3. Wählen Sie im Kontextmenü, das angezeigt wird, sobald Sie das Element in der globalen Vorlage von `team` platzieren, den Befehl **Tabelle erstellen**. Daraufhin wird das Dialogfeld "Dynamische Tabelle erstellen" (*Abbildung unten*) angezeigt.



4. Heben Sie in der Liste der Attribute/Elemente die Auswahl für `@id`, `department` und `telephone` (*siehe Abbildung*) auf und klicken Sie auf **OK**. Die dynamische Tabelle wird erstellt.
5. Platzieren Sie den Cursor in eine Zelle des Tabellenkörpers und geben Sie in der Seitenleiste "Eigenschaften" [49], während `Tabelle` in der Spalte "Eigenschaften für" ausgewählt ist, die Tabelleneigenschaften, wie in der Abbildung unten gezeigt, an.

© 2019-2025 Altova GmbH



6. Stellen Sie in den Seitenleisten "Eigenschaften" und "Stil" weitere Eigenschaften ein. So können Sie z.B. für eine Kopfzeile eine Hintergrundfarbe auswählen. Platzieren Sie dazu den Cursor in die Kopfzeile, während in der Seitenleiste "Stil" in der Spalte "Stile für" der Eintrag "`trow`" ausgewählt ist und definieren Sie einen Wert für die Eigenschaft `background-color` (Gruppe *Farbe*). Sie können auch die statischen Textstrings der Kopfzeile ändern. Wenn der Platzhalter `Inhalt` des Elements `team` in der globalen Vorlage immer noch vorhanden ist, so löschen Sie ihn.

Die HTML-Vorschau der Tabelle sieht ungefähr folgendermaßen aus:

First	Last	Email
Andrew	Bentinck	a.bentinck@nanonull.com
Nadia	Edwards	n.edwards@nanonull.com
John	Edwards	j.edwards@nanonull.com
Janet	Ashe	j.ashe@nanonull.com

3.8 Fertig!

Herzlichen Glückwunsch, Sie haben das Tutorial nun abgeschlossen. Sie sind nun mit den wichtigsten Schritten beim Erstellen eines SPS vertraut:

- Wie man die Struktur des Dokuments erstellt 61 (Hauptvorlage 61 und globale Vorlagen 92).
- Wie man unter Verwendung der verschiedensten dynamischen und statischen SPS-Komponenten dynamischen 61 und statischen 68 Inhalt in das Design einfügt.
- Wie man CSS-Stile 61 in externen Stylesheets 76, globalen Stilregeln 76 und in lokalen Stilregeln 78 verwendet.
- Wie man mittels einer automatischen Berechnung 80 zusätzliche Informationen aus den verfügbaren XML-Daten extrahiert.
- Wie man die XML-Daten mittels Bedingungen 84 filtert und wie man je nach den Werten in den XML-Daten unterschiedliche Ausgabeergebnisse erhält.
- Wien man globale Vorlagen 93 und "restlicher Inhalt" 92 verwendet.

Eine ausführlichere Beschreibung dieser Funktionen finden Sie in den folgenden entsprechenden Abschnitten:

- SPS-Datei: Inhalt 110
- SPS-Datei: Struktur 184
- SPS-Datei: Komplexe Funktionen 256
- SPS-Datei: Darstellung 327
- SPS-Datei: Zusätzliche Funktionen 361

Diese Abschnitte enthalten außerdem Beschreibungen einiger anderer StyleVision Funktionen, die im Schnellstart-Tutorial nicht behandelt werden.

© 2019-2025 Altova GmbH

4 Übersicht über die Verwendung

Ziele

SPS -Dokumente, die Sie in StyleVision erstellen, können zur Generierung von XSLT Stylesheets für HTML-Dokumente verwendet werden. Ein anhand eines SPS generiertes Stylesheet kann auch zum Transformieren jedes beliebigen XML-Dokuments verwendet werden, das auf demselben Schema wie das SPS basiert.

Schritte beim Erstellen eines SPS

Im Folgenden sind die einzelnen Schritte beim Erstellen eines neuen SPS aufgelistet:

1. Zuweisen eines Schemas 38 zu einem neu erstellten leeren SPS. Beim Schema kann es sich um: (i) eine Schemadatei (DTD oder XML-Schema); (ii) ein anhand von einer Datenbank generiertes XML-Schema (*nur Enterprise und Professional Edition*), (iii) ein auf einer XBRL-Taxonomie basierendes Schema (*nur Enterprise Edition*) oder (iii) ein benutzerdefiniertes (direkt in StyleVision erstelltes) Schema handeln. Dies erfolgt in der Seitenleiste "Design-Übersicht" 36. Alternativ dazu kann mit Hilfe des Befehls **Datei | Neu** ein neues SPS direkt mit einem Schema erstellt werden.
2. Zuweisen einer XML-Arbeitsdatei 38 zum SPS. Die XML-Arbeitsdatei 23 stellt die XML-Daten zur Verfügung, die vom SPS beim Generieren der Ausgabeansichten verarbeitet werden. Die XML-Arbeitsdatei 23 wird in der Seitenleiste "Design-Übersicht" 36 zugewiesen. Mit Hilfe der XML-Arbeitsdatei können Sie eine Vorschau der Ausgabe in StyleVision anzeigen. Über die XML-Arbeitsdatei können Sie eine Vorschau der Ausgabe in StyleVision anzeigen.
3. Auswählen der erforderlichen XSLT-Version 100. Für die Textausgabe muss XSLT-Version 2.0 oder 3.0 verwendet werden.
4. Auswahl der Internet Explorer-Kompatibilität 101. Diese muss mit der Version des installierten Internet Explorers übereinstimmen.
5. Das SPS Dokument wird in der Design-Ansicht 29 unter Verwendung der verschiedenen verfügbaren Design-Komponenten erstellt. Der Design-Prozess 99 besteht darin, dass eine Dokumentstruktur erstellt wird und Darstellungseigenschaften 327 definiert werden.
6. Die Ausgabedokumente werden überprüft. Falls Änderungen am Design erforderlich sind, werden diese durchgeführt und das SPS-Dokument wird erneut überprüft.
7. Wenn XSLT-Dateien oder Ausgabedateien 103 benötigt werden, werden diese generiert 103.

4.1 SPS und Quellen

Erstellen einer neuen SPS-Datei

Um ein neues SPS Dokument zu erstellen, wählen Sie eine Option unter **Datei | Neu (Strg+N)** [468] oder klicken Sie in der Standardsymbolleiste [466] auf das Symbol **Neues Design** . Es wird ein neues SPS Dokument erstellt und in der Design-Ansicht angezeigt. Das neue Dokument erhält den vorläufigen Namen `SPSX.sps`, wobei `X` eine Ganzzahl ist, die der Position des SPS Dokuments in der Reihenfolge der neuen seit dem Start der Applikation erstellten Dokumente entspricht.

Nachdem ein neues SPS Dokument erstellt wurde, müssen die Quelldateien für das SPS zugewiesen werden.

Zuweisen von Quelldateien zum SPS

Es gibt zwei Arten von Quelldateien, die einem SPS zugewiesen werden können:

- Schemaquellen [98]
- XML-Arbeitsdatei [98]

Diese Zuweisungen erfolgen in der Seitenleiste "Design-Übersicht" [36]. Eine Beschreibung dazu finden Sie im Abschnitt Design-Übersicht [36]. Die wichtigsten Punkte zu den einzelnen Arten von Quelldateien finden Sie im Folgenden.

Schemaquellen

Einem SPS muss eine Schemaquelldatei zugewiesen werden, damit für das Design-Dokument eine Struktur erstellt werden kann. Schemaquellen werden in der Seitenleiste "Design-Übersicht" [36] zugewiesen. Beim Schema kann es sich um eine XML-Schemadatei (`.xsd` Datei), ein von einer XML-Datei generiertes XML-Schema, eine DTD oder ein benutzerdefiniertes Schema handeln. Für jedes Schema kann eine optionale XML-Arbeitsdatei [98] zugewiesen werden.

Anmerkung: Wenn Sie einen Namespace zu einem SPS oder einem anhand eines SPS generierten XSLT-Stylesheet hinzufügen möchten, muss der Namespace zum `schema`-Element der obersten Ebene des XML-Schemas, auf dem das SPS basiert, hinzugefügt werden.

XML-Arbeitsdatei

kann optional eine damit verknüpfte XML-Arbeitsdatei [23] haben. Die Funktion der XML-Arbeitsdatei [23] ist es, die XML-Datenquelle für Ausgabevorschauen in StyleVision zur Verfügung zu stellen. Sie muss daher entsprechend dem damit verknüpften Schema gültig sein. Die XML-Arbeitsdatei [23] wird in der Seitenleiste "Design-Übersicht" [36] zugewiesen.

© 2019-2025 Altova GmbH

4.2 Erstellen des Designs

Im SPS Design werden folgende Punkte definiert:

1. Welcher Inhalt 99 (aus dem XML-Dokument bzw. der Datenbank) ausgegeben werden soll; zusätzlicher Inhalt kann direkt in das SPS eingefügt werden, um in das Ausgabedokument übernommen zu werden;
2. Wie das Ausgabedokument strukturiert werden soll 99; und
3. Welche Darstellungseigenschaften (Formatierung) 99 auf die verschiedenen Teile des Ausgabedokuments angewendet werden sollen.

Inhalt für die Ausgabe

Der Inhalt für die Ausgabe kann aus folgenden Quellen stammen:

1. dem XML-Dokument , auf das/die das SPS angewendet wird. Inhalt aus dem XML-Dokument 24 wird durch Ziehen des gewünschten XML-Daten-Node aus der jeweiligen Schemastruktur in der Seitenleiste "Schema-Struktur" 38 an die gewünschte Stelle ins SPS eingefügt.
2. einem externen XML-Dokument, auf das die Applikation (also StyleVision) Zugriff hat. Mit Hilfe der doc() Funktion von XPath 2.0 in einer automatischen Berechnung kann Inhalt aus externen XML-Dokumentquellen aufgerufen werden. Ein XML-Dokument, auf das über die doc() Funktion eines XPath-Ausdrucks aufgerufen wird, muss nicht über die Schemaquellen 38 Assoziation referenziert werden.
3. dem SPS selbst. Text und anderer Inhalt (wie z.B. Bilder und Tabellen) kann entweder über die Tastatur oder andere Funktionen der Benutzeroberfläche direkt in das SPS eingefügt werden. Solche Eingaben sind unabhängig vom XML-Dokument.
4. bearbeiteten dynamischen (XML-Quell)-Daten, wobei die Bearbeitungen mittels XPath-Ausdrücken vorgenommen werden. Bearbeitungen werden normalerweise durch automatische Berechnungen 257 vorgenommen.
5. Bei der HTML-Ausgabe kann Inhalt mittels JavaScript-Funktionen 403 generiert werden.

Struktur der Ausgabe

Im SPS Design kann die Struktur des Ausgabedokuments 230 auf folgende Arten gesteuert werden: (i) entweder durch einen prozeduralen Ansatz. Dabei wird die Struktur des Ausgabedokuments in einer Vorlage 230 (der StyleVision Hauptvorlage 230) definiert und kann unabhängig von der Struktur des XML-Dokuments sein; (ii) oder durch einen deklarativen Ansatz, wobei die Vorlagenregeln für verschiedene Nodes deklariert werden 230 (globale Vorlagen 230 in StyleVision), wodurch ein Ausgabedokument generiert wird, dessen Struktur die des XML-Dokuments wiedergibt; oder (iii) durch eine Kombination des prozeduralen und des deklarativen Ansatzes. In der Design-Ansicht können Sie die gewünschte Struktur für das Ausgabedokument durch eine Kombination aus der Hauptvorlage 230 und globalen Vorlagen 230 erstellen. Modulare SPSs 216 und Design-Fragmente 242 bieten mehr Flexibilität bei der Strukturierung eines SPS.

Darstellung (oder Formatierung) des Ausgabedokuments

In der Design-Ansicht werden Darstellungseigenschaften mittels CSS-Stilen auf Design-Komponenten angewendet. Stile können lokal für die Komponente, für auf Dokumentebene deklarierte HTML-Selektoren und für in einem externen CSS Stylesheet deklarierte HTML-Selektoren definiert werden. Zusätzlich dazu können bestimmte HTML-Elemente mittels vordefinierter Formate 328 auf Komponenten angewendet werden. Die Definition von Darstellungseigenschaften wird im Abschnitt Darstellungseigenschaften 327 näher beschrieben.

4.3 XSLT- und XPath-Versionen

Ein SPS ist im Grunde ein XSLT Stylesheet. Für jedes SPS muss die XSLT-Version angegeben werden: 1.0, 2.0 oder 3.0. Dies geschieht, indem Sie in der Symbolleiste auf die entsprechende Schaltfläche klicken: oder oder . Die Auswahl, die Sie vornehmen, legt zweierlei fest:

- Welcher der drei XSLT-Prozessoren in StyleVision für Transformationen verwendet wird; StyleVision hat einen separaten XSLT 1.0-, XSLT 2.0- und XSLT 3.0-Prozessor.
- Welche XSLT-Funktionalität (1.0, 2.0 oder 3.0) auf der Benutzeroberfläche angezeigt und im SPS zugelassen werden soll. So wird von XSLT 3.0 z.B. XPath 3.0 verwendet, eine viel umfangreichere Sprache als XPath 1.0 (welches in XSLT 1.0 verwendet wird) oder XPath 2.0 (welches in XSLT 2.0 verwendet wird). Zusätzlich dazu stehen einige SPS-Features wie z.B. die Funktion "Inhaltsverzeichnis" nur bei XSLT 2.0 und XSLT 3.0 zur Verfügung.

Anmerkung: Für die Generierung der Textausgabe muss XSLT 2.0 oder XSTL 3.0 verwendet werden.

XSLT-Transformationen

XSLT-Transformationen in StyleVision dienen dazu um: (i) auf der Benutzeroberfläche eine Ausgabevorschau 31 zu erstellen; und (ii) um über die Benutzeroberfläche 486 und über StyleVision Server Ausgabedateien zu generieren und zu speichern 103 (HTML). Der für die Transformationen verwendete XSLT-Prozessor (Altova XSLT 1.0, 2.0 oder 3.0-Prozessor) entspricht der im SPS ausgewählten XSLT-Version.

XSLT-Funktionalitäten auf der Benutzeroberfläche

Die für die jeweilige XSLT-Version geeigneten Funktionalitäten stehen hauptsächlich mit der korrekten XPath-Version (XPath 1.0 für XSLT 1.0, XPath 2.0 für XSLT 2.0, XPath 3.0 für XSLT 3.0) in Zusammenhang. XPath-Ausdrücke werden in StyleVision häufig verwendet - v.a. in Funktionen wie automatischen Berechnungen 257 und Bedingungsvorlagen 262. Es gibt Methoden, für die XPath-Ausdrücke erforderlich sind und die Ihnen beim Erstellen solcher Ausdrücke behilflich sind. Die Funktionalität der korrekten XPath-Version steht abhängig von der gewählten XSLT-Version automatisch auf der Benutzeroberfläche zur Verfügung.

© 2019-2025 Altova GmbH

4.4 Kompatibilität mit Internet Explorer

Damit das SPS-Design (in der Design-Ansicht) und die Ausgabevorschau (in der HTML-Vorschau) korrekt angezeigt wird, muss auf Ihrem StyleVision-Rechner Internet Explorer installiert sein. In Folgenden finden Sie Angaben zu den unterstützten IE-Versionen:

- Internet Explorer 5.5 oder höher
- Internet Explorer 6.0 und höher bietet bessere XML-Unterstützung und wird empfohlen.
- Internet Explorer 9 (IE9) oder höher bietet zusätzliche Features wie z.B. Unterstützung für weitere Bildformate und neue CSS-Stile. Wenn Sie diese zusätzlichen Features in Ihrem Design verwenden möchten, sollten Sie eventuell IE9 installieren.

Unterstützung für die Funktionen von IE9 in StyleVision

Die folgenden Funktionen von IE9 oder höheren Versionen werden in StyleVision unterstützt:

- Zusätzlich unterstützte Bildformate: TIFF, JPEG XR und SVG. (SVG-Dokumente müssen im XML-Format gespeichert und im SVG Namespace sein.) Diese Bildformate werden in IE9, nicht aber in älteren Versionen von IE angezeigt. Eine vollständige Liste der in den verschiedenen Ausgabeformaten unterstützten Bildformate finden Sie unter Bildtypen und Ausgabe 154.
- Unterstützung für die unten aufgelisteten neuen CSS-Stile (u.a. auch CSS3-Stile). Die Anwendung dieser Stile ist auf die HTML-Ausgabe beschränkt.

  - `background-clip`
  - `background-origin`
  - `background-size`
  - `box-sizing`
  - `box-shadow`
  - `border-radius (border-*-radius)`
  - `font-stretch`
  - `ruby-align`
  - `ruby-overhang`
  - `ruby-position`
  - `overflow-x, overflow-y`
  - `outline (outline-color, outline-style, outline-width)`
  - `text-align-last (partial)`
  - `text-overflow (partial)`

- Unterstützung für die neue CSS-Längenfunktion `calc()`
- Unterstützung für die neuen CSS-Farbfunktionen `rgba()`, `hsl()` und `hsla()`
- Unterstützung für die neuen CSS-Längeneinheiten `rem`, `vw`, `vm`, `vh` und `ch`
- HTML5-Elemente, die von IE9 unterstützt werden, können als benutzerdefinierte Elemente 123 in das Design eingefügt werden.

Design-Ansicht und IE-Versionen

Sie können die Design-Ansicht für eine bestimmte IE-Version konfigurieren. Definieren Sie dazu im Dialogfeld "Eigenschaften 489" die IE-Version, mit der die Design-Ansicht kompatibel sein soll. Dies wirkt sich folgendermaßen aus:

- Alle CSS-Stile, die von der ausgewählten IE-Version dargestellt werden können, werden automatisch in der Seitenleiste "Stil" von StyleVision angezeigt. (Beachten Sie allerdings, dass IE9 bei Auswahl von IE9 installiert sein muss, damit die unterstützten CSS-Stile auf der Design-Oberfläche zur Verfügung stehen.) Wenn z.B. IE9 installiert und als kompatible Version ausgewählt ist, so stehen auf der Design-Oberfläche die in IE9 unterstützten CSS3-Stile zur Verfügung.
- Die HTML-Elemente der entsprechenden ausgewählten IE-Version können als vordefinierte Formate 113 oder als benutzerdefinierte Elemente 123 eingegeben werden. Das HTML-Element wird in der HTML-Vorschau so dargestellt, wie die installierte IE-Version das Element unterstützt. Wenn z.B. IE9 installiert und als kompatible Version ausgewählt ist, so werden die unterstützten HTML5-Elemente in der HTML-Vorschau dargestellt.

Konfigurieren des Design-Ansicht für eine bestimmte IE-Version

Um die Design-Ansicht für eine bestimmte IE-Version zu konfigurieren, wählen Sie den Menübefehl Datei | Eigenschaften und wählen Sie auf dem Register "Ausgabe" die gewünschte IE-Version (Kompatibilität) aus. Nähere Informationen dazu finden Sie unter Datei | Eigenschaften. 489

Kompatibilität älterer SPS-Designs mit IE9

Wenn Sie ein SPS-Design öffnen, das für eine ältere IE-Version erstellt wurde und wenn die neuere IE Version 9 oder eine höhere Version auf Ihrem StyleVision-Rechner installiert ist, so erkennt StyleVision die neuere Version und fragt Sie in einem Dialogfeld, ob Sie die Kompatibilität in IE9-Kompatibilität ändern möchten. Wenn Sie zur neuen Kompatibilität wechseln, stehen die oben aufgelisteten Optionen zur Verfügung. Die Darstellung des Dokuments in der Design-Ansichtund der HTML-Ausgabe bleibt mit Ausnahme von **Tabellenspalten,** die in IE9 anders behandelt werden, unverändert. Wenn Sie die Kompatibilität in IE9-Kompatibilität umändern, überprüfen Sie, ob die Tabellenspalten so generiert werden, wie dies gewünscht wird. Falls dies nicht der Fall ist, können Sie die Eigenschaften der Tabellenspalten ändern oder die IE-Kompatibilität über das Dialogfeld Eigenschaften 489 wieder in die zuvor ausgewählte Version ändern.

© 2019-2025 Altova GmbH

4.5 Generierte Dateien

In StyleVision können XSLT Stylesheets und Ausgabedateien über den Befehl **Datei | Generierte Dateien speichern** 486 oder über StyleVision Server generiert werden.

In StyleVision können die folgenden Dateien generiert werden:

- XSLT Stylesheets auf Basis des SPS Designs.
- Ausgabedateien, die durch Verarbeitung der im SPS zugewiesenen XML-Arbeitsdatei 23 generiert werden, wobei anhand des SPS das XSLT Stylesheet generiert wird.

Die Markup-Tags für die Ausgabe sind im SPS enthalten. Die Daten für die Ausgabe stammen aus dem XML-Dokument. Im XSLT Stylesheet für die Ausgabe werden die Markup-Tags und die Daten zusammengebracht. Sie können die XSLT Stylesheets und die aktuelle Ausgabedatei in StyleVision in den Ausgabevorschauen 31 anzeigen.

Anmerkung: Wenn Sie einen Namespace zu einem SPS oder einem anhand eines SPS generierten XSLT-Stylesheet hinzufügen möchten, muss der Namespace zum `schema`-Element der obersten Ebene des XML-Schemas, auf dem das SPS basiert, hinzugefügt werden.

Altova Website: XML-Berichterstellung

Ausgabedokumente

Nachfolgend einige wichtige Anmerkungen zu den generierten Dokumenten:

- ***HTML-Ausgabe und Stylesheets:*** (1) Formatierung und Layout der generierten HTML-Ausgabe entsprechen exakt der HTML-Vorschau von StyleVision. (2) Die Dateneingabeelemente (Texteingabefelder, Kontrollkästchen usw.) in der HTML-Datei gestatten keine Eingabe. Diese Eingabeelemente sind für die Eingabe von XML-Daten in der Authentic-Ansicht gedacht. Sie werden zwar unverändert in das grafische HTML-Äquivalent übersetzt, können im HTML-Dokument aber nicht zur Dateneingabe verwendet werden.
  Altova Website: XML in HTML

4.6 Kataloge in StyleVision

Mit Hilfe des XML-Katalogmechanismus können Dateien aus lokalen Ordnern abgerufen und so die Verarbeitungsgeschwindigkeit gesteigert werden. Gleichzeitig wird dadurch die Übertragbarkeit von Dokumenten verbessert, da in diesem Fall nur die Katalogdatei-URIs geändert werden müssen. Nähere Informationen dazu finden Sie im Abschnitt Funktionsweise von Katalogen 104.

In den XML-Produkten von Altova kommt ein Katalogmechanismus zur Anwendung, mit dem gemeinsam verwendete Dateien wie DTDS und XML-Schemas schnell aufgerufen und geladen werden können. Dieser Katalogmechanismus kann vom Benutzer angepasst und erweitert werden und ist in den Abschnitten Katalogstruktur in StyleVision 105 und Anpassen von Katalogen 107 beschrieben. Im Abschnitt Variablen für Windows-Systempfade 108 sind Windows-Variablen für gemeinsam benutzte Systempfade aufgelistet. Diese Variablen können in Katalogdateien verwendet werden, um gemeinsam verwendete Ordner aufzurufen.

Dieser Abschnitt ist in die folgenden Unterabschnitte gegliedert:

- Funktionsweise von Katalogen 104
- Katalogstruktur in StyleVision 105
- Anpassen von Katalogen 107
- Variablen für Windows-Systempfade 108

Nähere Informationen zu Katalogen finden Sie in der XML-Katalogspezifikation.

4.6.1 Funktionsweise von Katalogen

Mit Hilfe von Katalogen können Umleitungen sowohl zu DTDs als auch XML-Schemas definiert werden. Das Prinzip des Mechanismus ist in beiden Fällen dasselbe, doch unterscheidet er sich in einigen weiter unten beschriebenen Details.

DTDs

Kataloge dienen normalerweise dazu, einen Aufruf von einer DTD an eine lokale URI umzuleiten. Dies geschieht in der Katalogdatei durch Mappen von Public und System Identifiern auf die gewünschte lokale URI. Wenn also die **DOCTYPE**-Deklaration in einer XML-Datei gelesen wird, findet ihr Public oder System Identifier über das Katalogdatei-Mapping die gewünschte lokale Ressource.

Für gebräuchliche Schemas ist der **PUBLIC** Identifier normalerweise vordefiniert, sodass für die URI in der Katalogdatei nur der **PUBLIC** Identifier auf den korrekten lokalen Identifier gemappt werden muss. Wenn das XML-Dokument geparst wird, wird der **PUBLIC** Identifier darin gelesen. Wenn dieser Identifier in einer Katalogdatei gefunden wird, wird die entsprechende URL in der Katalogdatei nachgeschlagen und das Schema wird von dort aus gelesen. Wenn also die folgende SVG-Datei in StyleVision geöffnet wird:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg width="20" height="20" xml:space="preserve">
  <g style="fill:red; stroke:#000000">
     <rect x="0" y="0" width="15" height="15"/>
```

 © 2019-2025 Altova GmbH

```
    <rect x="5" y="5" width="15" height="15"/>
  </g>
</svg>
```

Der Katalog wird nach dem **PUBLIC** Identifier dieser SVG-Datei durchsucht. Angenommen, die Katalogdatei enthält den folgenden Eintrag:

```
<catalog>
  ...
    <public publicId="-//W3C//DTD SVG 1.1//EN" uri="schemas/svg/svg11.dtd"/>
  ...
</catalog>
```

In diesem Fall gibt es einen Treffer für den **PUBLIC** Identifier, sodass der Lookup-Mechanismus für die SVG DTD an die URL **schemas/svg/svg11.dtd** umgeleitet wird; (dieser Pfad ist relativ zur Katalogdatei). Diese lokale Datei wird dann als DTD für die SVG-Datei verwendet. Wenn im Katalog kein passender Treffer für den **Public** Identifier gefunden wird, wird die URL im XML-Dokument verwendet (im SVG Beispiel oben ist dies die Internet URL **http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd**).

XML-Schemas

In StyleVision können Sie eine **Umleitung zu einem XML-Schema** auch mit Hilfe von Katalogen durchführen. In der XML-Instanzdatei erfolgt die Referenz zum Schema im **xsi:schemaLocation** Attribut des Elements der obersten Ebene des XML-Dokuments. Beispiel:

```
xsi:schemaLocation="http://www.xmlspy.com/schemas/orgchart OrgChart.xsd"
```

Der Wert des **xsi:schemaLocation**-Attributs besteht aus zwei Teilen: einem Namespace-Teil (oben grün) und einem URI-Teil (markiert). Anhand des Namespace-Teils erfolgt im Katalog das Mapping auf die alternative Ressource. So erfolgt etwa im folgenden Katalogeintrag eine Umleitung der Schemareferenz oben auf ein Schema unter einem anderen Pfad.

```
<uri name="http://www.xmlspy.com/schemas/orgchart" uri="C:\MySchemas\OrgChart.xsd"/>
```

Normalerweise ist der URI-Teil des Attributwerts von **xsi:schemaLocation** ein Pfad zum aktuellen Schema. Wenn das Schema jedoch über einen Katalog referenziert wird, muss der URI-Teil nicht auf ein tatsächliches XML-Schema verweisen, muss aber vorhanden sein, damit das Attribut **xsi:schemaLocation** lexikalisch gültig ist. So würde z.B. der Wert `foo` als URI-Teil des Attributwerts genügen, um gültig zu sein.

4.6.2 Katalogstruktur in StyleVision

StyleVision lädt beim Start eine Datei namens **RootCatalog.xml**(*Struktur siehe unten*), die eine Liste von Katalogdateien enthält, die durchsucht werden. Sie können diese Datei bearbeiten und beliebig viele Katalogdateien definieren, die durchsucht werden sollen. Jede davon wird in einem **nextCatalog** referenziert. Diese Katalogdateien werden durchsucht und die URIs darin werden entsprechend ihren Mappings aufgelöst.

Codefragment von RootCatalog.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<catalog xmlns="urn:oasis:names:tc:entity:xmlns:xml:catalog"
   xmlns:spy="http://www.altova.com/catalog_ext"
```

```
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:entity:xmlns:xml:catalog Catalog.xsd">
  <nextCatalog catalog="%PersonalFolder%/Altova/%AppAndVersionName%/CustomCatalog.xml"/>
  <!-- Include all catalogs under common schemas folder on the first directory level -->
  <nextCatalog spy:recurseFrom="%CommonSchemasFolder%" catalog="catalog.xml"
spy:depth="1"/>
  <nextCatalog spy:recurseFrom="%ApplicationWritableDataFolder%/pkgs/.cache"
catalog="remapping.xml" spy:depth="0"/>
  <nextCatalog catalog="CoreCatalog.xml"/>
</catalog>
```

Im obigen Codefragment wird ein benutzerdefinierter Katalog namens **CustomCatalog.xml** und eine Gruppe von Katalogen, die gebräuchliche Schemas (wie z.B. W3C XML-Schemas udn das SVG-Schema) referenzieren, referenziert.

- **CustomCatalog.xml** befindet sich in Ihrem persönlichen Ordner (der über die Variable **%PersonalFolder%**) gefunden wird. erstellen. Diese Datei ist eine Skelettdatei, in der Sie Ihre eigenen Mappings erstellen können. Sie können für jedes gewünschte Schema, das nicht in den Katalogdateien im Altova Ordner "Common Schemas" enthalten ist, Mappings zu **CustomCatalog.xml** hinzufügen. Verwenden Sie dazu die unterstützten Elemente des OASIS-Katalogmechanismus (*siehe nächster Abschnitt*).
- Der durch die Variable **%CommonSchemasFolder%** definierte Ordner "Common Schemas" enthält eine Reihe gebräuchlicher Schemas. Innerhalb dieser einzelnen Schema-Ordner befindet sich eine **catalog.xml**-Datei, die Public und/oder System Identifier auf URIs mappt, die auf lokal gespeicherte Kopien des jeweiligen Schemas verweisen.
- **CoreCatalog.xml** befindet sich im StyleVision-Applikationsordner und dient zum Auffinden von Schemas und Stylesheets, die von StyleVision-spezifischen Prozessen wie z.B. StyleVision Power Stylesheets, anhand derer die Altova-Authentic-Ansicht von XML-Dokumenten generiert wird.

Pfadvariablen
Die in **RootCatalog.xml** verwendeten Variablen (*Codefragment oben*) haben die folgenden Werte:

%PersonalFolder%	Der persönliche Ordner des aktuellen Benutzers, z.B. C:\Benutzer\<Name>\Dokumente
%CommonSchemasFolder%	C:\ProgramData\Altova\Common2025\Schemas
%ApplicationWritableDataFolder%	C:\ProgramData\Altova

Speicherpfad von Katalogdateien und Schemas
Beachten Sie die Pfade der verschiedenen Katalogdateien.

- Die Dateien **RootCatalog.xml** und **CoreCatalog.xml** befinden sich im StyleVision Applikationsordner.
- **CustomCatalog.xml** befindet sich in Ihrem Ordner **EigeneDokumente\Altova\StyleVision**.
- Die **catalog.xml**-Dateien befinden sich jeweils in einem eigenen Schemaordner, wobei sich diese Schemaordner innerhalb des Ordners "Common Schemas" befinden:

© 2019-2025 Altova GmbH

4.6.3 Anpassen von Katalogen

Wenn Sie Einträge für **CustomCatalog.xml** (oder jede andere Katalogdatei, die von StyleVision gelesen werden soll) erstellen, verwenden Sie nur die folgenden Elemente der OASIS-Katalogspezifikation. Jedes der unten angeführten Elemente wird mit einer Erläuterung der Attributwerte aufgelistet. Eine ausführlichere Beschreibung finden Sie in der XML Catalogs Specification. Beachten Sie, dass jedes Element das Attribut **xml:base**, mit dem die Basis-URI dieses Elements definiert wird, erhalten kann.

- <public publicId="PublicID of Resource" uri="URL of local file"/>
- <**system** **systemId**="SystemID of Resource" **uri**="URL of local file"/>
- <**uri** **name**="filename" **uri**="URL of file identified by filename"/>
- <**rewriteURI** **uriStartString**="StartString of URI to rewrite" **rewritePrefix**="String to replace StartString"/>
- <**rewriteSystem** **systemIdStartString**="StartString of SystemID" **rewritePrefix**="Replacement string to locate resource locally"/>

Beachten Sie die folgenden Punkte:

- In Fällen, in denen es keinen Public Identifier gibt, kann der System Identifier einer URL wie z.B. bei den meisten Stylesheets über das **system** Element direkt auf eine URL gemappt werden.
- Eine URI kann über das uri Element auf eine andere URI gemappt werden.
- Mit Hilfe der Elemente rewriteURI und rewriteSystem kann der Anfangsteil einer URI bzw. eines System Identifiers neu geschrieben werden. Dadurch kann der Anfang eines Dateipfads ersetzt werden, sodass ein anderes Verzeichnis als Ziel gewählt werden kann. Nähere Informationen zu diesen Elementen finden Sie in der XML Catalogs Specification.

Ab Release 2014 entspricht StyleVision weitgehend der XML Catalogs Specification (OASIS Standard V1.1, 7 Oktober 2005). In dieser Spezifikation wird streng zwischen externen Identifier Look-ups (jenen mit einer öffentlichen ID oder einer System-ID) und URI Look-ups (URIs, die keine öffentlichen IDS oder System-IDs sind) getrennt. Namespace URIs müssen daher einfach als URIs - und nicht Public IDs oder System-IDs - behandelt werden und folglich als URI Look-ups anstelle von externen Identifier Look-ups verwendet werden. In StyleVision Versionen vor Version 2014 wurden Schema Namespace URIs über **<public>** Mappings übersetzt. Ab Version 2014 müssen **<uri>** Mappings verwendet werden.

Vor v2014: <public **publicID**="http://www.MyMapping.com/ref"
uri="file:///C:/MyDocs/Catalog/test.xsd"/>
Ab V-2014: <**uri** **name**="http://www.MyMapping.com/ref"
uri="file:///C:/MyDocs/Catalog/test.xsd"/>

Wie findet StyleVision ein referenziertes Schema

Ein Schema wird in einem XML-Dokument über das Attribut **xsi:scemaLocation** (*siehe unten*) referenziert. Der Wert des **xsi:schemaLocation**-Attributs besteht aus zwei Teilen: einem Namespace-Teil (grün) und einem URI-Teil (markiert).

xsi:schemaLocation="**http://www.xmlspy.com/schemas/orgchart** **OrgChart.xsd**"

Im Folgenden wird Schritt für Schritt beschrieben, wie StyleVision vorgeht, um ein referenziertes Schema zu finden: Das Schema wird beim ersten erfolgreichen Schritt geladen.

1. Durchsuchen des Katalogs nach dem URI-Teil des **xsi:schemaLocation**-Werts. Wenn ein Mapping

gefunden wird - darunter auch in **rewriteURI** Mappings wird die erzeugte URI verwendet, um das Schema zu laden.
2. Durchsuchen des Katalogs nach dem Namespace-Teil des **xsi:schemaLocation**-Werts. Wenn ein Mapping gefunden wird - darunter auch in **rewriteURI** Mappings wird die erzeugte URI verwendet, um das Schema zu laden.
3. Der URI-Teil des **xsi:schemaLocation**-Werts wird zum Laden des Schemas verwendet.

XML-Schema-Spezifikationen

Die XML-Schemaspezifikationsinformationen sind in StyleVision integriert und die Gültigkeit von XML-Schema-(`.xsd`)-Dokumenten wird anhand dieser internen Informationen überprüft. Daher sollte in einem XML-Schema-Dokument kein Verweis auf ein Schema, das die XML-Schema-Spezifikation definiert, vorgenommen werden.

Die Datei **catalog.xml** im Ordner `%AltovaCommonSchemasFolder%\Schemas\schema` enthält Referenzen auf DTDs, die ältere XML-Schema-Spezifikationen implementieren. Sie sollten Ihre XML-Schema-Dokumente nicht anhand dieser Schemas validieren. Zweck dieser beiden DTDs ist es einzig und allein, für die Eingabehilfen von StyleVision zu Bearbeitungszwecken Informationen bereitzustellen, falls Sie Dateien gemäß diesen älteren Empfehlungen erstellen wollen.

4.6.4 Variablen für Windows-Systempfade

Shell-Umgebungsvariablen können im `nextCatalog` Element verwendet werden, um den Pfad zu Systemordnern zu definieren (*siehe RootCatalog.xml-Liste oben*). Es werden die folgenden Shell-Umgebungsvariablen unterstützt:

`%PersonalFolder%`	Vollständiger Pfad zum persönlichen Ordner des aktuellen Benutzers, z.B. `C:\Benutzer\<Name>\Dokumente`
`%CommonSchemasFolder%`	`C:\ProgramData\Altova\Common2025\Schemas`
`%ApplicationWritableDataFolder%`	`C:\ProgramData\Altova`
`%AltovaCommonFolder%`	C:\Programme\Altova\Common2025
`%DesktopFolder%`	Vollständiger Pfad des Desktop-Ordners des aktuellen Benutzers.
`%ProgramMenuFolder%`	Vollständiger Pfad zum Ordner 'Programme" des aktuellen Benutzers.
`%StartMenuFolder%`	Vollständiger Pfad zum Startmenüordner des aktuellen Benutzers.
`%StartUpFolder%`	Vollständiger Pfad zum Startordner des aktuellen Benutzers.
`%TemplateFolder%`	Vollständiger Pfad des Template-Ordners des aktuellen Benutzers.

© 2019-2025 Altova GmbH

%AdminToolsFolder%	Vollständiger Pfad zum Dateisystemverzeichnis, in dem Verwaltungstools des aktuellen Benutzers gespeichert sind.
%AppDataFolder%	Vollständiger Pfad zum Ordner "Anwendungsdaten" des aktuellen Benutzers.
%CommonAppDataFolder%	Vollständiger Pfad zum Dateisystem, das die Applikationsdaten aller Benutzer enthält.
%FavoritesFolder%	Vollständiger Pfad zum Ordner 'Favoriten' des aktuellen Benutzers.
%PersonalFolder%	Vollständiger Pfad zum Ordner "Personal" des aktuellen Benutzers.
%SendToFolder%	Vollständiger Pfad zum SendTo-Ordner des aktuellen Benutzers.
%FontsFolder%	Vollständiger Pfad zum Ordner "Systemschriftarten".
%ProgramFilesFolder%	Vollständiger Pfad zum Ordner 'Programme" des aktuellen Benutzers.
%CommonFilesFolder%	Vollständiger Pfad zum Ordner "Gemeinsame Dateien" des aktuellen Benutzers.
%WindowsFolder%	Vollständiger Pfad zum Ordner "Windows" des aktuellen Benutzers.
%SystemFolder%	Vollständiger Pfad zum Ordner "System" des aktuellen Benutzers.
%LocalAppDataFolder%	Vollständiger Pfad zum Dateisystemverzeichnis, das als Datenspeicher für lokale (nicht-Roaming) Applikationen dient.
%MyPicturesFolder%	Vollständiger Pfad zum Ordner "Meine Bilder".

5 SPS-Inhalt

In diesem Abschnitt werden die grundlegenden Verfahren zum Erstellen und Bearbeiten der Komponenten eines SPS im Detail beschrieben. Nachfolgend sehen Sie eine Liste der Verfahren, die in den Unterabschnitten dieses Abschnitts näher beschrieben werden. Die Verfahren dienen zum Erstellen jeder Art von Vorlage: Hauptvorlage[230], globale Vorlage[231] oder Design-Fragment[242].

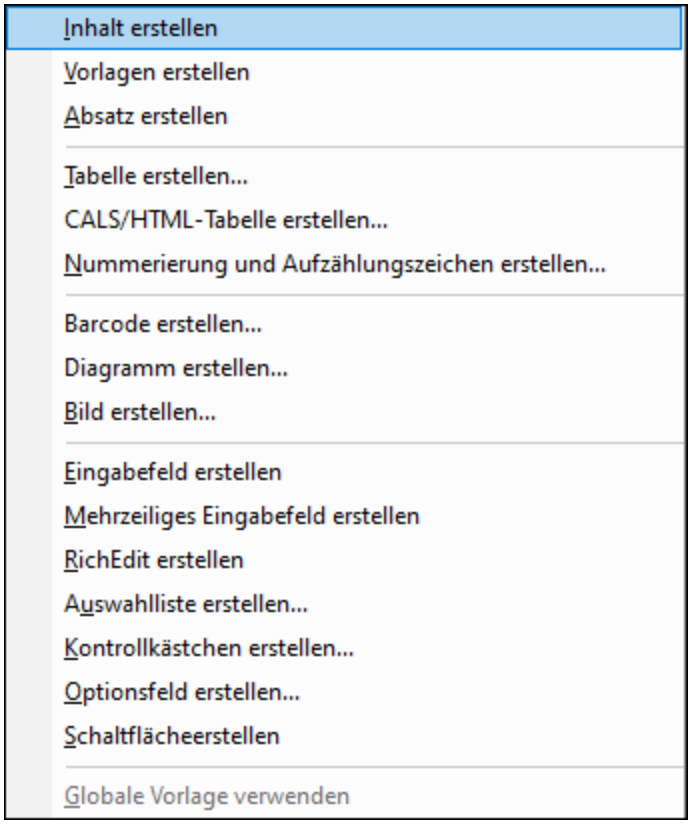
- Einfügen von XML-Inhalt als Text[111]. Sie können XML-Daten durch Ziehen der relevanten Nodes (Element, Attribut, Typ oder CDATA) in das Design und Erstellen der Nodes als `(Inhalt)` oder `(restlicher Inhalt)` einfügen.
- Einfügen von Inhalt aus MS Word[115]
- Arbeiten mit Tabellen[126]. Tabellen können (i) vom Ersteller des SPS (durch direktes Einfügen in das SPS Design (statische Tabellen) oder unter Verwendung von XML-Dokument-Substrukturen) und (ii) vom Benutzer der Authentic-Ansicht eingefügt werden.
- Erstellen von Listen[147]. Statische Listen, bei denen die Listenstruktur in das SPS Design eingegeben wird, und dynamische Listen, bei denen die Substruktur eines XML-Dokuments als Liste erstellt wird, bieten umfangreiche Funktionen zum Ordnen von Daten.
- Verwendung von Grafiken[152]: Grafiken können mit den verschiedensten Methoden zur Definition der Ziel-URI (statisch, dynamisch, eine Kombination aus beiden und ungeparste Entity URIs) in das SPS Design eingefügt werden
- Verwendung von Dateneingabe- (oder Formularsteuer) -Elementen[158]. XML-Daten können in der Authentic-Ansicht über Dateneingabeelemente wie Eingabefelder und Auswahllisten eingegeben werden. Auf diese Art stehen dem Benutzer Hilfsmittel zur Verfügung, die die Eingabemöglichkeiten einschränken. Einzelne Nodes im XML-Dokument können als Dateneingabeelemente erstellt werden.

- Links[165]
- Barcodes[166]
- Layout-Module[171]
  - Die Funktion "Ändern in"[181]. Mit Hilfe dieser Funktion können Sie einen anderen Node für eine Vorlage auswählen. Außerdem kann ein Node in einen anderen Content Type umgewandelt werden.

© 2019-2025 Altova GmbH

5.1 Einfügen von XML-Inhalt als Text

Daten aus einem Node im XML-Dokument werden im Design durch Ziehen des entsprechenden Schema-Node aus dem Fenster "Schema-Struktur" in das Design eingefügt. Beim Loslassen der Maustaste erscheint ein Popup-Menü mit Optionen, wie der Node im Design erstellt werden soll (*Abbildung unten*).



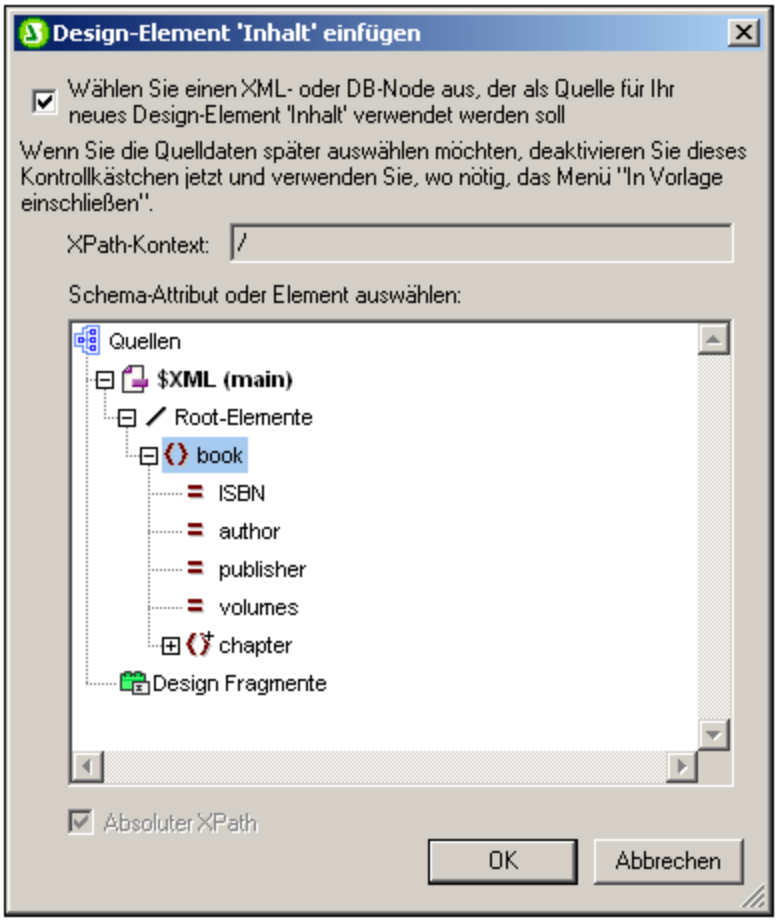
Typen von Schema-Nodes

Schema-Nodes, die aus der Seitenleiste "Schema-Struktur" in das Design gezogen werden können, können drei Typen angehören: (i) Element-Nodes; (ii) Attribut-Nodes; und (iii) Datentyp-Nodes.

Verwendung der Symbolleisten-Schaltfläche "Inhalt einfügen"

Auch über die Schaltfläche **Inhalt einfügen** der Symbolleiste "Design-Elemente einfügen" 462 können Sie den Inhalt eines Node in das Design einfügen. Gehen Sie dabei folgendermaßen vor:

1. Klicken Sie auf die Schaltfläche "Inhalt einfügen".
2. Klicken Sie auf die Stelle im Design, an der der Inhalt eingefügt werden soll. Daraufhin erscheint der Selektor zum Einfügen des Inhalts. (*Abbildung unten*).



3. Der Kontext der Einfügestelle im Design wird im Feld *XPath-Kontext* angezeigt. Wählen Sie den Node, für den Sie Inhalt erstellen möchten.
4. Klicken Sie auf **OK**. Daraufhin wird der Platzhalter `Inhalt` erstellt. Wenn es sich beim ausgewählten Node um einen anderen als den Kontext-Node handelt, werden rund um den Platzhalter `Inhalt` zusätzliche Vorlagen-Tags mit dem Pfad zum ausgewählten Node eingefügt.

Ausgabe des Textinhalts von Nodes

Um Textinhalt des Node auszugeben, sollte der Node als "Inhalt" erstellt werden. Wenn ein Node als Inhalt erstellt wird, sieht er im Design-Dokument in etwa folgendermaßen aus.:

In der Abbildung oben wurde das Element `Desc` als Inhalt erstellt. In der Ausgabe wird der Textinhalt von `Desc` angezeigt. Wenn `Desc` untergeordnete Elemente hat wie z.B. `Bold` und `Italic`, dann wird der Textinhalt der untergeordneten Elemente ebenfalls als Teil des Inhalts von `Desc` ausgegeben. Beachten Sie, dass Attribut-Nodes von `Desc` nicht als Child Nodes betrachtet werden, daher wird der Inhalt der Attribut-Nodes nicht als Teil des Inhalts von `Desc` ausgegeben. Attribut-Nodes müssen explizit eingefügt werden, um verarbeitet zu werden.

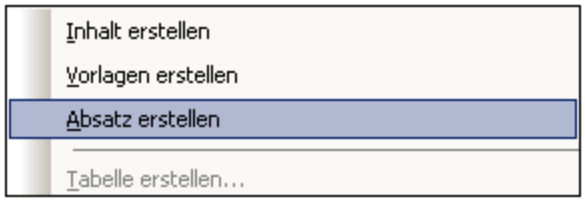
© 2019-2025 Altova GmbH

CDATA-Abschnitte

Wenn im XML-Dokument CDATA-Abschnitte vorhanden sind, werden sie ausgeben.

5.1.1 Einfügen von Inhalt mit einem vordefinierten Format

Der Textinhalt eines Node kann direkt mit den Markup-Tags eines der in StyleVision vordefinierten Formate eingefügt werden. Ziehen Sie den Node dazu aus dem Fenster "Schema-Struktur" an die gewünschte Stelle. Wählen Sie im Popup-Menü den Befehl **Absatz erstellen** (*Abbildung unten*).



Das vordefinierte Format kann geändert werden, indem Sie den Tag für das vordefinierte Format auswählen und dann in der Symbolleiste aus der Auswahlliste der Formatierungen[459] (*Abbildung unten*) ein anderes vordefiniertes Format auswählen oder den Menübefehl **Einfügen | Format** auswählen.

Das vordefinierte Format kann auch im Fenster "Eigenschaften" durch Ändern der Eigenschaft `Absatztyp` in der Eigenschaftsgruppe `Absatz` oder durch Ändern des Absatztyps über den Kontextmenübefehl der Node-Vorlage **Einschließen in | Spezieller Absatz**[239] geändert werden.

Jeder Absatztyp hat bestimmte nützliche Formatierungsfunktionen. Beachten Sie: Mit Hilfe des Formattyps `pre` können Zeilenschaltungen und Umbrüche als solche ausgegeben werden anstatt als Leerzeichen.

5.1.2 Restlicher Inhalt

Mit dem Platzhalter `restlicher Inhalt` werden auf alle restlichen Child-Elemente des Elements, für das die Vorlage erstellt wurde, Vorlagen angewendet. Dazu folgendes Beispiel:

- Ein Element `parent` hat 4 Child-Elemente `child1` bis `child4`.
- In der Vorlage für das Element `parent` wurden explizit für die Child-Elemente `child1` und `child4` bestimmte Anweisungen definiert.

Dies führt dazu, dass nur die Child-Elemente `child1` und `child4` verarbeitet werden. Die Elemente `child2` und `child3` werden nicht verarbeitet. Wenn nun der Platzhalter `restlicher Inhalt` in die Vorlage für `parent` eingefügt wird, werden nicht nur die Elemente `child1` und `child4` mit den explizit definierten Verarbeitungsregeln in der Vorlage verarbeitet. Die Vorlagen werden zusätzlich auch auf die Child-Elemente `child2` und `child3` angewendet. Wenn für diese globale Vorlagen[230] definiert sind, werden die globalen

Vorlagen verwendet. Andernfalls werden die vordefinierten Standardvorlagen (für Element-, Attribut- und Text-Nodes) angewendet.

Wichtig: Beachten Sie bitte, welche Nodes für `restlicher Inhalt` ausgewählt werden.

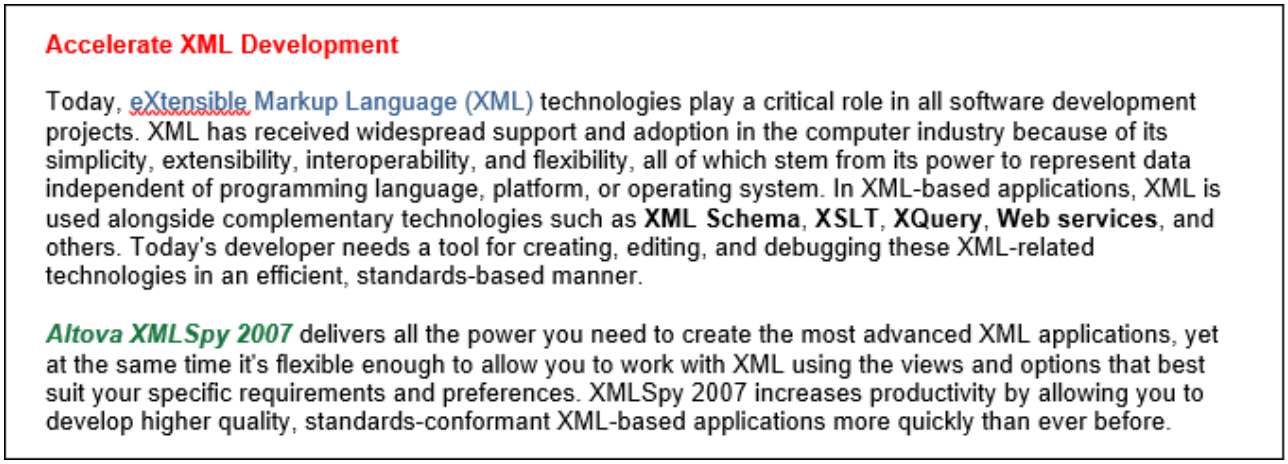
- Wie im Beispiel oben beschrieben, werden vom Platzhalter `restlicher Inhalt` alle Child-Element-Nodes und Child-Text-Nodes ausgewählt. (Sogar ungültige Child Nodes im XML-Dokument werden verarbeitet).
- Attribut-Nodes werden nicht ausgewählt, da diese in XPath-Ausdrücken nicht als Child Nodes gelten.
- Wenn in einer Parent-Vorlage eine globale Vorlage eines Child-Elements verwendet wird, zählt das Child-Element nicht als ein lokal verwendetes. Daher werden mit dem Platzhalter `restlicher Inhalt` auch solche Child-Elemente ausgewählt. Wenn allerdings eine globale Vorlage eines Child-Elements "lokal kopiert" wurde, gilt diese Verwendung als lokale Verwendung und das Child-Element wird vom Platzhalter `restlicher Inhalt` nicht ausgewählt.

Anmerkung: Durch Löschen des `(Inhalt)` Platzhalters können Sie eine **leere Vorlagenregel** erstellen. Mit Hilfe einer leeren Vorlagenregel können Sie festlegen, dass ein bestimmter Node nicht verarbeitet werden soll, d.h. dass dafür keine Ausgabe erzeugt werden soll.

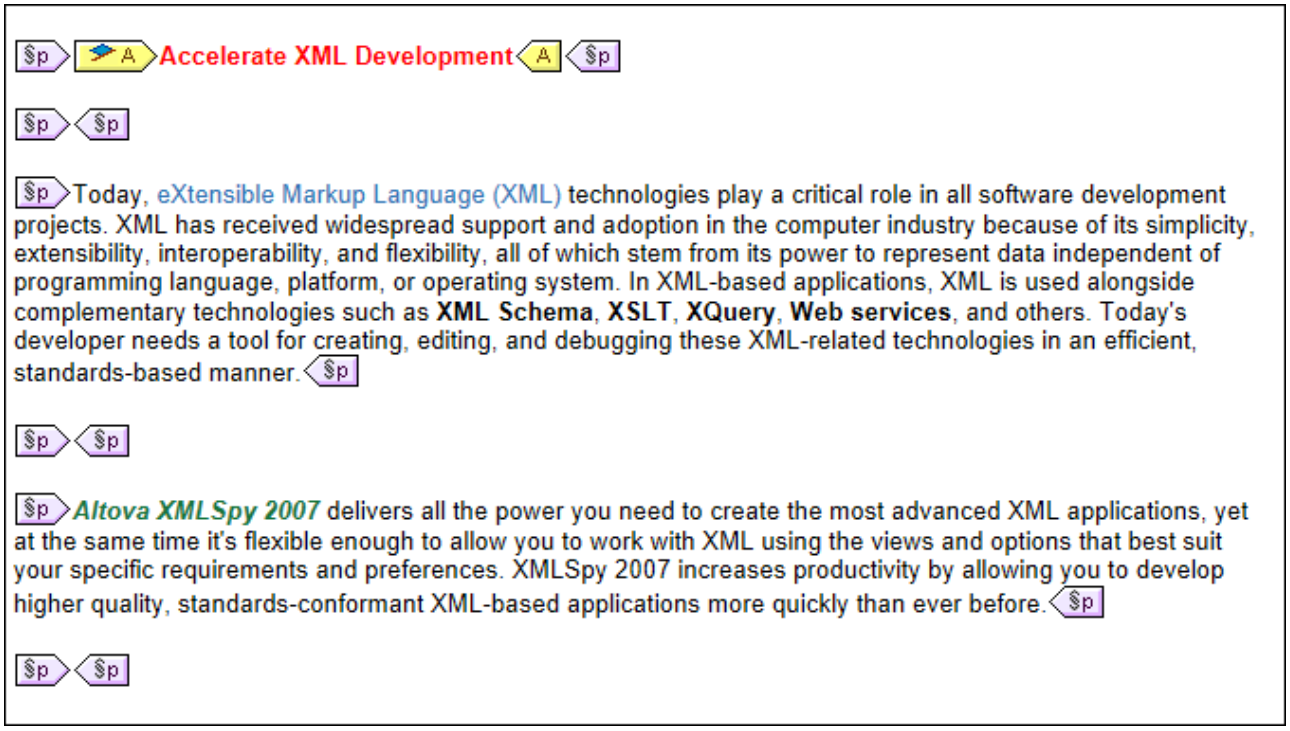
© 2019-2025 Altova GmbH

5.2 Einfügen von MS Word-Inhalt

Wenn Microsoft Word 2007+ auf Ihrem Rechner installiert ist, kann Inhalt als **statischer Inhalt** aus Word-Dokumenten in Ihr Design eingefügt werden. Der Word-Inhalt wird in passende Design-Komponenten eingefügt, wobei die Textformateigenschaften aus dem Word-Inhalt übernommen werden. So wird z.B. Textinhalt aus einem Absatz in Word in eine Absatzkomponente[113] eingefügt und die Formatierung des Texts beibehalten (*siehe Abbildung unten*).



Word-Inhalt



In ein Design eingefügter Word-Inhalt. Es wurde ein passendes Absatzformat angewendet und die Textformatierung wurde beibehalten.

Anmerkung: Neben Word-Inhalt kann *jeder Inhalt, der in ein Word-Dokument eingefügt werden kann*, auch in ein StyleVision-Design eingefügt werden. Dazu gehören auch **MS Excel-Tabellen** und **HTML-Seiteninhalt**.

Anmerkung: Um ein SPS zu erstellen, das statischen Inhalt aus einem ganzen Word-Dokument enthalten soll, erstellen Sie ein neues SPS mit dem Befehl **Datei | Neu | Neu von Word 2007+** 468.

Unterstützte Word-Funktionalitäten

Die folgenden Word-Strukturen und Formate werden unterstützt, wenn Sie Word-Inhalt in ein Design einfügen:

- Formatierter Text
  - *Unterschiedliche Schriftarten, Schriftgröße, -breite, unterschiedliche Stile, Textverzierung, usw.*
  - *Farbe*
  - *Hintergrundfarbe*
  - *Textumrandung*
- Absätze
- Seitenumbrüche
- Horizontale Linie
- Hyperlinks
- Lesezeichen
- Tabellen
  - *Zeilenabschnitte, Spaltenabschnitte*
  - *Formatiert/Rich Content*
  - *Verschachtelte Tabellen*
  - *Kopfzeilen, Fußzeilen*

© 2019-2025 Altova GmbH

- Listen, Unterlisten
  - *Aufzählungszeichen: verschiedene Stile*
  - *Nummerierung: verschiedene Stile*
- Bilder

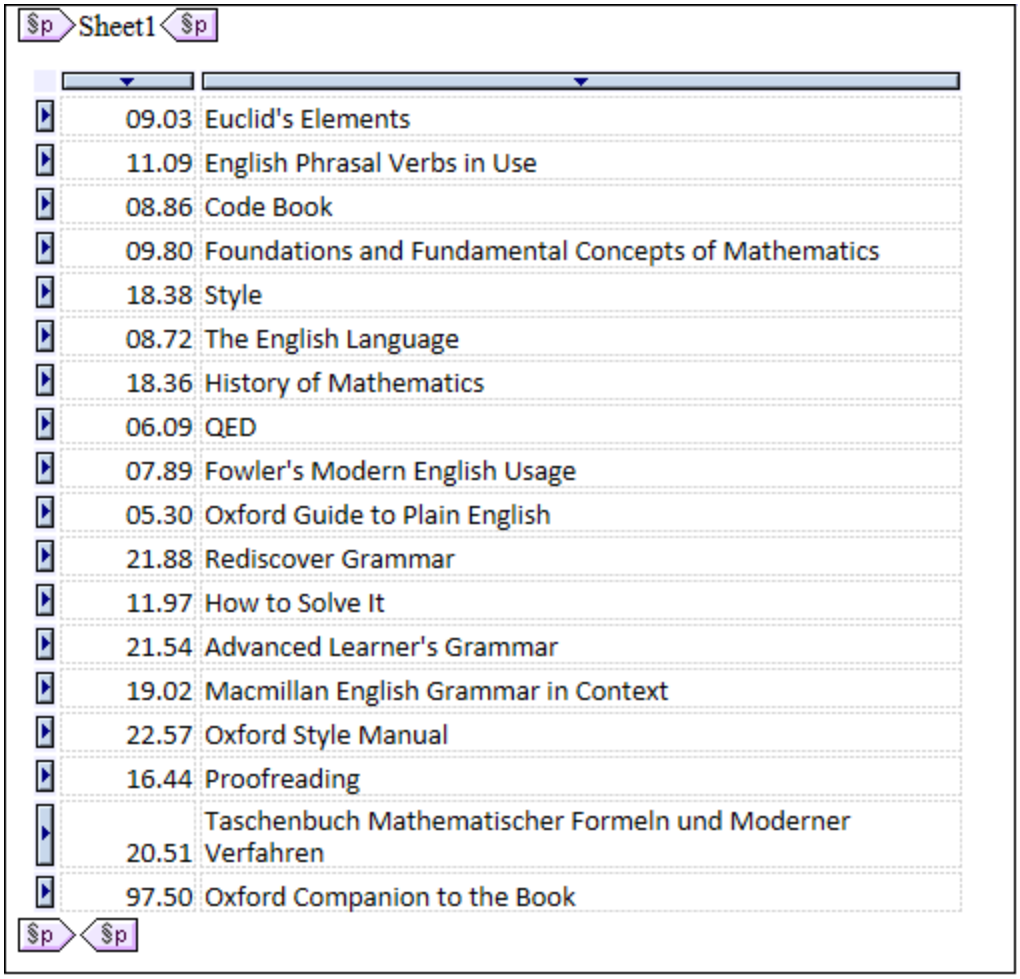
5.3 Einfügen von MS Excel-Inhalt

Wenn Microsoft Excel 2007+ auf Ihrem Rechner installiert ist, kann Inhalt aus Excel-Dokumenten als **statischer Inhalt** in Ihr Design eingefügt werden. Der Excel-Inhalt wird in Form statischer Tabellen oder anderer geeigneter Designkomponenten eingefügt. Die Formatierungseigenschaften bleiben dabei erhalten (*siehe Abbildungen unten*). Jedes Excel-Blatt wird als separate statische Tabelle eingefügt.

	A	B
1	09.03	Euclid's Elements
2	11.09	English Phrasal Verbs in Use
3	08.86	Code Book
4	09.80	Foundations and Fundamental Concepts of Mathematics
5	18.38	Style
6	08.72	The English Language
7	18.36	History of Mathematics
8	06.09	QED
9	07.89	Fowler's Modern English Usage
10	05.30	Oxford Guide to Plain English
11	21.88	Rediscover Grammar
12	11.97	How to Solve It
13	21.54	Advanced Learner's Grammar
14	19.02	Macmillan English Grammar in Context
15	22.57	Oxford Style Manual
16	16.44	Proofreading
17	20.51	Taschenbuch Mathematischer Formeln und Moderner Verfahren
18	97.50	Oxford Companion to the Book

Excel-Blatt

© 2019-2025 Altova GmbH



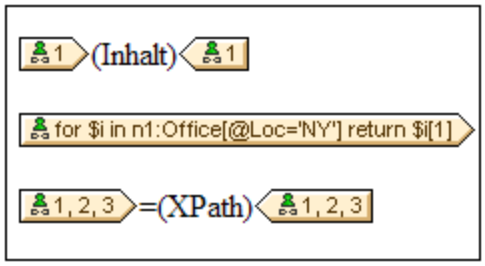
Excel-Inhalt, der als statische Tabelle unter Beibehaltung der Textformatierung in ein Design importiert wurde.

Anmerkung: Neben Excel-Inhalt kann *jeder Inhalt, der in ein Excel-Dokument eingefügt werden kann*, auch in ein StyleVision-Design eingefügt werden. Dazu gehören auch **MS Word-Tabellen** und **HTML-Seiteninhalt**.

Anmerkung: Um ein SPS zu erstellen, das statischen Inhalt aus einem ganzen Excel-Dokument enthalten soll, erstellen Sie ein neues SPS mit dem Befehl **Datei | Neu | Neu von Excel 2007+** 468.

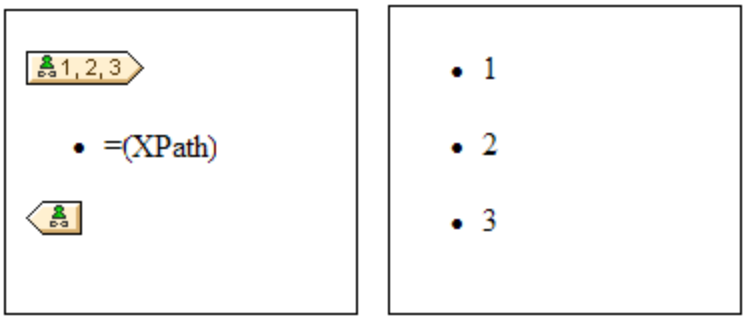
5.4 Benutzerdefinierte Vorlagen

Benutzerdefinierte Vorlagen sind Vorlagen für Elemente, die durch einen von Ihnen definierten XPath-Ausdruck generiert wurden. Dabei kann es sich um atomare Werte oder Nodes handeln. Beachten Sie in der Abbildung unten, in der Sie drei benutzerdefinierte Vorlagen sehen, das Symbol für eine benutzerdefinierte Vorlage (grüne Figur) auf der linken Seite der Tags. Benutzerdefinierte Vorlagen sind sehr nützlich, da sie beim Erstellen von Vorlagen besonders viel Flexibilität bieten.



Der XPath-Ausdruck in den drei in der obigen Abbildung gezeigten benutzerdefinierten Vorlagen bewirkt Folgendes:

- Ein Node in einem Quellschema wird ausgewählt. Mit Hilfe eines XPath-Ausdrucks kann jeder beliebige Node in jeder beliebigen Schemaquelle von jedem Kontext-Node aus ausgewählt werden. Wenn es einen eindeutigen Node gibt, der von StyleVision ausgewählt werden kann, so wird die Vorlage automatisch von einer benutzerdefinierten Vorlage in eine normale Vorlage umgewandelt. Wenn es sich um eine benutzerdefinierte Vorlage handelt, so wird dies durch ein grünes Symbol für eine benutzerdefinierte Vorlage am linken Rand des Vorlagen-Tags gekennzeichnet.
- Ein Node wird ausgewählt, der eine durch das `for` Konstrukt von XPath 2.0 und 3.0 definierte Bedingung erfüllt. Solche Vorlage können niemals zu normalen Vorlagen aufgelöst werden (sondern bleiben benutzerdefinierte Vorlagen), da das `for` Konstrukt es nicht zulässt, dass StyleVision das Ziel anhand der derzeit verfügbaren Schemainformationen eindeutig aufgelöst wird.
- Eine Sequenz atomarer Werte `{1, 2, 3}` wird ausgewählt. Es ist zwar zulässig, eine Vorlage für einen atomaren Wert zu erstellen, doch kann in einer solchen Vorlage der Platzhalter `Inhalt` nicht verwendet werden. Der Grund hierfür ist, dass die Anweisung `xsl:apply-templates` (also das, was der Platzhalter `Inhalt` generiert) nur auf Node-Elemente (nicht aber auf atomare Werte) angewendet werden kann. Sie könnten jedoch eine automatische Berechnung in Kombination mit einem Design-Element wie z.B. einer Liste, verwenden. So würde die benutzerdefinierte Vorlage in der Abbildung auf der linken Seite die Ausgabe auf der rechten Seite generieren.



Anmerkung: Wenn im SPS XSLT 1.0 verwendet wird, so muss der von Ihnen eingegebene XPath-Ausdruck ein Node-Set zurückgeben, da sonst ein Fehler gemeldet wird.

 © 2019-2025 Altova GmbH

Vorteil der Verwendung von XPath zur Auswahl eines Vorlagen-Node

Der Vorteil der Auswahl eines Schema-Node mittels eines XPath-Ausdrucks (benutzerdefinierte Vorlagen) ist, dass man mit Hilfe des XPath-Selektors jeden beliebigen Node, bzw. jede beliebige Elementsequenz auswählen kann und Filter und Bedingungen bei der Node-Auswahl einsetzen kann. Dadurch können spezifische XML-Dokument-Nodes für jede angegebene Vorlage adressiert werden. So wählt z.B. der XPath-Ausdruck `//Office/Department[@Location="NY"]` nur diejenigen `Department` Nodes aus, die ein `Location` Attribut mit dem Wert `NY` haben. Siehe auch die anderen Beispiele in diesem Abschnitt.

Anmerkung: Wenn ein XPath-Ausdruck mehrere location path-Schritte enthält, dann macht es - v.a. beim Gruppieren und Sortieren - einen Unterschied, ob die verschiedenen location path-Schritte in Klammern gesetzt werden oder nicht. So wird der XPath-Ausdruck `/Org/Office/Dept` z.B. anders verarbeitet als `(/Org/Office/Dept)`. Beim ersten Ausdruck (ohne Klammern) führt der Prozessor eine Schleifenverarbeitung der einzelnen location-Schritte durch. Beim zweiten Ausdruck (mit Klammern) werden alle `Dept` Elemente aller `Office` Elemente in einem einzigen Nodeset zurückgegeben.

Klammern	Zugrunde liegende XSLT-Methode	Ergebnis

Nein	`<xsl:for-each select="Org">`
	`  <xsl:for-each select="Office">`
	`  <xsl:for-each select="Dept">`
	`  ...`
	`  </xsl:for-each>`
	`  </xsl:for-each>`
	`</xsl:for-each>`
		Jedes `Office` Element hat seine eigene `Dept` Population. Dadurch kann die Gruppierung und Sortierung innerhalb der einzelnen `Office`Elemente durchgeführt werden.

Ja	`<xsl:for-each`
	`select="/Org/Office/Dept">`
	`  ...`
	`</xsl:for-each>`
		Die `Dept` Population umfasst alle `Office` Elements und alle `Org` Elemente.

Dieser Unterschied in der Auswertung von XPath-Ausdrücken kann sich auf die Gruppierung und Sortierung auswirken.

Einfügen einer benutzerdefinierten Vorlage

Um eine benutzerdefinierte Vorlage einzufügen, gehen Sie folgendermaßen vor:

1. Klicken Sie in der Symbolleiste "Design-Elemente" auf die Schaltfläche **Benutzerdefinierte Vorlage einfügen** und klicken Sie anschließend an die Stelle im Design, an der die Vorlage eingefügt werden soll. Als Alternative können Sie auch mit der rechten Maustaste an die gewünschte Stelle im Design klicken und aus dem Kontextmenü anschließend den Befehl **Benutzerdefinierte Vorlage einfügen** auswählen.
2. Geben Sie im Dialogfeld XPath-Ausdruck bearbeiten 440, das nun angezeigt wird, den gewünschten XPath-Ausdruck ein und klicken Sie auf **OK**. Beachten Sie, dass der Kontext-Node des XPath-Ausdrucks derjenige Node sein wird, in den Sie hineingeklickt haben. Daraufhin wird eine leere Node-Vorlage erstellt. Manchmal wird ein verbundener Node erstellt. Wenn ein Node verbunden wird, werden die Ziel-Instanznodes so ausgewählt, als befänden sie sich auf einer einzigen Ebene, während die Node-Auswahl bei nicht verbundenen Nodes (d.h. wenn er auf mehrere hierarchische Ebenen aufgeteilt

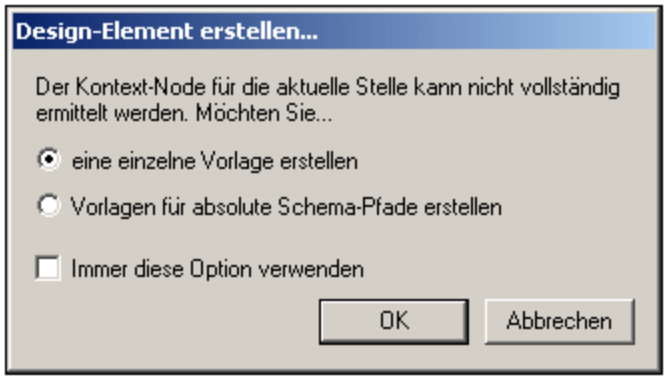
ist) durch Schleifenverarbeitung der einzelnen Instanz-Nodes auf den einzelnen Ebenen erfolgt. Das zurückgegebene Nodeset ist in beiden Fällen (bei verbundenen und getrennten Nodes) dasselbe, es sei denn ein Gruppierungs- oder Sortierkriterium wurde definiert. Nähere Informationen darüber, wie sich verbundene Nodes auf die Gruppierung und Sortierung auswirken, finden Sie unter Node-Vorlagen-Operationen 239.

Bearbeiten einer Vorlagenauswahl

Die Node-Auswahl jeder Node-Vorlage (benutzerdefiniert oder normal) kann über einen XPath-Ausdruck geändert werden, sodass ein neuer Suchausdruck ausgewählt wird. Um die Vorlagenauswahl einer Node-Vorlage zu bearbeiten, klicken Sie mit der rechten Maustaste auf die Node-Vorlage und wählen Sie den Befehl **Vorlagenauswahl ändern**. Daraufhin wird das Dialogfeld "XPath-Ausdruck bearbeiten" angezeigt, in dem Sie den XPath-Ausdruck zur Auswahl des neuen Node eingeben können. Klicken Sie anschließend auf **OK**.

Hinzufügen von Nodes zu benutzerdefinierten Vorlagen

Wenn ein Node aus der Schemastruktur zu einer benutzerdefinierten Vorlage hinzugefügt wird, ist der Kontext für den neuen Node unbekannt, wenn die benutzerdefinierte Vorlage für einen Node oder eine Sequenz erstellt wurde, der/die nicht in den Kontext der Schemaquelle des SPS platziert werden kann. Sie werden daher gefragt (*Abbildung unten*), wie der neue Node referenziert werden soll: (i) durch seinen Namen (im Wesentlichen ein relativer Pfad) oder (ii) durch einen vollständigen Pfad von der Root der Schemaquelle aus.



Standardmäßig werden Sie gefragt, wie der Node referenziert werden soll. Sie können dieses Standardverhalten auf dem Register "Design" des Dialogfelds Extras | Optionen 552 ändern.

 © 2019-2025 Altova GmbH

5.5 Benutzerdefinierte Elemente, XML-Textblöcke

Mit Hilfe von benutzerdefinierten Elementen[123] und benutzerdefinierten XML-Textblöcken[124] können Sie (i) jedes beliebige Element bzw. (ii) jeden beliebigen XML-Textblock in das Design einfügen. Der Vorteil dieser Funktion ist, dass nicht nur XML-Elemente und Design-Elemente aus Quellschemas und der Palette der StyleVision Design-Elemente hinzugefügt werden können. Es können (i) Vorlagen für vom Benutzer definierte Elemente (benutzerdefinierte Elemente) und (ii) unabhängiger eigenständiger XML-Code (benutzerdefinierte Blöcke) erstellt werden, mit dem Objekte (z.B. ActiveX-Objekte) unabhängig erstellt werden können.

Zwischen benutzerdefinierten Elementen und benutzerdefinierten XML-Textblöcken besteht ein grundlegender Unterschied. Ein benutzerdefiniertes Element wird im Design als Vorlagen-Node für ein einzelnes XML-Element (mit Attributen) angelegt. Der gesamte Inhalt dieser Vorlage muss explizit erzeugt werden. Dieser Inhalt besteht aus den verschiedenen Design-Elementen, die für das SPS zur Verfügung stehen. Ein benutzerdefinierter XML-Textblock darf kein Design-Element enthalten; es ist ein unabhängiger, eigenständiger Block. Da benutzerdefinierte Elemente leer erstellt werden, eignen sie sich nicht zur Erstellung von Objekten, für die mehrere Codezeilen erforderlich sind. Für derartige Objekte sollten benutzerdefinierte XML-Textblöcke verwendet werden.

Anmerkung: Benutzerdefinierte Elemente und benutzerdefinierte Textblöcke werden in der Authentic-Ansicht nur in der Enterprise Edition von Altova-Produkten unterstützt.

5.5.1 Benutzerdefinierte Elemente

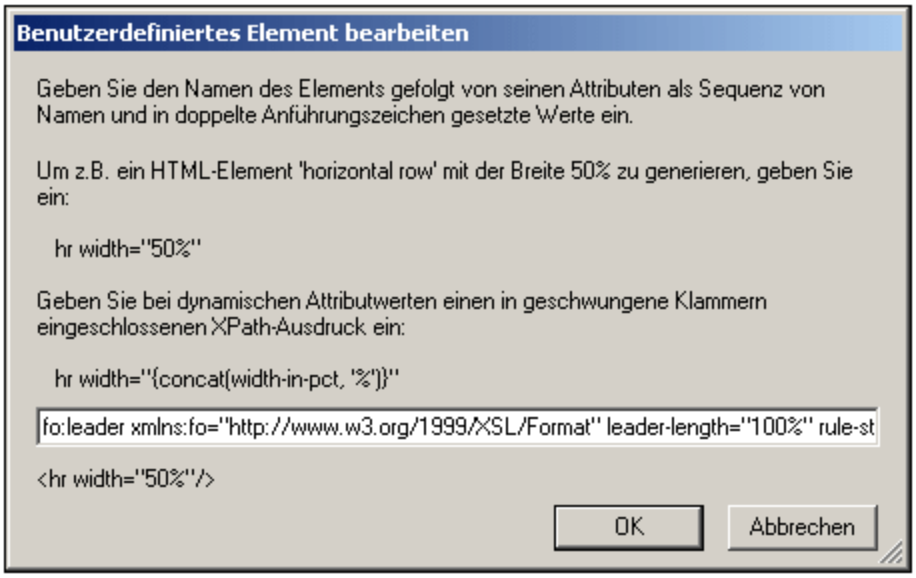
Benutzerdefinierte Elemente sind Elemente, die Sie in der Ausgabe generieren können, ohne dass diese Elemente in einer der Schemaquellen des SPS enthalten sein müssen. Das bedeutet, dass ein Element aus jedem beliebigen Namespace (HTML oder XSL-FO z.B.) an jeder beliebigen Stelle im Design eingefügt werden kann. Anschließend können in das eingefügte Element SPS Design-Elemente eingefügt werden.

Anmerkung: Benutzerdefinierte Elemente werden in der Authentic-Ansicht nur in der Enterprise Edition von Altova-Produkten unterstützt.

Einfügen benutzerdefinierter Elemente

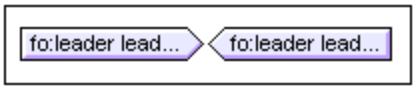
Um benutzerdefinierte Elemente zu verwenden, gehen Sie folgendermaßen vor:

1. Klicken Sie mit der rechten Maustaste an die Stelle im Design, an der das benutzerdefinierte Element eingefügt werden soll.
2. Wählen Sie nun im Kontextmenü den Befehl **Benutzerdefiniertes Element einfügen| Benutzerdefiniertes Element**.
3. Geben Sie im Dialogfeld, das daraufhin angezeigt wird (*Abbildung unten*), den Elementnamen, die gewünschten Attributwertpaare und eine Namespace-Deklaration für das Element ein, wenn das Dokument keine enthält.



In der Abbildung oben wird ein XSL-FO-Element mit dem Namen `leader` erstellt. Es hat das Präfix `fo:`, das an die Namespace-Deklaration `xmlns:fo="http://www.w3.org/1999/XSL/Format"` gebunden ist. Das Element hat eine Reihe von Attributen, wie z.B. `leader-length` und `rule-style`, jedes mit einem entsprechenden Wert. Das Element, seine Attribute und seine Namespace-Deklaration müssen ohne die eckigen Tag-Klammern eingegeben werden.

4. Klicken Sie auf **OK,** um das Element in das Design einzufügen. Das Element wird im Design als leere Vorlage mit einem Start- und einem End-Tag angezeigt (*Abbildung unten*).



5. Sie können nun wie bei jeder anderen Vorlage Inhalt zur Vorlage hinzufügen. Das benutzerdefinierte Element kann statischen Inhalt, dynamischen Inhalt aus dem XML-Dokument sowie weitere benutzerdefinierte Elemente enthalten.

Anmerkung: Ein benutzerdefiniertes Element, das für eine bestimmte Ausgabe verwendet werden soll, sollte in eine passende ausgabebasierte Bedingung eingeschlossen werden, um unerwartete Ergebnisse in anderen Ausgabedokumenten zu vermeiden.

5.5.2 Benutzerdefinierte XML-Textblöcke

Ein benutzerdefinierter XML-Textblock ist ein XML-Fragment, das in den vom SPS generierten XSLT-Code eingefügt wird. Er wird als eigenständiger Block, zu dem kein Design-Element hinzugefügt werden kann, in das SPS Design eingefügt. Ein solcher XML-Textblock sollte daher an der Stelle im Stylesheet, an der er vorkommt, als XSLT-Code angewendet werden können.

Der Vorteil dieser Funktion liegt darin, dass damit eine Möglichkeit besteht, XSLT-Fragmente und benutzerdefinierten Code in das Design einzufügen. So kann z.B. damit ein ActiveX-Objekt in ein HTML HTML `SCRIPT`-Element eingefügt werden.

 © 2019-2025 Altova GmbH

Anmerkung: Diese Funktion ist **nur in der Enterprise Edition der Authentic-Ansicht** (d.h. in der Enterprise Edition von StyleVision, Authentic Desktop, Authentic Browser und XMLSpy) aktiviert.

Einfügen benutzerdefinierter XML-Textblöcke

Um einen XML-Textblock einzufügen, gehen Sie folgendermaßen vor:

1. Klicken Sie mit der rechten Maustaste an die Stelle im Design, an der der benutzerdefinierte Block eingefügt werden soll.
2. Wählen Sie nun im Kontextmenü den Befehl **Benutzerdefiniertes Element einfügen| Benutzerdefinierter Block.**
3. Geben Sie im Dialogfeld, das daraufhin angezeigt wird (*Abbildung unten*), den gewünschten XML-Textblock ein. Beachten Sie, dass der XML-Textblock wohlgeformten XML-Code enthalten sollte, damit er im Dialogfeld verwendet werden kann.

   In der Abbildung oben wird ein XML-Textblock hinzugefügt, der eine HTML-Liste erstellt.
4. Klicken Sie auf **OK,** um das Element in das Design einzufügen. Der XML-Textblock wird im Design als Textblock angezeigt.

Anmerkung: Ein XML-Textblock, der für eine bestimmte Ausgabe verwendet werden soll, sollte in eine passende ausgabebasierte Bedingung eingeschlossen werden, um unerwartete Ergebnisse in anderen Ausgabedokumenten zu vermeiden.

5.6 Tabellen

In einem SPS-Design können zwei Arten von Tabellen verwendet werden: **SPS-Tabellen** und **CALS/HTML-Tabellen**. Diese zwei Arten von Tabellen unterscheiden sich in wichtigen Punkten. In diesem Abschnitt finden Sie eine detaillierte Anleitung zur Verwendung diese beiden Tabellentypen.

SPS-Tabellen

Eine **SPS-Tabelle** ist eine Komponente eines SPS. Sie wird im Design strukturiert und formatiert und kann an jeder beliebigen Stelle im Design erstellt werden. Es können beliebig viele SPS-Tabellen erstellt werden

SPS-Tabellen dienen ausschließlich zu Darstellungszwecken und werden mit Hilfe der Darstellungsmittel des Ausgabeformats wiedergegeben. Die **Struktur** einer SPS-Tabelle **wird nicht durch Nodes im XML-Dokument** vorgegeben, obwohl der Inhalt der Tabellen aus Nodes im XML-Dokument stammen kann.

Es gibt zwei Arten von SPS-Tabellen:

- **Statische Tabellen** werden von der Person, die das SPS entwirft, Schritt für Schritt erstellt. Nach Erstellung der Tabellenstruktur wird der Inhalt der einzelnen Zellen separat definiert. Der Zelleninhalt kann aus den unterschiedlichsten willkürlich in der Schemastruktur gewählten Nodes stammen und kann sogar unterschiedlicher Art sein. Beachten Sie, dass die Zeilen einer statischen Tabelle keine sich wiederholende Datenstruktur darstellen. Aus diesem Grund wird die Tabelle als statisch bezeichnet: Sie hat eine feste Struktur, die sich nicht ändert, wenn sich der XML-Inhalt ändert.
- **Dynamische Tabellen** dienen zur Darstellung von Datenstrukturen im XML-Dokument, die sich wiederholen. Sie können für Schemaelemente erstellt werden, die eine Substruktur haben, d.h. Elemente, die mindestens ein Child-Attribut oder Element haben. Jedes Element mit einer Substruktur wiederholt sich, wenn es davon mehr als eine Instanz gibt. Jede Instanz des Elements würde in einer dynamischen Tabelle als Zeile dargestellt und alle oder einige der Child-Elemente oder -Attribute würden als Spalten der Tabelle dargestellt werden. Die Struktur einer dynamischen Tabelle gibt daher den Inhalt der XML-Datei wieder und ändert sich dynamisch, wenn sich der XML-Inhalt ändert.

CALS/HTML-Tabellen

Das Content Model einer CALS- oder HTML-Tabelle ist im XML-Dokument - mittels Erweiterung in der DTD oder im Schema - definiert und entspricht der jeweiligen Spezifikation (CALS oder HTML). Anschließend können Sie im SPS-Design festlegen, dass CALS/HTML-Tabellen als Tabellen zu verarbeiten sind. Anhand der XML-Datenstruktur, die für die CALS/HTML-Tabelle definiert ist, wird der Tabellen-Markup-Code für das jeweilige Ausgabeformat generiert. Die Formatierung der CALS/HTML-Tabelle kann im XML-Instanzdokument oder im SPS oder auch in beiden definiert werden.

Unten sehen Sie die HTML-Vorschau einer HTML-Tabelle.

Name	Phone
John Merrimack	6517890
Joe Concord	6402387

Das HTML-Codefragment für die oben gezeigte XML-Tabelle folgendermaßen aus:

```
<table border="1" width="40%">
  <tbody>
```


© 2019-2025 Altova GmbH

```
    <tr>
      <td>Name</td>
      <td>Phone</td>
    </tr>
    <tr>
      <td>John Merrimack</td>
      <td>6517890</td>
    </tr>
    <tr>
      <td>Joe Concord</td>
      <td>6402387</td>
    </tr>
  </tbody>
</table>
```

Das ursprüngliche XML-Dokument könnte folgendermaßen aussehen:

```
<phonelist border="1" width="40%">
  <items>
    <person>
      <data>Name</data>
      <data>Phone</data>
    </person>
    <person>
      <data>John Merrimack</data>
      <data>6517890</data>
    </person>
    <person>
      <data>Joe Concord</data>
      <data>6402387</data>
    </person>
  </items>
</phonelist>
```

Beachten Sie, dass die Elementnamen im XML-Dokument keine Tabellensemantik aufweisen müssen; die **Struktur der Tabelle** muss jedoch dem HTML- oder CALS-Tabellenmodell entsprechen. Beachten Sie außerdem:

- Beachten Sie, dass nur ein XML-Element dem HTML-Spaltenelement `<td/>` entsprechen kann.
- Eine CALS/HTML-Tabelle kann an jeder beliebigen Stelle im XML-Dokument eingefügt werden, an der das Element, das dem Element `table` entspricht, gemäß dem Schema zulässig ist.
- In der Authentic-Ansicht werden Daten direkt in die Tabellenzellen eingegeben. Diese Daten werden als Inhalt des entsprechenden CALS/HTML-Tabellenelements gespeichert.
- Die Formatierungseigenschaften einer CALS/HTML-Tabelle können direkt aus dem XML-Dokument übernommen oder im SPS-Design definiert werden.

Zusammenfassung für den SPS-Designer

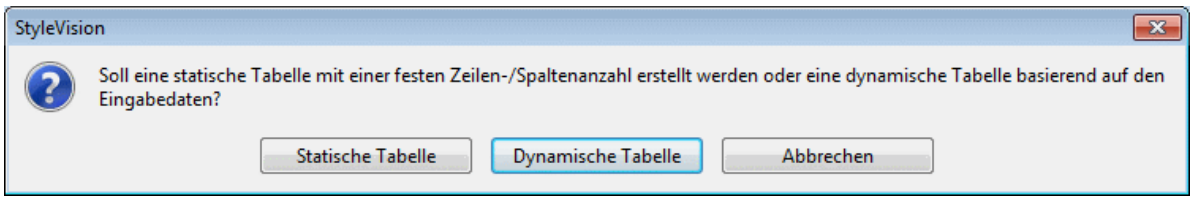
Für denjenigen, der das Dokument entwirft, sind die folgenden Punkte zu beachten:

- **Die Struktur** einer SPS-Tabelle ist im SPS definiert. Die Struktur einer CALS/HTML-Tabelle ist hingegen im Schema definiert und muss der des CALS/HTML-Tabellenmodells entsprechen; allerdings müssen die Elementnamen im Schema nicht mit denen im CALS- oder HTML-Tabellenmodell übereinstimmen.
- **Colspans (Anzahl der Spalten) und Rowspans (Anzahl der Zeilen)** in SPS-Tabellen werden im SPS definiert. In CALS/HTML-Tabellen werden Colspans und Rowspans jedoch im XML-Instanzdokument definiert.
- **Die Tabellenformatierung** von SPS-Tabellen wird im SPS definiert. Die Formatierung von CALS/HTML-Tabellen erfolgt im XML-Instanzdokument und/oder im SPS.

5.6.1 Statische Tabellen

Um eine statische Tabelle einzufügen, gehen Sie folgendermaßen vor:

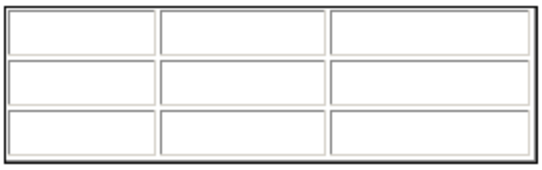
1. Verwenden Sie einen der folgenden Befehle: **Tabelle | Tabelle einfügen** oder **Einfügen | Tabelle** oder klicken Sie in der Symbolleiste "Design-Element einfügen" auf die Schaltfläche **Tabelle einfügen**.
2. Alle diese Befehle rufen das Dialogfeld "Tabelle einfügen" auf (*Abbildung unten*).



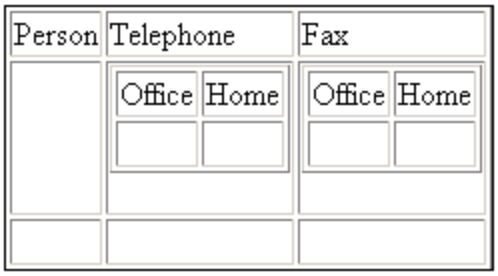
Klicken Sie auf **Statische Tabelle**.

3. Daraufhin wird das Dialogfeld "Tabelle einfügen" (*Abbildung unten*) angezeigt, in dem Sie die Größe der Tabelle definieren können und festlegen können, ob die Tabelle über die gesamte verfügbare Breite gehen soll.

4. Sobald Sie auf OK klicken, wird eine leere Tabelle mit den angegebenen Dimensionen wie unten gezeigt erstellt.



5. Sie können nun mit Hilfe der regulären StyleVision Funktionen Inhalt in die Tabelle eingeben. Dabei kann es sich um Text, Elemente, die aus der Schema-Struktur hineingezogen werden, oder Objekte wie z.B. Bilder oder geschachtelte Tabellen handeln. In der Abbildung unten sehen Sie eine Tabelle, die geschachtelte Tabellen enthält.



© 2019-2025 Altova GmbH

Statische SPS-Tabellen eignen sich vor allem zum Gliedern von XML-Daten, die sich an unterschiedlichen Stellen in der Schemahierarchie befinden, oder für statischen Inhalt (Inhalt, der nicht aus einer XML-Quelle stammt).

Löschen von Spalten, Zeilen und Tabellen

Um eine Spalte, Zeile oder Tabelle zu löschen, platzieren Sie den Cursor in die gewünschte Spalte, Zeile oder Tabelle und wählen Sie den Menübefehl **Tabelle | Spalte löschen**, **Tabelle | Zeile löschen** bzw. **Tabelle | Tabelle löschen**. Bei geschachtelten Tabellen wird mit diesen Befehlen die Spalte, Zeile bzw. Tabelle, die den Cursor enthält, gelöscht.

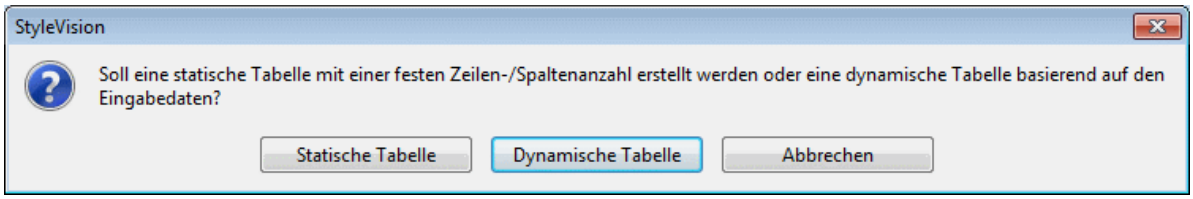
Bearbeitungsschaltflächen in der Symbolleiste

Die Bearbeitungsschaltflächen, die sich standardmäßig in der zweiten Reihe der Symbolleiste befinden, sind Shortcuts zu den Befehlen des Menüs **Tabelle**. Mit Hilfe dieser Befehle können Sie die Struktur der statischen Tabelle löschen, bearbeiten und Zeilen/Spalten einfügen und ihr Formatierungseigenschaften zuweisen. Diese Schaltflächen können auch für dynamische SPS-Tabellen verwendet werden.

5.6.2 Dynamische Tabellen

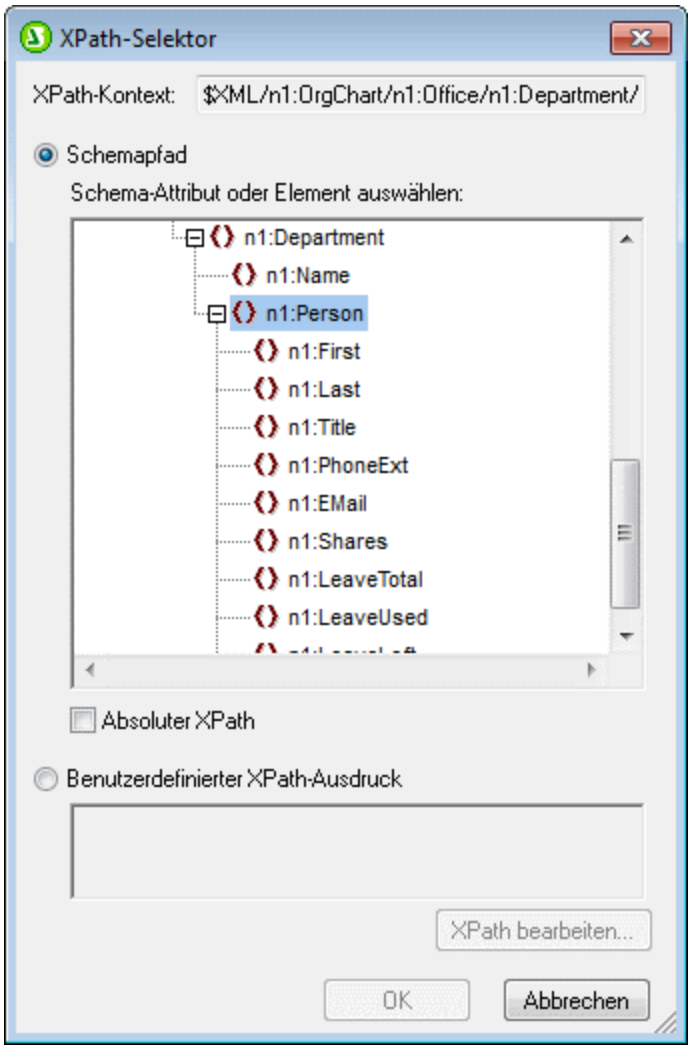
Um eine dynamische Tabelle einzufügen, gehen Sie folgendermaßen vor:

1. Verwenden Sie einen der folgenden Befehle: **Tabelle | Tabelle einfügen** oder **Einfügen | Tabelle** oder klicken Sie in der Symbolleiste "Design-Element einfügen" auf die Schaltfläche **Tabelle einfügen**.
2. Alle diese Befehle rufen das Dialogfeld "Tabelle einfügen" auf (*Abbildung unten*). Wenn Sie in der Symbolleiste auf die Schaltfläche "Tabelle einfügen" geklickt haben, wird das Dialogfeld "Tabelle einfügen" aufgerufen, wenn Sie im Design an die Stelle klicken, an der die Tabelle eingefügt werden soll.



Klicken Sie auf **Dynamische Tabelle**

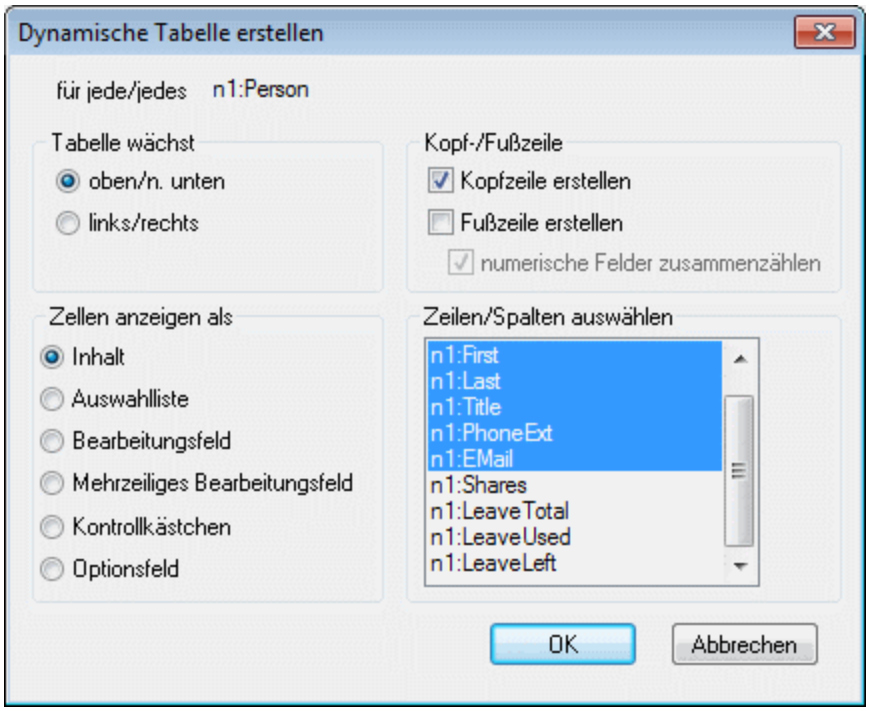
3. Beachten Sie im Dialogfeld XPath-Selektor (*Abbildung unten*), dass der XPath-Kontext der Kontext der Einfügestelle ist und im Dialogfeld nicht geändert werden kann. Wählen Sie den Node aus, der als dynamische Tabelle erstellt werden soll. In der Abbildung unten ist der Kontext-Node `n1:Department` und der Node `n1:Person` wurde als der Node ausgewählt, der in Form einer Tabelle dargestellt werden soll.



Wenn Sie die Option "benutzerdefinierter XPath" auswählen, können Sie einen XPath-Ausdruck eingeben, um den Node auszuwählen, der als dynamische Tabelle erstellt werden soll.

4. Klicken Sie auf **OK**. Daraufhin wird das Dialogfeld "Dynamische Tabelle erstellen" (*Abbildung unten*) angezeigt.

© 2019-2025 Altova GmbH



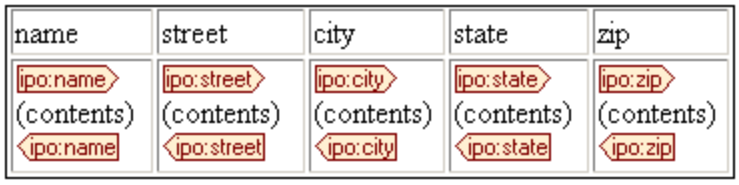
5. Die Child-Elemente und -Attribute des Elements, das in des Design-Fenster gezogen wurde, werden in der Liste "Zeilen/Spalten auswählen" angezeigt und können als Spalten der Tabelle erstellt werden. Heben Sie die Auswahl für die Child Nodes auf, die nicht übernommen werden sollen, und wählen Sie alle Attribute/Elemente, die als Spalten inkludiert werden sollen, aus. (In der Abbildung oben wurden die Elemente `Shares`, `LeaveTotal`, `LeaveUsed` und `LeaveLeft` deaktiviert.) Weiter unten finden Sie eine Erläuterung der anderen Optionen. Beachten Sie, dass Spalten nur für Child-Elemente und Attribute erstellt werden, nicht aber für untergeordnete Elemente auf niedrigeren Ebenen.

Anmerkung: Wenn Sie einen benutzerdefinierten XPath-Ausdruck zu Auswahl des als dynamische Tabelle zu erstellenden Node definiert haben, ist es für StyleVision wahrscheinlich nicht eindeutig, welcher Node das Ziel sein soll. Infolgedessen zeigt das Dialogfeld "Dynamische Tabelle erstellen" keine Liste von untergeordneten Attributen/Elementen an, die als Felder (Spalten) der Tabelle ausgewählt werden können. Die erstellte Tabelle muss daher manuell mit Node-Inhalt aus den Child-Attributen/Elementen des ausgewählten Node befüllt werden.

Anmerkung: Eine andere Methode, einen Schema-Node als Tabelle zu erstellen ist, den Node aus der Schema-Struktur in das Design zu ziehen und beim Loslassen der Maustaste zu definieren, dass der Node als Tabelle erstellt werden soll.

Tabelle wächst nach unten oder nach rechts

Wenn eine Tabelle von oben nach unten wächst, sieht Sie folgendermaßen aus:



Wenn eine Tabelle von links nach rechts wächst, sieht sie folgendermaßen aus:

Kopf- und Fußzeilen

Spalten und Zeilen können eine Überschriftszeile oder Kopfzeile erhalten, die die Namen der Spalten- und Zeilenelemente enthält. Spaltenüberschriften werden am oberen Rand der einzelnen Spalten erstellt. Zeilenüberschriften werden links von einer Zeile erstellt. Um eine Kopfzeile einzufügen, aktivieren Sie das Kontrollkästchen "Kopfzeile erstellen". Wenn die Tabelle von oben nach unten wächst, wird bei Erstellung einer Kopfzeile eine Überschriftszeile oberhalb des Tabellenkörpers erstellt. Wenn die Tabelle von links nach rechts wächst, wird bei Erstellung einer Überschriftszeile eine Spaltenüberschrift links vom Tabellenkörper erstellt.

Um Fußzeilen einzufügen, aktivieren Sie das Kontrollkästchen "Fußzeile erstellen". Fußzeilen können wie Kopfzeilen sowohl für Spalten (um unteren Rand der Spalte) als auch für Zeilen (auf der rechten Seite der Zeile) erstellt werden. In der Fußzeile von numerischen Spalten oder Zeilen wird außerdem jede einzelne der Spalten bzw. Zeilen summiert, wenn das Kontrollkästchen "*Numerische Felder zusammenzählen*" aktiviert ist.

Kopf- und Fußzeilen können über das Menü **Tabelle** zusammengelegt und aufgeteilt werden, es können Zeilen und Spalten eingefügt, angehängt und gelöscht werden; auf diese Art können Sie beim Strukturieren von Kopf- und Fußzeilen einigermaßen flexibel vorgehen. Zusätzlich dazu können Kopf- und Fußzeilen jede Art von statischem oder dynamischem Inhalt enthalten, z.B. auch Bedingungsvorlagen und automatische Berechnungen.

Anmerkung: Die Kopf- und Fußzeile muss beim Definieren der dynamischen Tabelle erstellt werden. Dies erfolgt durch Aktivieren der Optionen "Kopfzeile erstellen" und "Fußzeile erstellen" im Dialogfeld "Dynamische Tabelle erstellen". Wenn ein Zeile in einer dynamischen Tabelle eingefügt oder angehängt wird, werden dadurch keine Kopf- und Fußzeilen erstellt, sondern eine zusätzliche Zeile. Dies ist ein großer Unterschied. Mit dem Befehl "Kopf-/Fußzeile erstellen" werden echte Kopf- und Fußzeilen am oberen bzw. unteren Ende der Tabelle hinzugefügt. Wenn eine Zeile eingefügt oder angehängt wird, so wird für jede Instanz des Elements, das als dynamische Tabelle erstellt wurde, eine Zeile angezeigt.

Geschachtelte dynamische Tabellen

Sie können eine dynamische Tabelle in eine andere dynamische Tabelle einfügen, wenn das Element, für das die geschachtelte dynamische Tabelle erstellt werden soll, ein Child-Element des Elements ist, das als die enthaltende dynamische Tabelle erstellt wurde. Gehen Sie folgendermaßen vor:

1. Erstellen Sie die äußere dynamische Tabelle sodass das Child-Element, das als dynamische Tabelle erstellt werden soll, als Spalte angelegt wird.
2. Klicken Sie in der Design-Ansicht in der dynamischen Tabelle mit der rechten Maustaste auf das Child-Element.
3. Wählen Sie **Ändern in | Tabelle**. Daraufhin wird das Dialogfeld "Dynamische Tabelle erstellen" angezeigt.
4. Definieren Sie die Eigenschaften der geschachtelten dynamischen Tabelle.

© 2019-2025 Altova GmbH

Um eine dynamische Tabelle in eine statische Tabelle als geschachtelte Tabelle einzufügen, ziehen Sie das Element, das als dynamische Tabelle erstellt werden soll, in die gewünschte Zelle der statischen Tabelle. Wenn Sie die Maustaste loslassen, wählen Sie im Kontextmenü den Befehl **Tabelle erstellen.**

Tabellen für Elemente mit Textinhalt

Um Spalten (oder Zeilen) für Child-Elemente anzulegen, muss das als Tabelle erstellte Element ein **Child-Element oder einen Attribut-Node** haben. Wenn es einen **Child-Text-Node** hat, funktioniert dies nicht. Erstellen Sie in diesem Fall ein Child-Element z.B. mit dem Namen `Text` und setzen Sie Ihren Text in die `TableElement/Text` Elemente. Nun können Sie `TableElement` als dynamische Tabelle erstellen. Diese Tabelle enthält eine Spalte für `Text` Elemente. Jede Zeile enthält daher eine Zelle, die den Text-Node in `Text` enthält, und die Zeilen der Tabelle entsprechen den Instanzen des Elements `TableElement`.

Inhalt von Zellen des Tabellenhauptteils

Beim Erstellen einer dynamischen Tabelle können Sie den Inhalt von Nodes als eine aus einer Reihe von StyleVision Komponenten erstellen. In den obigen Beispielen wurden die Zellen des Tabellenhauptteils als Inhalt erstellt. Im Dialogfeld "Dynamische Tabelle erstellen" ist die Option "Zellen anzeigen als" *Inhalt.* Sie hätten auch als Dateneingabeelemente erstellt werden können. Beachten Sie die folgenden beiden Punkte:

- Die gewählte Einstellung ist eine globale Einstellung für alle Zellen des Tabellenhauptteils. Wenn eine einzelne Zelle anders angezeigt werden soll, bearbeiten Sie die Zelle, nachdem Sie die Tabelle erstellt haben: Rechtsklicken Sie in die Zelle und wählen Sie im Kontextmenü den Befehl "Ändern in" und anschließend den gewünschten Inhaltstyp aus.
- Wenn Sie Zellen als Elementinhalt erstellen und das Element untergeordnete Elemente hat, wird als Inhalt der Zelle eine Verkettung aus den Textstrings des Elements und aller seiner Nachfahren angezeigt.

Löschen von Spalten, Zeilen und Tabellen

Um eine Spalte, Zeile oder Tabelle zu löschen, platzieren Sie den Cursor in die gewünschte Spalte, Zeile oder Tabelle und wählen Sie den Menübefehl **Tabelle | Spalte löschen**, **Tabelle | Zeile löschen** bzw. **Tabelle | Tabelle löschen**. Bei geschachtelten Tabellen wird mit dem Befehl **Tabelle | Tabelle löschen** die Tabelle, die den Cursor enthält, gelöscht.

Bearbeitungsschaltflächen in der Symbolleiste

Die Bearbeitungsschaltflächen in der Symbolleiste sind Shortcuts zu den Befehlen des Menüs **Tabelle**. Mit Hilfe dieser Befehle können Sie die Struktur der dynamischen Tabelle löschen, bearbeiten und Zeilen/Spalten einfügen und ihr Formatierungseigenschaften zuweisen. Diese Schaltflächen können auch für statische Tabellen verwendet werden.

Erstellen von dynamischen Tabellen in globalen Vorlagen

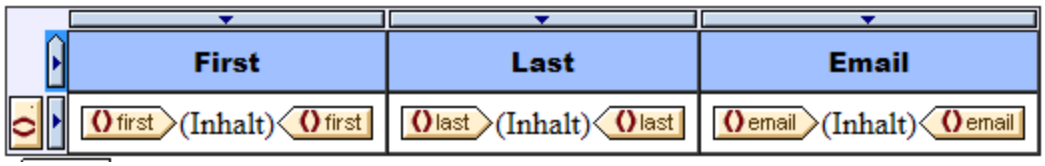
Sie können dynamische Tabellen auch in Elementen in globalen Vorlagen erstellen. Die Methode ist dieselbe wie bei den Elementen der Hauptvorlage (siehe oben). Beachten Sie, dass eine dynamische Tabelle in einer globalen Vorlage nur für **untergeordnete Elemente** des globalen Vorlagen-Node erstellt werden kann; sie kann nicht für den globalen Vorlagen-Node selbst erstellt werden. Wenn Sie z.B. in einer globalen Vorlage eine dynamische Tabelle für das Element `authors` erstellen möchten, so muss diese globale Tabelle in der globalen Vorlage des Parent-Elements von `authors` erstellt werden. Sie kann nicht in der globalen Vorlage des Elements `authors` erstellt werden.

5.6.3 Bedingte Verarbeitung in Tabellen

Eine bedingte Verarbeitung kann für einzelne Spalten und Zeilen von statischen und dynamischen Tabellen sowie für Spalten- und Zeilenüberschriften definiert werden, um einzelne Spalten, Zeilen oder Spaltenüberschriften ein- oder auszublenden, je nachdem, ob die Bedingung wahr oder falsch ist. Wenn das Ergebnis der Bedingung "true" ist, wird die Spalte, Zeile oder Überschrift angezeigt, andernfalls wird sie ausgeblendet.

Hinzufügen und Bearbeiten einer bedingten Verarbeitung

Um eine bedingte Verarbeitungsanweisung zu einer Spalte, Zeile oder Überschrift hinzuzufügen, klicken Sie mit der rechten Maustaste auf die entsprechende Design-Komponente und wählen Sie den Befehl **Bedingte Verarbeitung bearbeiten**. (In der Abbildung unten ist die Spaltenüberschrift-Design-Komponente links oben blau markiert; die Design-Komponente für die zweite Spalte ist blau umrandet; die einzige Zeilenkomponente befindet sich unterhalb der Spaltenüberschrift-Design-Komponente.)



Wenn Sie den Befehl **Bedingte Verarbeitung bearbeiten** auswählen, wird das Dialogfeld "XPath-Ausdruck bearbeiten" 440 angezeigt, in das Sie den XPath-Ausdruck der Bedingung eingeben können. Einige Verwendungsmöglichkeiten für eine bedingte Verarbeitung:

- Geben Sie für eine Spalte, Zeile oder Tabelle den XPath-Ausdruck `false()` ein, um die Spalte auszublenden, `true()` um sie anzuzeigen.
- Eine Spalte wird nur dann angezeigt, wenn die Summe aller Werte in dieser Spalte größer ist als ein bestimmter Ganzzahlwert.
- Eine Spalte oder Zeile wird nur dann ausgegeben, wenn keine Zelle in dieser Spalte bzw. Zeile leer ist.
- Eine Spalte oder Zeile wird nur dann ausgegeben, wenn diese Spalte bzw. Zeile einen bestimmten Zellwert enthält.

Um eine bereits erstellte Bedingung zu bearbeiten, klicken Sie mit der rechten Maustaste auf die entsprechende Design-Komponente und wählen Sie den Befehl **Bedingte Verarbeitung bearbeiten**. Bearbeiten Sie den XPath-Ausdruck, der prüft, ob die Bedingung zutrifft, im Dialogfeld "XPath-Ausdruck bearbeiten" 440, das daraufhin angezeigt wird.

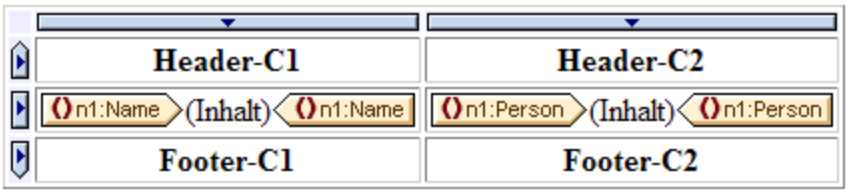
Entfernen einer bedingten Verarbeitung

Um die bedingte Verarbeitung einer Spalte, Zeile oder Spaltenüberschrift zu entfernen, klicken Sie mit der rechten Maustaste auf die entsprechende Design-Komponente und wählen Sie den Befehl **Bedingte Verarbeitung löschen**.

© 2019-2025 Altova GmbH

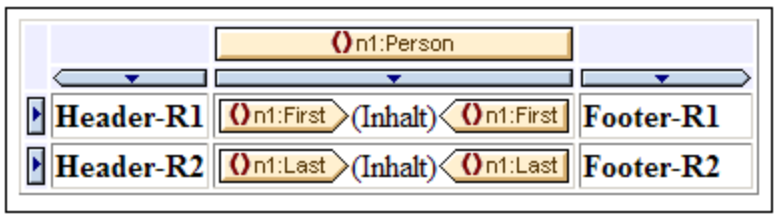
5.6.4 Tabellen in der Design-Ansicht

In der Abbildung unten sehen Sie die Hauptkomponenten von statischen und dynamischen SPS-Tabellen, wobei der Tabellen-Markup-Code eingeblendet ist (**Tabelle | Tabellen-Markup-Code anzeigen).**



In der Abbildung oben sehen Sie eine einfache Tabelle, die von oben nach unten wächst und eine Kopf- und eine Fußzeile hat.

- Eine Spalte wird durch ein Rechteck mit einem Abwärtspfeil gekennzeichnet. Am oberen Rand von Spalten befinden sich Spaltenindikatoren. Klicken Sie auf den Spaltenindikator einer Spalte, um diese gesamte Spalte auszuwählen - z.B. um dieser Spalte eine Formatierungseigenschaft zuzuweisen.
- Eine Zeile wird durch ein Rechteck mit einem Rechtspfeil gekennzeichnet. Klicken Sie auf einen Zeilenindikator, um die gesamte Zeile auszuwählen.
- In Tabellen, die von oben nach unten wachsen (*Abbildung oben*) sind Kopf- und Fußzeilen durch Symbole gekennzeichnet, die nach oben bzw. unten zeigen. In Tabellen, die von links nach rechts wachsen, sind Kopf- und Fußzeilen durch Symbole gekennzeichnet, die nach links bzw. rechts zeigen (*Abbildung unten*).
- Um eine gesamte Tabelle auszuwählen, klicken Sie in die linke obere Ecke der Tabelle (in den Abbildungen oben und unten auf die Stelle, auf die der Cursorpfeil zeigt).
- Wenn eine Tabellenzeile oder - spalte ausgewählt ist, wird diese mit einem dunkelblauen Hintergrund angezeigt. In der Abbildung oben ist die Fußzeile ausgewählt.
- In Tabellen, die von oben nach unten wachsen, wird das Element, für das die Tabellen erstellt wurde ganz links außerhalb des Spalten-Zeilenrasters angezeigt (*Abbildung oben*). In Tabellen, die von links nach rechts wachsen, wird das Element, für das die Tabelle erstellt wurde, ganz oben außerhalb des Spalten-Zeilen-Rasters angezeigt (*Abbildung unten*).



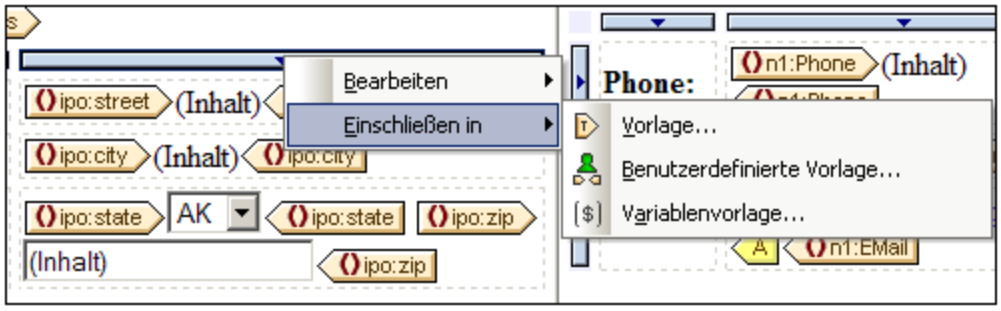
Nach Auswahl einer Spalte, Zeile oder Tabelle können in den Fenstern "Stile" und "Eigenschaften" Stile und/oder Eigenschaften für die Auswahl definiert werden.

Drag-and-Drop-Funktion

Die Spalten und Zeilen einer SPS-Tabelle (statisch oder dynamisch) können an eine andere Stelle innerhalb derselben Tabelle gezogen werden.

Einschließen von Zeilen und Spalten in Vorlagen und Entfernen der Vorlagen

Wenn Sie mit der rechten Maustaste auf eine Zeilen- bzw. Spaltenmarkierung klicken und im Kontextmenü (*Abbildung unten*) den Befehl **Einschließen in | Vorlage** oder **Einschließen in | Benutzerdefinierte Vorlage** auswählen, können Sie die Zeile bzw. Spalte in eine Vorlage einschließen. Im nächsten Schritt können Sie einen Node aus der Schema-Struktur auswählen oder einen XPath-Ausdruck für eine benutzerdefinierte Vorlage[235] eingeben. Daraufhin wird rund um die Zeile bzw. Spalte eine Vorlage angelegt.



Eine um eine Zeile bzw. Spalte angelegte Vorlage kann auch entfernt werden, wobei die Zeile bzw. Spalte davon nicht betroffen ist. Klicken Sie dazu auf den Vorlagen-Tag und drücken Sie die **Entf**-Taste.

Die Funktionen zum Einschließen in eine Vorlage und zum Entfernen der Vorlage sind nützlich, wenn Sie eine Vorlage löschen möchten, ohne dabei auch den Inhalt einer Zeile oder Spalte zu löschen. Sie können die Zeile/Spalte anschließend bei Bedarf in eine andere Vorlage einschließen. Wenn Sie die Zeile/Spalte in eine benutzerdefinierte Vorlage[235] einschließen, können Sie durch Filtern der Zeile bzw. Spalte auch interessante Ergebnisse erzielen (z.B. mit Hilfe von automatischen Berechnungen).

5.6.5 Formatieren von Tabellen

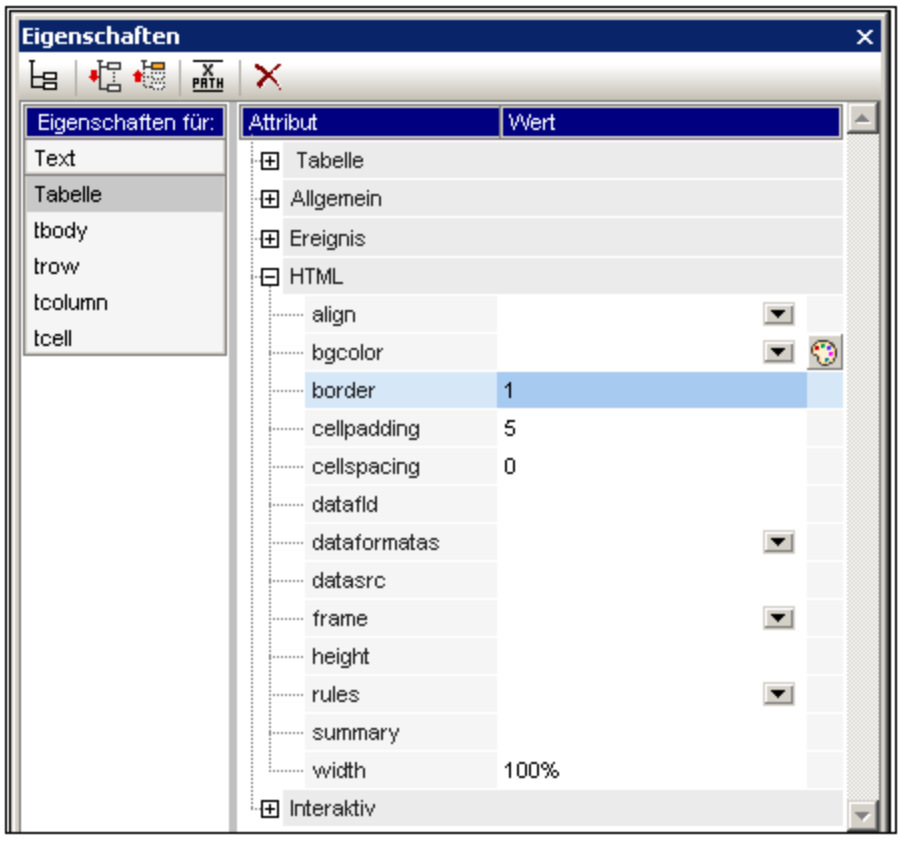
Statische und dynamische Tabellen

- können mit Hilfe von HTML-Tabellenformatierungseigenschaften (in der Seitenleiste "Eigenschaften")
- und CSS-Eigenschaften zum Definieren von Stilen (in der Seitenleiste "Stile") formatiert werden.

Seitenleiste "Eigenschaften"

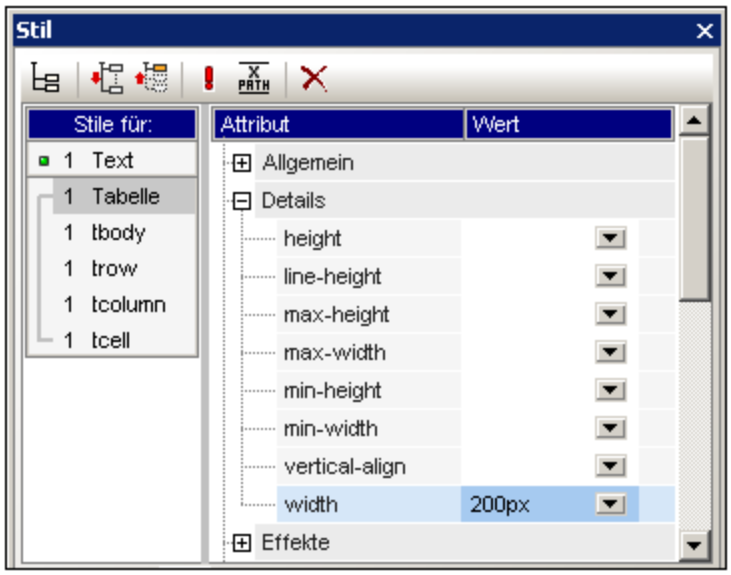
Die HTML-Tabellenformatierungseigenschaften stehen in der Seitenleiste "Eigenschaften" (*Abbildung unten*) zur Verfügung. Diese Eigenschaften sind in der Eigenschaftsgruppe "*HTML*" für die Tabellenkomponente und ihre Unterkomponenten (Body, Zeile, Spalte und Zelle) verfügbar.

© 2019-2025 Altova GmbH



Seitenleiste "Stil"

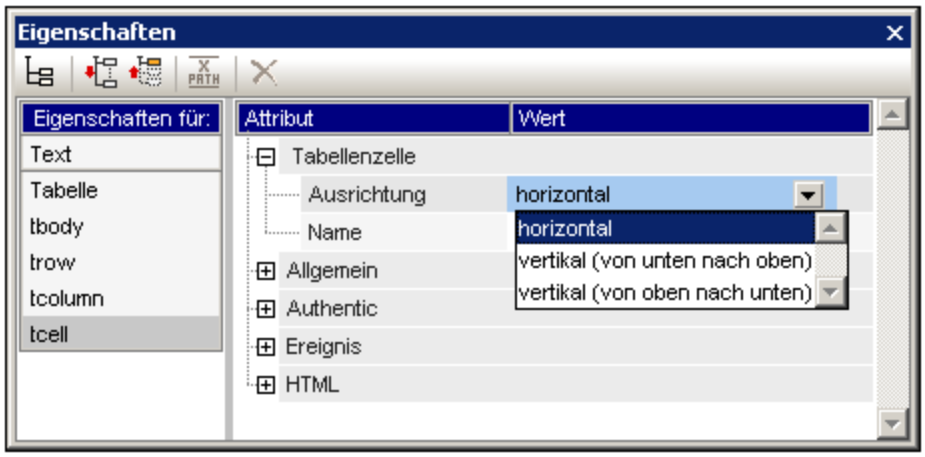
Die CSS-Tabellenformatierungseigenschaften stehen in der Seitenleiste "Stil" (*Abbildung unten*) zur Verfügung. CSS-Eigenschaften sind für die Tabellenkomponente und ihre Unterkomponenten verfügbar (Body, Zeile, Spalte und Zelle).



Anmerkung: Wenn alle Tabellenzellen einer Zeile leer sind, wird die Zeile im Internet Explorer reduziert, sodass die Zeile eventuell nicht sichtbar ist. Setzen Sie in diesem Fall ein geschütztes Leerzeichen in die entsprechende(n) Zelle(n).

Vertikaler Text

Text in Tabellenzellen kann um 90 Grad im oder entgegen dem Uhrzeigersinn gedreht werden, sodass der Text vertikal angezeigt wird und von oben nach unten bzw. von unten nach oben gelesen wird. Wählen Sie dazu im Design die gewünschte Tabellenzelle aus und wählen Sie in der Seitenleiste "Eigenschaften" (*Abbildung unten*) den Eintrag `tcell` aus. Wählen Sie in der Eigenschaftsgruppe *Tabellenzelle* für die Eigenschaft *Ausrichtung* den gewünschten Wert aus.



Beachten Sie die folgenden Punkte:

- Die Drehung wird an der Ausgabe vorgenommen, im Design aber nicht angezeigt.
- Diese Eigenschaft ist für die Anwendung auf Text gedacht und sollte nicht für anderen Inhalt verwendet werden.
- Die Eigenschaft kann nicht nur auf Text in Tabellenzellen, sondern auch auf Text in Textfeldern[177] angewendet werden.

Tabellenformatierung über Eigenschaften und Stile

Einige Formatierungseigenschaften stehen sowohl in der Seitenleiste "Eigenschaften" als auch in der Seitenleiste "Stile" zur Verfügung. In der nachfolgenden Tabelle sind einige der wichtigeren Tabelleneigenschaften aus den beiden Seitenleisten aufgelistet.

Tabellenkomponente	Seitenleiste "Eigenschaften"	Seitenleiste "Stil"
Tabelle	border, frame, rules; cellpadding, cellspacing; bgcolor; height, width *(wird durch height, width in der Seitenleiste "Stile" überschrieben, falls vorhanden); align*	Umrandungen (border) und Abstände (padding) in Feldstilen; height, width in der Gruppe "Details" (*überschreibt height und width in der Seitenleiste "Eigenschaften"*); Farbe, Schriftart und Textstile

© 2019-2025 Altova GmbH

Body	align, valign	height, vertical-align; Farbe, Schriftart und Textstile
Spalte	align, valign	width, vertical-align; Farbe, Schriftart und Textstile; Feldstile
Zeile	align, valign	height, vertical-align; Farbe, Schriftart und Textstile; Feldstile
Cell	align, valign	height, width, vertical-align; Farbe, Schriftart und Textstile; Feldstile

Höhe und Breite

Die Höhe und Breite von Tabellen, Zeilen, Spalten und Zellen muss in der Seitenleiste "Stile" (in der Stilgruppe "Details") definiert werden. Wenn die Größe einer Tabelle, Spalte oder Zeile in der Anzeige durch Ziehen mit der Maus angepasst wird, werden die geänderten Werte automatisch in der Seitenleiste "Stile" im entsprechenden Stil übernommen. Beachten Sie aber, dass die Stile "Höhe" und "Breite" bei Zellen, die einen Span-Wert haben (rowspan oder colspan), nicht unterstützt werden.

Zentrieren einer Tabelle

Um eine Tabelle zu zentrieren, setzen Sie die Eigenschaft `align` in der Gruppe "HTML" der Tabelleneigenschaften auf `center`. Die Eigenschaft `align` wird durch Auswahl der Tabelle und anschließende Auswahl des Menübefehls **Tabelle | Tabelleneigenschaften** aufgerufen. Alternativ dazu steht die Eigenschaft auch in der Seitenleiste "Eigenschaften" über die Eigenschaftsgruppe "HTML" zur Verfügung.

Um die Tabelle in der PDF-Ausgabe zu zentrieren, sind je nachdem, welchen FOP-Prozessor Sie verwenden, zusätzliche Einstellungen erforderlich. Laut FO-Spezifikation ist die richtige Methode zum Zentrieren einer Tabelle, das Element `fo:table` in ein `fo:table-and-caption` Element einzuschließen und das Attribut `text-align` des Elements `fo:table-and-caption` auf `center` zu setzen. Stylevision erstellt nicht automatisch ein `fo:table-and-caption` Element, wenn eine Tabelle in das Design eingefügt wird, doch Sie können dieses Element als benutzerdefiniertes Element 123 einfügen. Beachten Sie allerdings: Wenn Sie den Apache FOP-Prozessor verwenden, hängt es von der verwendeten FOP-Version ab, ob das Element `fo:table-and-caption` unterstützt wird oder nicht. In diesem Fall gibt es eine einfache Umgehungslösung: Machen Sie die Tabelle zu einer Tabelle mit fester Breite. Definieren Sie dazu als Wert der Eigenschaft `width` der Tabelleneigenschaftsgruppe "HTML" (Aufruf über **Tabelle | Tabelleneigenschaften**) einen Längenwert wie z.B. `4in` oder `120mm`.

Zuweisen unterschiedlicher Hintergrundfarben für die einzelnen Zeilen

Wenn Sie für die Zeilen Ihrer dynamischen Tabelle unterschiedliche Hintergrundfarben definieren möchten, gehen Sie folgendermaßen vor:

1. Wählen Sie den Zeilenanzeiger der Zeile aus, für die abwechselnd zwei Hintergrundfarben benötigt werden. Bedenken Sie, dass es sich um eine dynamische Tabelle handelt, in der ein Element als Zeile definiert ist. Das Design enthält eine einzige Zeile, die für das Element steht, das als Tabelle angezeigt werden soll.
2. Klicken Sie, während der Zeilenanzeiger ausgewählt ist, in der Seitenleiste "Eigenschaften" auf die *Eigenschaften für:* `trow`.
3. Wählen Sie die Eigenschaft `bgcolor` aus.

4. Klicken Sie in der Symbolleiste des Fensters "Eigenschaften" auf die Schaltfläche "XPath" und geben Sie im daraufhin angezeigten Dialogfeld XPath-Ausdruck bearbeiten[440] einen XPath-Ausdruck ein, der in etwa folgendermaßen aussieht:

```
if ( position() mod 2 = 0 ) then "white" else "gray"
```

   Mit diesem XPath-Ausdruck wird eine `bgcolor` (Hintergrundfarbe) Weiß für Zeilen mit geraden Zahlen und eine `bgcolor` Grau für Zeilen mit ungerader Nummerierung definiert.

Mit der oben beschriebenen Methode können Sie noch komplexere Formatierungen definieren.

Nummerieren der Zeilen einer dynamischen Tabelle

Sie können die Zeilen einer dynamischen Tabelle mit Hilfe der `position()` Funktion von XPath nummerieren. Fügen Sie dazu zuerst eine Spalte in die Tabelle ein, die die Nummerierung enthalten soll, und fügen Sie anschließend in die Zelle dieser Spalte eine automatische Berechnung mit einem XPath: `position()` ein. Da der Kontext-Node das Element ist, das der Zeile der dynamischen Tabelle entspricht, gibt die `position()` Funktion die Position der einzelnen Zeilenelemente innerhalb der Gruppe der Zeilenelemente zurück.

Tabellenkopf- und -fußzeilen in der PDF-Ausgabe

Wenn die Tabelle mehr als eine Seite einnimmt, wird auf jeder Seite, die die Tabelle enthält, die Tabellenkopf- und Fußzeile angezeigt. Beachten Sie die folgenden Punkte:

- Wenn die Fußzeile automatische Berechnungen enthält, enthält die Fußzeile, die am Ende des Tabellensegments jeder Seite angezeigt wird, die automatische Berechnung für die gesamte Tabelle - nicht nur die Berechnung für das Tabellensegment auf dieser Seite.
- Die Kopf- und Fußzeile werden für einzelne Seiten nicht deaktiviert (z.B. wenn Sie nur am Ende der Tabelle und nicht am Ende jeder Seite eine Fußzeile möchten).

Damit die Kopf- oder Fußzeile nicht an jedem Seitenende angezeigt wird, verwenden Sie im `table`-Element die Eigenschaften `table-omit-header-at-break` und/oder `table-omit-footer-at-break` (Attribute). Diese Eigenschaften stehen in der Seitenleiste "Stile" in der Eigenschaftsgruppe XSL-FO für die Tabelle zur Verfügung. Um die Kopf- oder Fußzeile am Seitenende wegzulassen, geben Sie für das jeweilige Attribut einen Wert `true` ein. (Beachten Sie, dass der Standardwert `false` ist. Wenn Sie diese Eigenschaften daher nicht definieren, wird die Kopf- und Fußzeile bei jedem Seitenumbruch eingefügt.)

Abteilungszeichen für den Inhalt von Tabellenzellen

Wenn der Text in Tabellenzellen Ihrer PDF-Ausgabe abgeteilt werden soll, beachten Sie, dass dafür in der XSL:FO-Spezifikation das Attribut **`hyphenate`** des Elements **`fo.block`** vorgesehen ist. Um Text also abzuteilen, müssen Sie die Eigenschaft **`hyphenate`** der entsprechenden **`fo:block`**-Elemente im FO-Dokument explizit auf **`true`** setzen.

5.6.6 Anzeigen von Zeilen und Spalten

Für Tabellen stehen **nur** in der **HTML-Ausgabe** die folgenden Optionen zur Anzeige von Zeilen und Spalten zur Verfügung. Diese Funktionen werden in der **Authentic-Ansicht nicht unterstützt.** Um diese Funktionen verwenden zu können, muss als XSLT-Version des SPS XSLT 2.0 oder XSLT 3.0 ausgewählt sein.

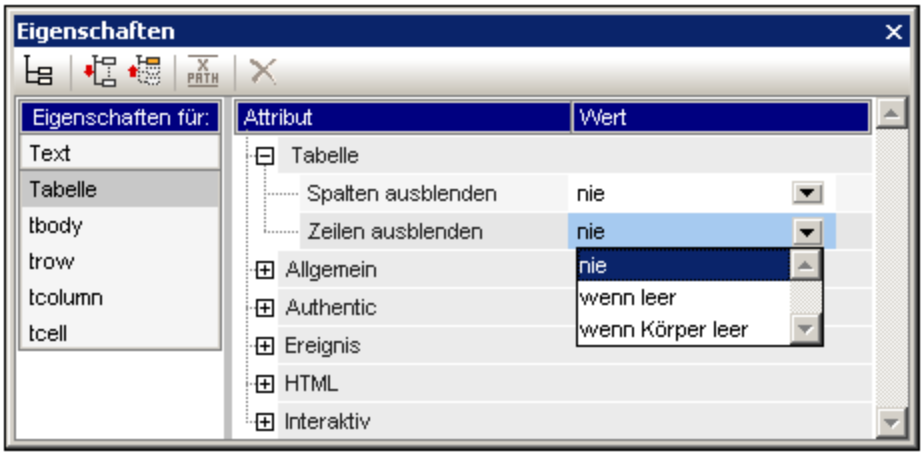
© 2019-2025 Altova GmbH

- Leere Zeilen und Spalten können automatisch ausgeblendet werden
- Jede Spalte kann eine **Schließen**-Schaltfläche haben, mit der die einzelnen Spalten ausgeblendet werden können
- Zeilenelemente mit untergeordneten Elemente können mit Erweitern/Reduzieren-Schaltflächen angezeigt werden

Standardmäßiges Ausblenden leerer Zeilen und Spalten

Um leere Zeilen und/oder Spalten in der HTML-Ausgabe auszublenden, gehen Sie folgendermaßen vor:

1. Wählen Sie die Tabelle oder einen Teil davon (Spalte, Zeile, Zelle) in der Design-Ansicht aus.
2. Wählen Sie in der Seitenleiste "Eigenschaften" die Eigenschaften für *Tabelle* aus und die Eigenschaftsgruppe *Tabelle* (*Abbildung unten*).



3. Wählen Sie für die Eigenschaften *Spalten ausblenden* und *Zeilen ausblenden* die gewünschten Werte aus. Die Optionen für jede dieser beiden Eigenschaften sind dieselben: *Nie*, Wenn leer und *Wenn Körper leer*. Mit der Option *Wenn leer* wird die Spalte oder Zeile ausgeblendet, wenn die gesamte Spalte/Zeile (einschließlich Kopf- und Fußzeile) leer ist. Bei *Wenn Körper leer* muss nur der Hauptteil leer sein.

Anmerkung: Wenn sich in einer Nicht-XBRL-Tabelle Zeilen oder Spalten über mehrere Zeilen bzw. Spalten erstrecken, lassen sich leere Zeilen/Spalten unter Umständen nicht ausblenden.

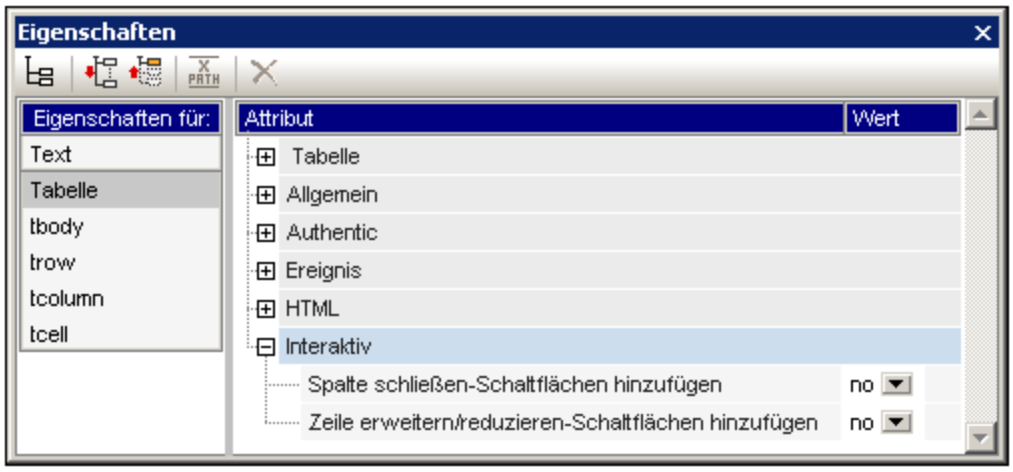
Ausblenden von Spalten und Erweitern/Reduzieren von Zeilen durch den Benutzer

Sie können im Design definieren, dass jede einzelne Tabellenspalte in der HTML-Ausgabe (*siehe Abbildung unten*) eine **Schließen**-Schaltfläche enthält. Wenn der Benutzer auf diese Schaltfläche klickt, kann er einzelne Spalten ausblenden. Nachdem die Spalte ausgeblendet wurde, wird in der ersten Spalte ein Plus-Symbol angezeigt (*siehe Abbildung unten*). Wenn Sie auf dieses Symbol klicken, werden alle ausgeblendeten Spalten wieder angezeigt.

Balance Sheet (in Millions)	2004-09-30	2004-07-01 - 2004-09-30	2003-12-31	2004-01-01
Assets, Total	€21.49		€24.02	
Current Assets, Total	€10.65		€12.32	
Non Current Assets, Total	€10.85		€11.7	
Liabilities and Equity, Total	€21.49		€24.02	
Liabilities, Total	€8.9		€10.79	
Minority Interests				
Equity, Total	€12.59		€13.23	
Issued Capital and Reserves	€12.59		€13.23	

Auch Zeilenelemente, die untergeordnete Elemente haben, können in der HTML-Ausgabe mit einem Erweitern/Reduzieren-Symbol (Plus/Minus) daneben (*siehe Abbildung oben*) angezeigt werden. Wenn Sie in der HTML-Ausgabe auf dieses Symbol klicken, wird dieses Zeilenelement erweitert bzw. reduziert. Sie können die Einrückung für einzelne Zeilen im Design über CSS-Eigenschaften definieren.

Die Einstellungen für diese beiden Funktionen werden in den Eigenschaften *Tabelle* in der Eigenschaftsgruppe *Interaktiv* vorgenommen (*Abbildung oben*).



Die Optionen für beide Eigenschaften sind *Yes* (um die Funktion hinzuzufügen) und *No* (um die Funktion nicht hinzuzufügen).

5.6.7 CALS/HTML-Tabellen

Eine XML-Tabelle ist eine hierarchische XML-Struktur, deren Elemente (i) die Struktur einer CALS- oder HTML-Tabelle definieren, (ii) die Formatierung dieser Tabelle festlegen und (iii) den Zelleninhalt der Tabelle enthalten. Diese XML-Struktur muss genau dem CALS- bzw. HTML-Tabellenmodell entsprechen.

Um eine CALS/HTML-Tabelle im Design zu erstellen, gehen Sie folgendermaßen vor:

1. Definieren Sie die XML-Struktur als CALS/HTML-Tabellenstruktur 143
2. Definieren Sie die Formatierungsstile für die Tabelle 145
3. Fügen Sie die CALS/HTML-Tabellen in das SPS-Design ein 145

© 2019-2025 Altova GmbH

Aktivieren der CALS/HTML-Tabellenstrukturen für die Ausgabe

Ein XML-Dokument kann eine Datenstruktur haben, in der die Struktur und der Inhalt einer Tabelle definiert sind. So entspricht etwa die folgende XML-Datenstruktur dem HTML-Tabellenmodell und weist auch dieselben Elementnamen wie das HTML-Tabellenmodell auf:

```
<table>
  <tbody>
    <tr>
      <td/>
    </tr>
  </tbody>
</table>
```

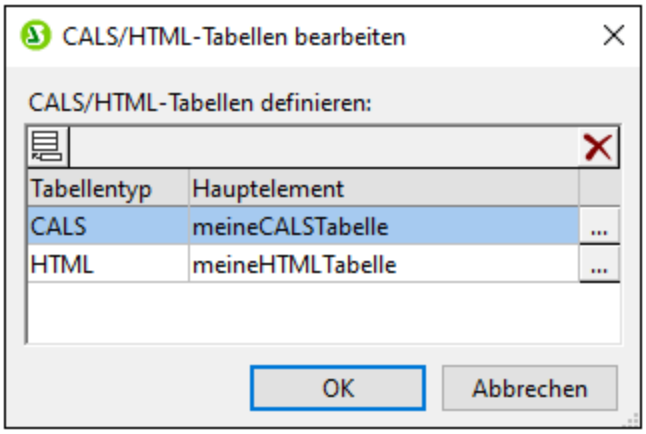
Alternativ dazu könnte die XML-Datenstruktur der einer HTML-Tabelle entsprechenden, mit dem Unterschied, dass die Elementnamen sich von denen im HTML-Tabellenmodell unterscheiden. Beispiel:

```
<semester>
  <subject>
    <class>
      <student/>
    </class>
  </subject>
</semester>
```

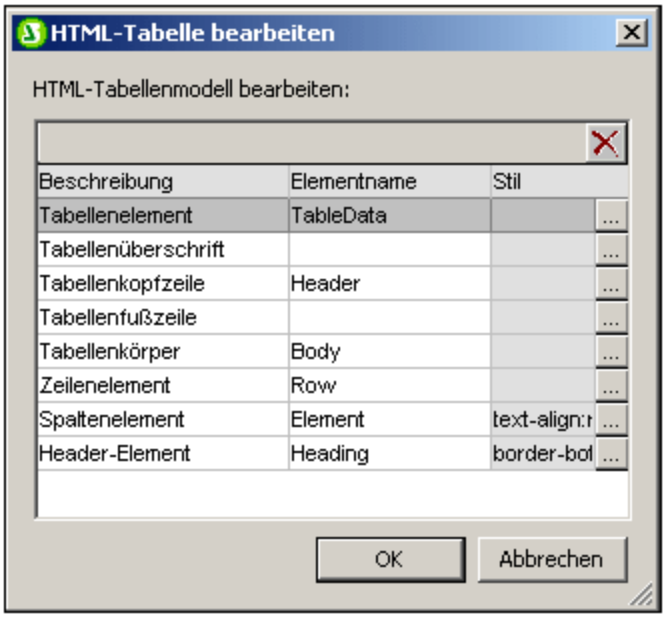
Anhand dieser im XML-Dokument definierten Tabellenstruktur kann auf direktem Weg eine Tabelle in den verschiedenen Ausgabeformaten generiert werden. Dazu muss diese XML-Datenstruktur als CALS- oder HTML-Tabelle definiert werden. Wenn die XML-Datenstruktur nicht als CALS- oder HTML-Format (Standardeinstellung) definiert ist, werden die Elemente in der Datenstruktur als gewöhnliche Nicht-Tabellenelemente behandelt und es wird kein Tabellen-Markup-Code zum Ausgabedokument hinzugefügt.

Um die Ausgabe von CALS/HTML-Tabellen-Markup-Code in der Ausgabe zu aktivieren, gehen Sie folgendermaßen vor:

1. Wählen Sie den Befehl **Tabelle | CALS/HTML-Tabellen bearbeiten** aus.
2. Fügen Sie im Dialogfeld, das daraufhin angezeigt wird (*Abbildung unten*) einen Eintrag für die Datenstruktur hinzu, die Sie als CALS/HTML-Tabelle verwenden möchten, je nachdem ob die Datenstruktur dem CALS- oder dem HTML-Tabellenmodell entspricht. (Informationen zum CALS-Tabellenmodell finden Sie unter dem CALS-Tabellenmodell bei OASIS. Ein Beispiel für ein `table`-Element mit einer HTML-Tabellenstruktur finden Sie in `HTMLTable1.sps`, das sich (im Projektfenster der Benutzeroberfläche) im Ordner `Basics` des Projektordners `Examples` befindet. Wenn Sie also ein Element in Ihre Schema als CALS- oder als ein HTML-Tabellenelement aktivieren möchten, klicken Sie im linken oberen Bereich des Dialogfelds auf die Tabellenschaltfläche "CALS/HTML-Tabelle hinzufügen" und wählen Sie anschließend einen der beiden Befehle **CALS-Tabelle hinzufügen** oder **HTML-Tabelle hinzufügen** aus. (In der Abbildung unten wurden die Elemente `table` und `informaltable` als CALS-Tabellen (sowie als HTML-Tabellen) aktiviert. Klicken Sie zur Bestätigung auf **OK**.



3. Es wird ein Dialogfeld (CALS-Tabelle bearbeiten bzw. HTML-Tabelle bearbeiten) angezeigt, in dem die Elemente des ausgewählten Tabellentyps angezeigt werden (*Abbildung unten*). Die in diesem Dialogfeld aufgelisteten Elemente (CALS oder HTML) sind standardmäßig die Elementnamen im ausgewählten Tabellenmodell. Wenn das SPS-Schema Elemente dieses Namens enthält, so werden diese schwarz angezeigt (*wie in der Abbildung unten*). Wenn einer der aufgelisteten Elementnamen im SPS-Schema nicht vorhanden ist, so wird dieser Elementname rot angezeigt. Sie können den Namen eines aufgelisteten Elements ändern, sodass er einem Schemanamen entspricht. Doppelklicken Sie dazu auf das entsprechende *Elementnamensfeld* und bearbeiten Sie den Namen.



4. Klicken Sie auf **OK,** um diese XML-Datenstruktur als CALS- oder HTML-Tabelle zu definieren.
5. Sie können Einträge für beliebig viele XML-Datenstrukturen hinzufügen (*siehe Abbildung oben in Schritt 2)*. Dasselbe Hauptelement kann einmal für CALS- und einmal für HTML-Tabellentypen verwendet werden.
6. Nachdem Sie die gewünschten Datenstrukturen, die als CALS/HTML-Tabellen aktiviert werden sollen, fertig bearbeitet haben, klicken Sie zur Bestätigung auf **OK.**

Wenn eine CALS/HTML-Tabelle definiert wurde und die XML-Datenstruktur korrekt als CALS/HTML-Tabelle eingefügt wurde[145], so wird die Datenstruktur als Tabelle an die Ausgabegesendet. Um eine CALS/HTML-

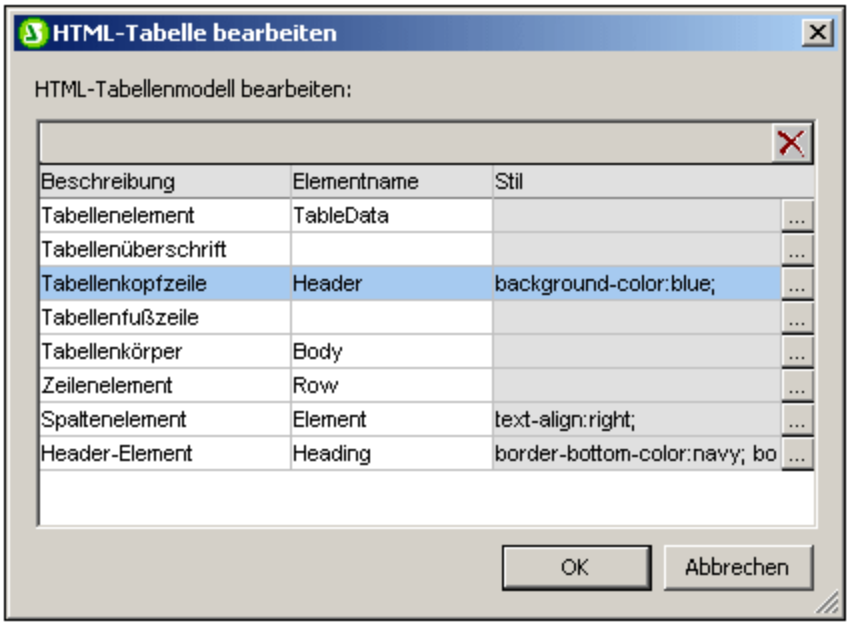
 © 2019-2025 Altova GmbH

Tabellendefinition zu entfernen, wählen Sie die zu löschende Definition im Dialogfeld "CALS/HTML-Tabelle" aus und klicken Sie rechts oben im Fenster "CALS/HTML-Tabellen" auf die Schaltfläche **Löschen.**

Tabellenformatierung

Es gibt zwei Arten um CALS/HTML-Tabellen zu formatieren:

1. Mittels Formatierungsattributen im XML-Quelldokument. Das CALS- und das HTML-Tabellenmodell sieht Formatierungsattribute vor. Falls solche Attribute im XML-Quelldokument vorhanden sind, werden sie an die Darstellungsattribute des Tabellen-Markup-Codes der Ausgabe übergeben.
2. Jedes einzelne Element in der Tabelle kann im Dialogfeld "CALS-Tabelle bearbeiten" bzw. "HTML-Tabelle bearbeiten" in der Spalte "Stile" formatiert werden (*siehe Abbildung unten*).



Um einem bestimmten Element einen Stil zuzuweisen, klicken Sie auf die Schaltfläche **Stile hinzufügen** für das jeweilige Element und weisen Sie die gewünschten Stile über die daraufhin angezeigte Seitenleiste "Stile" 47 zu. Jeder Stil wird als eigenes CSS-Attribut zum jeweiligen Element hinzugefügt. Beachten Sie, dass ein Stil, der über das Attribut `style` hinzugefügt wurde, eine höhere Priorität hat, als wenn der Stil über als eigenes CSS-Attribut (wie z.B. `bgcolor`) zugewiesen wurde. So hat z.B. in `<thead style="background-color: red" bgcolor="blue"/>` das Attribut`style="background-color: red"` Vorrang vor dem Attribut `bgcolor="blue"`.

Um einen Stil zu entfernen, der einem Element in der CALS/HTML-Tabellendefinition zugewiesen wurde, wählen Sie das entsprechende Element (z.B. in der Abbildung oben das Element `thead`) aus und klicken Sie auf die Schaltfläche **Löschen**. Die Stile für dieses Element werden daraufhin entfernt.

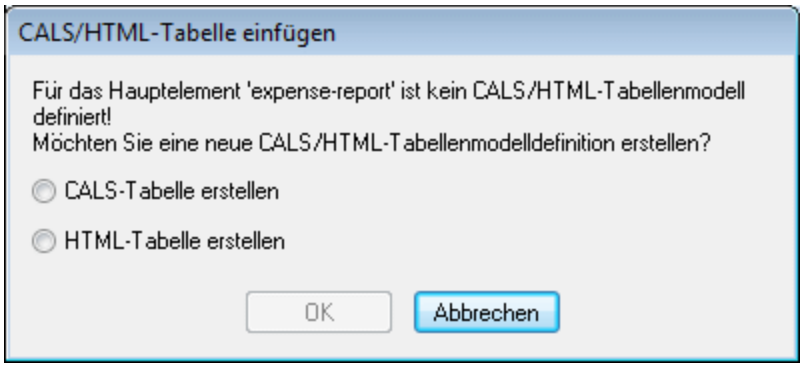
Einfügen einer CALS/HTML-Tabelle in das Design

Eine CALS/HTML-Tabellenstruktur kann auf zwei Arten in das Design eingefügt werden:

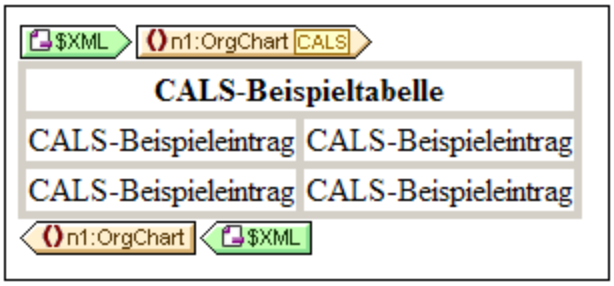
1. Durch Einfügen des Parent-Elements des Tabellenelements in das Design als `(Inhalt)`. Wenn der Inhalt des Parent-Elements verarbeitet wird, wird das Tabellenelement verarbeitet. Wenn die

CALS/HTML-Tabellenausgabe aktiviert ist, so wird das Element in Form einer Tabelle ausgegeben. Andernfalls wird es als Text ausgegeben.

2. Indem Sie das Tabellenelement aus der Schema-Struktur ins Design ziehen. Wenn Sie die Maustaste an der gewünschten Stelle im Design loslassen, kann das Element (mit dem Befehl **CALS/HTML-Tabelle erstellen**) als CALS/HTML-Tabelle angelegt werden. Wenn das Element nicht als CALS/HTML-Tabellenelement definiert wurde[143], so wird das Dialogfeld "CALS/HTML-Tabelle einfügen" (*Abbildung unten*) aufgerufen und Sie können das CALS- oder HTML-Tabelleelement definieren.



Wenn das Element im Design als CALS/HTML-Tabellenelement erstellt wurde, wird an der Stelle ein Platzhalter für das CALS/HTML-Tabellendesignelement eingefügt (*Abbildung unten).*



Globale Vorlagen von Tabellenelementen

Wenn globale Vorlagen[231] der folgenden Tabellenelemente erstellt werden, werden diese in der CALS/HTML-Tabellenausgabe verwendet. Für CALS-Tabellen sind dies die Elemente `title` und `entry`. Für HTML-Tabellen sind dies die Elemente `caption`, `th` und `td`.

Beispieldateien

Beispieldateien befinden sich (im Projektfenster der Benutzeroberfläche) im Projektordner `Examples`.

© 2019-2025 Altova GmbH

5.7 Listen

Im SPS können zwei Arten von Listen erstellt werden:

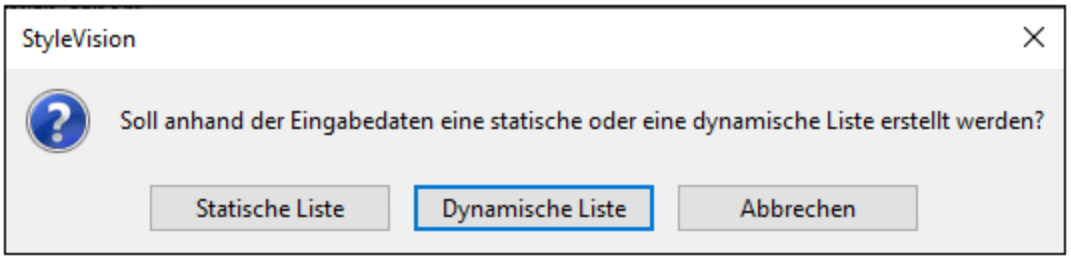
- Statische Listen[147]: Dies sind Listen, deren Inhalt direkt in das SPS eingegeben wird. Die Listenstruktur wird nicht dynamisch anhand der Struktur des XML-Dokuments erstellt.
- Dynamische Listen[149]: Dies sind Listen, deren Struktur und Inhalt dynamisch anhand des XML-Dokuments erstellt wird.

Anleitung zur Erstellung dieser beiden Listenarten finden Sie in den Unterabschnitten dieses Abschnitts.

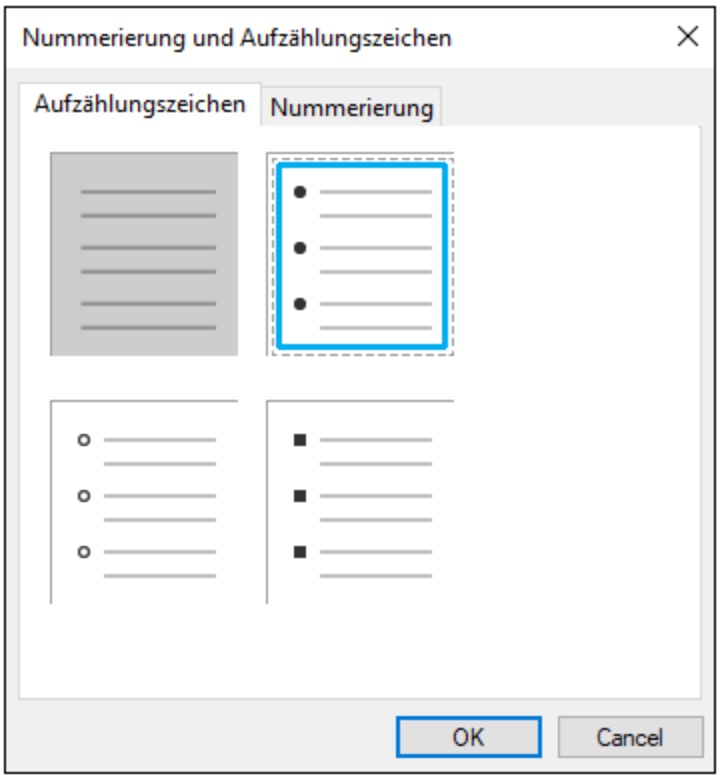
5.7.1 Statische Listen

Bei einer statischen Liste wird der Inhalt der Liste direkt in das SPS eingegeben. Um eine statische Liste zu erstellen, gehen Sie folgendermaßen vor:

1. Platzieren Sie den Cursor an die Stelle im Design, an der die statische Liste erstellt werden soll und wählen Sie den Menübefehl **Einfügen | Nummerierung und Aufzählungszeichen**[513] (oder klicken Sie in der Symbolleiste "Design-Elemente einfügen"[462] auf die Schaltfläche "Nummerierung und Aufzählungszeichen"). Daraufhin wird ein Dialogfeld angezeigt, in dem Sie gefragt werden, ob Sie eine statische oder eine dynamische Liste erstellen möchten (*Abbildung unten*).



2. Klicken Sie auf **Statische Liste**. Daraufhin wird das Dialogfeld "Nummerierung und Aufzählungszeichen" (*Abbildung unten*) angezeigt.



3. Wählen Sie die gewünschte Kennzeichnung aus und klicken Sie auf **OK.** Es wird nun eine leere Liste erstellt.
4. Geben Sie den Text für den ersten Eintrag in der Liste ein.
5. Drücken Sie die **Eingabetaste,** um einen neuen Listeneintrag zu erstellen.

Um eine verschachtelte Liste zu erstellen, platzieren Sie den Cursor in den Listeneintrag, der die untergeordnete Liste enthalten soll, und wählen Sie den Menübefehl **Einfügen | Nummerierung und Aufzählungszeichen**[513]. Gehen Sie dann vor, wie oben beschrieben.

Anmerkung: Sie können die statische Liste auch durch Platzieren des Cursors an die gewünschte Stelle im Design und Klicken entweder auf die Schaltfläche "Aufzählungszeichen" auf die Schaltfläche "Nummerierung" in der Symbolleiste Format[459] erstellen. Der erste Listeneintrag wird am Cursoreinfügepunkt erstellt.

Umwandeln von statischem Text in eine Liste

So ändern Sie statischen Text in eine Liste um:

Markieren Sie den gewünschten Text, klicken Sie auf **Einschließen in | Nummerierung und Aufzählungszeichen**[513], wählen Sie die gewünschte Art der Listenkennzeichnung aus und klicken Sie auf **OK.** Wenn der Text CR-LF, Zeilenschaltungen und/oder Zeilenvorschubzeichen enthält (die durch Drücken der **Eingabetaste** eingefügt wurden), wird für jedes Textfragment, das durch ein CR-LF-Zeichen getrennt ist, ein separater Listeneintrag erstellt. Wenn Sie ein Textfragment innerhalb einer Zeile markiert haben, so wird dieser Text als Listeneintrag einer aus einem einzigen Eintrag bestehenden Liste angelegt; durch Drücken der **Eingabetaste** können Sie so oft Sie möchten beliebig viele Listeneinträge hinzufügen. Beachten Sie, dass Sie den Befehl **Einschließen in | Nummerierung und Aufzählungszeichen**[513] auch über das Kontextmenü aufrufen können.

© 2019-2025 Altova GmbH

5.7.2 Dynamische Listen

In einer dynamischen Liste wird der Inhalt einer Gruppe von gleichrangigen Nodes desselben Namens angezeigt, wobei jeder Node, als Listeneintrag in der Liste dargestellt wird. Das Element, dessen Instanzen als Einträge der Liste erscheinen sollen, wird als Liste erstellt. Vorgehensweise und Verwendung sind im Folgenden erklärt.

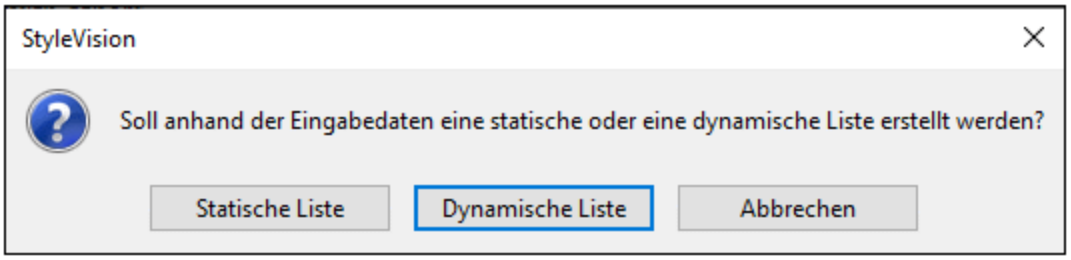
Grundprinzip

- Es kann jedes Element als Liste erstellt werden.
- Wenn ein Element als Liste erstellt wird, werden die Instanzen dieses Elements als Listeneinträge erstellt. Wenn z.B. das Element `department` mehrere `person` Elemente (d.h. Instanzen) enthält und Sie eine Liste aller Mitarbeiter der Abteilung erstellen möchten, müssen Sie das Element `person` als Liste erstellen.
- Nachdem die Liste für das Element erstellt wurde, können Sie das Aussehen oder den Inhalt der Liste bzw. des Listeneintrags durch Einfügen zusätzlichen statischen oder dynamischen Inhalts wie z.B. Text, automatischen Berechnungen, dynamischen Inhalt usw. ändern.

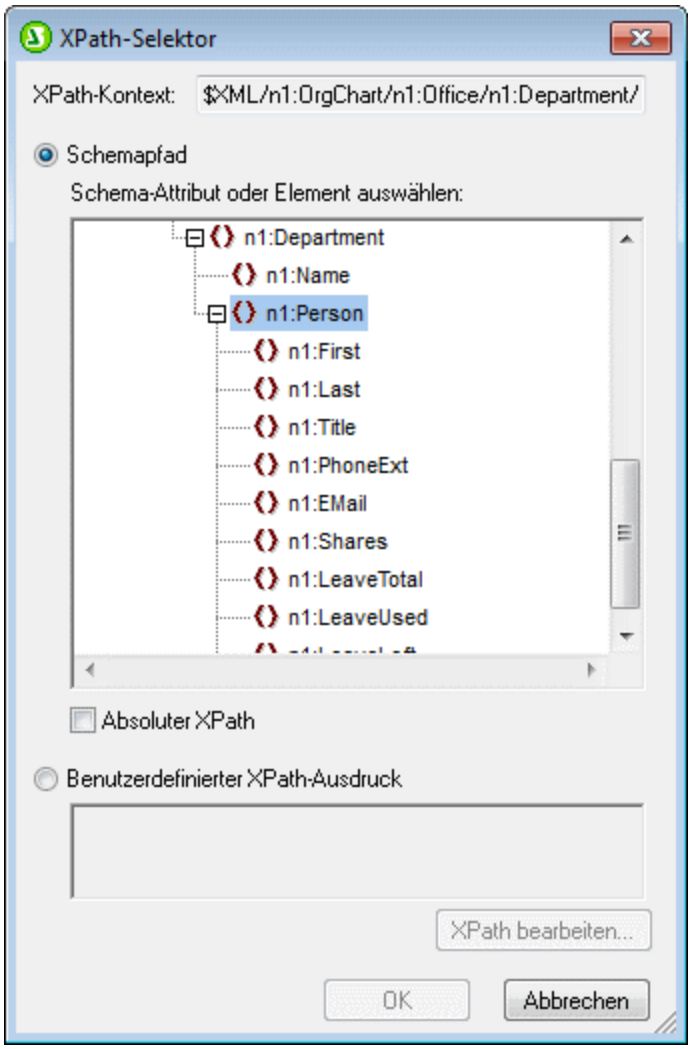
Erstellen einer dynamischen Liste

Um eine dynamische Liste zu erstellen, gehen Sie folgendermaßen vor:

1. Platzieren Sie den Cursor an die Stelle im Design, an der die dynamische Liste erstellt werden soll und wählen Sie den Menübefehl **Einfügen | Nummerierung und Aufzählungszeichen**[513]. Daraufhin wird ein Dialogfeld angezeigt, in dem Sie gefragt werden, ob Sie eine statische oder eine dynamische Liste erstellen möchten (*Abbildung unten*).



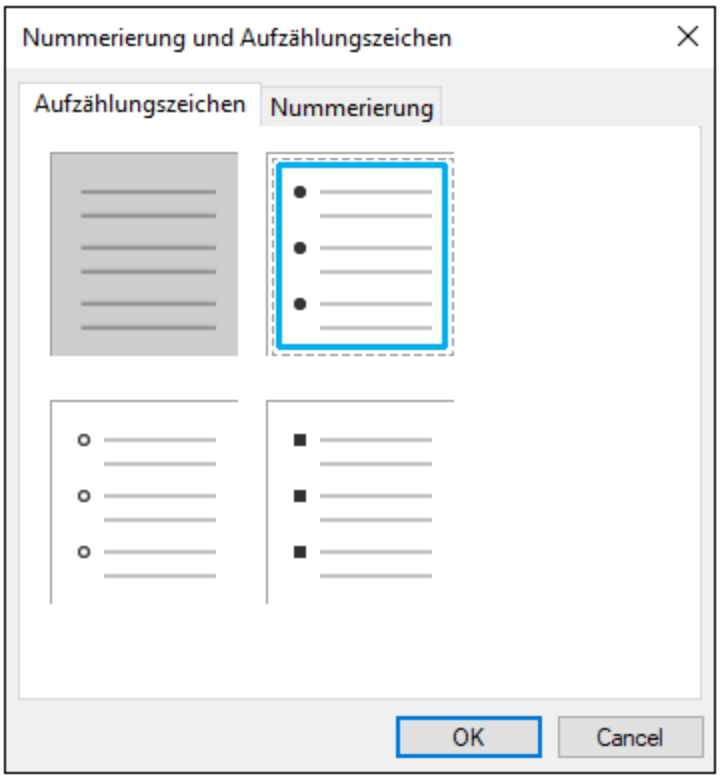
2. Klicken Sie auf **Dynamische Liste**. Daraufhin wird das Dialogfeld "XPath-Selektor" (*Abbildung unten)* angezeigt.
3. Beachten Sie, dass der XPath-Kontext im Dialogfeld "XPath-Selektor" der Kontext an der Einfügestelle ist und dass dieser Kontext im Dialogfeld nicht geändert werden kann. Wählen Sie den Node aus, der als dynamische Liste erstellt werden soll. In der Abbildung unten ist der Kontext-Node `n1:Department` und der Node `n1:Person` wurde als der Node ausgewählt, der als Liste erstellt werden soll. D.h. der Inhalt der einzelnen `n1:Person` Nodes wird in Form von Listeneinträgen erstellt.



Wenn Sie die Option "Benutzerdefinierter XPath-Ausdruck" auswählen, können Sie einen XPath-Ausdruck eingeben, der den Node auswählt, der als dynamische Tabelle erstellt werden soll. Wenn Sie auf **OK** klicken, wird das im nächsten Schritt beschriebene Dialogfeld "Nummerierung und Aufzählungszeichen" angezeigt.

4. Wählen Sie hier den gewünschten Listentyp aus. Sie haben die Wahl zwischen einer Liste mit Aufzählungszeichen (mit den Aufzählungszeichen Punkte, Kreise oder Quadrate) oder einer nummerierten Liste. Klicken Sie auf **OK**, um die Liste mit der auswählten Variante zu erstellen).

© 2019-2025 Altova GmbH


Nummerierung und Aufzählungszeichen
Aufzählungszeichen
Nummerierung
OK
Cancel

5.8 Grafiken

Wenn Sie Bilder in das Design-Dokument einfügen, kann der Pfad zum Bild direkt im SPS angegeben werden (vom Ersteller des SPS) oder er kann aus einem Node im XML-Dokument geholt oder abgeleitet werden. Eine Beschreibung, wie man den Pfad zum Bild definiert, finden Sie im Abschnitt Bild-URIs 152. Eine Liste der Bildtypen, die in den verschiedenen Ausgabedateien unterstützt werden, finden Sie im Abschnitt Bildtypen und Ausgabe 154.

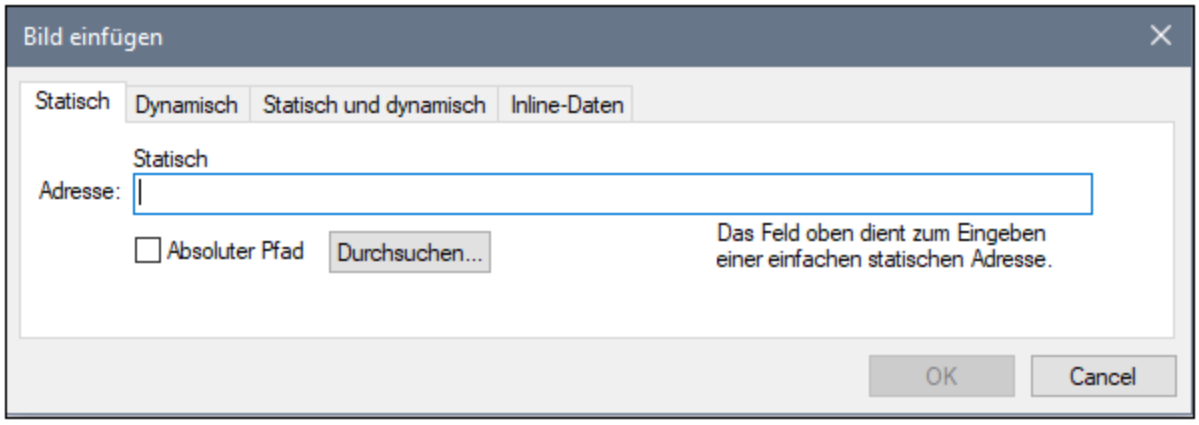
Bildeigenschaften

Bilder können im Fenster "Eigenschaften" definiert werden. Wählen Sie dazu das Bild im Design aus. Wählen Sie anschließend im Fenster "Eigenschaften" (i) das *Bild* in den Eigenschaften für die Spalte aus, (ii) wählen Sie die gewünschte Eigenschaftsgruppe aus und (iii) wählen Sie in der ausgewählten Eigenschaftsgruppe die gewünschte Eigenschaft aus. Um z.B. die Höhe und Breite des Bild einzustellen, definieren Sie in der Eigenschaftsgruppe *HTML* die Eigenschaften `Höhe` und `Breite`.

5.8.1 Bilder: URIs und Inline-Daten

Sie können an jeder Stelle im Design-Dokument Bilder einfügen. Diese Bilder werden in den Ausgabedokumenten angezeigt; in der Design-Ansicht eingefügte Bilder werden durch Miniaturansichten oder Platzhalter gekennzeichnet.

Um ein Bild einzufügen, klicken Sie auf den Menübefehl **Einfügen | Bild einfügen** 509, woraufhin das Dialogfeld "Bild einfügen" (*Abbildung unten*) angezeigt wird.



Sie können auf zwei Arten auf Bilder zugreifen:

- Das Bild ist eine Datei, auf die zugegriffen wird, indem ihre URI in das Dialogfeld "Bild einfügen" eingegebne wird.
- Das Bild ist als Base 16- oder Base 64-Text in einer XML-Datei kodiert.

Einfügen einer Bilddatei

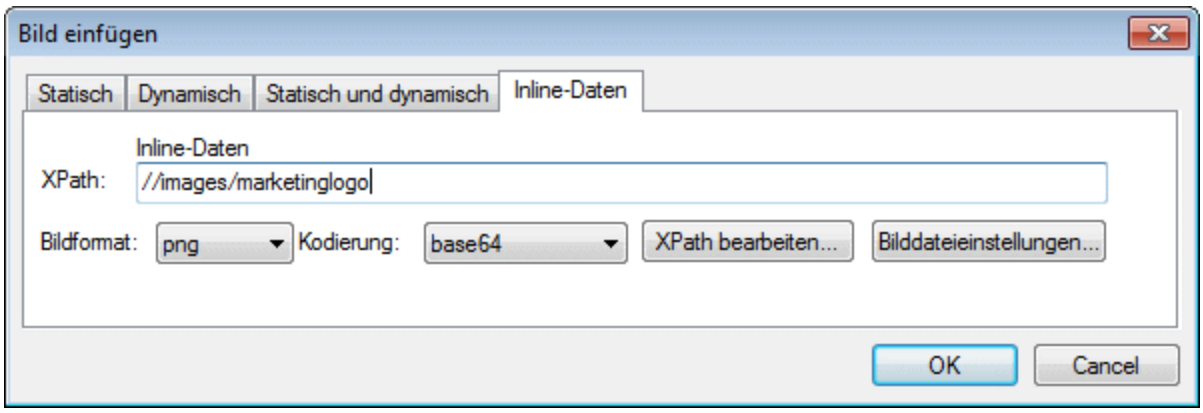
Ein Bilddatei wird in das Design eingefügt, indem ihre URI angegeben wird. Diese Datei wird während der Laufzeit aufgerufen und im Dokument platziert. Die Bild-URI kann auf drei Arten im Dialogfeld "Bild einfügen" (*Abbildung oben*) eingegeben werden:

© 2019-2025 Altova GmbH

- auf dem Register "Statisch" wird die URI direkt als absolute oder relative URI, z.B. `nanonull.gif` (*relative URI; siehe Abschnitt unten*[154]) und `C:\images\nanonull.gif` (*absolute URI*).
- auf dem Register "Dynamisch" als XPath-Ausdruck, der einen Node auswählt, der entweder (i) eine (absolute oder relative) URI enthält oder (ii) einen Namen einer ungeparsten Entity[377]. So würde z.B. mit dem Eintrag `image/@location` das Attribut `location` des Elements `image` ausgewählt, welches das Child des Kontext-Node (also des Node, in den das Bild eingefügt ist) ist. Der Node "location" im XML-Dokument würde die URI des Bildes enthalten. Eine Beschreibung zur Verwendung von ungeparsten Entities finden Sie im Abschnitt URIs von ungeparsten Entities[377].
- Auf dem Register "Statisch" und "Dynamisch" können einem XPath-Ausdruck im dynamischen Teil statische Einträge (Text) voran- und/oder nachgestellt werden. Das statische Präfix könnte z.B. lauten `C:/XYZCompany/Personnel/Photos/`; der dynamische Teil könnte lauten `concat(First, Last)`; und das statische Suffix könnte lauten `.png`. Das Ergebnis wäre eine absolute URI, die in etwa folgendermaßen aussieht: `C:/XYZCompany/Personnel/Photos/JohnDoe.png`.

Einfügen eines Bildes, das als kodierter Text vorliegt

Ein Bild kann in einer XML-Datei als Base 16- oder Base 64-kodierter Text gespeichert sein. Der Vorteil liegt darin, dass das Bild nicht aus einer separaten Datei aufgerufen (auf die verwiesen wird) werden muss, sondern als Text im XML-Quelldokument vorhanden ist. Um ein Bild, das als kodierter Text in der XML-Quelle verfügbar ist, einzufügen, verwenden Sie das Register "Inline-Daten" des Dialogfelds "Bild einfügen" (*siehe Abbildung unten*).



Verwenden Sie einen XPath-Ausdruck, um den Node im XML-Dokument anzugeben, der den kodierten Text des Bildes enthält. Wählen Sie eine Option aus der Auswahlliste "Bildformat", um anzugeben, in welchem Format die Bilddatei generiert werden muss. (Aus den Daten des kodierten Texts wird eine Bilddatei generiert und diese Datei wird dann im Ausgabedokument verwendet.) In der Auswahlliste "Kodierung" wählen Sie die Kodierung, die in der XML-Quelle verwendet wurde. Dadurch kann StyleVision den kodierten Text korrekt auslesen (indem das Kodierungsformat verwendet wird, das Sie angeben).

Das Dialogfeld "Einstellungen für generierte Bilddatei" (aufgerufen durch Klicken auf die Schaltfläche **Bilddateneinstellungen**) ermöglicht Ihnen, der Bilddatei, die erstellt wird, einen Namen zu geben. Sie können auch keinen Namen angeben, in diesem Fall wird standardmäßig ein Name von StyleVision generiert.

Um ein Inline-Bild in die HTML-Ausgabe einzubetten, aktivieren Sie im Dialogfeld Eigenschaften[489] des SPS die Option *Bildeinbettung mit Daten-Uri-Schema* aus.

Aufrufen des Bildes für die Ausgabe

Das Bild wird bei der Generierung der verschiedenen Ausgabedokumente auf verschiedene Arten und zu verschiedenen Zeitpunkten aufgerufen. Beachten Sie die folgenden Punkte:

- Beachten Sie die Ausgabeformate für Ihre einzelnen Editionen: (i) HTML in der Basic Edition; (ii) HTML und RTF in der Professional Edition; (iii) HTML, RTF, PDF und Word 2007+ in der Enterprise Edition).
- In der Design-Ansicht können Sie im Dialogfeld Eigenschaften 489 festlegen, ob relative Pfade zu Bildern relativ zum SPS oder zur XML-Datei sein sollen.
- Bei der HTML-Ausgabe wird die URI des Bildes an die HTML-Datei übergeben, und das Bild wird vom Browser aufgerufen. Wenn der Pfad zum Bild relativ ist, muss er relativ zum Ordner der HTML-Datei sein. Für die HTML-Vorschau in StyleVision wird im selben Ordner wie die SPS-Datei eine temporäre HTML-Datei erstellt. Daher müssen zur Darstellung der HTML-Vorschau relative Pfade relativ zu diesem Ordner sein.
- Egal, ob die URI relativ oder absolut ist, der Prozess benötigt physischen Zugriff auf das Bild, um es wiedergeben zu können.

Bearbeiten von Bildeigenschaften

Um ein Bild zu bearbeiten, klicken Sie mit der rechten Maustaste in der Design-Ansicht auf den Platzhalter für das Bild und wählen Sie im Kontextmenü den Befehl "**URL bearbeiten**" aus. Daraufhin erscheint das Dialogfeld "Bild bearbeiten", welches mit dem Dialogfeld "Bild einfügen" (*Abbildung oben*) identisch ist und in dem Sie die gewünschten Änderungen vornehmen können. Sie können das Dialogfeld "Bild bearbeiten" auch über die Eigenschaft `URL` in der Eigenschaftsgruppe *Bild* im Fenster "Eigenschaften" aufrufen. Die Eigenschaftsgruppe *Bild* enthält auch die Eigenschaft `alt`, die zum Definieren eines Alternativtexts für das Bild dient.

Löschen von Bildern

Um ein Bild zu löschen, wählen Sie es aus und drücken Sie die Taste "Löschen".

5.8.2 Bildtypen und Ausgabe

In der Tabelle unten finden Sie Informationen darüber, welche der von StyleVision unterstützten Bildtypen in den verschiedenen Ausgabeformaten von StyleVision unterstützt werden. Beachten Sie, dass die verschiedenen Editions von <%STYLEVISION% auch unterschiedliche Ausgabeformate unterstützen: *Enterprise Edition*, HTML, Authentic, RTF, PDF und Word 2007+; *Professional Edition*, HTML, Authentic, RTF; *Basic Edition*, HTML.

Bildtyp	Authentic	HTML	RTF	PDF	Word 2007+
JPEG	Ja	Ja	Ja	Ja	Ja
GIF	Ja	Ja	Ja	Ja	Ja
PNG	Ja	Ja	Ja	Ja	Ja
BMP	Ja	Ja	Ja	Ja	Ja

© 2019-2025 Altova GmbH

Bildtyp	Authentic	HTML	RTF	PDF	Word 2007+
TIFF	Ja*	Ja*	Ja	Ja	Ja
SVG	Ja*	Ja*	Nein	Ja	Nein
JPEG XR	Ja	Ja	Nein	Nein	Nein

** siehe Anmerkungen unterhalb*

Beachten Sie die folgenden Punkte:

- Bilder werden in der Design-Ansicht nur angezeigt, wenn es sich beim Pfad um eine statische URL (also eine direkt in das SPS eingegebene URL) handelt.
- Für die Anzeige von TIFF- und SVG-Bildern in der Authentic-Ansicht und der HTML-Ansicht wird Internet Explorer 9 oder höher benötigt.
- In der RTF-Ausgabe können TIFF-Bilder nur mittels Link, nicht aber durch Einbettung verwendet werden. Daher kann ihre Größe nicht angepasst werden.
- SVG-Dokumente müssen im XML-Format und im SVG Namespace sein.
- FOP gibt einen Fehler aus, wenn eine Bilddatei nicht gefunden wird und generiert kein PDF-Dokument.
- Bei Verwendung von FOP zum Generieren von PDF-Dokumenten muss zur Wiedergabe von PNG-Bildern die JIMI-Bildbibliothek installiert sein und es muss Zugriff darauf bestehen.
- Nähere Informationen zur Behandlung von Grafiken durch FOP finden Sie auf der FOP Website.

Beispieldatei

Eine Beispieldatei, nämlich `Images.sps`, befindest sich im Ordner:

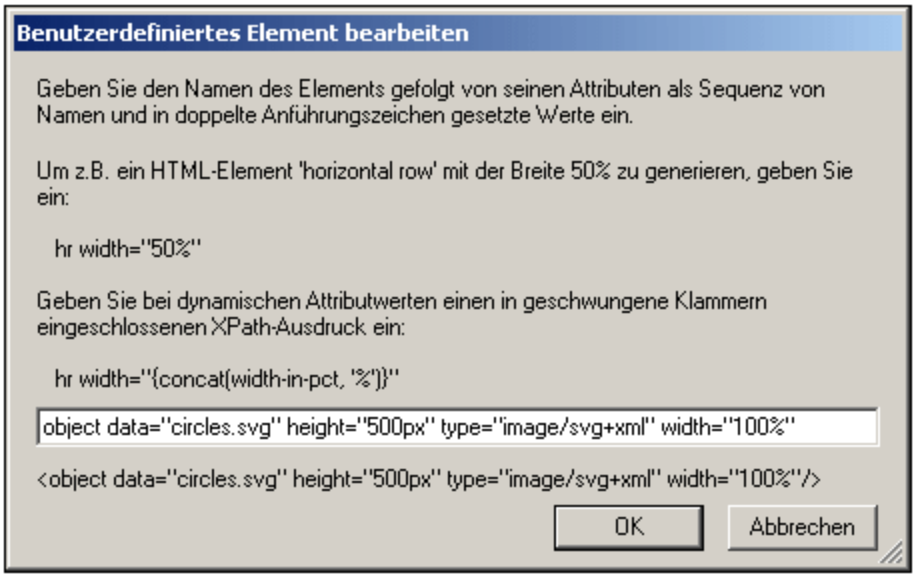
```
C:\Documents and Settings\<username>\My
Documents\Altova\StyleVision2025\StyleVisionExamples/Tutorial/Images
```

SVG-Bilder in HTML

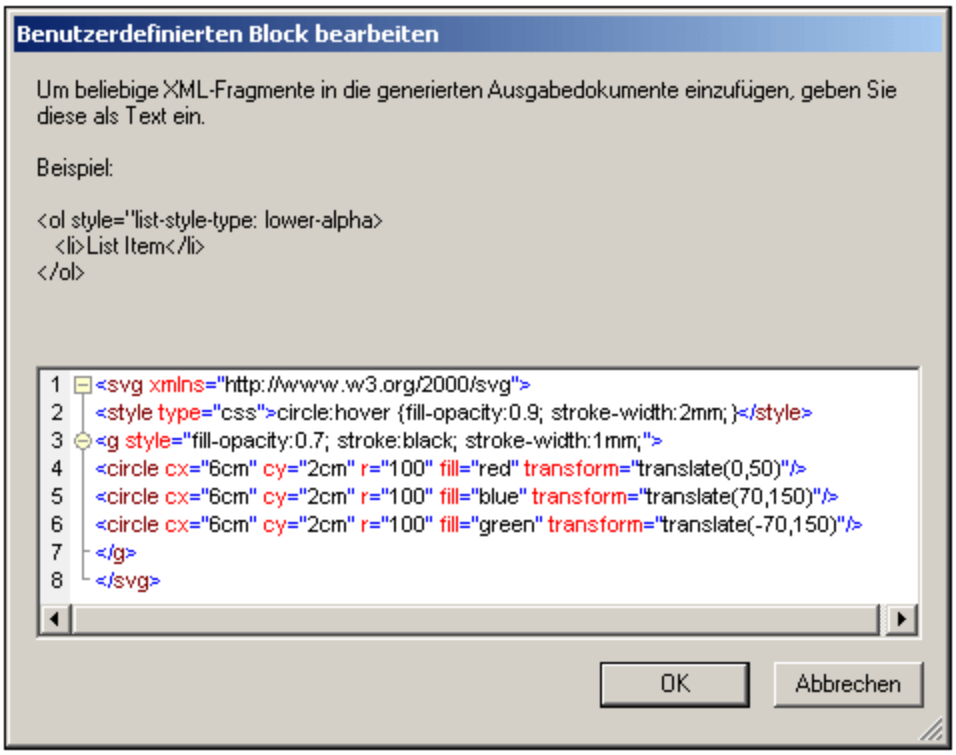
Wenn als Bild eine externe SVG-Datei mit Code für Maus-Events verwendet wird, so wird die SVG-Datei innerhalb des Bilds dargestellt und ist nicht mehr interaktiv. Um diese Einschränkung zu umgehen, können Sie die externe SVG-Bilddatei als Objekt verwenden oder das SVG-Codefragment als benutzerdefinierten XML-Block inkludieren.

Im Folgenden sind drei Methoden beschrieben, um SVG-Bilder in eine Webseite zu inkludieren.

1. Die externe SVG-Datei wird als Bild eingefügt 152: Dadurch wird in der generierten HTML-Datei ein `<img>` generiert. Die Interaktivität geht verloren.
2. Die externe SVG-Datei wird mittels der Funktion benutzerdefiniertes Element 123 als Objekt eingefügt (*siehe Abbildung unten*). Stellen Sie sicher, dass das `type` Attribut korrekt eingefügt wird: ke `type="image/svg+xml"`. Wenn die Datei auf diese Art eingefügt wird, bleibt das SVG-Objekt interaktiv und die Mouse-Hover-Funktion funktioniert.



3. Eine Inline-SVG-Datei wird über einen benutzerdefinierten XML-Block[124] eingefügt. Ein Beispiel für ein SVG-Codefragment sehen Sie in der Abbildung unten. In diesem Fall funktioniert die Interaktivität. Beachten Sie, dass das `svg` Element sich nicht im SVG-Namespace befinden muss, wenn die Ausgabemethode HTML 4.0 oder 5.0 ist, dass der Namespace aber bei Verwendung von XHTML als Ausgabemethode erforderlich ist.



 © 2019-2025 Altova GmbH

5.8.3 Beispiel: Eine Vorlage für Grafiken

Das StyleVision Paket beinhaltet eine SPS-Datei, die ein Beispiel für die Verwendung von Grafiken in StyleVision liefert. Diese Datei befindet sich im Verzeichnis (Eigene) Dateien[25]: `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2025\StyleVisionExamples\Tutorial\Images\Images.sps`. Das Bild-Dokument (`Images.xml` und `Images.sps`) besteht aus drei Teilen:

- Der zweite Teil enthält eine Tabelle mit den in den verschiedenen StyleVision Ausgabeformaten unterstützten Bildformaten. Beachten Sie, dass die Ausgabeformate RTF und PDF nur in der Enterprise Edition von StyleVision zur Verfügung stehen. In der Design-Ansicht werden nur Bilder mit statischen URIs angezeigt. Alle in der Tabelle aufgelisteten Bildformate werden in Teil 3 des Bilddokuments aufgelistet.
- In Teil 3 werden alle gängigen von StyleVision unterstützten Bildformate untereinander aufgelistet. Sie können nach dem Öffnen der Datei `Images.sps` in StyleVision zwischen den verschiedenen Vorschauen in StyleVision wechseln, um zu sehen, wie die einzelnen Bilder in der jeweiligen Vorschau angezeigt werden. Da es sich beim Pfad zum Bild um einen XML Node handelt, können Sie in der Authentic-Ansicht auch den Pfad zu Ihren eigenen Grafiken eingeben und sich die Grafiken anschließend in den Vorschaufenstern ansehen.

5.9 Formularsteuerelemente

Nodes im XML-Dokument können als Dateneingabeelemente erstellt werden (z.B. als Eingabefelder und Auswahllisten). In der HTML-Ausgabe wird ein Dateneingabeelement so dargestellt, wie es in Design-Ansichtangezeigt wird, oder es wird auf eine möglichst ähnliche Art wiedergegeben.Beachten Sie, dass Dateneingabeelemente in der HTML-Ausgabe nicht funktionieren.

Allgemeine Funktionsweise

Im Folgenden finden Sie eine Liste von in StyleVision verfügbaren Dateneingabeelementen sowie eine Erklärung, (i) wie die Daten bei jedem dieser Elemente in das XML-Dokument eingegeben werden und (ii) wie die Dateneingabeelemente im reinen Textformat ausgegeben werden.

Dateneingabeelement	Daten in der XML-Datei	Ausgabe ins reine Textformat
Eingabefeld (Textfeld)	Text, der vom Benutzer eingegeben wird	Text, der vom Benutzer eingegeben wird
Mehrzeiliges Eingabefeld	Text, der vom Benutzer eingegeben wird	Text, der vom Benutzer eingegeben wird
Auswahlliste	Der Auswahl des Benutzers wird ein Wert zugeordnet.	Ausgewählter Wert
Kontrollkästchen	Der Auswahl des Benutzers wird ein Wert zugeordnet.	`[]` (nicht ausgewählt); `[x]` (ausgewählt)
Optionsfeld	Der Auswahl des Benutzers wird ein Wert zugeordnet.	`()` (nicht ausgewählt); `[o]` (ausgewählt)
Schaltfläche	Der Auswahl des Benutzers wird ein Wert zugeordnet.	Schaltflächentext

Die in die Eingabefelder eingegebenen Textwerte werden als XML-Inhalt direkt in das XML-Dokument eingegeben. Bei den anderen Dateneingabeelementen wird die Auswahl des Benutzers der Authentic-Ansicht auf einen Wert gemappt. In StyleVision können Sie die Liste von Optionen definieren, die der Benutzer sieht und den XML-Wert, auf den die einzelnen Optionen gemappt sind. In der Regel werden die Optionen und deren entsprechende Werte in einem Dialogfeld definiert.
Im Folgenden sehen Sie eine Liste der in StyleVision verfügbaren Dateneingabeelemente:

Dateneingabeelement	Ausgabe ins reine Textformat
Eingabefeld (Textfeld)	Text, der vom Benutzer eingegeben wird
Mehrzeiliges Eingabefeld	Text, der vom Benutzer eingegeben wird
Auswahlliste	Ausgewählter Wert
Kontrollkästchen	`[]` (nicht ausgewählt); `[x]` (ausgewählt)

© 2019-2025 Altova GmbH

Optionsfeld	`()` (nicht ausgewählt); `[o]` (ausgewählt)
Schaltfläche	Schaltflächentext

Allgemeine Verwendung

Um ein Dateneingabelement zu erstellen, gehen Sie folgendermaßen vor:

1. Ziehen Sie einen Node aus dem Fenster "Schema-Struktur" an die gewünschte Stelle in der Design-Ansicht.
2. Wählen Sie im Kontextmenü das Dateneingabeelement aus, als das der Node erstellt werden soll.
3. Bei einigen Dateneingabeelementen wird ein Dialogfeld angezeigt. Geben Sie in diesem Fall die gewünschten Informationen in das Dialogfeld ein und klicken Sie auf OK.

Um die Eigenschaften eines Dateneingabeelements **erneut zu öffnen und zu bearbeiten,** wählen Sie das gewünschte Element aus (nicht den Node, der es enthält) und bearbeiten Sie dessen Eigenschaften in der Seitenleiste "Eigenschaften".

Anmerkung:

- Datenkönnen in der HTML-Ausgabe nicht in Dateneingabeelemente eingegeben werden. In der HTML-Ausgabe dienen Dateneingabeelemente rein als Alternativmethode zur Darstellung von Inhalt.
- Dateneingabeelemente können auch erstellt werden, indem Sie den aktuellen Komponententyp eines Node in ein Dateneingabeelement ändern. Klicken Sie dazu mit der rechten Maustaste auf den Node und wählen Sie **Ändern in**.
- In der HTML-Ausgabe wird der vom Benutzer ausgewählte Eintrag in der Ausgabe angezeigt. Wenn Sie den Wert eines Dateneingabeelements im HTML-Dokument ändern, wird der Textwert weder im XML-Dokument noch im HTML-Dokument geändert.

5.9.1 Eingabefelder, mehrzeilige Eingabefelder

Wenn Sie einen Node aus dem Fenster "Schemaquellen" in die Design-Ansicht ziehen, können Sie ein Eingabefeld oder ein mehrzeiliges Eingabefeld in Ihr SPS einfügen. Der Inhalt dieses Node wird im Eingabefeld oder im mehrzeiligen Eingabefeld angezeigt.

Bearbeiten der Eigenschaften von Eingabefeldern

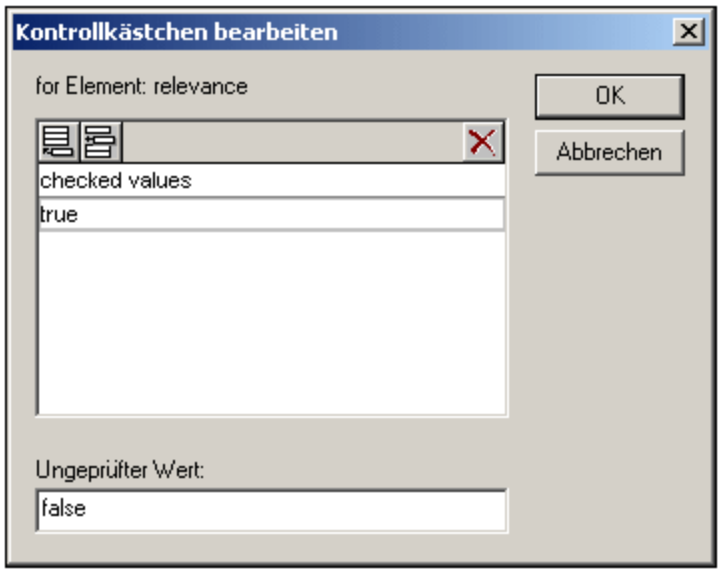
Sie können die HTML-Eigenschaften von Eingabefelder ändern, indem Sie das Eingabefeld auswählen und seine HTML-Eigenschaften in der Seitenleiste "Eigenschaften) bearbeiten.

So können Sie z.B. wenn das Eingabefeld ausgewählt ist, im Fenster "Eigenschaften" den Eintrag `editfield` auswählen und in der Eigenschaftsgruppe `HTML` die Eigenschaft `maxlength` definieren, indem Sie auf ihr Wertefeld doppelklicken und einen Wert eingeben.

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-Ansicht als Komponenten des Textinhalts angezeigt werden

5.9.2 Kontrollkästchen

Sie können ein Kontrollkästchen als Dateneingabeelement erstellen. In der Basic Edition können Sie die Einstellungen im Dialogfeld "Kontrollkästchen bearbeiten" auf der Standardeinstellung belassen (da die Basic Edition die Authentic-Ansicht nicht unterstützt, daher kann auch kein Wert in die XML-Datei eingebeben werden.)



In der Abbildung oben wurde ein Element mit dem Namen `Name` als Kontrollkästchen erstellt. Wenn der Benutzer der Authentic-Ansicht das Kontrollkästchen aktiviert, wird der Wert `true` als Wert des Elements `Name` eingegeben. Wenn das Kästchen deaktiviert ist, wird der Wert `false` als XML-Wert von `Name` eingegeben (wie im Dialogfeld definiert).

Anmerkung: Kontrollkästchen werden in der Textausgabe in Form von eckigen Klammern angezeigt: `[]` für nicht ausgewählte Kontrollkästchen, `[x]` für ausgewählte Kontrollkästchen.

Aufrufen des Dialogfelds "Kontrollkästchen bearbeiten"

Beim Erstellen eines neuen Kontrollkästchens wird das Dialogfeld "Kontrollkästchen bearbeiten" angezeigt, wenn Sie den Node als Kontrollkästchen erstellen. Um dieses Dialogfeld später wieder aufzurufen, gehen Sie folgendermaßen vor:

1. Wählen Sie das Kontrollkästchen im Design aus.
2. Wählen Sie in der Seitenleiste "Eigenschaften" den Eintrag "Kontrollkästchen" und anschließend die Eigenschaftsgruppe `Kontrollkästchen` aus (*siehe Abbildung unten*).

1.

© 2019-2025 Altova GmbH

3. Klicken Sie auf die Bearbeiten-Schaltfläche ... der Eigenschaft `geprüfte Werte`. Daraufhin wird das Dialogfeld "Kontrollkästchen bearbeiten" angezeigt.

Anmerkung: Sie können die HTML-Eigenschaften eines Kontrollkästchens ändern, indem Sie es auswählen und die HTML-Eigenschaften anschließend in der Seitenleiste"Eigenschaften" ändern.

5.9.3 Auswahllisten

In einer Auswahlliste werden Einträge in Form einer Dropdown-Liste zur Verfügung gestellt. Sie können die Einträge in der Liste, wie unten beschrieben, auf drei verschiedene Arten auswählen. Diese Liste kann im generierten HTML-Dokument je nach Erfordernis verwendet werden; so kann der generierte HTML-Code z.B. später weiterverarbeitet werden, sodass die Auswahlliste Einträge für ein HTML-Formular zur Verfügung stellt.

- über die Schema-Enumerationen für den ausgewählten Node.
- über eine Liste, die im Dialogfeld "Auswahlliste bearbeiten" definiert wird. Sie geben den sichtbaren Eintrag und den dazugehörigen XML-Wert ein, welcher ein anderer sein kann.Der XML-Wert gilt für die Enterprise und die Professional Edition und bezieht sich auf den XML-Wert, dem der in der Authentic-Ansicht ausgewählte Wert zugeordnet wird. Benutzer der Basic Edition können diese Spalte leer lassen (da die Authentic-Ansicht in der Basic Edition nicht unterstützt wird.
- über die Ergebnissequenz eines XPath-Ausdrucks zum aktuellen Node. Die Einträge in der Ergebnissequenz werden als die Einträge der Dropdown-Liste angezeigt. Diese Methode eignet sich v.a. zur Verwendung dynamischer Einträge in der Auswahlliste. Es ist wichtig, welchen Node Sie als Auswahlliste erstellen. Angenommen Sie haben z.B. ein Element `NameList`, das eine unbegrenzte Anzahl an `Name` Elementen enthalten kann, die wiederum die Child-Elemente `First` und `Last` enthalten. Wenn Sie z.B. das Element `Name` als Auswahlliste erstellen und in der Auswahlliste das Child-Element `Last` auswählen, so erhalten Sie so viele Auswahllisten wie es `Name` Elemente gibt und jede Auswahlliste hat das Child-Element `Last` als Eintrag des Dropdown-Menüs. Um eine einzige Auswahlliste zu erstellen, die in der Dropdown-Liste alle `Last` Elemente enthält, müssen Sie das einzige `NameList` Element als Auswahlliste auswählen und im XPath-Ausdruck das Element `Last` auswählen..

Aufrufen des Dialogfelds "Auswahlliste bearbeiten"

Beim Erstellen einer neuen Auswahlliste wird das Dialogfeld "Auswahlliste bearbeiten" angezeigt, wenn Sie den Node als Auswahlliste erstellen. Sie können eine Auswahlliste auch mit dem Menübefehl **Einfügen |**

Formularsteuerelement einfügen | Auswahlliste einfügen. Um das Dialogfeld "Auswahlliste bearbeiten" später wieder aufzurufen, gehen Sie folgendermaßen vor:

1. Wählen Sie im Design die Auswahlliste aus.
2. Wählen Sie in der Seitenleiste "Eigenschaften" den Eintrag "Auswahlliste" und anschließend die Eigenschaftsgruppe `Auswahlliste` aus (*siehe Abbildung unten*).

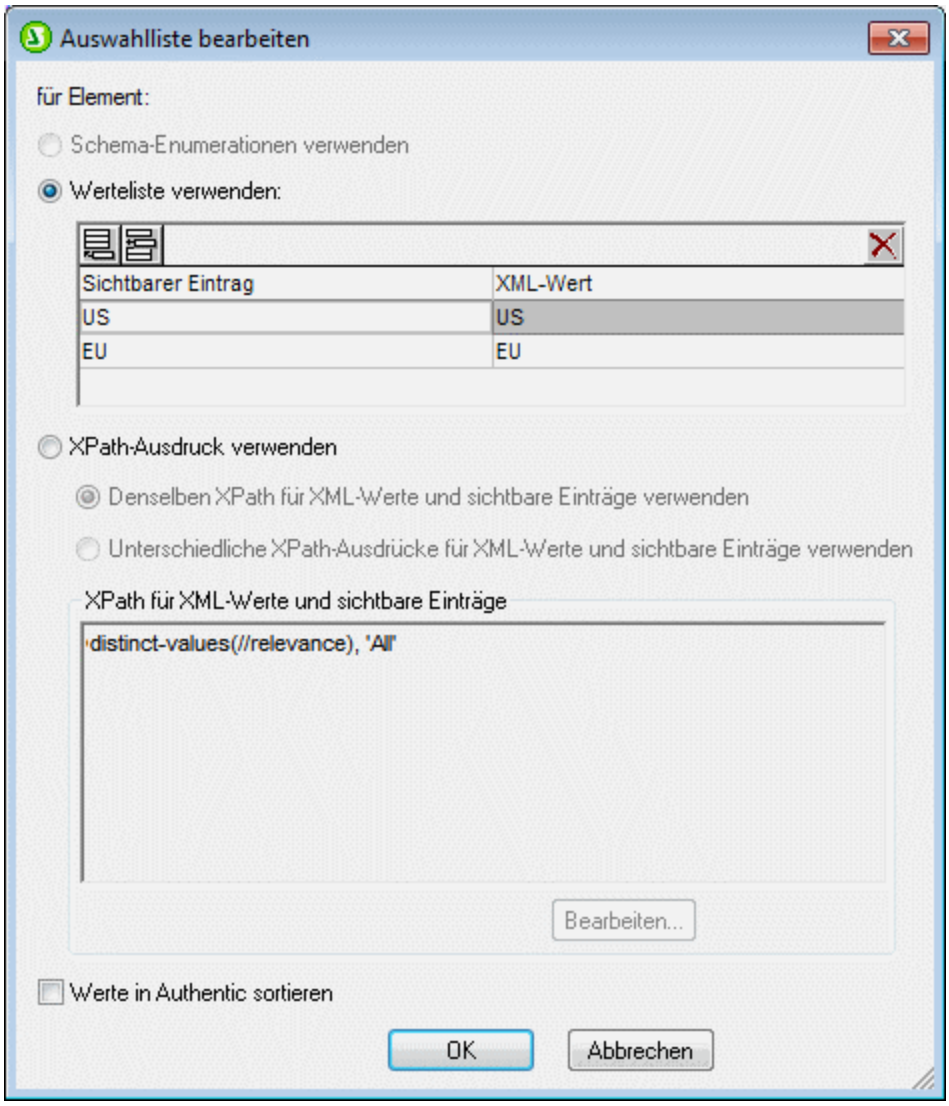
1.

3. Klicken Sie auf die Bearbeiten-Schaltfläche ... der Eigenschaft `Ursprung des Inhalts`. Daraufhin wird das Dialogfeld "Auswahlliste bearbeiten" angezeigt.

Verwenden des Dialogfelds "Auswahlliste bearbeiten"

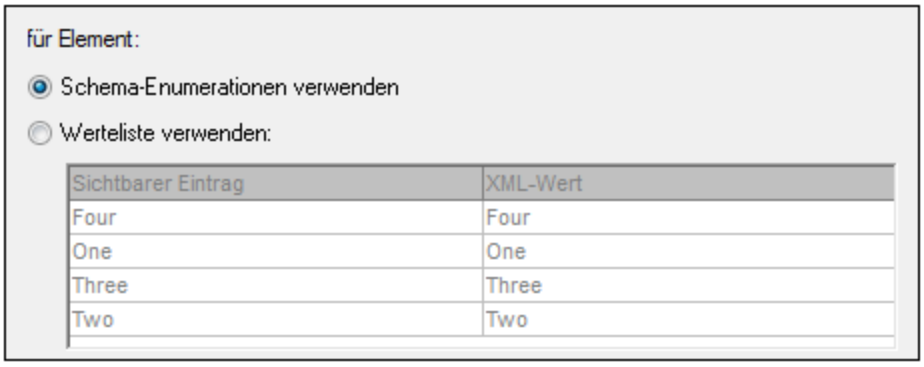
Unten sehen Sie das Dialogfeld "Auswahlliste bearbeiten".

© 2019-2025 Altova GmbH



Um die Einträge und Werte für die Auswahlliste zu definieren, gehen Sie folgendermaßen vor:

1. Wählen Sie die Methode, mit der Sie die Einträge und Werte definieren möchten, durch Klicken auf das entsprechende Optionsfeld aus: (i) Schema-Enumerationen, (ii) Werteliste oder (iii) XPath-Ausdrücke zur Auswahl der Werte.
2. Wenn Sie die Option "*Schema-Enumerationen verwenden*" auswählen, werden die diesem Node zugewiesenen Enumerationen automatisch (i) als die sichtbaren Einträge der Dropdown-Liste der Auswahlliste und (ii) die entsprechenden XML-Werte dazu (*Abbildung unten*) eingegeben. Die sichtbaren Einträge sind die Einträge in der Dropdown-Liste der Auswahlliste. Jeder Eintrag in der Dropdown-Liste hat als Entsprechung einen XML-Wert. Der XML-Wert, der dem sichtbaren vom Authentic-Benutzer ausgewählten Eintrag entspricht, ist der XML-Wert, der in die XML-Datei eingegeben wird. Sowohl sichtbare Einträge als auch XML-Werte sind in der Werteliste ausgegraut, da sie beide von den Schema-Enumerationen abgeleitet werden und nicht bearbeitet werden können.



Bei Auswahl der Option "*Werteliste verwenden*" können Sie beliebig viele Einträge für die Dropdown-Liste der Auswahlliste sowie für die zugehörigen XML-Werte einfügen, anhängen, bearbeiten und löschen. Diese Bearbeitungen werden im Feld unterhalb der Optionsschaltfläche *Werteliste verwenden* durchgeführt. Sie können die sichtbaren Einträge und deren XML-Werte auch mit Hilfe eines XPath-Ausdrucks ermitteln. Die Einträge in der Sequenz, die vom XPath-Ausdruck zurückgegeben werden, werden als sichtbare Einträge und XML-Werte verwendet. Sie können festlegen, (i) dass derselbe XPath-Ausdruck für die sichtbaren Einträge und die XML-Werte verwendet werden soll oder (ii) dass unterschiedliche XPath-Ausdrücke verwendet werden sollen. In zweiterem Fall wird die Entsprechung zwischen dem sichtbaren Eintrag und dem XML-Wert durch ein 1:1-Index-Mapping festgelegt. Wenn die Anzahl der Einträge in den beiden Sequenzen nicht gleich ist, wird ein Fehler ausgegeben.

3. Wenn die Einträge in der Authentic-Ansicht in der Dropdown-Liste der Auswahlliste sortiert angezeigt werden sollen, aktivieren Sie das Kontrollkästchen *Werte in Authentic sortieren*.
4. Klicken Sie zum Fertigstellen auf **OK**.

Anmerkung

- Bei Verwendung von XPath zur Auswahl der Einträge der Dropdown-Auswahlliste können Sie Auswahllisten mit dynamischen Einträgen aus der XML-Datei selbst erstellen.
- Wenn die Einträge in der Dropdown-Liste der Auswahlliste aus Schema-Enumerationen stammen, werden sie standardmäßig alphabetisch sortiert. Wenn die Einträge aus einer XML-Datendatei stammen, erscheinen sie standardmäßig in der Reihenfolge, in der sie im Dokument vorkommen.
- Auswahllisten werden in der Textausgabe als ausgewählter Wert angezeigt.

5.9.4 Optionsfelder, Schaltflächen

Es gibt zwei Arten von Schaltflächen: Optionsfelder und Schaltflächen. Optionsfelder eigenen sich zur Eingabe in Formulare oder zum Auslösen von Ereignissen in der HTML-Ausgabe. Zweiteres erfolgt durch Verknüpfung von Skripts mit dem Butten Event.

Anmerkung: Sie können die HTML-Eigenschaften eines Optionsfelds oder einer Schaltfläche ändern, indem Sie sie auswählen und die HTML-Eigenschaften anschließend in der Seitenleiste "Eigenschaften" ändern.

Anmerkung: Optionsfelder werden in der Textausgabe als Klammern angezeigt: `()` für nicht ausgewählte Optionsfelder; `(o)` für ausgewählte Optionsfelder. Bei Schaltflächen wird in der Textausgabe nur der Schaltflächentext generiert.

 © 2019-2025 Altova GmbH

5.10 Links

Links (oder Hyperlinks) können sowohl zu Lesezeichen im Dokument als auch zu externen Ressourcen wie Webseiten angelegt werden. Links können auch zu dynamisch erstellten Markierungen angelegt werden. StyleVision bietet erhebliche Flexibilität bei der Art, wie Ziel-URIs für Hyperlinks erstellt werden können.

Im Abschnitt Lesezeichen und Hyperlinks 320 wird beschrieben, wie man im Dokument dynamische und statische Hyperlinks anlegt und wie man sowohl zu Lesenzeichen als auch zu externen Dokumenten Links erstellt.

Anmerkung: Links werden in der Textausgabe nicht dargestellt.

5.11 Barcodes

Das Barcode-Design-Element wird im **XSLT 2.0- oder 3.0-Modus**, nicht aber im XSLT 1.0-Modus unterstützt. Mit Hilfe des Design-Elements "Barcode" können im Ausgabedokument Barcodes (*Abbildung unten*) generiert werden. Fügen Sie das Barcode-Design-Element 166 an der gewünschten Stelle im Design-Dokument ein und definieren Sie seine Eigenschaften 168.

Wichtig: Damit Barcodes funktionieren, muss eine Java Runtime-Umgebung installiert sein. Diese muss zumindest Version 1.4 sein und in einer Bit-Version vorhanden sein, die der Bit-Version des auf Ihrem System installierten StyleVision-Pakets entspricht: 34-Bit oder 64-Bit.

Wichtig: Damit in der Ausgabe Barcodes generiert werden, **muss für die Generierung der Altova XSLT-Prozessor verwendet werden,** da die Barcodes in einem SPS durch Aufruf spezieller Java-Erweiterungsfunktionen, die nicht Teil des XSLT-Standards bilden, generiert werden. Die XSLT-Prozessoren von Altova unterstützen diese spezifischen Erweiterungsfunktionen, während andere XSLT-Prozessoren dies höchstwahrscheinlich nicht tun. Infolgedessen werden keine Barcodes generiert, wenn das SPS mit einem nicht von Altova stammenden XSLT-Prozessor verarbeitet wird. Der Altova XSLT-Prozessor ist in StyleVision integriert und wird automatisch aufgerufen, wenn Sie die Ausgabe mit den **Generierungsbefehlen** aus dem Menü **Datei** generieren. Alternativ dazu können Sie dazu auch RaptorXML Server, einen eigenständigen XSLT-Prozessor von Altova, verwenden.

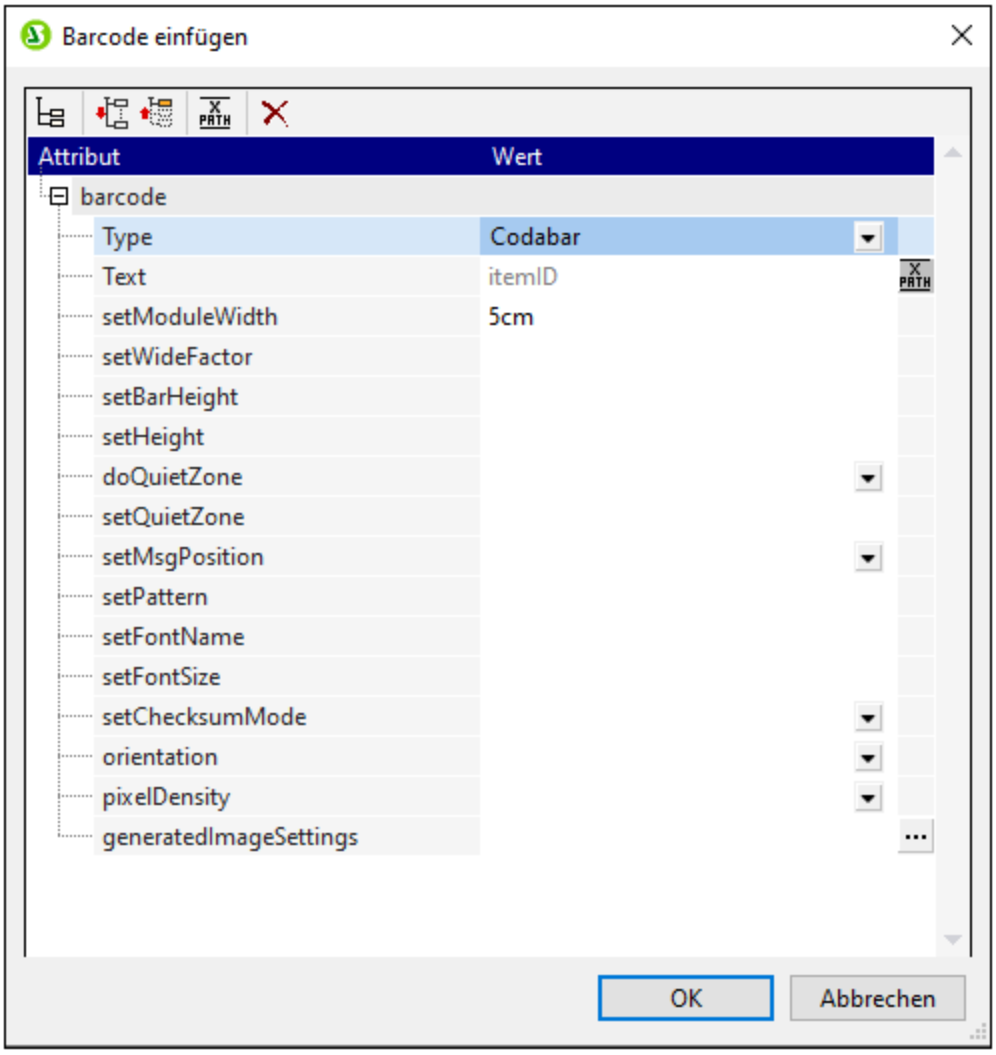
Anmerkung: Barcodes werden in der Textausgabe nicht dargestellt.

Einfügen eines Barcode

Um einen Barcode in Ihr Design einzufügen, gehen Sie folgendermaßen vor:

1. Klicken Sie an der gewünschten Stelle im Design-Dokument auf die rechte Maustaste und wählen Sie den Befehl **Barcode einfügen.** Alternativ dazu können Sie auch den Befehl **Einfügen | Barcode einfügen** wählen oder auf das Barcode-Symbol aus der Symbolleiste und anschließend auf die gewünschte Stelle im Design klicken. Sie können auch ein Element aus der Schema-Struktur in die Design-Ansicht ziehen und dann 'Barcode erstellen' auswählen. Daraufhin wird das Barcode-Dialogfeld "Barcode einfügen" angezeigt (*Abbildung unten*).

© 2019-2025 Altova GmbH



2. Zwei Eigenschaften, nämlich *Type* und Text sind verpflichtend; die anderen Eigenschaften sind optional und/oder haben passende Standardwerte. Mit der Eigenschaft *Type*, deren Wert aus einer Dropdown-Liste ausgewählt werden kann (*siehe Abbildung oben*), wird die Art des Barcode definiert, z.B. EAN-13 (ein Typ, zu dem auch ISBN-Barcodes gehören) und UPC-A. Mit der Eigenschaft *Text* wird der Wert definiert, der den Barcode generiert, z.B. eine ISBN-Nummer. Die verschiedenen Barcode-Eigenschaften sind weiter unten beschrieben 168. Definieren Sie die benötigten Eigenschaften und alle anderen gewünschten Eigenschaften. Beachten Sie: Wenn Sie einen Wert aus der XML-Datei als Wert einer Eigenschaft verwenden möchten, können Sie den gewünschten XML-Node mit Hilfe eines XPath-Ausdrucks 52 aufrufen. Gehen Sie dazu folgendermaßen vor: Wählen Sie die Eigenschaft aus, aktivieren Sie in der Symbolleiste des Dialogfelds "Eigenschaften" die Schaltfläche **XPath** und geben Sie anschließend den XPath-Ausdruck in das Dailogfeld "XPath-Ausdruck bearbeiten 440" ein. Der XPath-Ausdruck wird innerhalb des aktuellen Kontext-Node ausgewertet.
3. Nachdem Sie die Eigenschaften definiert haben, klicken Sie auf **OK**. Daraufhin wird das Barcodebild eingefügt. Der generierte Barcode (*siehe Abbildung unten*) kann sofort in jeder der Ausgabevorschauen angezeigt werden.

Anmerkung: Barcode-Bilder werden als PNG-Dateien generiert.

Barcode-Eigenschaften

Die folgenden Barcode-Eigenschaften können definiert werden. Die Eigenschaften *Type* und *Text* müssen definiert werden; die anderen Eigenschaften sind optional. Beachten Sie, dass für unterschiedliche Barcodetypen unterschiedliche Eigenschaften zur Verfügung stehen.

- *Type:* Das Barcode-System, mit dem die Message interpretiert wird, z.B. EAN und UPC.
- *Text:* Der Wert, der zur Generierung des Barcode-Musters verwendet wird.
- *SetModuleWidth:* Die Breite der Striche im Code,
- *SetBarHeight:* Die Höhe der Striche.
- *SetHeight:* Die Höhe der Barcode-Grafik.
- *DoQuietZone:* Die Werte `Yes` oder `No` bestimmen, ob die "Quiet Zone" (oder der Abstand) rund um den Barcode, die in den Eigenschaften *SetQuietZone* und *SetVerticalQuietZone* angegeben ist, angewendet wird.
- *SetQuietZone:* Definiert die "Quiet Zone" (oder den Abstand) rund um den Barcode. Im Fall von eindimensionalen Barcodes wird der hier angegebene Wert auf die horizontale Dimension angewendet. Bei zweidimensionalen Barcodes wird der Wert sowohl auf die horizontale als auch auf die vertikale Dimension angewendet. Der Wert der vertikalen Dimension kann mit dem Wert, der in der Eigenschaft *SetVerticalQuietZone* angegeben wird, überschrieben werden. Eine Längeneinheit in Millimetern (`mm`) ist erforderlich. Beispiel: `2mm`.
- *SetVerticalQuietZone:* Definiert die "Quiet Zone" (oder den Abstand) für die vertikale Dimension bei zweidimensionalen Barcodes. Eine Längeneinheit in Millimetern (`mm`) ist erforderlich. Beispiel: `2mm`.
- *SetMsgPosition:* Legt fest, wo die Nachricht in Bezug auf den Barcode erscheint. Werte sind `top`, `bottom` und `none` (keine Nachricht wird generiert).
- *SetPattern:* Legt ein Muster für die Nachricht fest, so dass der Text lesbar ist. Eine lange Zeichenfolge von, zum Beispiel, Zahlen kann schwierig zu lesen sein. Die Syntax für Muster ist unten angeführt.
- *SetFontName:* Die Schriftart, in der der Text angezeigt werden soll.
- *SetChecksumMode:* Folgende Werte sind verfügbar: (i) *Add*: die Checksumme wird automatisch zur Nachricht hinzugefügt; (ii) *Check*: die Checksumme wird geprüft, während der Barcode erzeugt wird (Vorhandensein der Checksumme wird vorausgesetzt); (iii) *Ignore*: Keine Checksummen-Verarbeitung wird durchgeführt; (iv) *Auto*: aktiviert das Standardverhalten entsprechend dem Barcode-Typ.
- *Orientation:* Ob der Barcode horizontal oder vertikal dargestellt werden soll. Die Optionen sind 90°-Schritte gegen den Uhrzeigersinn.
- *PixelDensity:* Gibt die Pixeldichte im Barcode-Bild an. Höhere Pixeldichte führt zu schärferen Bildern.
- *GeneratedImageSettings:* Ermöglicht Ihnen, einen Namen für die generierte Barcode-Bilddatei anzugeben. Wenn kein Name angegeben wird, wird von StyleVision automatisch ein Name generiert.

Muster-Syntax

Muster werden verwendet, um die Eingabe-Zeichenfolge im Barcode lesbarer zu machen. In den Mustern wird jedes Zeichen im Text der Eingabenachricht durch einen Unterstrich "_" dargestellt. Alle anderen Zeichen, die im Muster enthalten sind, werden an den zugehörigen Orten in den Text der Ausgabenachricht eingefügt. Der Gegenschrägstrich "\" ist ein Escape-Symbol. Die Kombination '\?' fügt also das Zeichen '?' in den Text der Ausgabenachricht ein, wobei '?' durch ein beliebiges Zeichen ersetzt werden kann. Das Zeichen '#' kann

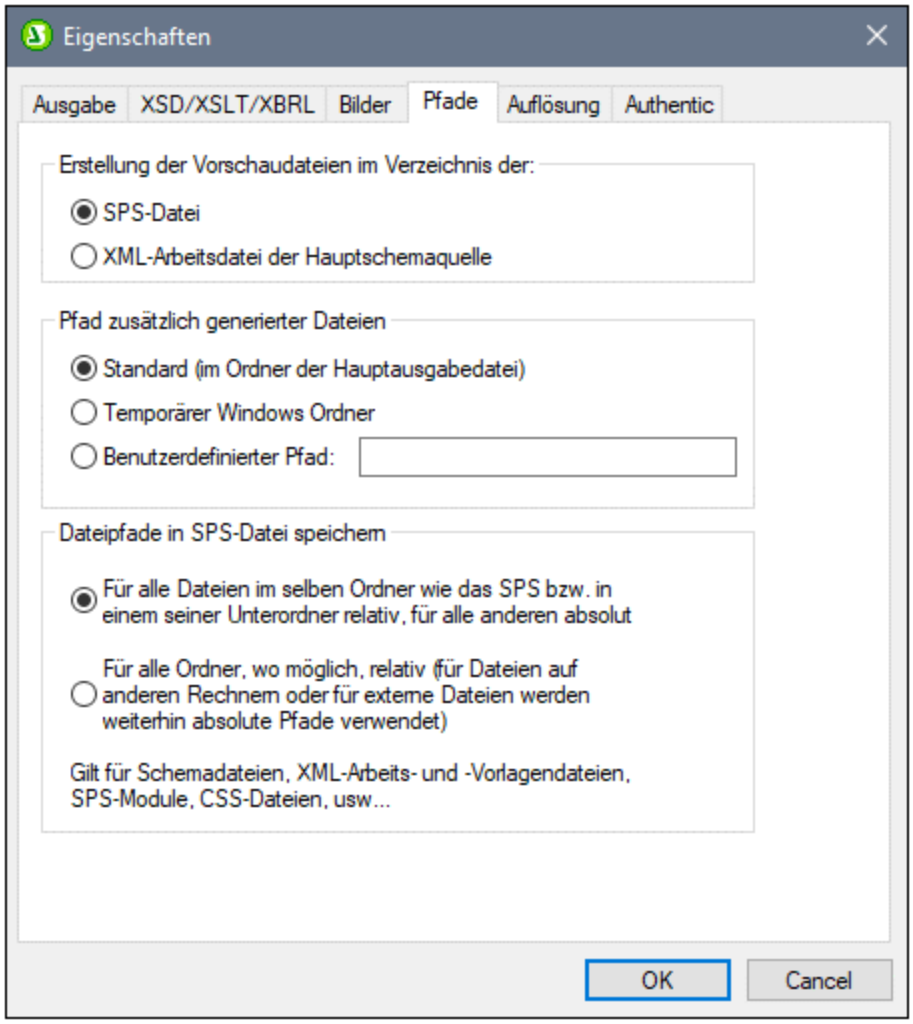
 © 2019-2025 Altova GmbH

verwendet werden, um ein Zeichen aus der Ausgabenachricht zu löschen. Diese Punkte der Muster-Syntax werden mit den unten stehenden Beispielen veranschaulicht.

Text der Eingabenachricht	Muster	Text der Ausgabenachricht
123456	__ __ __	12 34 56
15032011094655	________ __:__:__ UTC	15\03\2011 09:46:55 UTC
15-03-2011	__#/__#/____	15/03/2011

Generieren von Ausgabedateien

Die für die Ausgabe generierten Barcode-Bilddateien werden an Orten gespeichert, die auf dem Register "Pfade" des Dialogfelds "Eigenschaften" (*Abbildung unten*) angegeben werden. Dieses Dialogfeld wird über den Menübefehl **Datei | Eigenschaften** aufgerufen.



Barcode-Bilddateien für Vorschauen können im gleichen Verzeichnis wie die SPS-Datei oder die XML-Arbeitsdatei erstellt werden. Es handelt sich dabei um temporäre Dateien, die gelöscht werden, wenn das SPS

geschlossen wird. Barcode-Bilddateien, die mit Hilfe des Befehle **Datei | Generierte Dateien speichern** erstellt werden, wenn die Ausgabe generiert wird, können an jedem beliebigen Ort erstellt werden. Ihr Zielort wird im Bereich "Pfad zusätzlich generierter Dateien" (*siehe Abbildung oben*) angegeben.

© 2019-2025 Altova GmbH

5.12 Layout-Module

Layout-Module sind Objekte, die ein Layout enthalten. Das Modul als Ganzes wird in das SPS Design eingefügt und kommt innerhalb des Dokuments als Block vor. In einem Layout-Modul können je nach Design-Anforderung mehrere Layout-Felder, von denen jedes SPS Standard-Design-Elemente enthält, eingefügt werden. Mit Hilfe von Layout-Modulen kann daher wie mit einer Grafikdesignapplikation ein Layout erstellt werden.

Gehen Sie beim Erstellen eines Layout-Moduls folgendermaßen vor:

1. Fügen Sie einen Layout Container[171] ein. Der Layout Container kann die gesamte Breite einer Seite einnehmen oder jede beliebige Größe haben. Er kann mit einer Grafikvorlage des Designs hinterlegt werden und (über die Seitenleiste "Stile") mit Hilfe der Stile für den Layout Container formatiert werden.
2. Fügen Sie ein oder mehrere Layout-Felder[174] in den Layout Container ein. Layout-Felder können mehrere Design-Elemente (u.a. statischen Text, Schema-Nodes, automatische Berechnungen, Bilder, Listen, usw.) enthalten und (über die Seitenleiste "Stile") mit Hilfe der Stile für das Layout-Feld formatiert werden. Layout-Felder können innerhalb des Layout Containers auch relativ zueinander verschoben und vor oder hinter einem anderen Layout-Feld positioniert werden.
3. Linien[178] können gezeichnet, formatiert, positioniert und im Stapel der Layout-Objekte (Layout-Felder und andere Linien) nach vorne oder hinten gebracht werden.

Anmerkung: Layout-Module werden in der Textausgabe nicht dargestellt.

Formularbasierte Designs

Beim Erstellen eines neuen SPS[468] haben Sie die Möglichkeit, ein frei fließendes Design oder ein formularbasiertes Design zu erstellen. Ein formularbasiertes Design ist nichts anderes als ein SPS, das aus einem Layout Container besteht.

Anmerkung: Layout-Module werden in der Authentic-Ansicht nur in der Enterprise Edition von Altova-Produkten unterstützt.

5.12.1 Layout Container

Ein Layout Container hat die folgenden Eigenschaften:

- Er kann innerhalb eines Dokumentablaufs, also innerhalb einer Vorlage, eingefügt[172] werden. Oder er kann als der Container eingefügt werden, in dem das Dokument-Design erstellt wird.
- Er kann dieselbe Größe wie die für diesen Abschnitt definierte Seite haben (Layout Container-Eigenschaft "Automatisch an Papiergröße anpassen"), oder er kann eine beliebige andere von Ihnen definierte Größe haben. Nähere Informationen dazu finden Sie im Abschnitt Layout Container-Größe[172] weiter unten.
- Ein Layout-Raster[172] und eine Zoom-Funktion[173] erleichtern das Positionieren von Objekten im Layout Container.
- Er kann Stileigenschaften[173] wie z.B. Ränder, Hintergrundfarben, Schriftarteigenschaften, usw. haben.

- Er kann Layout-Felder und Linien[174] enthalten, aber kein anderes Design-Element. (Alle Design-Elemente müssen in Layout-Felder platziert werden.)
- Er kann eine Grafikvorlage[174] enthalten. Dies ist ein Bild, mit dem der Zeichenbereich hinterlegt wird und das als Grafikvorlage für den Designer dient. Das Design kann anschließend genau entsprechend der Grafikvorlage nachgezeichnet werden.

Anmerkung: Layout Container werden in der Authentic-Ansicht nur in der Enterprise Edition von Altova-Produkten unterstützt.

Einfügen eines Layout Containers

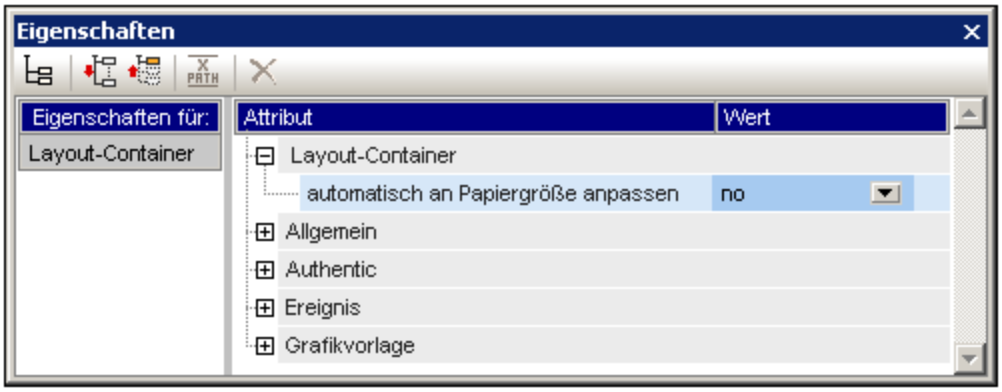
Um einen Layout Container einzufügen, klicken Sie in der Symbolleiste "Design-Elemente einfügen"[462] auf die Schaltfläche **Layout Container einfügen** und klicken Sie im Design auf die gewünschte Stelle**.** Daraufhin erscheint ein Dialogfeld, in dem Sie gefragt werden, ob Sie den Layout Container automatisch an die Papiergröße anpassen möchten. Wenn Sie auf **Ja** klicken, erhält der Layout Container dieselbe Größe wie die in den Seitenlayouteigenschaften für diesen Dokumentabschnitt definierte Seite. Wenn Sie auf **Nein** klicken, wird ein Layout Container mit der Standardgröße 3,5" x 5,0" erstellt.

Beachten Sie, dass ein Layout Container auch zum Zeitpunkt der Erstellung eines SPS erstellt werden kann.

Layout Container-Größe

Es gibt zwei Gruppen von Eigenschaften, die sich auf die Größe des Layout Containers auswirken:

- Die Eigenschaft *Automatisch an Papiergröße anpassen* (Seitenleiste "Eigenschaften", *Abbildung unten*) kann auf *yes* gesetzt werden, sodass ein Layout Container mit denselben Abmessungen wir den der im Dokumentabschnitt definierten Seiten erstellt wird. Bei einem Wert `no` für diese Eigenschaft wird ein Layout Container mit einer benutzerdefinierten Größe erstellt.



- Mit den Eigenschaften *height* und *width* der Gruppe "Details" der Layout Container-Stile (in der Seitenleiste "Stil") wird die Größe des Layout Containers definiert. Die Größe kann auch direkt im Design geändert werden. Ziehen Sie dazu den rechten und den unteren Rand des Layout Containers mit der Maus auf die gewünschte Größe. Beachten Sie, dass die Eigenschaften *height* und *width* nur dann wirksam werden, wenn die Eigenschaft *Automatisch an Papiergröße anpassen* den Wert `no` hat.

Layout Container-Raster

Der Layout Container hat als Hilfe zur gleichmäßigen Platzierung von Elementen im Layout ein Raster. Mit den folgenden Einstellungen können Sie die Verwendung des Rasters definieren:

© 2019-2025 Altova GmbH

- *Raster ein-/ausblenden:* Ein Ein/Aus-Befehl in der Symbolleiste "Design-Elemente einfügen" blendet das Raster ein bzw. aus.
- *Rastergröße:* Auf dem Register "Design" des Dialogfelds "Optionen" können Einheiten für die vertikale und horizontale Länge definiert werden. Beachten Sie: Wenn Sie sehr große Längeneinheiten auswählen, ist das Raster möglicherweise nicht mehr vollständig zu sehen.
- *Am Raster ausrichten:* Ein Ein/Aus-Befehl in der Symbolleiste "Design-Elemente einfügen" aktiviert bzw. deaktiviert diese Funktion. Wenn die Funktion aktiviert ist, werden die linke und die obere Kante von Layout-Feldern und die Endpunkte von Layout-Linien an den Rasterlinien bzw. Punkten ausgerichtet.

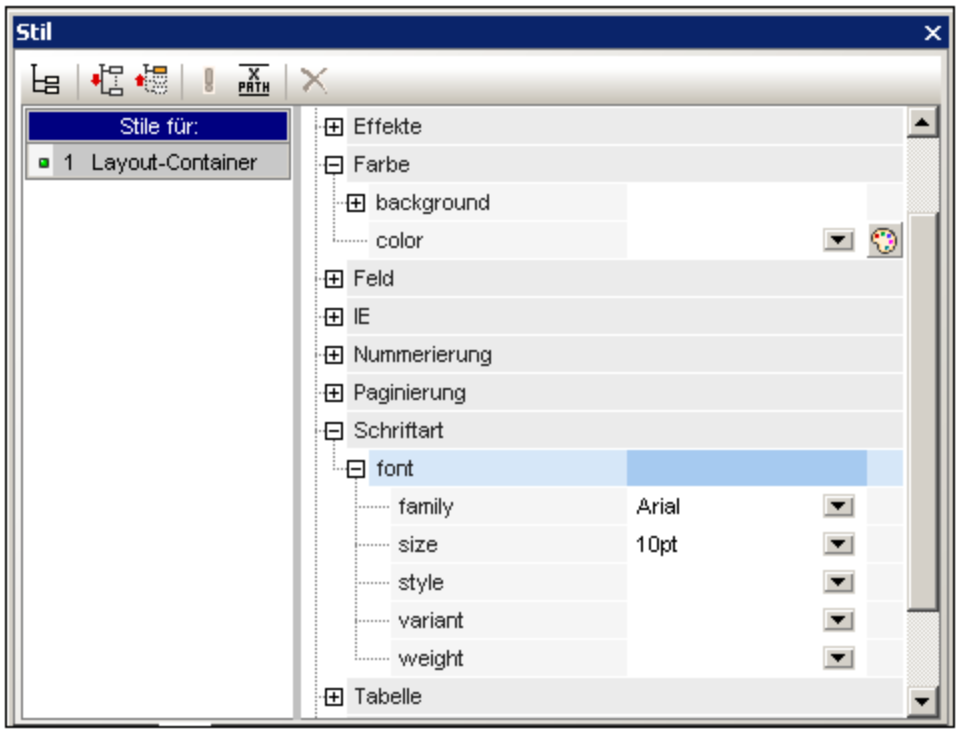
Zoomen

Um Objekte genauer positionieren zu können, können Sie die Ansicht vergrößern. Ändern Sie dazu in der Zoom-Auswahlliste (in der Standard-Symbolleiste) den Zoomfaktor oder drücken Sie die **Strg**-Taste und scrollen Sie dabei mit dem Mausrad.

Layout Container-Stileigenschaften

Es gibt zwei Arten von Stileigenschaften, die auf Layout Container angewendet werden können:

- Solche, die auf den Layout Container alleine angewendet werden und die nicht vererbt werden können, wie z.B. die Eigenschaften *border* und *background-color*.
- Solche, die von den Layout-Feldern im Layout Container weitervererbt werden können, wie z.B. Schriftarteigenschaften.



Die Stileigenschaften eines Layout Containers werden in der Seitenleiste "Stil" in den *Layout Container*-Stilen definiert (*Abbildung oben*).

Layout Container-Inhalt

Die einzigen Design-Elemente, die ein Layout Container enthalten kann, sind Layout-Felder und Linien. Zusätzlich dazu kann der Container als Design-Hilfe mit einer Grafikvorlage (welche kein Design-Element ist) hinterlegt werden. Alle Design-Elemente müssen in ein Layout-Feld platziert werden.

Grafikvorlagen

Als Hilfe beim Erstellen des SPS kann ein Layout Container mit jeweils einer Grafikvorlage hinterlegt werden. Dabei handelt es sich um eine Bilddatei, die so platziert werden kann, dass sie genau die Größe des Layout Containers erhält. Wenn die Grafikvorlage kleiner als der Layout Container ist, kann sie mit einem Versatzabstand an die gewünschte Stelle platziert werden (siehe Grafikvorlage in der *Abbildung "Eigenschaften" unten*). Der Designer kann diese Grafik als Vorlage verwenden, indem er das SPS Design über der Grafikvorlage reproduziert. Auf diese Art können Design-Elemente im Layout exakt wie in der Grafikvorlage platziert werden. Die Grafikvorlage wird nur in der Design-Ansicht angezeigt, **nicht** aber in einer Ausgabeansicht, da sie nur als Vorlage beim Entwurf des SPS verwendet wird.

Die Eigenschaften der Grafikvorlage können (in der Seitenleiste "Eigenschaften", *Abbildung unten)* über die Eigenschaftsgruppe *Grafikvorlage* der Layout Container-Eigenschaften definiert werden.

Sie können die Opazität der Grafikvorlage im Layout Container definieren, damit die Grafikvorlage die Anzeige des Designs nicht beeinträchtigt. Die Grafikvorlage kann gegebenenfalls über die Eigenschaft *Bild anzeigen* auch ausgeblendet werden.

5.12.2 Layout-Felder

Jedes Design-Element in einem Layout (wie z.B. statischer Text, Schema-Nodes, automatische Berechnungen, Bilder, Listen, usw.) muss in ein Layout-Feld platziert werden. Die Layout-Felder mit den darin enthaltenen Design-Elementen werden im Layout Container wie gewünscht angeordnet und mit einem Layout

© 2019-2025 Altova GmbH

versehen. Beachten Sie, dass ein Design-Element nicht direkt in einen Layout Container gesetzt werden kann; es muss in ein Layout-Feld platziert werden.

In diesem Abschnitt wird beschrieben, wie Layout-Felder verwendet werden. Der Abschnitt ist in die folgenden Unterabschnitte gegliedert:

- Einfügen von Layout-Feldern 175
- Auswählen und Verschieben von Layout-Feldern 175
- Anpassen der Größe eines Layout-Felds 175
- Definieren von Stileigenschaften für ein Layout-Feld 176
- Einfügen von Inhalt in das Layout-Feld 177
- Stapelreihenfolge von Layout-Feldern 177

Einfügen eines Layout-Felds

Ein Layout-Feld kann nur in einen Layout Container 171 eingefügt werden. Um ein Layout-Feld hinzuzufügen, klicken Sie zuerst in der Symbolleiste "Design-Elemente einfügen" 462 auf die Schaltfläche "Layout-Feld einfügen" und anschließend an die Stelle im Layout Container, an der das Layout-Feld eingefügt werden soll. Daraufhin wird ein Layout-Feld eingefügt, wobei dessen linke obere Ecke an die Stelle platziert wird, auf die Sie geklickt haben. Das Feld wird durchsichtig, ohne Ränder und mit Standardtext darin angezeigt.

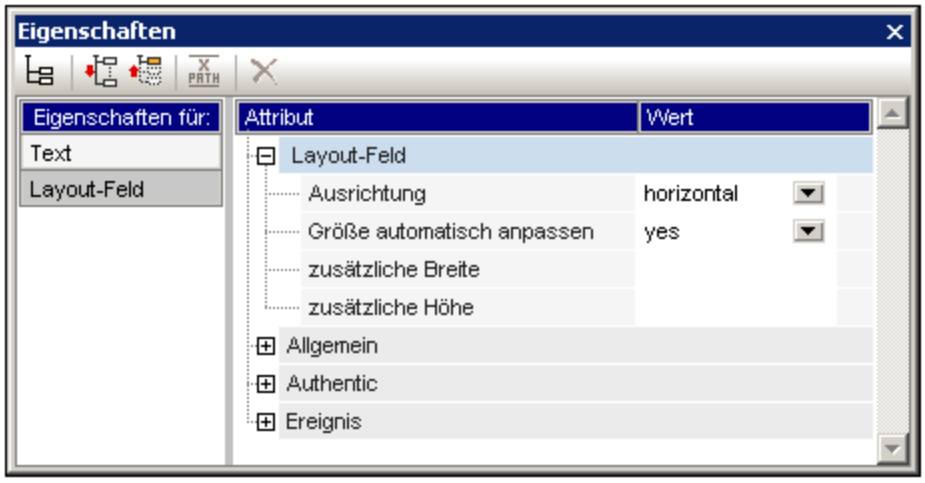
Auswählen und Verschieben von Layout-Feldern

Um ein Layout-Feld auszuwählen, platzieren Sie den Cursor über den linken oder oberen Rand des Layout-Felds, sodass der Cursor als Doppelpfeilkreuz angezeigt wird und klicken Sie um das Layout-Feld auszuwählen. Wenn Sie die Maustaste gedrückt halten, können Sie das Layout-Feld an eine andere Stelle im Layout Container ziehen. Sie können ein Layout-Feld nach links, rechts, oben oder unten verschieben, indem Sie es auswählen und anschließend die entsprechende Cursor-Taste drücken. Wenn das Layout-Feld ausgewählt ist, werden seine Eigenschaften und Stile in den entsprechenden Seitenleisten angezeigt.

Layout-Feldgröße

Jedes Layout-Feld hat eine Eigenschaft namens *Größe automatisch anpassen (siehe Abbildung unten*). Wenn der Wert dieser Eigenschaft auf `yes` gesetzt ist, wird die Größe des Felds automatisch an den in die Design-Ansicht eingefügten statischen Inhalt (einschließlich Markup-Tags) angepasst. Wenn der Wert dieser Eigenschaft auf no gesetzt ist, ändert sich die Größe des Layout-Felds nicht automatisch, wenn Inhalt eingefügt wird.

Um die Größe des Layout-Felds automatisch anzupassen, ziehen Sie den rechten und unteren Rand mit der Maus auf die richtige Größe. Die Größe eines Layout-Felds kann auch durch Verschieben des rechten und unteren Rands des Felds mit Hilfe der Cursor-Tasten geändert werden. Wählen Sie das Layout-Feld dazu zuerst aus 175. Gehen Sie anschließend folgendermaßen vor: (i) um den rechten Rand zu verschieben, halten Sie die **Umschalttaste** gedrückt und drücken Sie die rechte oder linke Cursor-Taste bis das Feld die gewünschte Größe hat; (ii) um den unteren Rand zu verschieben, halten Sie die **Umschalttaste** gedrückt und drücken Sie die nach oben oder nach unten-Taste (Cursor-Taste).

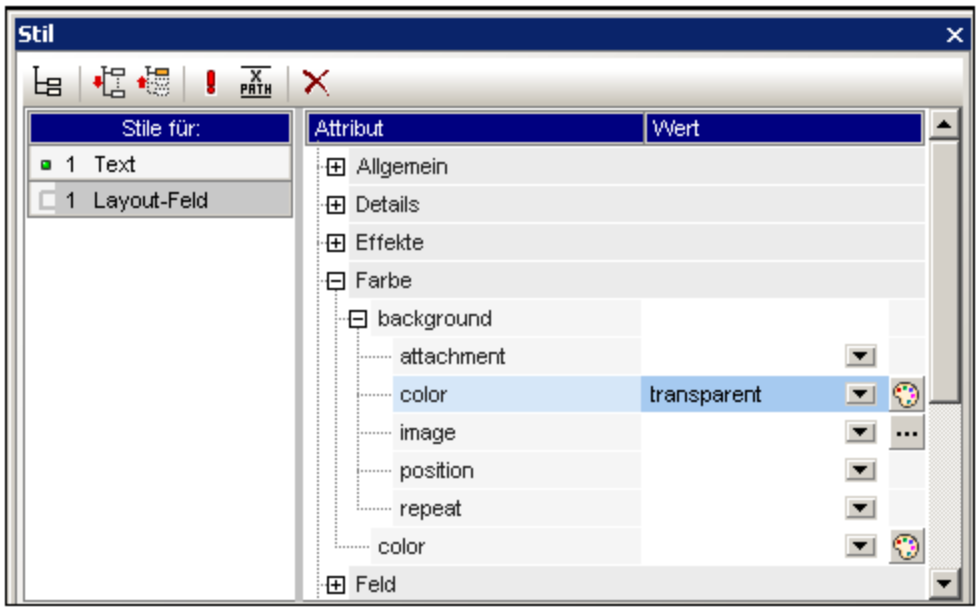


Die Eigenschaften *zusätzliche Höhe* und *zusätzliche Breite* geben die Längen an, die zusätzlich zur optimalen Größe (bei Auswahl der Option "Automatisch an die optimale Größe anpassen") anfallen. Diese zusätzlichen Längen erhält man, wenn man die Größe eines Layout-Felds manuell anpasst. Wenn Sie umgekehrt die Werte dieser beiden Eigenschaften ändern, ändert sich die Größe des Layout-Felds.

Anmerkung: Durch Drücken der **Eingabetaste** können Sie einen Zeilenumbruch in ein Layout-Feld einfügen. Dies ist deshalb von Bedeutung, da die aktuelle Zeile bei Hinzufügen von Inhalt ohne Zeilenumbruch, länger wird, wodurch die optimale Breite des Layout-Felds vergrößert wird, was sich auch auf den Wert *Additional Width* auswirkt, der anhand der optimalen Breite berechnet wird.

Stileigenschaften des Layout-Felds

Die Stileigenschaften eines Layout-Felds werden in der Seitenleiste "Stil" in den Layout-Feld-Stilen (*Abbildung unten*) definiert. Die Stile werden angezeigt, wenn ein Layout-Feld ausgewählt wird, und können anschließend bearbeitet werden.



 © 2019-2025 Altova GmbH

Anmerkung: Der *background-color* Wert `transparent` kann in der Dropdown-Liste der Eigenschaftsauswahlliste ausgewählt werden (er steht nicht in der Farbpalette zur Verfügung). Weiter unten wird erklärt, welche Bedeutung dieser Wert in einer Situation hat, in der das Layout-Feld Teil eines Stapels ist.

Einfügen von Inhalt in ein Layout-Feld

In ein Layout-Feld kann jede Art von Design-Element eingefügt werden. Dies erfolgt auf die gleiche Weise wie in einem SPS. Beachten Sie allerdings, dass weder ein Layout Container[171] noch eine Layout-Linie[178] in ein Layout-Feld eingefügt werden kann. Beachten Sie dazu die folgenden Punkte:

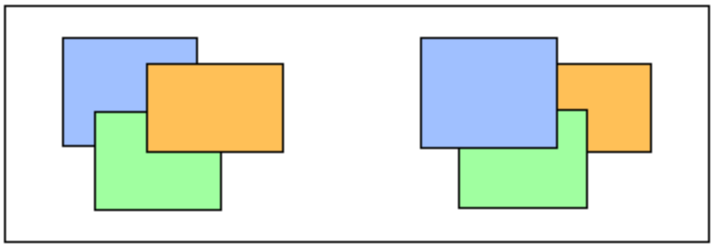
- Wenn Design-Elemente eingefügt werden, für die ein Kontext-Node benötigt wird, wird als Kontext-Node der aktuelle Node verwendet. Der aktuelle Node ist der Node, in dem das Layout-Modul erstellt wurde.
- Text in einem Layout-Feld kann um 90 Grad im oder entgegen dem Uhrzeigersinn gedreht werden, sodass der Text vertikal angezeigt wird und von oben nach unten bzw. von unten nach oben gelesen wird. Wählen Sie dazu im Design den gewünschten Text aus und wählen Sie in der Seitenleiste "Eigenschaften" (*Abbildung unten)* den Eintrag `LayoutBox` aus. Wählen Sie in der Eigenschaftsgruppe *Layout-Feld* für die Eigenschaft *Ausrichtung* den gewünschten Wert aus.

Beachten Sie dazu die folgenden Punkte:

- Die Drehung wird an der Ausgabe vorgenommen, im Design aber nicht angezeigt.
- Die Eigenschaft kann nicht nur auf Text in Tabellenzellen auch auf Text in Textfeldern[177] angewendet werden.

Stapelreihenfolgen von Layout-Feldern

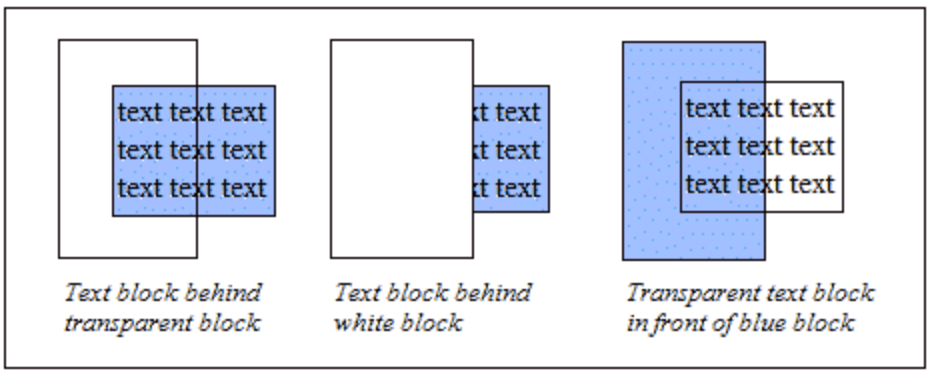
Layout-Felder können übereinander platziert werden. Wenn ein Layout-Feld über ein anderes gelegt wird, wird dadurch der darunterliegende Teil des anderen Layout-Felds verdeckt, wenn das obere undurchsichtig ist. Dieses Verhalten kann auch auf einen Stapel mehrerer Layout-Felder angewendet werden. In einem solchen Stapel ist nur das oberste Layout-Feld zur Gänze sichtbar; die anderen werden zum Teil oder zur Gänze verdeckt.



Mit Hilfe der Befehle aus dem Menü **Reihenfolge** im Kontextmenü des ausgewählten Layout-Felds kann ein Feld nach vorne oder nach hinten gebracht werden. Mit Hilfe dieser Befehle kann ein Layout-Feld relativ zu (i) seinem unmittelbaren Nachbarfeld im Stapel angeordnet werden (Befehle **Eine Eben nach vorne** und **Eine Ebene nach hinten**) oder (ii) relativ zum gesamten Stapel verschoben werden (Befehle **In den Vordergrund** und **In den Hintergrund**). In der Abbildung oben ist die Reihenfolge im Stapel von vorne nach hinten die folgende:

- *Linker Stapel:* orange, grün, blau
- *Rechter Stapel:* blau, grün, orange

Beachten Sie, dass es bei Layout-Feldern mit transparentem Hintergrund (dem Standardhintergrund von Layout-Feldern) so aussehen kann, als ob sich die Felder relativ zueinander nicht verschoben hätten. Dies ist vor allem dann der Fall, wenn mehrere Felder im Stapel transparent sind und die Felder keine Umrandung haben. In der Abbildung unten sehen Sie, wie sich Transparenz auf den Stapel auswirken kann.



Anmerkung: Sie können zu einem Stapel von Layout-Feldern auch Layout-Linien[178] hinzufügen. Jede Linie kann relativ zu den anderen Elementen im Stapel verschoben werden.

5.12.3 Linien

Linien können in einen Layout Containter eingefügt werden[178] (nicht aber in Layout-Felder) und anschließend ausgewählt, in der Größe angepasst und im Layout Container verschoben werden[179]. Den Linien können Eigenschaften zugewiesen werden[179] und die Linien können in einem Stapel von Layout-Elementen bestehend aus Layout-Feldern und anderen Linien nach vorne oder hinten verschoben werden[180].

Einfügen einer Linie

Um eine Linie zu einem Layout Container hinzuzufügen, gehen Sie folgendermaßen vor:

© 2019-2025 Altova GmbH

1. Klicken Sie in der Symbolleiste Design-Elemente einfügen[462] auf die Schaltfläche "Linie einfügen".
2. Klicken Sie an die Stelle im Layout Container, an der Sie den Anfangspunkt der Linie einfügen möchten.
3. Ziehen Sie die Maus bei gedrückter Maustaste bis zu dem Punkt, an dem die Linie enden soll. Lassen Sie anschließend die Maustaste los.

Daraufhin wird eine schwarze Linie eingefügt, wobei ein Punkt an jedem Ende den Anfangs- bzw. Endpunkt der Linie markiert.

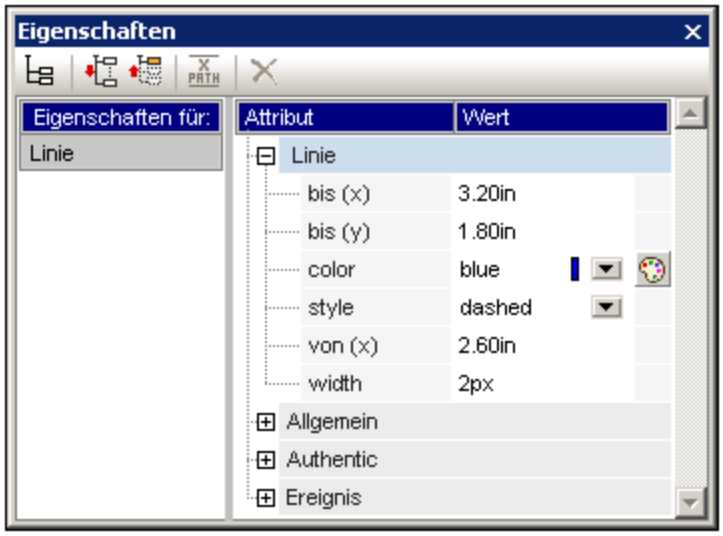
Auswählen, Verschieben, Anpassen der Größe einer Linie

Sie können im Hauptfenster die folgenden Drag-and-Drop-Funktionen ausführen:

- Um eine Linie auszuwählen, klicken Sie auf einen beliebigen Teil der Linie (der Cursor verwandelt sich in einen gekreuzten Doppelpfeil, wenn er sich über der Linie befindet). Sobald eine Linie ausgewählt ist, werden ihre Eigenschaften in der Seitenleiste "Eigenschaften" angezeigt und können hier bearbeitet werden (*siehe unten*).
- Um eine Linie zu verschieben, wählen Sie sie aus und ziehen Sie sie an die gewünschte Stelle. Sie können eine Linie auch nach links, rechts, oben oder unten verschieben, indem Sie sie auswählen und die Cursor-Taste für die entsprechende Richtung drücken.
- Um die Größe einer Linie anzupassen oder ihre Ausrichtung zu ändern, wählen Sie entweder den Anfangs- oder den Endpunkt der Linie aus und ziehen Sie ihn an eine andere Stelle um die Größe bzw. Ausrichtung der Linie zu ändern. Sie können die Größe oder Ausrichtung einer Linie auch anpassen, indem Sie die **Umschalttaste** gedrückt halten und die Cursor-Taste drücken. Mit der rechten und linken Cursor-Taste können Sie den rechten Endpunkt nach rechts bzw. nach links verschieben. Mit der nach oben- und der nach unten-Taste können Sie den rechten Endpunkt nach oben bzw. nach unten verschieben.

Linieneigenschaften

Wenn eine Linie ausgewählt ist, werden ihre Eigenschaften in der Seitenleiste "Eigenschaften" (*Abbildung unten*) angezeigt und die Eigenschaften (unten aufgelistet) können in der Seitenleiste bearbeitet werden. Oder Sie klicken mit der rechten Maustaste auf eine Linie, um die Seitenleiste "Eigenschaften" mit den Eigenschaften der Linie aufzurufen.

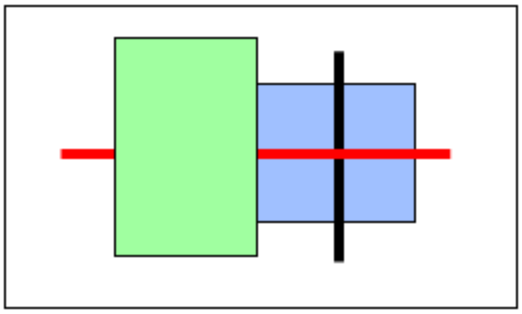


Die folgenden Linieneigenschaften können in der Seitenleiste "Eigenschaften" bearbeitet werden:

- *Color:* Definiert eine Farbe für die Linie. Die Standardfarbe ist Schwarz.
- *Größe und Position:* Die Position des Anfangs- und Endpunkts der Linie können mit Hilfe eines X-Y-Koordinationsystems (horizontal - vertikal) definiert werden. Das Referenzrahmen ist der Layout Container, wobei die linke obere Ecke des Layout Containers die Koordinaten `(x=0, y=0)` hat.
- *Width:* Definiert die Linienstärke.

Linien und Stapelreihenfolge

Mit Hilfe der Befehle aus dem Menü **Reihenfolge** im Kontextmenü der ausgewählten Linie kann eine Linie in einem Stapel aus Layout-Feldern und anderen Linien nach vorne oder nach hinten gebracht werden. Mit Hilfe dieser Befehle kann eine Linie relativ zu (i) ihrem unmittelbaren Nachbarelement im Stapel angeordnet werden (Befehle **Eine Eben nach vorne** und **Eine Ebene nach hinten**) oder (ii) relativ zum gesamten Stapel verschoben werden (Befehle **In den Vordergrund** und **In den Hintergrund**).



In der Abbildung oben ist die Reihenfolge im Stapel von vorne nach hinten die folgende: grünes Feld, schwarze Linie, blaues Feld.

© 2019-2025 Altova GmbH

5.13 Die Funktion "Ändern in"

Die Funktion **Ändern in** steht zur Verfügung, wenn eine Vorlage oder der Inhalt einer Vorlage ausgewählt ist und dient zu folgenden Zwecken: (i) Ändern des Node, für den diese Vorlage angewendet wird, oder (ii) Ändern der Art, wie der Node im Design erstellt wird.

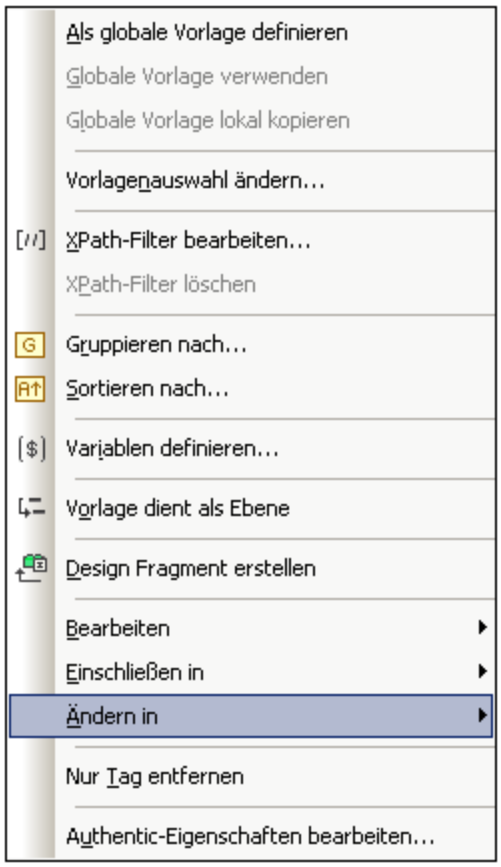
Was kann mit der Funktion "Ändern in" geändert werden

Sie können entweder einen Node oder dessen Inhalt ändern. In der Abbildung auf der linken Seite ist der Node ausgewählt. In der Abbildung auf der rechten Seite ist der Inhalt des Node ausgewählt.

Das `n1:Name` Element in der Abbildung oben wurde als `(Inhalt)` erstellt. Daher steht der Platzhalter `(Inhalt)` für den Inhalt des Node. Alternativ dazu hätte der Node als ein anderer Inhaltstyp erstellt werden können, z.B. als Eingabefeld oder Auswahlliste. Es können auch andere Inhaltstypen ausgewählt werden.

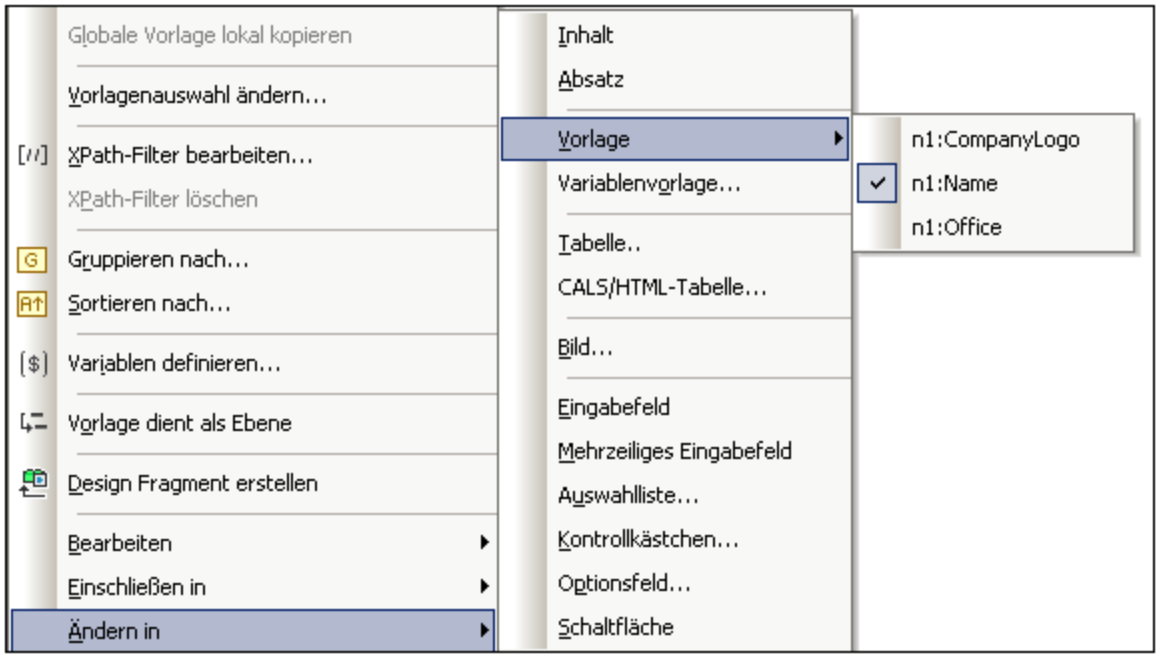
Der Befehl "Ändern in"

Rufen Sie den Befehl durch Rechtsklick auf Ihre Auswahl auf. Wählen Sie im Kontextmenü, das angezeigt wird, den Befehl **Ändern in** (*Abbildung unten*).



Ändern der Nodes, auf die die Vorlage angewendet werden kann

Wenn eine Vorlage ausgewählt ist, können Sie den Node wechseln, auf den diese Vorlage angewendet wird. Dies ist nützlich, wenn sich z.B. der Name eines Elements im Schema geändert hat. Wenn Sie die Maus über den Befehl "Ändern in" platzieren und im Untermenü, das angezeigt wird, "Vorlage" auswählen, erhalten Sie eine Liste aller Nodes, die als Child Node des übergeordneten Elements des ausgewählten Node eingefügt werden können. Klicken Sie auf einen dieser Nodes, um die Vorlage auf diesen Node anzuwenden.



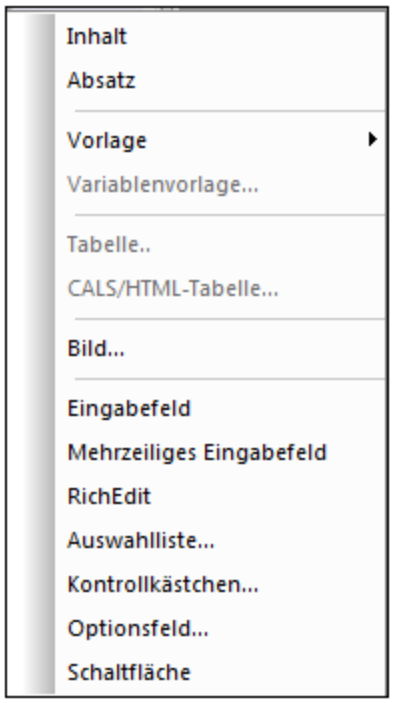
Wenn der ausgewählte Node ein Content Model hat, das nicht mit dem in der Vorlage beschriebenen übereinstimmt, kommt es zu strukturellen Ungereimtheiten. Dabei handelt es sich um Fehler, die durch rote durchgestrichene Node Tags, die ungültig sind, gekennzeichnet werden.

Sie können auch die Vorlagenübereinstimmung ändern, so dass nicht ein Node sondern eine Variablenvorlage 238 übereinstimmen soll.

Ändern des Inhaltstyps des Node

Wenn eine Vorlage oder ihr Inhalt ausgewählt werden, können Sie den Inhaltstyp, als der der Node erstellt wird, ändern. Wenn Sie den Mauszeiger im Kontextmenü über den Befehl "Ändern in" platzieren, wird in den Optionen des Untermenüs der Inhaltstyp angezeigt, in den der ausgewählte Node geändert werden kann (*Abbildung unten*).

© 2019-2025 Altova GmbH



Der obige Screenshot wurde gemacht, während eine Auswahlliste ausgewählt war.

6 SPS-Struktur

Die Struktur eines SPS Dokuments ist sowohl input- als auch outputgesteuert und wird von folgenden Faktoren gesteuert:

- Schemaquellen 186
- Modulare SPSs 216
- Vorlagen und Design-Fragmente 230

Input-gesteuerte Struktur: Schemas und modulare SPS-Dateien

Mit Input-gesteuert ist gemeint, dass die Struktur des Input-Dokuments/der Input-Dokumente durch die Quellschemas von SPS-Dateien bestimmt wird und dass dies die Struktur ist, auf der das SPS-Dokument basiert. Wenn z.B. in einem Quellschema eine Struktur definiert ist, die eine Sequenz von `Office` Elementen ist, dann könnte das SPS Design eine Vorlage für das `Office` Element haben. Während der Verarbeitung wird diese Vorlage wiederum auf die einzelnen `Office` Elemente im Quelldatendokument angewendet.

Ein weiteres Beispiel dafür, wie die Struktur des Quelldokuments das Design der SPS Datei bestimmt, sehen Sie in der Verwendung von Tabellen. Angenommen ein `Office` Element enthält mehrere `Person` Element-Children und jedes `Person` Element enthält eine Gruppe von Child-Elementen wie z.B. `Name`, `Address`, `Telephone`, usw. In diesem Fall kann eine Vorlage in Form einer Tabelle für das `Person` Element erstellt werden. Jedes `Perrson` Element kann in einer separaten Tabellenzeile dargestellt werden (*Abbildung unten*), wobei die Spalten die einzelnen Child-Elemente von `Person` sind.

First	Last	Title (sorted by)
Loby	Matise	Accounting Manager
Frank	Further	Accounts Receivable
Vernon	**Callaby**	Office Manager

Eine solche Vorlage ist auf Grund der Struktur des `Person` Elements möglich und weil die `Person` Elemente gleichrangige Elemente sind. In der Tabellenvorlage wurde eine einzelne Zeile für das `Person` Element erstellt. Diese Verarbeitung (das Zeilendesign) wird nacheinander auf die einzelnen `Person` Elemente im Quelldokument angewendet, sodass für jedes `Person` Element eine neue Zeile erstellt wird, wobei die Child-Elemente die Spalten der Tabelle bilden.

Eine Beschreibung, wie die verschiedenen Arten von Schemaquellen verwendet werden, finden Sie im Abschnitt Schemaquellen 186.

Außerdem können SPSs in StyleVision als Module mit anderen SPSs wiederverwendet werden. Auf diese Weise können Module in eine Struktur inkludiert werden und diese verändern. Allerdings muss eine in einem Modul enthaltene Schemastruktur auch mit der Struktur des zugrunde liegenden Schemas des SPS, in dem es enthalten ist, zusammenpassen. Eine Beschreibung zu Arbeiten mit modularen SPSs finden Sie im Abschnitt Modulare SPSs 216.

© 2019-2025 Altova GmbH

Output-gesteuerte Struktur: Vorlagen und Design-Fragmente

Während die Schemaquelle die Struktur des Input-Datendokuments bestimmt, ist das tatsächliche Design des Ausgabedokuments davon abhängig, was im SPS-Dokument definiert ist. Dieses Design ist in einer Dokumentenvorlage, der Hauptvorlage, enthalten. Die Hauptvorlage enthält normalerweise mehrere Komponentenvorlagen und kann globale Vorlagen referenzieren. Eine Beschreibung von Vorlagen finden Sie im Abschnitt Vorlagen und Design-Fragmente 230.

Dieses Baukastensystem (zusammengesetzt aus mehreren Vorlagen) wird noch durch die StyleVision-Funktion "Design-Fragmente" verbessert. Mit dieser Funktion kann einem wiederverwendbaren Dokument-Fragment eine bestimmt Verarbeitung zugewiesen werden. Ein Design-Fragment unterscheidet sich insofern von einer globalen Vorlage als es (i) aus mehreren Vorlagen zusammengesetzt werden kann und (ii) in separaten Design-Fragmenten, von denen je nach Bedarf ein bestimmtes in einer Vorlage verwendet werden kann, identischer Inhalt mit unterschiedlicher Verarbeitung erstellt werden kann. So wird z.B. in einigen Situationen ein `Email` Node als Link benötigt, über den eine leere E-Mail geöffnet wird; in anderen Fällen soll das `Email` Element rot und fett gedruckt werden. Für die entsprechende Verarbeitung können zwei verschiedene Design-Fragmente erstellt werden, von denen beide bei Bedarf wiederverwendet werden können.

Design-Fragmente werden im Abschnitt Design-Fragmente 242 näher beschrieben.

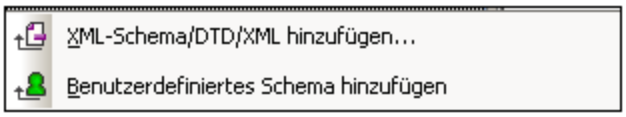
6.1 Schemaquellen

Die Schemaquellen bilden den Ausgangspunkt für das Design und die Designstrukur kann durch folgende Faktoren beeinflusst werden: (i) die Auswahl, die Sie bei der Auswahl des Schemas treffen und (ii) die Root-Elemente, die Sie im Schema auswählen.

Auswahl des Schemas

Die Auswahl des Schemas für eine neue SPS-Datei kann auf folgende Arten erfolgen:

1. Klicken Sie auf **Datei | Neu** und wählen Sie eine Schemaquelle, die mittels einer der im Popup-Menü verfügbaren Methoden (mit Ausnahme von "Neu (leer)" hinzugefügt werden soll, direkt aus.
2. Klicken Sie auf **Datei | Neu** und wählen Sie im Popup-Menü den Befehl "Neu (leer)" aus. Nachdem das neue SPS erstellt wurde und auf der Benutzeroberfläche angezeigt wird, wählen Sie in der Seitenleiste "Design-Übersicht" [36] den Befehl **Neues Schema hinzufügen** aus. Daraufhin wird ein Menü aufgerufen, in dem die Methoden aufgelistet werden, mit denen Sie verschiedene Schemaarten hinzufügen können (*Abbildung unten*). Die einzelnen Befehle in diesem Menü werden in den Unterabschnitten dieses Abschnitts beschrieben.



Bei der Schemaquelle kann es sich um eine Datei oder ein benutzerdefiniertes Schema handeln. Ein wichtiger Punkt ist auch die Frage, ob globale Vorlagen verwenden werden und ob Elemente, die Sie als globale Vorlagen erstellen möchten, im Schema als globale Elemente definiert sind. Wenn Sie anhand einer Datei eine DTD hinzufügen möchten, denken Sie daran, dass alle in der DTD definierten Elemente globale Elemente sind. Wenn Sie ein XML-Schema anhand einer Datei hinzufügen, sollten Sie überprüfen, welche Elemente als globale Elemente definiert sind. Sollten Sie Änderungen am Schema vornehmen wollen, überprüfen Sie, ob dies in Ihrer XML-Umgebung zulässig ist.

Anmerkung: Wenn Sie einen Namespace zu einem SPS oder einem anhand eines SPS generierten XSLT-Stylesheet hinzufügen möchten, muss der Namespace zum `schema`-Element der obersten Ebene des XML-Schemas, auf dem das SPS basiert, hinzugefügt werden.

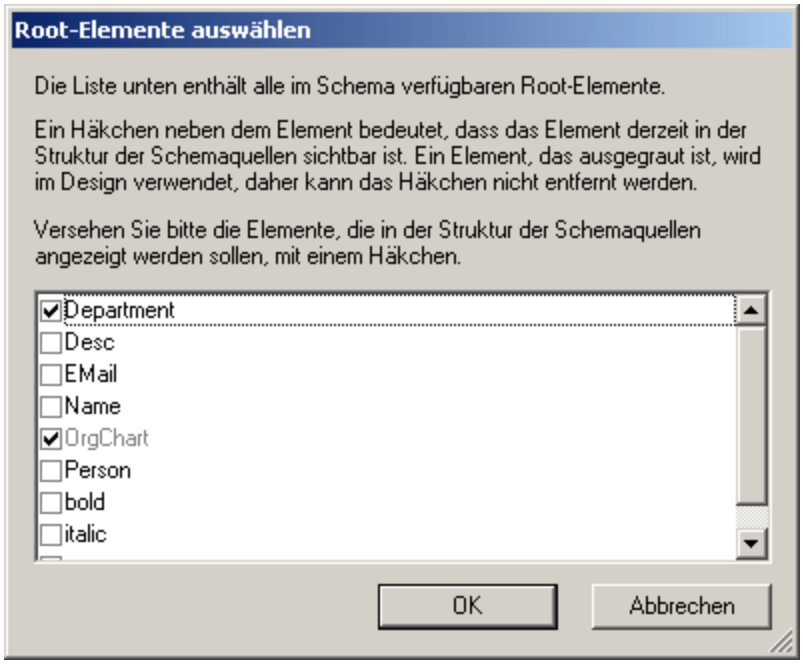
Root-Elemente

Wenn eine Schemaquelle mehrere globale Elemente [22] hat, so können mehrere Root-Elemente (Dokumentelemente [22]) für die Verwendung im Design ausgewählt werden. Auf diese Art können Sie für das SPS Design Vorlagen bereitstellen, die mehreren Dokumentelementen entsprechen. Der Vorteil dabei ist, dass, wenn ein SPS, z.B. `UniversalSPS.sps` basierend auf der Datei `UniversalSchema.xsd` je eine Vorlage für jedes seiner zwei Root-Elemente, `Element-A` und `Element-B`, hat, dann dieses eine SPS mit einem XML-Instanzdokument verwendet werden kann, welches `Element-A` als sein Dokumentelement hat, und genauso auch mit einem anderen XML-Instanzdokument, welches `Element-B` als Dokumentelement hat. Für jede XML-Instanz wird die entsprechende Vorlage verwendet und die andere ignoriert. Dies liegt daran, dass es für des Dokumentelement jedes einzelnen XML-Instanzdokuments im SPS nur eine Vorlage gibt, die mit diesem Dokumentelement übereinstimmt. So entspricht etwa dem Dokumentelement `/Element-A` die Vorlage, die `/Element-A` auswählt, nicht aber jene, welche `/Element-B` auswählt. Beachten Sie in diesem Zusammenhang auch Folgendes: Wenn im Schema mehrere globale Elemente definiert sind, ist ein XML-

 © 2019-2025 Altova GmbH

Dokument, das irgendeines dieser globalen Elemente enthält, gültig (vorausgesetzt die Substruktur ist entsprechend dem Schema gültig).

Um das SPS für die Verwendung mehrerer Root-Elemente (Dokumentelemente[22]) einzurichten, klicken Sie auf die [...] Schaltfläche rechts vom Eintrag `/Root-Elemente` des Schemas. Daraufhin wird das folgende Dialogfeld angezeigt.



In diesem Dialogfeld werden alle globalen Elemente im Schema aufgelistet. Wählen Sie jene globalen Elemente aus, die als Root-Elemente (Dokumentelemente[22]) zur Verfügung stehen sollen und klicken Sie auf OK. Das/Die ausgewählte/n Element/e steht/stehen nun als Dokument-Root-Elemente zur Verfügung und wird/werden in der Liste der Root-Elemente angezeigt. Für jedes dieser Dokumentelemente kann nun eine Vorlage angelegt werden. Jede dieser Vorlagen dient als alternative Root-Element-Vorlage. Wenn ein XML-Dokument mit diesem SPS verarbeitet wird, wird immer nur eine der Root-Elementvorlagen verwendet, nämlich diejenige, die mit dem Root (oder Dokument-)-Element des XML-Dokuments übereinstimmt.

Wenn daher ein XML-Dokument verarbeitet wird, das `Element-A` als Dokument-Element hat, so wird automatisch die Root-Vorlage in dem SPS verwendet, die mit `Element-A` übereinstimmt, während alle anderen Root-Element-Vorlagen im SPS ignoriert werden. Wenn ein XML-Dokument verarbeitet wird, das `Element-B` als Dokument-Element hat, so wird automatisch die Root-Vorlage in dem SPS verwendet, die mit `Element-B` übereinstimmt, während alle anderen Root-Element-Vorlagen im SPS ignoriert werden. Auf diese Weise kann man zwei oder mehrere XML-Dokumente mit jeweils unterschiedlichen Root- (oder Dokument-) Elementen mit Hilfe eines einzigen SPS verarbeiten.

6.1.1 DTDs und XML-Schemas

Ein SPS kann auf einem XML-Schema oder einer DTD basieren. Ein XML-Schema oder eine DTD kann auf eine der folgenden Arten als Schemaquelle erstellt werden:

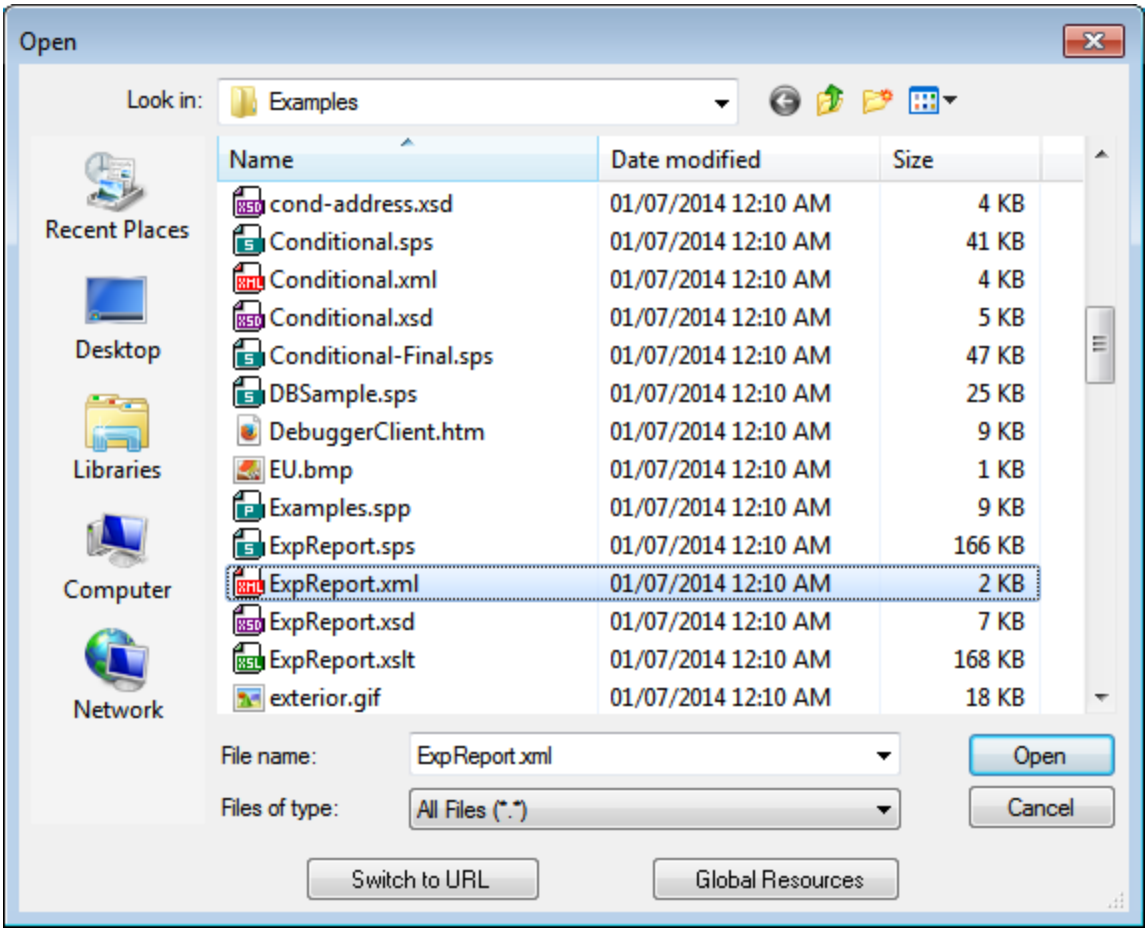
- Das XML-Schema/die DTD wird direkt bei der Erstellung des SPS (**Datei | Neu | Neu von XML-Schema / DTD / XML**) als Schemaquelle erstellt.
- Das XML-Schema oder die DTD wird (in der Seitenleiste "Design-Übersicht" 36 zu einem leeren SPS hinzugefügt .

Bei Aufruf des jeweiligen Befehls werden Sie aufgefordert, das XML-Schema oder die DTD zu suchen. Wenn das Schema gültig ist, wird es in der Seitenleiste "Schema-Struktur" in der Baumstruktur der Schemaquellen als Schemaquelle erstellt. Alternativ dazu kann eine XML-Datei ausgewählt werden. Wenn ein XML-Schema (`.xsd`) oder eine DTD-Datei mit der XML-Datei verknüpft wird, so werden das XML-Schema bzw. die DTD als Quellschema und die XML-Datei als XML-Arbeitsdatei geladen. Wenn kein Schema mit der XML-Datei verknüpft ist, wird ein Dialogfeld angezeigt, in dem Sie gefragt werden, ob auf Basis der Struktur und des Inhalts der XML-Datei ein XML-Schema generiert werden soll oder ob Sie ein bestehendes Schema auswählen möchten. Wenn Sie ein Schema generieren, wird dieses als Quellschema geladen und die XML-Datei wird als XML-Arbeitsdatei geladen.

▼ 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.

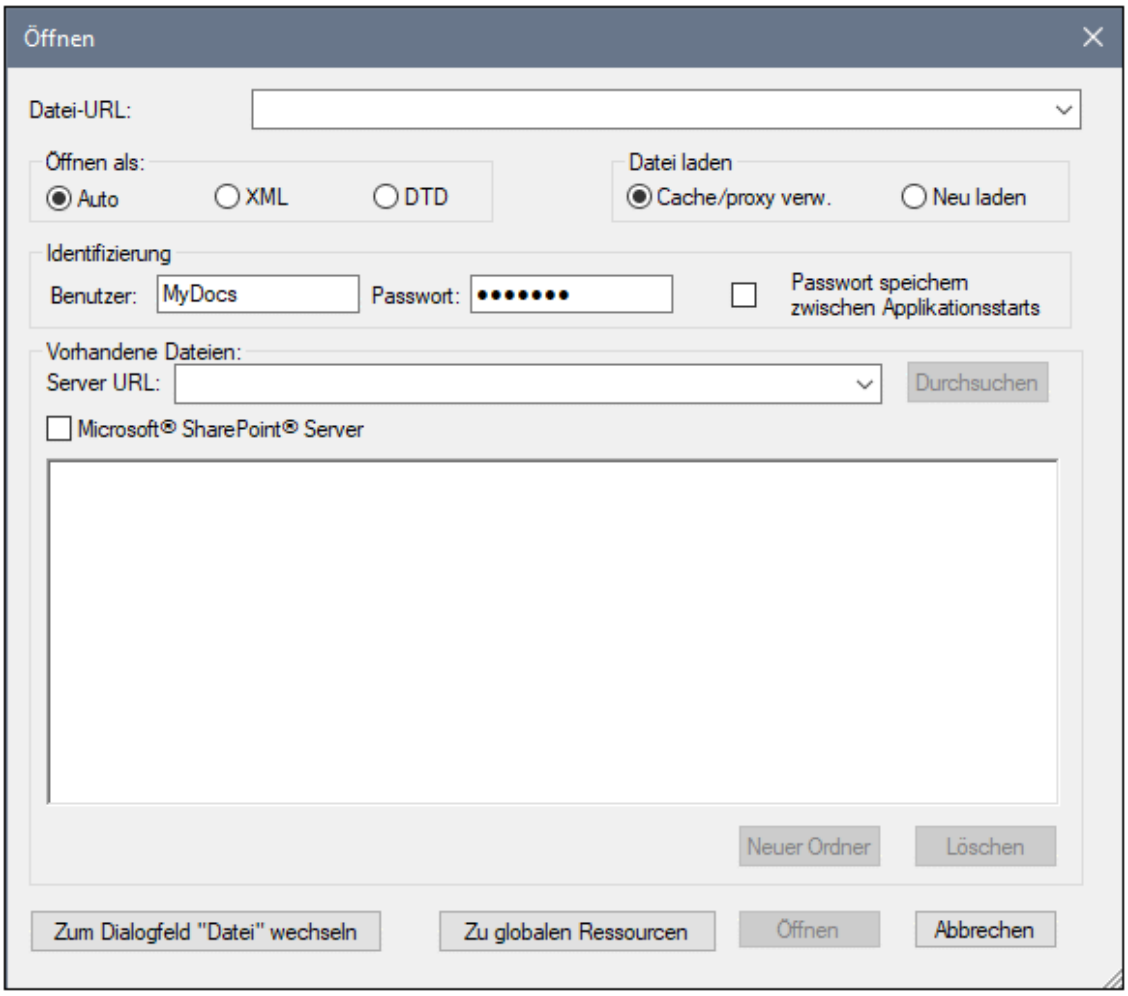
© 2019-2025 Altova GmbH



Auswählen von Dateien über URLs

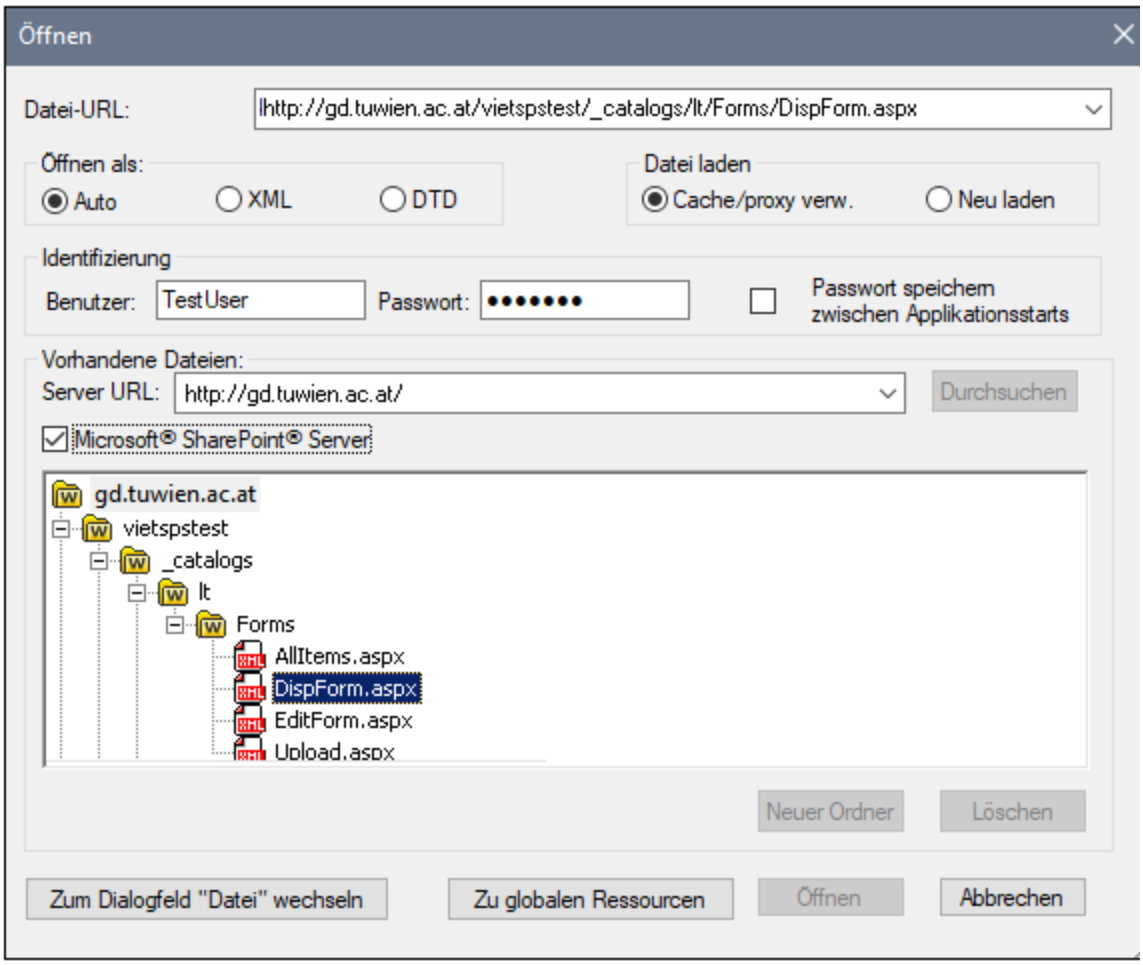
Um eine Datei über eine URL auszuwählen, gehen Sie folgendermaßen vor:

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.

© 2019-2025 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:

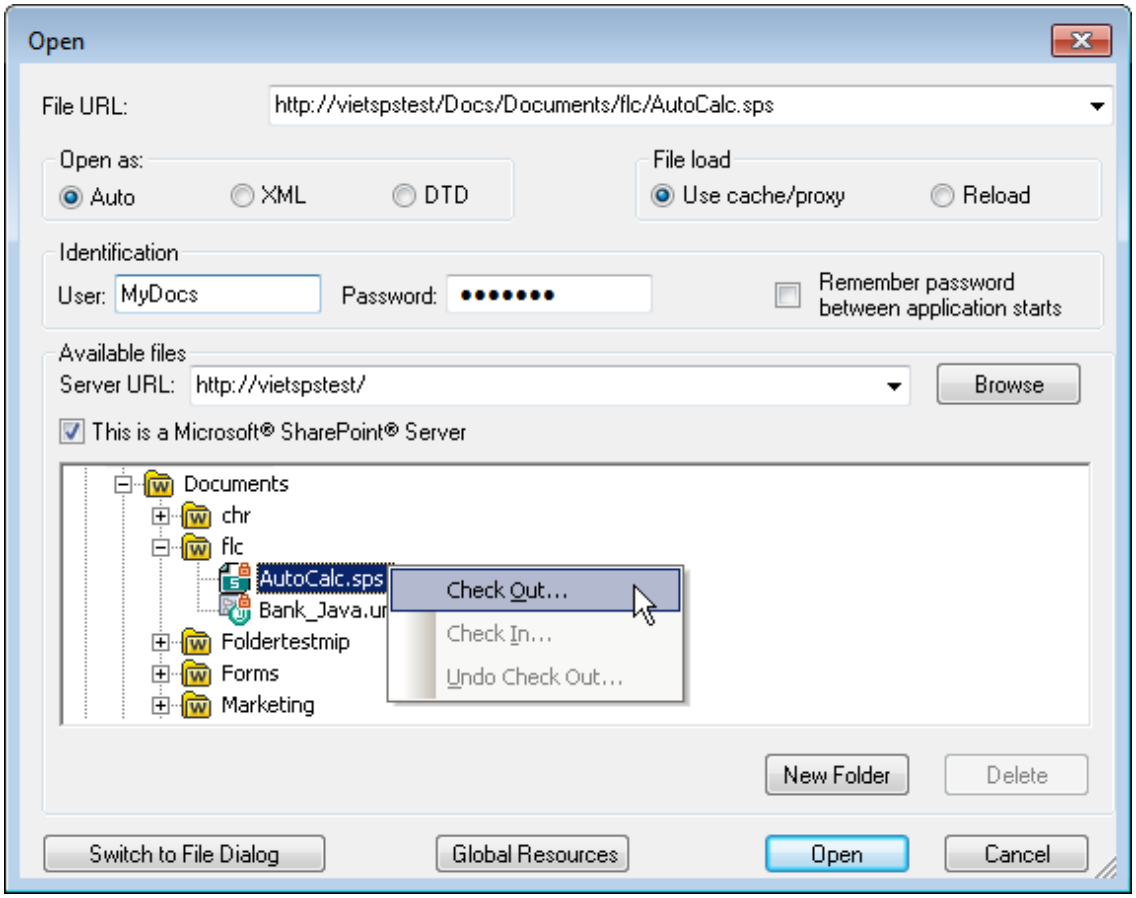
- Die Durchsuchen-Funktion steht nur auf Servern zur Verfügung, die WebDAV und Microsoft SharePoint Server unterstützen. Die unterstützten Protokolle sind FTP, HTTP und HTTPS.
- Damit Sie mehr Kontrolle über den Ladevorgang haben, können Sie auswählen, ob die Datei über den lokalen Cache-Speicher oder 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.

© 2019-2025 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 näher beschrieben. Eine allgemeine Beschreibung globaler Ressourcen finden Sie in der Dokumentation im Abschnitt.

Der anyType-Datentyp von XML-Schema

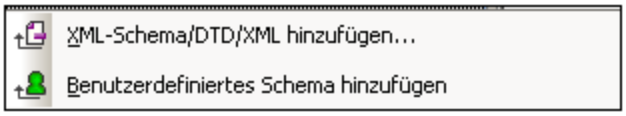
Wenn einem Element im XML-Schema der Datentyp `anyType` von XML-Schema zugewiesen wurde, dann werden für dieses Element in der Schema-Struktur alle globalen Elemente dieses Schemas als mögliche Subelemente angezeigt. Wenn z.B. einem Element namens `email` kein Datentyp zugewiesen wurde, wird es in der Schema-Struktur mit allen globalen Elementen als möglichen Subelementen (z.B. `person`, `address`, `city`, `tel`, usw.) angezeigt. Um dies zu vermeiden, weisen Sie dem Element `email` z.B. den Datentyp `xs:string` zu.

6.1.2 Benutzerdefinierte Schemas

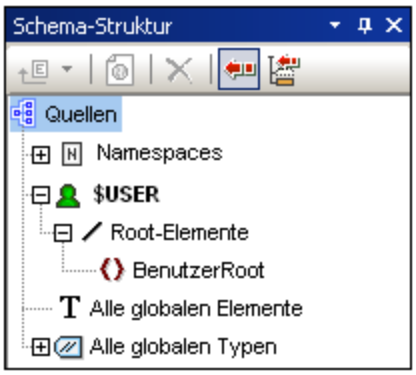
In der Seitenleiste "Schema-Struktur" [38] können Sie ein benutzerdefiniertes Schema schnell erstellen. Dies ist vor allem dann nützlich, wenn Sie ein XML-Dokument haben, das auf keinem Schema basiert und Sie möchten ein SPS für dieses XML-Dokument erstellen.

Um ein benutzerdefiniertes Schema zu erstellen und in der Seitenleiste "Schema-Struktur" hinzuzufügen, gehen Sie folgendermaßen vor:

1. Klicken Sie auf **Datei | Neu | Neu (leer)**. Klicken Sie in der Seitenleiste "Design-Übersicht" [36] (unter der Überschrift "Quellen") auf den Befehl **Neue Quelle hinzufügen** und wählen Sie den Befehl **Benutzerdefiniertes Schema hinzufügen** (*Abbildung unten*).

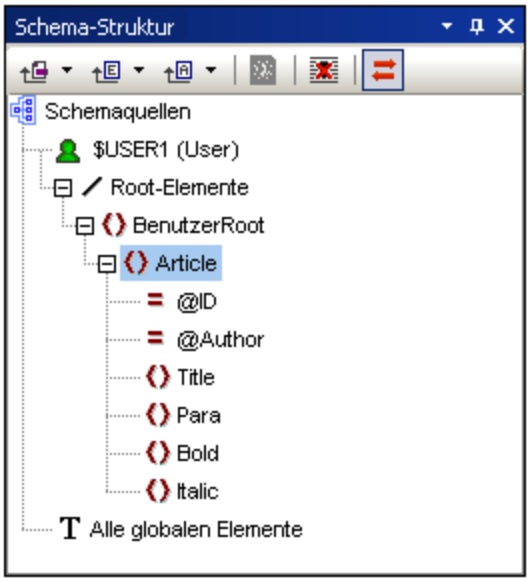


Das neue Schema wird erstellt und erhält die Bezeichnung `$USER` (*Abbildung unten*).

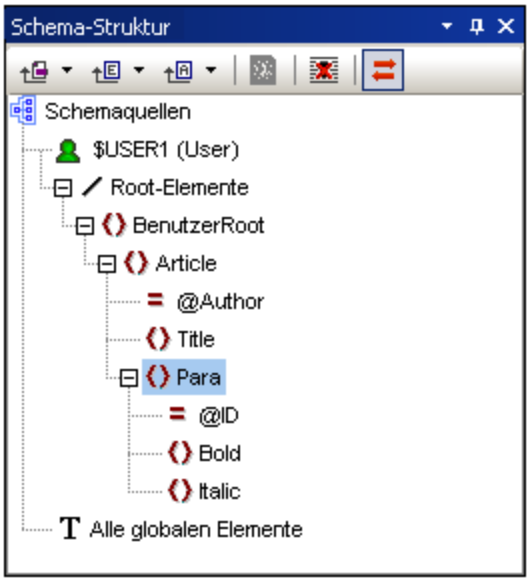


2. In der Baumstruktur des Root-Elements gibt es ein einziges Root-Element (Dokumentelemente)[22] mit der Bezeichnung `BenutzerRoot`.
3. Doppelklicken Sie auf Benutzer`Root` und benennen Sie es um, damit der Name dem Dokumentelement[22] des XML-Dokuments entspricht, für das Sie dieses Schema erstellen.
4. Um dem Dokumentelement ein Child-Element oder ein Attribut zuzuweisen, wählen Sie das Dokumentelement (`BenutzerRoot`) aus und klicken Sie (i) in der Symbolleiste der Seitenleiste "Schema-Struktur"[38] auf das Symbol "Neues Element anhängen" und (ii) auf den Auswahllistenpfeil des Befehls "Neues Element anhängen" und wählen Sie den Befehl "Element anhängen" | "Attribut anhängen" . Alternativ dazu können Sie rechtsklicken und den gewünschten Befehl aus dem Kontextmenü auswählen. Wenn ein Element ausgewählt ist, wird das neue Element beim Anhängen und Einfügen eines Elements als gleichrangiges Element nach bzw. vor dem ausgewählten Element eingefügt. Sie können auch ein Child-Element und ein Child-Attribut hinzufügen. Wenn ein Attribut ausgewählt ist, können Sie vor bzw. nach dem ausgewählten Attribut ein weiteres Attribut einfügen bzw. anhängen. Nachdem des neue Element oder Attribut zur Struktur hinzugefügt wurde, geben Sie den gewünschten Namen ein. Sie können Nodes auch mit der Maus an die gewünschte Stelle ziehen (im nächsten Schritt beschrieben). In der Abbildung unten ist das Element `Article` das Dokumentelement. Die Elemente `Title`, `Para`, `Bold` und `Italic`, und die Attribute `ID` und `Author` wurden unterhalb von `Article` hinzugefügt.

© 2019-2025 Altova GmbH



5. Um die Elemente `Bold` und `Italic` und das Attribut `ID` auf die Ebene der Child-Elemente von `Para` zu verschieben, wählen Sie jedes einzeln aus und ziehen Sie es zum Element `Para`. Sobald ein nach unten weisender Pfeil angezeigt wird, lassen Sie die Maustaste los, um den Node abzulegen. Der Node wird als "Child" von `Para` erstellt (*Abbildung unten*).



6. Wenn ein anderes Element als das Dokumentelement ausgewählt wird, wird der neue Node beim Hinzufügen eines neuen Elements oder Attributs auf derselben Ebene wie das ausgewählte Element hinzugefügt. Um einen Node als "Child-Element" des Element-Nodes zu erstellen, ziehen Sie ihn (Element oder Attribut) in einen Element-Node.

Bearbeiten von Node-Namen und Löschen von Nodes

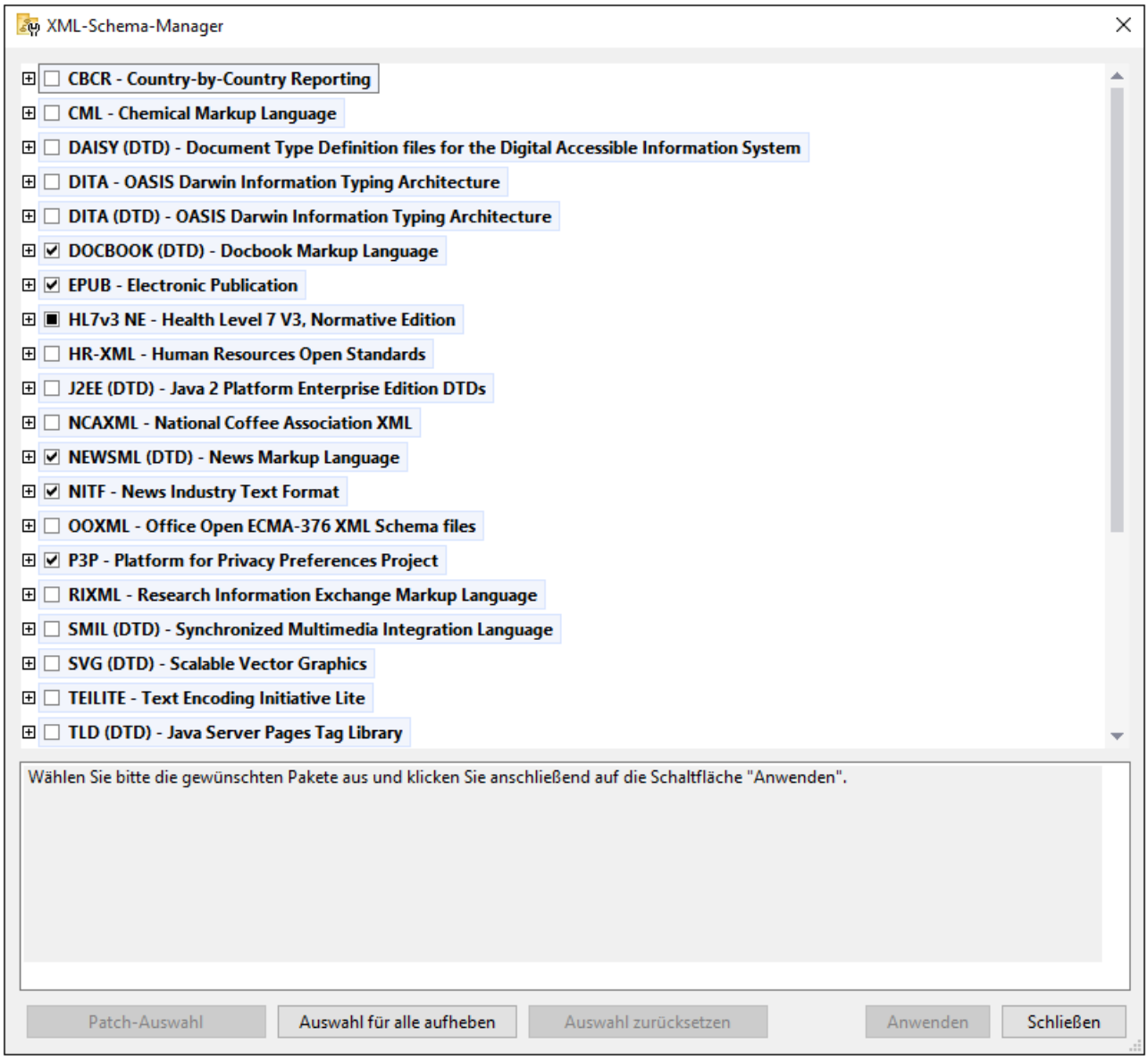
Um den Namen eines Elements oder Attributs zu bearbeiten, doppelklicken Sie in den Namen und bearbeiten Sie ihn. Um einen Node zu löschen, wählen Sie ihn aus und klicken Sie in der Symbolleiste auf das Symbol "Entfernen" . Alternativ dazu können Sie auch im Kontextmenü den Befehl **Entfernen** auswählen.

© 2019-2025 Altova GmbH

6.2 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-Schema-fähigen Applikationen von Altova einschließlich StyleVision 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 des Altova Mission Kit 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.
- Wenn Sie versuchen ein Dokument in einem Altova-Produkt anhand eines nicht installierten aber über Schema-Manager verfügbaren Schemas zu validieren, wird das Schema automatisch installiert. Wenn das Schema-Paket jedoch Namespace-Zuordnungen enthält, wird das Schema nicht automatisch installiert; in diesem Fall müssen Sie Schema-Manager starten, das/die gewünschte(n) Paket(e) auswählen und die Installation starten. Wenn Ihre offene Altova-Applikation nach der Installation nicht automatisch neu gestartet wird, müssen Sie sie manuell neu starten.

© 2019-2025 Altova GmbH

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.

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`[208] 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 StyleVision zum ersten Mal an einem bestimmten Kalendertag starten.
- Wenn StyleVision länger als 24 Stunden geöffnet ist, findet alle 24 Stunden eine Aktualisierung des Cache statt.
- Sie können den Cache auch durch Ausführung des `update`[211]-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`[209] der Befehlszeilenschnittstelle und (ii) anschließend den Befehl `initialize`[207] aus. (Führen Sie alternativ dazu den Befehl **reset** mit der Option **-i** aus).

6.2.1 Ausführen des Schema-Managers

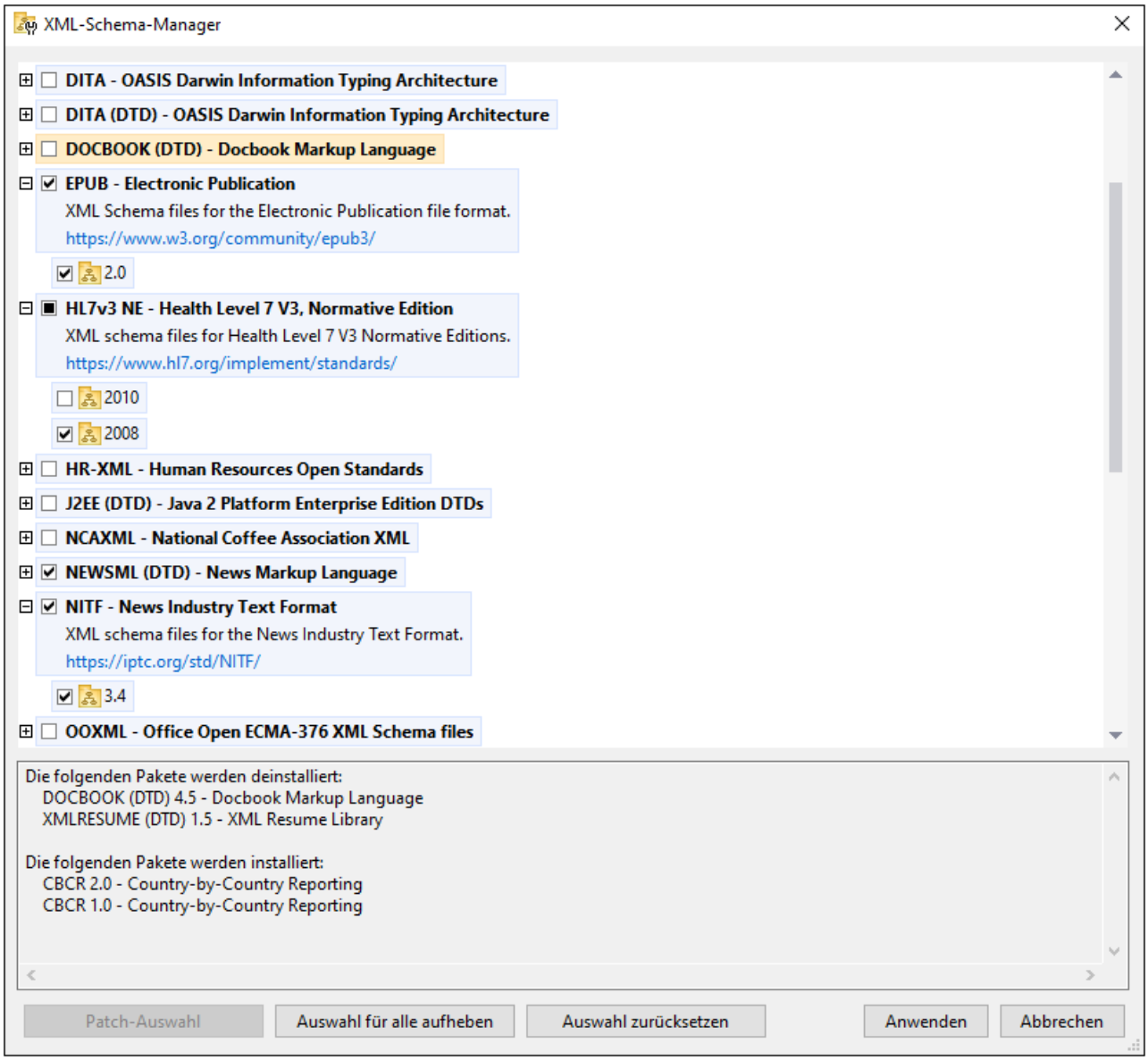
Grafische Benutzeroberfläche

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

- *Bei der Installationen von StyleVision:* 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 StyleVision:* 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.

© 2019-2025 Altova GmbH



Wenn Sie auf **Anwenden** klicken, wird der Fortschritt der Installation angezeigt. Bei Auftreten eines Fehlers (z.B. eines Verbindungsfehlers) wird eine Fehlermeldung angezeigt. Klicken Sie in diesem Fall auf die Schaltfläche **Erweitert**, die im Dialogfeld erscheint, überprüfen Sie die Schemaauswahl und versuchen Sie es erneut durch Klicken auf **Anwenden**.

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 205 aufgelisteten Befehle verwenden.

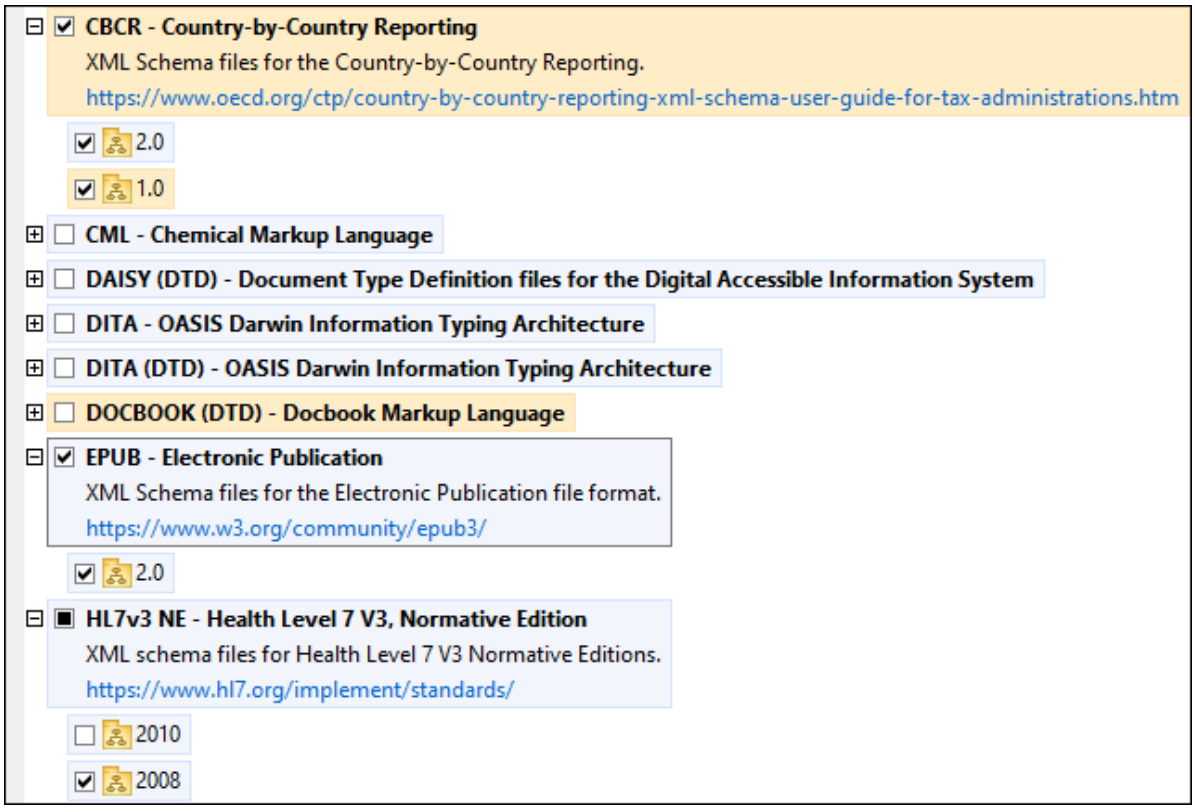
Um die Hilfe zu den Befehlen anzuzeigen, führen Sie den folgenden Befehl aus:

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

6.2.2 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.



 © 2019-2025 Altova GmbH

- *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` 208 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`

6.2.3 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* 202*).* 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` [208] aus. Daraufhin werden alle Schemas, für die Upgrades zur Verfügung stehen, aufgelistet.
2. Führen Sie den Befehl `upgrade` [211] 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` [208] 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` [208] ausführen.) Sie können beliebig viele Schemas eingeben. Nähere Informationen dazu finden Sie unter der Beschreibung des Befehls `install` [208].

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

Installation eines benötigten Schemas
Wenn Sie einen XML-Schema-Befehl in StyleVision ausführen und StyleVision 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.

6.2.4 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.

© 2019-2025 Altova GmbH

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` [210] 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` [208] ausführen.) Sie können beliebig viele Schemas eingeben. Nähere Informationen dazu finden Sie unter der Beschreibung des Befehls `uninstall` [210].

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` [209] aus.

Nachdem Sie diesen Befehl ausgeführt haben, muss der Befehl `initialize` [207] ausgeführt werden, um das Cache-Verzeichnis neu zu erstellen. Führen Sie alternativ dazu den Befehl `reset` [209] mit der Option `-i` aus.

Beachten Sie, dass mit `reset-i` [209] die Originalinstallation des Produkts wiederhergestellt wird, daher wird empfohlen, nach dem Zurücksetzen auch den Befehl `update` [211] auszuführen. Führen Sie alternativ dazu den Befehl `reset` [209] mit den Optionen `-i` und `-u` aus.

6.2.5 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.

6.2.5.1 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
```

6.2.5.2 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.

© 2019-2025 Altova GmbH

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[208].)
- 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
```

6.2.5.3 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
```

6.2.5.4 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[208] 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*[197].

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
```

6.2.5.5 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

© 2019-2025 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.

6.2.5.6 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 207 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 211 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.

6.2.5.7 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` 208 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* 197.

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`**.

 © 2019-2025 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
```

6.2.5.8 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`[209] und `initialize`[207] 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
```

6.2.5.9 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`[208].

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.

© 2019-2025 Altova GmbH

6.3 Zusammenführen von XML-Daten aus mehreren Quellen

Wenn als XSLT-Version des SPS die Version XSLT 2.0 oder XSLT 3.0 verwendet wird, so können XML-Daten aus mehreren XML-Quelldateien in einem Ausgabedokument zusammengeführt werden.

Normalerweise wird der Zusammenführung von Daten ein gemeinsamer Nenner in den Daten wie z.B. eine ID zugrunde gelegt. So kann die Personalabteilung z.B. die Personendaten eines Mitarbeiters in einem Unternehmen, der durch eine persönliche ID identifiziert wird, in mehreren XML-Dateien speichern: (i) Stammdaten, (ii) Lohnzettel, (iii) Arbeitszeiten und Urlaub, (iv) besuchte Schulungen usw. Die Daten aus diesen verschiedenen Dateien können unter Verwendung der persönlichen ID als Schlüssel in einem einzigen Ausgabedokument zusammengeführt werden.

Anmerkung: In der Enterprise Edition können Sie mehrere Quellschemas inkludieren, sodass XML-Nodes aus anderen Schemas über den Parameternamen für das entsprechende Schema ausgewählt werden können (wie dies im Beispiel unten der Fall ist). In der Professional und der Basic Edition können Sie die gewünschte XML-Datei und den XML-Node in dieser Datei mit Hilfe der Funktion `doc()` abrufen. Die `doc()` Funktion von XPath 2.0 ermöglicht den Zugriff auf die Dokument-Root von externen XML-Dokumenten, wodurch Node-Inhalt aus externen XML-Dokumenten in die Ausgabe eingefügt werden kann. Sie können die XML-Daten daher mit Hilfe einer automatischen Berechnung 257 unter Verwendung der `doc()` Funktion in der Ausgabe zusammenführen (*siehe Beispiel unten*).

Beispiel

Der Ordner "Eigene Dateien" 25, `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2025\StyleVisionExamples`, enthält eine Beispiel-SPS-Datei (`MergeData_2_Files.sps`), in der gezeigt wird, wie Daten aus verschiedenen XML-Quelldateien zusammengeführt werden können. Das SPS wählt Daten aus einer Bestellung (`MergeOrder.xml`, *siehe unten*) aus, die ein fiktiver Kunde abgegeben hat.

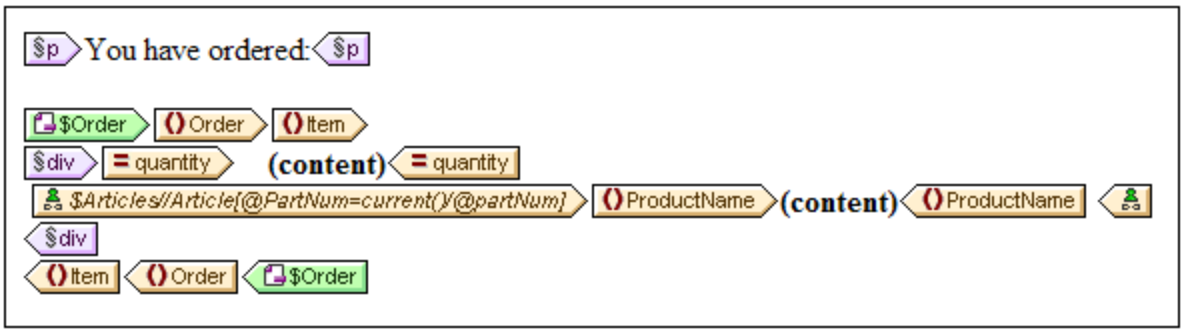
```
<?xml version="1.0" encoding="UTF-8"?>
<Order xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="MergeOrder.xsd">
   <Item partNum="238-KK" quantity="3" shipDate="2000-01-07" comment="With no inclusions,
please."/>
   <Item partNum="748-OT" quantity="1" shipDate="2000-02-14" comment="Valentine's day
packaging."/>
   <Item partNum="229-OB" quantity="1" shipDate="1999-12-05"/>
   <Item partNum="833-AA" quantity="2" shipDate="1999-12-05" comment="Need this for the
holidays!"/>
</Order>
```

Anhand des Werts des Attributs `/Order/Item/@partNum` in dieser Datei (*siehe oben*) werden die bestellten Produkte aus dem in einer anderen Datei gespeicherten Artikelkatalog `MergeArticles.xml` (*siehe unten)* ausgewählt.

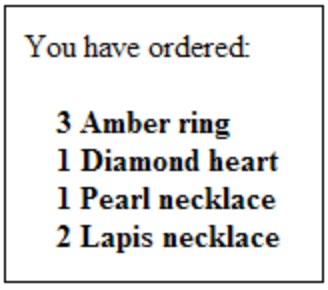
```
<?xml version="1.0" encoding="UTF-8"?>
<Articles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="MergeArticles.xsd">
   <Article PartNum="833-AA">
      <ProductName>Lapis necklace</ProductName>
      <Price>99.95</Price>
```

```
   </Article>
   <Article PartNum="748-OT">
      <ProductName>Diamond heart</ProductName>
      <Price>248.90</Price>
   </Article>
   <Article PartNum="783-KL">
      <ProductName>Uncut diamond</ProductName>
      <Price>79.90</Price>
   </Article>
   <Article PartNum="238-KK">
      <ProductName>Amber ring</ProductName>
      <Price>89.90</Price>
   </Article>
   <Article PartNum="229-OB">
      <ProductName>Pearl necklace</ProductName>
      <Price>4879.00</Price>
   </Article>
   <Article PartNum="128-UL">
      <ProductName>Jade earring</ProductName>
      <Price>179.90</Price>
</Article>
   ...
</Articles>
```

Um die Daten zusammenzuführen, wird innerhalb der Vorlage `/Order/Item` (*siehe Abbildung unten*) eine benutzerdefinierte Vorlage[235] angelegt, die das entsprechende `Article`-Element in der Datei `MergeArticles.xml` anhand der Artikelnummer des bestellten Artikels auswählt. Der XPath-Ausdruck (der sich im `/Order/Item` Kontext befindet) lautet: `$Articles//Article[@PartNum=current()/@partNum]`



Mit dieser Vorlage wird ein Ausgabedokument erzeugt, das in etwa wie in der Abbildung unten aussieht.

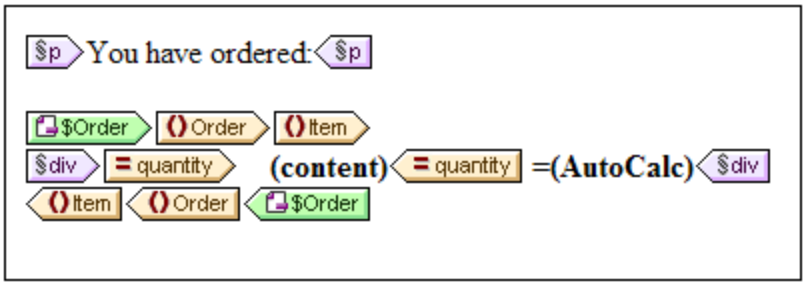


Beachten Sie Folgendes: Während die Anzahl der einzelnen bestellten Artikel aus der Datei `MergeOrder.xml` stammt, wird der Name des bestellten Artikels aus der Datei `MergeArticles.xml` abgerufen. Bitte beachten

© 2019-2025 Altova GmbH

Sie außerdem, wie der Node `ProductName` innerhalb des Kontexts der Vorlage `/Articles/Article` ausgewählt wird.

Dasselbe Ergebnis wie oben könnte auch mit Hilfe einer automatischen Berechnung[257] erzielt werden (*siehe Abbildung unten*). Ziehen Sie das `quantity` Attribut aus dem Fenster "Schema-Struktur" und erstellen Sie es als "Inhalt". Fügen Sie anschließend wie in der Abbildung gezeigt, eine automatische Berechnung hinzu und weisen Sie der automatischen Berechnung wie oben beschrieben einen XPath-Ausdruck zu.



Der XPath-Ausdruck der automatischen Berechnung könnte den benötigten Node entweder über den Parameter einer anderen Schemaquelle oder die `doc()` Funktion abrufen:

```
$Articles//Article[@PartNum=current()/@partNum]/ProductName
```

oder

```
doc('MergeArticles.xml')//Article[@PartNum=current()/@partNum]/ProductName
```

Beachten Sie Folgendes: Während der erste XPath-Ausdruck oben einen Parameter verwendet, um ein anderes XML-Schema zu referenzieren (eine Funktion, die nur in der Enterprise Edition zur Verfügung steht), verwendet der zweite Ausdruck die `doc()` Funktion von XPath 2.0 (eine Funktion, die auch in der Professional und der Basic Edition zur Verfügung steht).

6.4 Modulare SPSs

Die globalen Vorlagen eines SPS sowie die Design-Fragmente, JavaScript-Funktionen und Seitenlayout-Elemente können im Design eines anderen SPS verwendet werden. Dadurch haben Sie folgende Möglichkeiten:

1. Wiederverwendung globaler Vorlagen und anderer Komponenten in mehreren SPSs. Hauptvorteil dabei ist die Bearbeitung einer einzigen Quelle und eine einheitliche Ausgabe.
2. SPSs können in Modulform erstellt werden, wodurch die Struktur flexibler wird.

In jedem beliebigen SPS kann ein oder mehrere SPSs als Module hinzugefügt werden. Einige Komponententypen (oder Objekte) in diesen Modulen stehen dann dem importierenden (oder referenzierenden) SPS zur Verfügung.

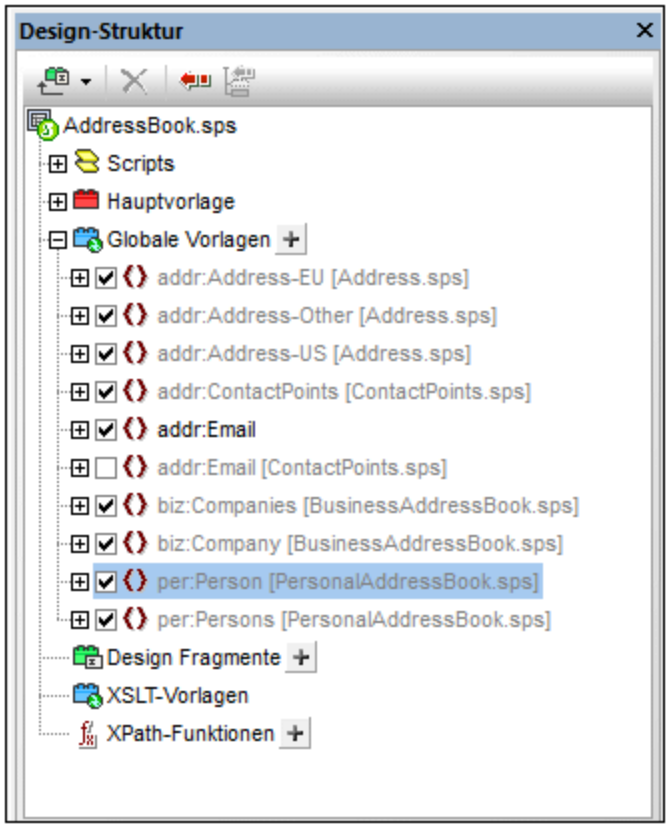
Verfügbare Modulobjekte

Im Abschnitt Verfügbare Modulobjekte 217 wird u.a. das Ausmaß beschrieben, bis zu dem verschiedene Komponenten eines SPS einem importierenden SPS zur Verfügung stehen, sowie die Bedingungen, unter denen dies der Fall ist. Des Weiteren werden jene Komponenten aufgelistet, die für ein importierendes SPS nicht zur Verfügung stehen. Beachten Sie bitte: Wenn ein hinzugefügtes Modul selbst Module enthält, werden diese rekursiv zum referenzierenden SPS hinzugefügt. Auf diese Art kann die Modularisierung auf mehrere Ebenen und über eine breite Designstruktur erweitert werden.

Erstellen eines modularen SPS

Um ein aus Modulen zusammengesetztes SPS zu erstellen, fügen Sie zuerst das benötigte SPS 221 zum Haupt-SPS als Modul hinzu. Alle JavaScript-Funktionen, globalen Vorlagen, Design-Fragmente, und XPath-Funktionen im hinzugefügten Modul stehen dem referenzierenden SPS zur Verfügung. Die einzelnen verfügbaren Objekte werden in der Design-Struktur unter der entsprechenden Überschrift aufgelistet (*Abbildung unten*) und können durch Aktivieren bzw. Deaktivieren ihres Kontrollkästchens aktiviert bzw. deaktiviert werden.

© 2019-2025 Altova GmbH



Diese Objekte können anschließend ihren jeweiligen Inkludierungsmechanismen entsprechend im referenzierenden SPS wiederverwendet werden. Globale Vorlagen müssen normalerweise nur aktiviert werden, um im referenzierenden SPS angewendet zu werden. Design-Fragmente müssen aus der Design-Struktur an die gewünschte Stelle gezogen werden. JavaScript-Funktionen werden über das Fenster "Eigenschaften" als Event Handler für die ausgewählte Design-Komponente zugewiesen. Verfügbare (aktivierte) XPath-Funktionen können in XPath-Ausdrücken verwendet werden.

Eine Anleitung zum Erstellen und Arbeiten mit einem modularen SPS finden Sie im Abschnitt Erstellen eines modularen SPS 221.

Terminologie

Wenn ein SPS innerhalb eines anderen Moduls verwendet wird, spricht man vom **Hinzufügen** eines <%SPS%. Die beiden SPSs werden als das ***hinzugefügte SPS Modul*** und das ***referenzierende SPS Modul*** bezeichnet. Wenn ein SPS-Modul hinzugefügt wird, werden seine **Objekte** zum referenzierenden SPS Modul hinzugefügt. Diese Objekte werden als ***Modulobjekte*** bezeichnet und gehören den folgenden Typen an: globale Vorlagen; Design-Fragmente; JavaScript-Funktionen und Seitenlayout-Elemente.

6.4.1 Verfügbare Modulobjekte

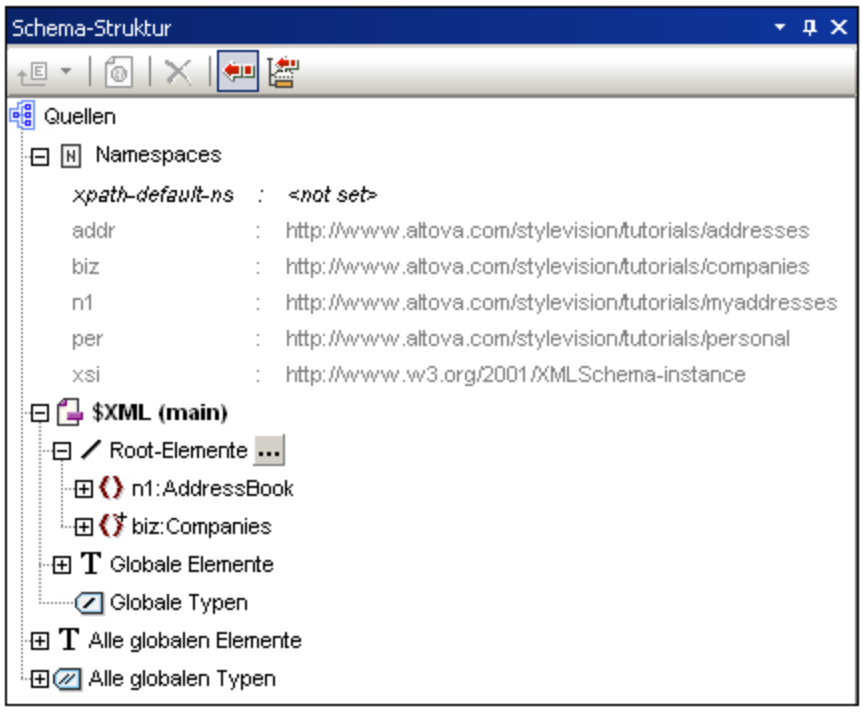
In diesem Abschnitt werden die Objekte in hinzugefügten SPS Modulen 217 aufgelistet, die dem referenzierenden SPS Modul 217 zur Verfügung stehen. In der Liste sehen Sie, auf welche Art die einzelnen Objekte dem referenzierenden SPS Modul zur Verfügung stehen und wie sie dort verwendet werden können.

Eine schrittweise Anleitung zum Erstellen eines modularen SPS finden Sie im nächsten Abschnitt Erstellen eines modularen SPS 221. Am Ende des Abschnitts finden Sie eine Liste von Objekten im hinzugefügten SPS, die dem referenzierenden SPS Modul nicht zur Verfügung stehen. Dies soll Ihnen helfen, besser zu verstehen, wie modulare SPSs funktionieren.

- Namespace-Deklarationen 218
- Globale Vorlagen 219
- Design-Fragmente 220
- Hinzugefügte Module 220
- Skripts 220
- CSS-Stile 220
- Seitenlayouts 220
- Nicht verfügbare modulare Objekte 220

Namespace-Deklarationen

Jedes SPS enthält eine Liste von Namespace URIs und ihren Präfixen. Wenn ein SPS als Modul hinzugefügt wird, werden die Namespaces darin mit den Namespaces in der/den Schemaquelle(n) des referenzierenden SPS verglichen. Wenn eine Namespace URI im hinzugefügten SPS mit einer Namespace URI in der/den Schemaquelle(n) des referenzierenden SPS übereinstimmt, wird das in der Schemaquelle des referenzierenden SPS verwendete Präfix als Präfix für diesen Namespace im hinzugefügten SPS übernommen. Wenn für eine Namespace URI im hinzugefügten SPS keine Übereinstimmung mit einer Namespace URI in der/den Schemaquelle(n) des referenzierenden SPS gefunden wird, wird eine entsprechende Fehlermeldung angezeigt.

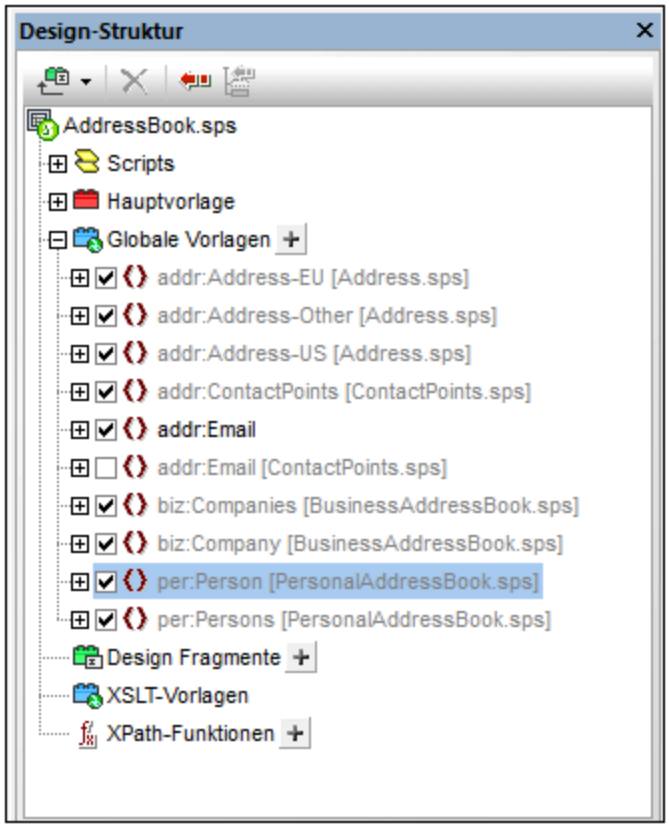


In der Abbildung oben werden die verschiedenen Namespaces in einem SPS mit ihren Präfixen angezeigt. Diese Namespaces stammen aus dem/den Quellschema(s) und können nicht bearbeitet werden.

© 2019-2025 Altova GmbH

Globale Vorlagen

Die globalen Vorlagen 231 des hinzugefügten SPS Moduls stehen dem referenzierenden SPS Modul zur Verfügung und werden in der Seitenleiste "Design-Struktur" 41 angezeigt (*Abbildung unten*). Sie sind je nach ihrem jeweiligen Aktivierungsstatus im hinzugefügten Modul standardmäßig aktiviert oder deaktivert (mit bzw. ohne Häkchen angezeigt). Wenn Sie eine globale Vorlage erstellen möchten, die eine globale Vorlage aus einem hinzugefügten Modul außer Kraft setzen soll, erstellen Sie durch Klicken auf das +-Symbol neben dem Eintrag "Globale Vorlagen" eine neue globale Vorlage. Wählen Sie nun im Dialogfeld "Neue globale Vorlage hinzufügen", das jetzt angezeigt wird, ein Element oder ein Attribut aus, für das sie die globale Vorlage erstellen möchten. Alternativ dazu können Sie einen XPath-Ausdruck eingeben, mit dem der gewünschte Node im Schema ausgewählt wird. Wenn Sie auf **OK** klicken, werden Sie gefragt, ob die neue globale Vorlage anstelle der globalen Vorlage im hinzugefügten Modul aktiviert werden soll. Je nachdem, welche Auswahl Sie treffen, wird entweder die neu erstellte globale Vorlage oder die globale Vorlage im hinzugefügten Modul aktiviert. Sie können die globale Vorlage jederzeit wechseln, indem Sie die andere der beiden globalen Vorlagen aktivieren.



Beachten Sie, dass die Hauptvorlage hinzugefügter Module nicht zur Verfügung steht. Wenn Sie daher planen, eine Vorlage als modulare Vorlage wiederzuverwenden, müssen Sie sie als globale Vorlage erstellen. Wenn für ein bestimmtes Element keine globale Vorlage definiert wurde und dieses Element verarbeitet werden soll, wird die Standardverarbeitung für dieses Element (die vordefinierte XSLT-Vorlagen) verwendet.

Design-Fragmente

Design-Fragmente[242] im hinzugefügten SPS Modul stehen dem referenzierenden SPS zur Verfügung und werden in der Seitenleiste Design-Struktur[41] (*Abbildung oben*) angezeigt. Achten Sie darauf, ein Design-Fragment, wenn Sie es einfügen, in den richtigen Kontext-Node im Design zu platzieren.

Hinzugefügte Module

Jedes hinzugefügte SPS Modul stellt dem referenzierenden SPS auch seine eigenen hinzugefügten Module und deren hinzugefügte Module usw. zur Verfügung. Auf diese Art stehen bei Hinzufügung eines Moduls alle diesem über mehrere Ebenen hinweg rekursiv hinzugefügten Module zur Verfügung. Natürlich müssen diese Module ein gültiges Content Model bilden, das gemäß der/den Schemaquelle(n) des referenzierenden SPS Moduls gültig ist. Module können in der Seitenleiste "Design-Übersicht"[36] angezeigt und verwaltet werden.

Skripts

Die Skripts in allen hinzugefügten SPS Modulen stehen im referenzierenden SPS zur Verwendung zur Verfügung und werden in der Seitenleiste "Design-Struktur"[403] angezeigt. Dabei werden die Skripts aller hinzugefügten Module in einer Bibliothek gesammelt, die nun - im referenzierenden SPS - im Dialogfeld "Eigenschaften" zur Auswahl zur Verfügung steht.

CSS-Stile

Die globalen Stile in hinzugefügten SPS Modulen werden als globale Stile im referenzierenden SPS übernommen und die Stilregeln werden in der Seitenleiste "Globale Stile"[343] angezeigt. Auf ähnliche Weise stehen auch externe CSS-Dateien, die dem hinzugefügten SPS Modul zur Verfügung standen, dem referenzierenden SPS Modul zur Verfügung und werden in der Seitenleiste "Design-Struktur"[403] angezeigt.

Seitenlayouts

Die Seitenlayouts eines hinzugefügten Moduls stehen dem referenzierenden SPS zur Verfügung .

Modulobjekte, die dem referenzierenden SPS nicht zur Verfügung stehen

Die folgenden Objekte des hinzugefügten Moduls stehen dem referenzierenden SPS nicht zur Verfügung:

- ***Parameterdefinitionen:*** werden ignoriert.
- ***Schemaquellen:*** Die Schemaquelle, auf der das hinzugefügte SPS basiert, wird ignoriert. Beachten Sie, dass das Content Model des Dokumentelements des hinzugefügten SPS im Content Model des referenzierenden SPS enthalten sein muss; andernfalls wäre es nicht möglich, das hinzugefügte SPS als Modul korrekt zu verwenden. Gegebenenfalls können Sie ein benutzerdefiniertes Schema zum referenzierenden SPS hinzufügen. Das zusätzliche Schema könnte das Content Model der hinzugefügten globalen Vorlage(n) enthalten.
- ***XML-Arbeitsdatei und XML-Vorlagendatei:*** Referenzen auf diese Dateien werden ignoriert. Das referenzierende SPS verwendet seine eigene XML-Arbeits- und Vorlagendatei.
- ***XPath-Standard-Namespaces:*** Wenn Sie für ein Modul festgelegt wurden, das importiert wird, werden sie nicht im importierenden SPS übernommen.

© 2019-2025 Altova GmbH

6.4.2 Erstellen eines modularen SPS

Die Erstellung eines modularen SPS erfolgt grundsätzlich in vier Schritten:

1. Erstellung und Speichern des hinzuzufügenden SPS Moduls[221].
2. Hinzufügen des Moduls[222] zum SPS, in dem es verwendet werden soll (d.h. zum referenzierenden SPS Modul).
3. Aktivieren bzw. Deaktivieren des hinzugefügten Objekts/der hinzugefügten Objekte[223], je nach Anforderung.
4. Anwenden des benötigten Objekts, wo erforderlich.

Das hinzuzufügende SPS Modul

Zu beachten sind beim Erstellen eines SPS, das zu einem anderen hinzugefügt werden soll, die folgenden Punkte:

1. Bei Vorlagen, die im referenzierenden SPS Modul[217] verwendet werden können, muss es sich um globale Vorlagen[231] handeln. Dies bedeutet, dass die Vorlagen, die wiederverwendet werden sollen, im hinzuzufügenden SPS Modul[217] als globale Vorlagen erstellt werden müssen.
2. Die im hinzuzufügenden SPS Modul definierte Dokumentstruktur muss gemäß dem im/in den Quellschema(s) des referenzierenden SPS[220] definierten Content Model gültig sein. Wenn eine hinzugefügte Vorlage nicht im Content Model enthalten ist, das für das Hauptschema des SPS definiert wurde, kann ihr Content Model in einem benutzerdefinierten Schema definiert werden.

Wenn Sie das hinzuzufügende SPS Modul erstellen, kann es sich beim Schema, das Sie dem SPS zugrunde legen, um eines der folgenden handeln:

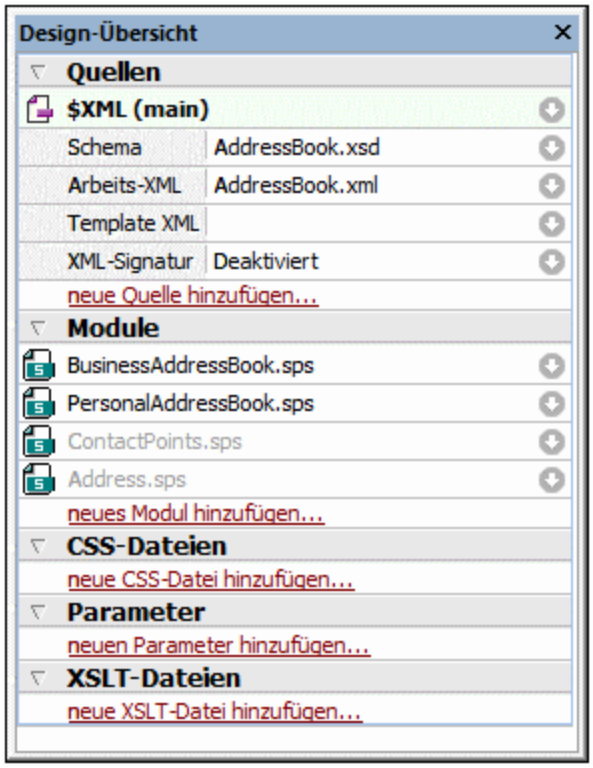
- das Hauptquellschema des referenzierenden SPS. In diesem Fall bilden die hinzugefügten globalen Vorlagen bei Hinzufügung des SPS einen Bestandteil des Content Model des Hauptschemas des referenzierenden SPS. Die Ausgabe dieser globalen Vorlagen kann daher in der Authentic-Ansicht bearbeitet werden.
- ein Schema, das ein Content Model definiert, das Teil des Content Model ist, das durch das Hauptschema des referenzierenden SPS definiert ist. In diesem Fall passen die globalen Vorlagen, die hinzugefügt werden, in das Content Model des Hauptschemas des referenzierenden SPS. Die Ausgabe dieser globalen Vorlagen kann in der Authentic-Ansicht bearbeitet werden.
- ein Schema, das ein Content Model definiert, das **nicht** Teil des durch das Hauptschema des referenzierenden SPS definierten Content Model ist. Wenn dieses SPS Modul hinzugefügt wird, bilden seine globalen Vorlagen nicht Teil des Content Model des Hauptschemas des referenzierenden SPS. Sie können jedoch eine Ausgabe anhand dieser Vorlagen erzeugen, wenn ein benutzerdefiniertes Schema verwendet wird, in dem ein Content Model definiert wird, das das Content Model der globalen Vorlage(n) enthält. Die Ausgabe dieser globalen Vorlagen kann jedoch in der Authentic-Ansicht nicht bearbeitet werden.

Achten Sie beim Definieren von Content Models in Ihren Schemas genau auf die verwendeten Namespaces[218], die festlegen, wie Node-Namen erweitert werden.

Sie können die Ausgabe des hinzuzufügenden SPS-Moduls mit Hilfe einer XML-Arbeitsdatei[23] testen. Die Referenz auf diese XML-Arbeitsdatei wird vom referenzierenden SPS ignoriert[220].

Hinzufügen des SPS Moduls

Um ein Modul zu einem SPS hinzuzufügen, klicken Sie in der Design-Übersicht[36] mit der rechten Maustaste auf den Eintrag "Neues Modul hinzufügen", suchen Sie im nun angezeigten Dialogfeld die benötigte SPS Datei, wählen Sie sie aus und klicken Sie auf **Öffnen.**



Das Modul wird zum SPS hinzugefügt und in der Design-Übersicht unter der Überschrift `Modules` aufgelistet. In der Abbildung oben wurden die Module `BusinessAddressBook.sps` und `PersonalAddressBook.sps` zum Modul `AddressBook.sps` (dem aktiven SPS)hinzugefügt. Alle hinzugefügten Modulobjekte werden in der Seitenleiste Design-Struktur angezeigt; hinzugefügte CSS-Dateien werden jedoch auch in der Design-Übersicht angezeigt. Wenn die hinzugefügten Module selbst Module referenzieren, werden diese indirekt importierten Module in Grau unter der Überschrift "Module" aufgelistet. Informationen darüber, welche Module ein indirekt importiertes Modul importieren, werden in einem Popup-Fenster angezeigt, das erscheint, wenn Sie die Maus über das indirekt importierte Modul platzieren.

Um eines der hinzugefügten oder indirekt importierten Module schnell in StyleVision zu öffnen, klicken Sie mit der rechten Maustaste auf das Modul und wählen Sie im Kontextmenü, das angezeigt wird, den Befehl **Definierendes Modul öffnen**.

Reihenfolge der hinzugefügten Module

Die Reihenfolge, in der Module hinzugefügt und aufgelistet werden, ist wesentlich für die Reihenfolge, in der CSS-Stile angewendet werden. CSS-Stilregeln haben - bei Beachtung der CSS-Stilreihenfolge - in einem relativ gesehen späteren Modul (also einem Modul, das weiter unten in der Liste vorkommt) Vorrang vor Stilen, die in einem relativ gesehen früheren Modul (also einem Modul weiter oben in der Liste) vorkommen. CSS-Stile im referenzierenden SPS Modul haben Vorrang vor denen in einem hinzugefügten Modul. Um die Position eines

© 2019-2025 Altova GmbH

hinzugefügten Moduls zu ändern, klicken Sie in der Design-Struktur mit der rechten Maustaste darauf und klicken Sie im Kontextmenü, je nach Wunsch auf den Befehl **Nach oben** oder **Nach unten.**

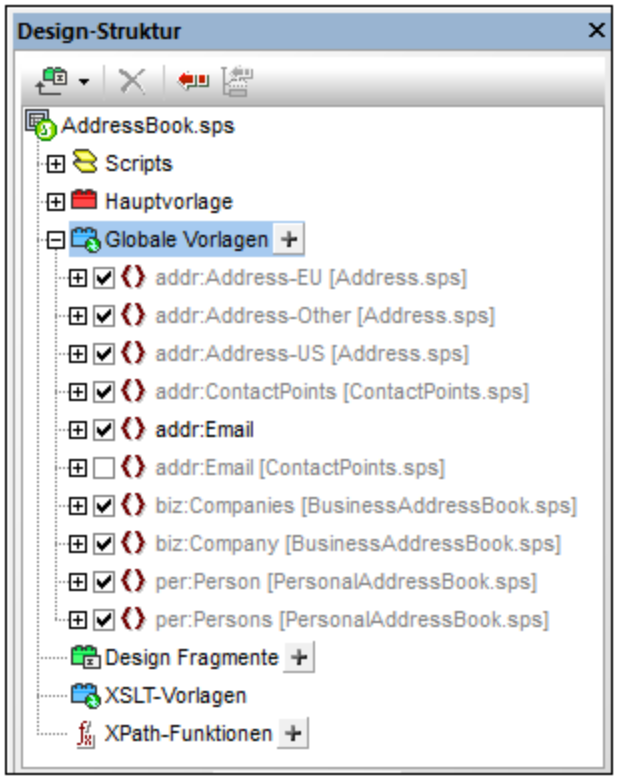
Die Modulreihenfolge spielt bei der Lösung von Konflikten zwischen Skripts, globalen Vorlagen, Design-Fragmenten und Seitenlayouteinträgen keine Rolle.

Warnmeldungen über geänderte Dateien

Wenn eine hinzugefügte Datei (ob es sich nun um ein SPS Modul, ein Schema oder eine XML-Arbeitsdatei handelt) geändert wird, nachdem das referenzierende SPS geöffnet wurde, erscheint eine Warnmeldung, in der Sie auf die Änderung aufmerksam gemacht werden und gefragt werden, ob das referenzierende SPS Modul mit den Änderungen aktualisiert werden soll.

Aktivieren/Deaktivieren des hinzugefügten Objekts

Alle Modulobjekte in allen hinzugefügten Modulen (egal, ob direkt oder indirekt hinzugefügte) werden zum referenzierenden SPS hinzugefügt und unter der entsprechenden Überschrift in der Design-Struktur aufgelistet: `Scripts`; `Globale Vorlagen`, `Design Fragmente`; `XSLT-Vorlagen` und `XPath-Funktionen`. Neben jedem dieser Objekte befindet sich ein Kontrollkästchen (*siehe Abbildung unten*), das Sie aktivieren oder deaktivieren können, um dieses Objekt zu aktivieren bzw. zu deaktivieren. Wenn ein Objekt deaktiviert wird, wird es aus dem SPS entfernt.



In der Abbildung oben werden alle im Modul `AddressBook.sps` verwendeten globalen Vorlagen unter der Überschrift "Globale Vorlagen" aufgelistet. Diejenigen, die über andere Module (direkt oder indirekt) hinzugefügt wurden, werden grau angezeigt. Die, die direkt in `AddressBook.sps` erstellt wurden, werden schwarz angezeigt. In der Abbildung sehen Sie, dass nur eine globale Vorlage `addr:Email` in `AddressBook.sps` selbst

erstellt wurde. Alle anderen globalen Vorlagen wurden über andere Module hinzugefügt und die Datei, in der diese Module jeweils definiert sind, wird neben ihrem Namen aufgelistet.

Beachten Sie, dass es für `addr:Email` zwei globale Vorlagen gibt: eine, die im referenzierenden SPS (`AddressBook.sps`) selbst erstellt wurde und die andere, die im hinzugefügten Modul `ContactPoints.sps` erstellt wurde. Wenn mehrere globale Vorlagen denselben (durch einen Namespace) erweiterten Namen haben, ist immer nur eine dieser Vorlagen aktiv. Durch Auswählen des entsprechenden Kontrollkästchens können Sie auswählen, welche dies sein soll. (Alternativ dazu können Sie die globale Vorlage über das Kontextmenü in der Design-Ansicht aktivieren.) Dieser Mechanismus ist nützlich, wenn Sie: (i) eine hinzugefügte globale Vorlage durch eine globale Vorlage außer Kraft setzen möchten, die Sie im referenzierenden SPS Modul erstellen, oder wenn Sie (ii) in einer Situation, in der in mehr als einem hinzugefügten Modul eine globale Vorlage für ein und dasselbe Element definiert wurde, angeben möchten, welche globale Vorlage verwendet werden soll.

Eine globale Vorlage, die im aktuellen SPS definiert wurde, kann gelöscht werden, indem Sie sie auswählen und auf die Schaltfläche **Entfernen** klicken. Globale Vorlagen, die in einem hinzugefügten Modul definiert wurden, können jedoch nicht aus dem referenzierenden SPS entfernt werden. Sie müssen durch Öffnen des hinzugefügten SPS und Entfernen der globalen Vorlage in diesem SPS entfernt werden.

Auch einzelne Skripts, Design-Fragmente und Seitenlayoutelemente können auf dieselbe Art aktiviert und deaktiviert werden.

Anwenden oder Verwenden von modularen Objekten

Ihre Vorlagen werden im referenzierenden SPS Modul[217] wie üblich erstellt. Jeder einzelne Typ von hinzugefügten Objekten wird unterschiedlich verwendet oder angewendet. Stellen Sie außerdem sicher, dass alle Modulobjekte, die Sie anwenden möchten, aktiviert wurden[223].

Globale Vorlagen

Wenn Sie eine globale Vorlage[231] aus einem der hinzugefügten SPS Module verwenden möchten, stellen Sie sicher, dass diese globale Vorlage auch wirklich angewendet wurde. Dies kann, je nachdem, welche für ihr Design verwendet werden muss, auf zwei Arten erfolgen:

- Definieren Sie in der Hauptvorlage, dass die Elementvorlage entweder die globale Vorlage für dieses Element verwendet oder diese globale Vorlage lokal kopiert. Diese beiden Befehle stehen im Kontextmenü, das erscheint, wenn Sie mit der rechten Maustaste auf dem Element-Tag im Design klicken, zur Verfügung.
- In der Hauptvorlage werden durch die Platzhalter "Inhalt" und "Restlicher Inhalt" Vorlagen angewendet, sodass die entsprechenden globalen Vorlagen verarbeitet werden.

Design-Fragmente

Um ein Design-Fragment zu verwenden, ziehen Sie es aus der Design-Struktur an die gewünschte Stelle in der Hauptvorlage oder der globalen Vorlage. Stellen Sie sicher, dass es sich dabei um den korrekten Kontext-Node für dieses Design-Fragment handelt. Nähere Informationen dazu finden Sie unter Design-Fragmente[242].

Skripts

All JavaScript-Funktionen (ob in einem hinzugefügten Modul oder ob sie im referenzierenden SPS erstellt wurden) stehen als Event Handler zur Verfügung und können über die Seitenleiste "Eigenschaften" für ein bestimmtes Event definiert werden[405].

© 2019-2025 Altova GmbH

6.4.3 Beispiel: Ein Adressbuch

Der Ordner (Eigene) Dateien 25 `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2025\StyleVisionExamples\Tutorials\ModularSPS` enthält Beispiele modularer SPSs. Die Beispieldateien in diesem Ordner enthalten ein Projekt, in dem ein Adressbuch, das Geschäftskontakte und persönliche Adressen enthält, in Module gegliedert wird. In diesem Beispiel wird nicht nur die Wirkungsweise der Modularisierung gezeigt, sondern Sie sehen auch die wichtigsten Gründe, warum ein SPS in Module aufgeteilt werden soll.

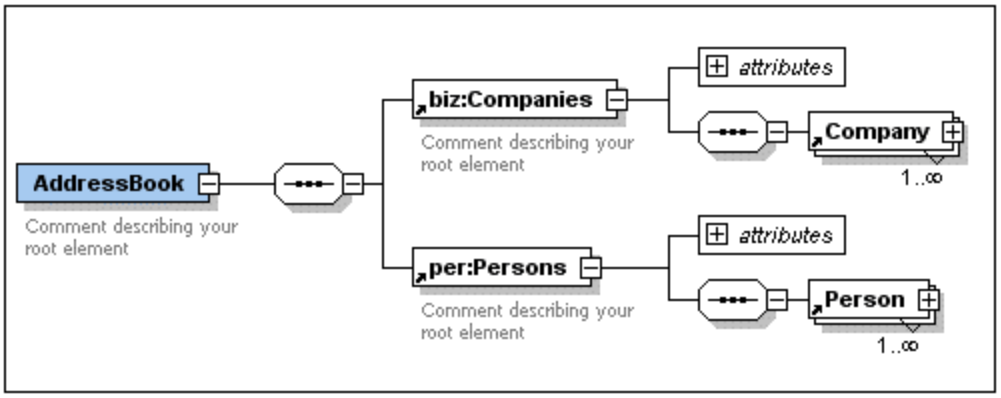
- Das komplette Adressbuch besteht aus zwei Modulen: (i) einem Geschäftsadressbuch und (ii) einem persönlichen Adressbuch. Jedes davon hat ein separates SPS, in dem unterschiedliche Designs definiert sind. Die beiden Module können zu einem einzigen Adressbuch kombiniert werden. In diesem Fall dient die Modularisierung dazu, Module zusammenstellen zu können: Die Module sind Komponenten einer größeren Einheit.
- Zwar unterscheiden sich die Content Models der einzelnen Module (Geschäftsadressbuch und persönliches Adressbuch) geringfügig voneinander, doch haben beide ein gemeinsames Modul, das Modul "ContactPoints", das aus den wichtigsten Adressbestandteilen besteht: Adresse, Telefonnummer, Fax und E-Mail. Das Modul "ContactPoints" kann daher in beiden Adressbüchern (Geschäftsadressen und Privatadressen) verwendet werden. In diesem Fall kann aufgrund der Modularisierung ein einziges Modul von mehreren anderen Einheiten gemeinsam genutzt werden.
- Das Modul "ContactPoints" kann zur größeren Flexibilität ebenfalls in Module aufgegliedert werden. Im Beispielprojekt haben wir ein separates Adressmodul für die Postadresse erstellt, welches eines von drei Content Models haben kann, je nachdem, ob es sich um eine Adresse in der EU, in den USA oder in einem anderen Land handelt. Die Ausgabe für alle drei Content Models wird in einem einzigen SPS definiert. Größere Flexibilität wäre gewährleistet worden, wenn diese Content Models in separaten SPSs definiert worden wären. In diesem Fall könnten die Module einfacher wiederverwendet werden.

Die Beschreibung des Projekts ist in die folgenden Teile gegliedert:

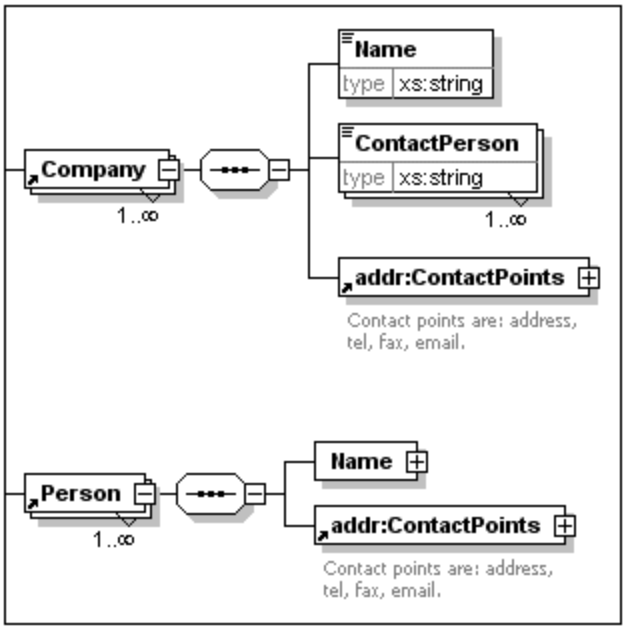
- die Schema-Dateien 225
- die XML-Datenquellen 227
- die SPS-Dateien 227

Die Schema-Dateien

Bei der Erstellung von Schemas für modulare SPSs muss unbedingt beachtet werden, dass die Elemente, die wiederverwendet werden sollen, als globale Elemente erstellt werden. Das Schema für das Adressbuch ist `AddressBook.xsd`. Dieses Schema wurde durch Import der Schemas für das Geschäftsadressbuch (`BusinessAddressBook.xsd`) und das persönliche Adressbuch (`PersonalAddressBook.xsd`) erstellt. Das Schema `BusinessAddressBook.xsd` enthält ein Content Model für Unternehmen. Das Schema `PersonalAddressBook.xsd` enthält ein Content Model für Personen. (*siehe Abbildung unten*)

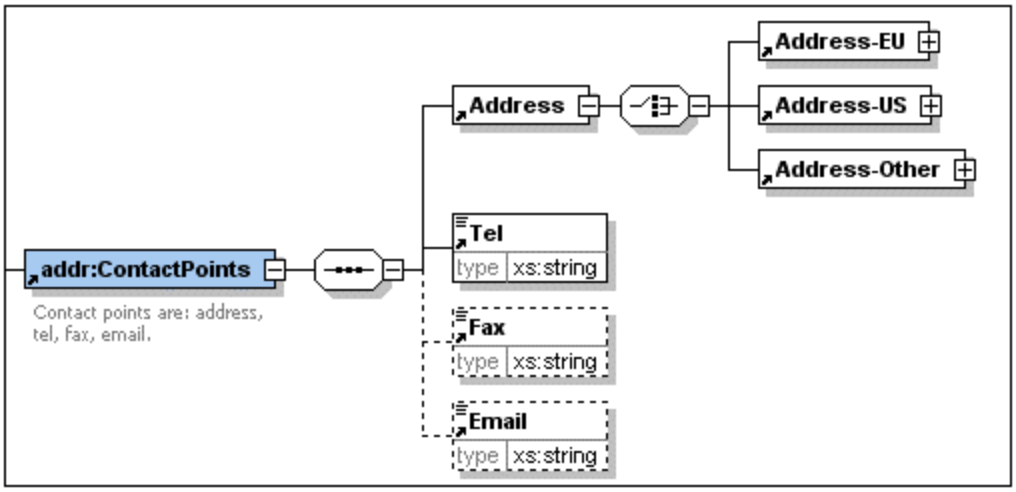


In beide Schemas wird das Schema `ContactPoints.xsd` importiert (*siehe Abbildung unten*), in dem ein Content Model für Adressdaten definiert ist.



Schließlich wird in das Schema `ContactPoints.xsd` (*siehe Abbildung unten*) das Schema `Address.xsd` inkludiert, in dem die drei Content Models für Adressarten definiert sind: EU, USA und andere Adressen.

© 2019-2025 Altova GmbH



Imports werden verwendet, wenn das importierte Schema zu einem anderen Namespace gehört als das importierende Schema. Includes werden verwendet, wenn das inkludierte Schema zum selben Namespace wie das inkludierenden Schema gehört.

Anmerkung: In den Abbildungen oben sehen Sie das Schema in der Schema-Ansicht von Altova XMLSpy.

Die XML-Datenquellen

Die XML-Daten sind in der Datei `AddressBook.xml` enthalten. Diese Datei ist so strukturiert, dass das Element `AddressBook` die Elemente `companies` und `persons` als Children enthält. Die Content Models dieser beiden Elemente sind in den Schemadateien `BusinessAddressBook.xsd` bzw. `PersonalAddressBook.xsd` definiert.

Zusätzlich gibt es für die beiden Schemas `BusinessAddressBook.xsd` und `PersonalAddressBook.xsd` die zwei XML-Dateien `BusinessAddress.xml` und `PersonalAddress.xml`, die als XML-Arbeitsdateien benutzt werden. Die XML-Arbeitsdatei für die entsprechenden SPS Module verwendet werden.

Die drei XML-Dateien sind die XML-Arbeitsdateien[21] für die folgenden SPS Module:

- `AddressBook.xml => AddressBook.sps, ContactPoints.sps, Address.sps`
- `BusinessAddressBook.xml => BusinessAddressBook.sps`
- `PersonalAddressBook.xml => PersonalAddressBook.sps`

Die SPS Module

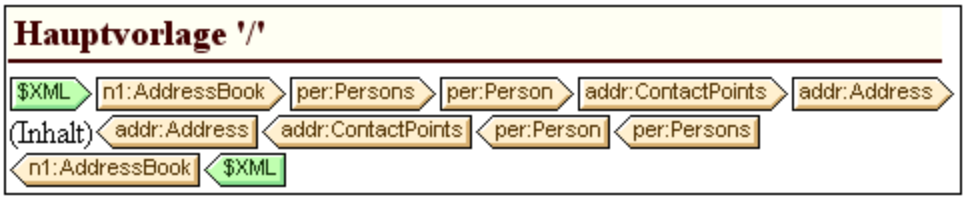
Die Beschreibung der SPS Module beginnt mit dem einfachsten Modul (`Address.sps`) und wird schrittweise bis zum vollständigen Adressbuch fortgesetzt (`AddressBook.sps`). In allen SPS Modulen wird `AddressBook.xsd` als Schema verwendet.

Address.sps
Die wichtigsten zu beachtenden Punkte sind die Verwendung des Schemas und der XML-Arbeitsdatei.

- `Address.sps` verwendet `AddressBook.xsd` als Schema. Genauso gut hätte aber auch `Address.xsd`, `ContactPoints.xsd`, `BusinessAddressBook.xsd` oder `PersonalAddressBook.xsd` verwendet werden können, da das Element `Address` in allen diesen Schemas vorhanden ist und als globales Element zur Verfügung steht. Wenn das SPS Modul zu einem anderen SPS Modul hinzugefügt wird, wird das

Schema des importierten Moduls ignoriert. Es ist daher beim Hinzufügen des SPS als Modul nicht wichtig, welches verwendet wird.

- Die XML-Arbeitsdatei ist `AddressBook.xml`. Beachten Sie, dass in der Hauptvorlage in `Address.sps` festgelegt ist, dass nur das Element `Address` verarbeitet werden soll und dass globale Vorlagen für `Address-EU`, `Address-US` und `Address-Other` definiert wurden.

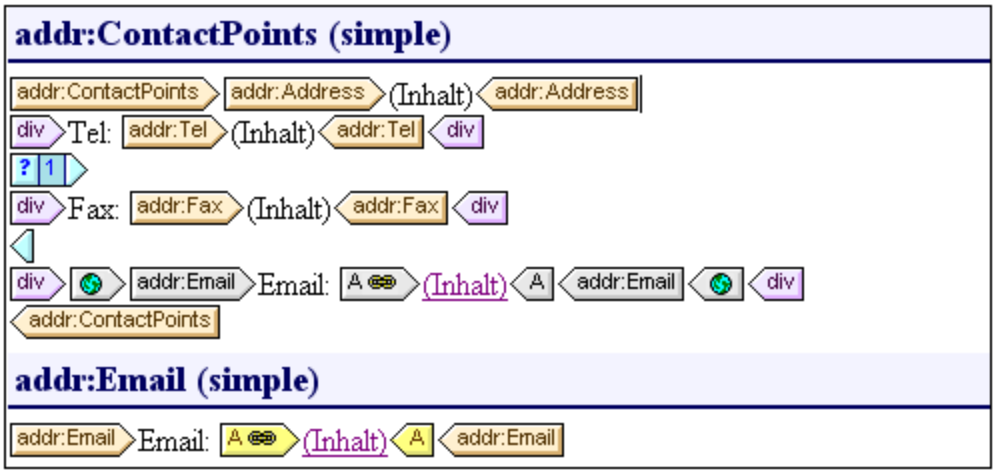


Da nur das Element `Address` verarbeitet wird, wird in den Ausgabevorschauen nur die Ausgabe von `Address` angezeigt. Wenn als Modul `Address.sps` verwendet wird, werden die globalen Vorlagen hinzugefügt und die Hauptvorlage wird ignoriert.

ContactPoints.sps

In dieses SPS wird ein Modul importiert. Beachten Sie die Verwendung von globalen Vorlagen in anderen globalen Vorlagen und der Hauptvorlage.

- `ContactPoints.sps` verwendet `AddressBook.xsd` als Schema und `AddressBook.xml` als XML-Arbeitsdatei.
- `Address.sps` wird als Modul hinzugefügt, sodass die globalen Vorlagen der Elemente `Address-EU`, `Address-US` und `Address-Other` zur Verfügung stehen.
- Es sind globale Vorlagen für die Elemente `ContactPoints` und `Email` definiert. Beachten Sie, dass in der `ContactPoints` Definition die globale Vorlage von `Email` verwendet wird (*Abbildung unten*).



- Die Hauptvorlage - die für die Vorschau benötigt wird - verwendet die globale Vorlage des Elements `ContactPoints` und ermöglicht dadurch die Vorschau auf die Ausgabe von `ContactPoints`.

BusinessAddressBook.sps und PersonalAddressBook.sps

Dieses SPS importieren jeweils ein Modul, das wiederum ein anderes importiert. Beachten Sie, dass die Hauptvorlage einfach globale Vorlagen anwendet.

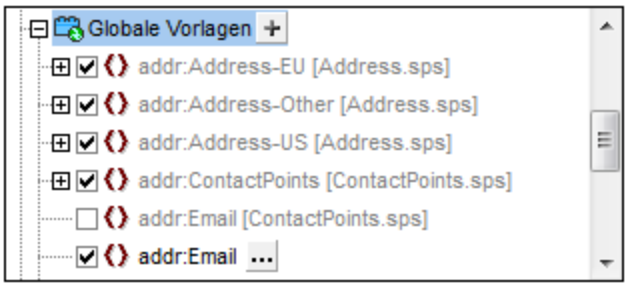
- Jedes dieser beiden Module verwendet `AddressBook.xsd` als Schema. Die XML-Arbeitsdateien sind `BusinessAddressBook.xml` bzw. `PersonalAddressBook.xml`.

© 2019-2025 Altova GmbH

- `ContactPoints.sps` wird als Modul hinzugefügt, wodurch `Address.sps` indirekt importiert wird. Alle globalen Vorlagen in diesen beiden Modulen stehen im referenzierenden SPS Modul zur Verfügung.
- In `BusinessAddressBook.sps` sind globale Vorlagen für die Elemente `Companies` und `Company` definiert. Beachten Sie, dass die `Company` Definition die globale Vorlage von `ContactPoints` verwendet.
- In Personal`AddressBook.sps` sind globale Vorlage für die Elemente `Person` und `Persons` definiert. Die `Person` Definition verwendet die globale Vorlage von `ContactPoints`.

AddressBook.sps
Für das Element `Email` gibt es zwei globale Vorlagen; jede davon kann aktiviert werden.

- `AddressBook.sps` verwendet `AddressBook.xsd` als Schema. Die XML-Arbeitsdatei ist `AddressBook.xml`.
- `BusinessAddressBook.sps` und `PersonalAddressBook.sps` werden als Module hinzugefügt, wodurch die Module `ContactPoints.sps` und `Address.sps` indirekt importiert werden.
- Für das Element `Email` ist eine globale Vorlage definiert. Dies bedeutet, dass es nun zwei globale Vorlagen für `Email` gibt, eine in `ContactPoints.sps` und die andere in `AddressBook.sps` (*siehe Abbildung unten*).



- In der Liste der globalen Vorlagen in der Design-Struktur (*Abbildung oben*) können Sie auswählen, welche der beiden globalen Vorlagen aktiv sein soll. Es kann immer nur eine davon aktiv sein. Die jeweils aktive Vorlage wird in der globalen Vorlage `ContactPoints` verwendet.
- Die Hauptvorlage enthält einigen statischen Inhalt für die Überschriften in der Ausgabe.

6.5 Vorlagen und Design-Fragmente

Das Design-Dokument besteht aus Vorlagen. Sie sollten die verschiedenen Vorlagenarten kennen, die verwendet werden können.

- *Hauptvorlagen und globale Vorlagen:* Das Design-Dokument besteht aus einer Hauptvorlage[23] und optional dazu einer oder mehreren globalen Vorlagen[231]. Globale Vorlagen können über die Hauptvorlage referenziert werden.
- *Node-Vorlagen und Variablen-Iteratoren:* Dabei handelt es sich um die Vorlagen, die die Hauptvorlage und die globalen Vorlagen bilden. Mit einer Node-Vorlage[239] wird ein Node in einer Schemaquelle ausgewählt.
- *Design-Fragmente:* Dabei handelt es sich um Vorlagen, die separat erstellt wurden und an verschiedenen Stellen des Designs (Hauptvorlage oder globale Vorlagen) wiederverwendet werden.

In diesem Abschnitt wird beschrieben, welche Rolle Vorlagen und Design-Fragmente in der Struktur des Designs spielen. Im Moment wollen wir uns nicht mit den Darstellungseigenschaften[327] im Design beschäftigen, sondern nur mit der Struktur.

Anmerkung:
Das SPS kann in der Design-Ansicht mehrere Vorlagen haben: die Hauptvorlage, globale Vorlagen und Design-Fragmente. Über die in Form von Symbolleistenschaltflächen[464] verfügbaren Vorlagenanzeigefilter[464] können Sie festlegen, welcher dieser Vorlagentypen in der Design-Ansicht angezeigt werden soll. Diese Anzeigefilter helfen Ihnen dabei, zwischen verschiedenen Anzeigearten Ihres SPS zu wechseln und die optimale Anzeigeart zu wählen.

6.5.1 Hauptvorlage

Die Hauptvorlage[23] legt die Struktur des Ausgabedokuments fest. D.h. die Reihenfolge des Layouts in der die Hauptvorlage entspricht der Reihenfolge des Layouts in der der Ausgabe. Im Entwicklerjargon wird dies als prozedurale Verarbeitung bezeichnet. Die Verarbeitung wird am Anfang der Vorlage begonnen und wird der Reihe nach bis zum Ende der Vorlage fortgesetzt. Während des Vorgangs werden Nodes aus dem XML-Dokument verarbeitet. Die Vorlagen, die zur Verarbeitung dieser Nodes verwendet werden, werden als lokale Vorlagen[22] bezeichnet. Nachdem eine lokale Vorlage verarbeitet wurde, wird zur nächsten Komponente in der Hauptvorlage übergegangen, usw. Gelegentlich kann ein Node zur Verarbeitung eine globale Vorlage[22] referenzieren. In diesen Fällen wird die Verabeitung nach Ausführung der globalen Vorlage an der Stelle im Hauptdokument fortgesetzt, von der aus die globale Vorlage aufgerufen wurde und die nächste Komponente wird verarbeitet.

Der Eintrittspunkt für die Hauptvorlage ist der Dokument-Node[22] desSchemas. StyleVision bietet auch die Option, mehrere Root-Elemente (Dokument-Elemente[22]) auszuwählen. Das bedeutet, dass es innerhalb der Hauptvorlage für jedes der aktiven Dokument-Elemente eine lokale Vorlage[22] geben kann. Die Vorlage, die bei der Verarbeitung ausgeführt wird, ist diejenige für das Element, welches das Dokument-Element des gerade verarbeiteten XML-Instanzdokuments ist.

© 2019-2025 Altova GmbH

6.5.2 Globale Vorlagen

Eine globale Vorlage[22] kann für jeden Node oder Typen im Schema oder für einen in einem XPath-Ausdruck definierten Node angelegt werden.

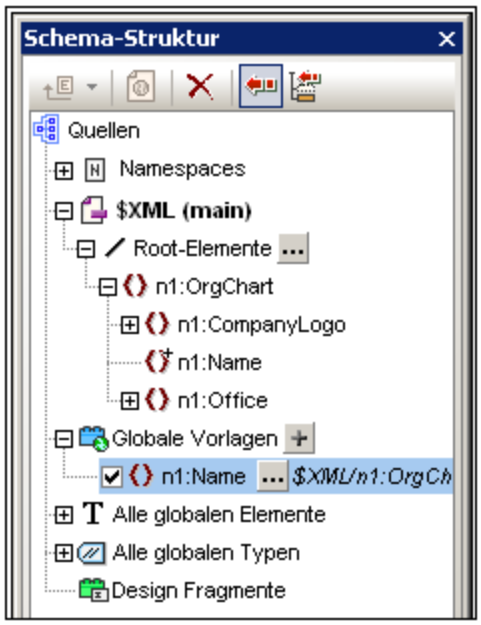
Eine globale Vorlage definiert Anweisungen für den ausgewählten Node oder Typ und wird über die Hauptvorlage[23], Design-Fragmente[242] oder andere globale Vorlagen aufgerufen. Das Verarbeitungsmodell ähnelt dem deklarativer Programmiersprachen, d.h. es wird eine einzige Vorlage definiert und mehrmals aufgerufen. Auf diese Art kann eine einzige Definition mehrmals wiederverwendet werden. Globale Vorlagen werden in zwei Situationen aufgerufen:

- wenn ein Node oder Typ in der Hauptvorlage[23] so definiert wurde, dass er die globale Vorlage referenziert (Indem Sie im Design mit der rechten Maustaste auf die Komponente klicken und den Befehl "Als globale Vorlage definieren" auswählen).
- wenn `(Inhalt)`[111] oder `(restlicher Inhalt)`[113] innerhalb eines Elements oder Typs in eine lokale Vorlage[22] eingefügt wird und der restliche Inhalt dieses Elements oder Typs einen Node oder Typ enthält, für den es eine globale Vorlage[22] gibt.

Globale Vorlagen sind nützlich, wenn ein Node (oder Typ) in verschiedenen Elementen oder an verschiedenen Stellen vorkommt und für alle Instanzen dieses Elements nur eine Gruppe von Anweisungen erforderlich ist. Angenommen, ein `para` Element muss immer gleich formatiert werden, egal ob es im Element `chapter`, `section`, `appendix` oder `blockquote` vorkommt. Eine effiziente Methode wäre, eine globale Vorlage für `para` zu definieren und in der Hauptvorlage[23] sicherzustellen, dass das `para` Element überall dort, wo notwendig, verarbeitet wird (z.B. durch Inkludieren von `//chapter/para` in die Hauptvorlage und Definieren, dass `para` seine globale Vorlage referenziert; oder durch Inkludieren von `//chapter/title` und anschließendes Inkludieren von `(Inhalt)`[111] oder `(restlicher Inhalt)`[113], so dass der restliche Inhalt des Elements `chapter` mit den verfügbaren globalen Vorlagen und Standardvorlagen verarbeitet wird.). Daher kann eine globale Vorlage auch für einen complexType (z.B. einen, der ein Adressmodell definiert) oder auch für einen simpleType (z.B. `xs:decimal`) definiert werden. In solchen Fällen werden alle Instanzen dieses Typs (complex oder simple), die die globale Vorlage für diesen Typ aufrufen, gemäß den in der globalen Vorlage definierten Regeln verarbeitet.

Erstellen einer globalen Vorlage

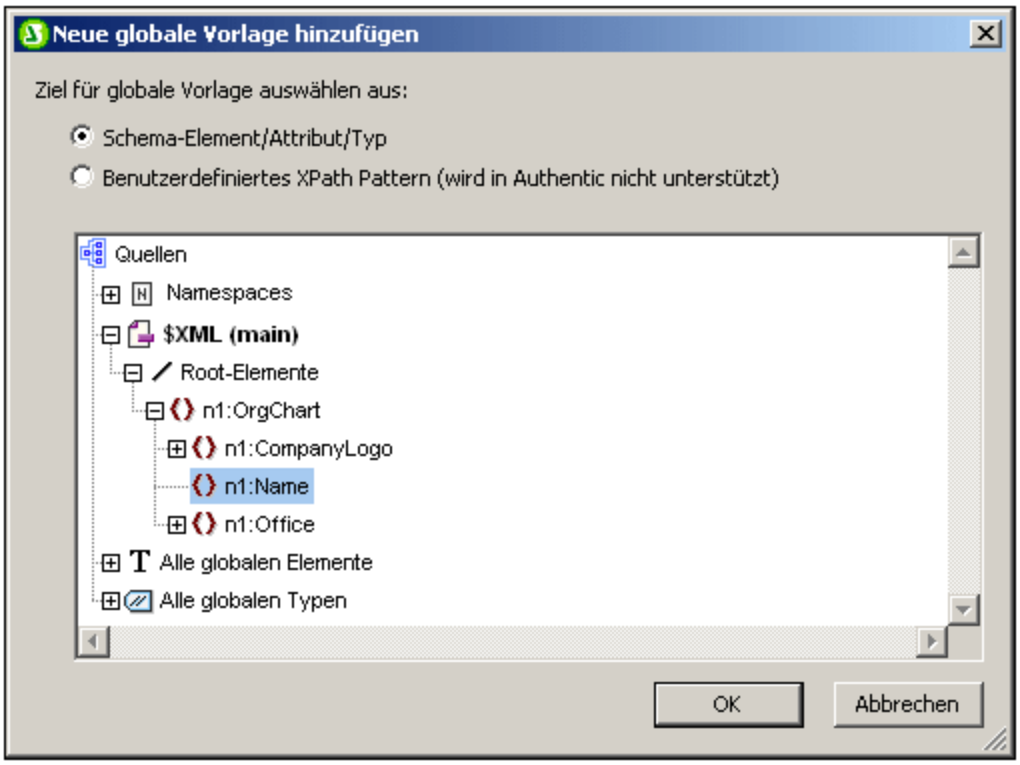
Globale Vorlagen können für jeden Node oder Typ im Schema oder für einen in einem XPath-Ausdruck definierten Node erstellt werden. Dies erfolgt über die Seitenleiste "Schema-Struktur" (*Abbildung unten*).



Eine globale Vorlage kann auf eine der folgenden Arten erstellt werden:

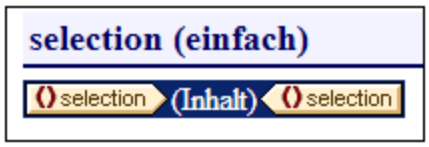
- Klicken Sie in der Schema-Struktur rechts vom Eintrag "Globale Vorlage" (*Siehe Abbildung oben*) auf die Schaltfläche "**Neue globale Vorlage hinzufügen**". Daraufhin wird das Dialogfeld "Neue globale Vorlage hinzufügen" (*Abbildung unten*) angezeigt. Sie können ein Element, ein Attribut oder einen Typ aus der im Dialog angezeigten Schema-Struktur auswählen oder Sie können einen XPath-Ausdruck eingeben. Dadurch wird der Node ausgewählt, der als globale Vorlage erstellt werden muss. Klicken Sie zum Fertigstellen auf **OK**. Die Vorlage wird erstellt und an die bereits vorhandenen Vorlagen in der Design-Ansicht angehängt. Sie kann nun bearbeitet werden. Der Schema-Node bzw. Schema-Typ wird in der Schema-Struktur durch ein vorangestelltes Pluszeichen gekennzeichnet.

© 2019-2025 Altova GmbH



- Rechtsklicken Sie in der Schema-Struktur (unter Root-Elemente, je nachdem unter Alle globalen Elemente oder alle globalen Typen) auf den Schema-Node oder die Typkomponente und wählen Sie den Befehl **Als global definieren/Globale Vorlage löschen**. Die Vorlage wird erstellt und an die bereits vorhandenen Vorlagen in der Design-Ansicht angehängt. Sie kann nun bearbeitet werden. Der Schema-Node bzw. Schema-Typ wird in der Schema-Struktur durch ein vorangestelltes Pluszeichen gekennzeichnet.
- Globale Vorlagen können in der Design-Ansicht auch anhand von Vorlagen in der Hauptvorlage erstellt werden. Klicken Sie mit der rechten Maustaste auf die Vorlage (entweder in der Design-Ansicht oder der Seitenleiste "Schema-Struktur") und wählen Sie den Befehl **Als globale Vorlage definieren**. Von der ausgewählten Vorlage wird eine globale Vorlage erstellt (sie wird an die Vorlagen in der Design-Ansicht angehängt) und in der Vorlage in der Hauptvorlage wird automatisch definiert, dass sie diese globale Vorlage **verwendet**. (Informationen zur Verwendung von globalen Vorlagen finden Sie weiter unten).

Eine globale Vorlage befindet sich in der Design-Ansicht unterhalb der Hauptvorlage. Sie wird durch eine lilafarbene Leiste mit dem Namen des Node, für den die globale Vorlage erstellt wurde, gefolgt von seinem Typ *(simple)* oder *(complex),* gekennzeichnet. In der Abbildung unten sehen Sie eine globale Vorlage.



Beachten Sie, dass die Verarbeitung der globalen Vorlage benutzerdefiniert ist und sowohl statische als auch dynamische Komponenten enthalten könnte und dass dafür alle für die Verarbeitung der Hauptvorlage verfügbaren Optionen zur Verfügung stehen.

Verwenden einer globalen Vorlage

Nachdem eine globale Vorlage erstellt wurde, kann sie verwendet werden, wenn ein Node mit demselben qualifizierten Namen in das Dokument eingefügt wird. (Wählen Sie, nachdem Sie den Node mit der Maus in das Design gezogen haben, im Menü, das daraufhin angezeigt wird, den Befehl **Globale Vorlage verwenden** aus). Wenn das Design eine lokale Vorlage enthält und es eine globale Vorlage desselben qualifizierten Namens für einen Node gibt, so kann alternativ dazu die globale Vorlage anstelle der lokalen verwendet werden. Um eine globale Vorlage für eine lokale Vorlage zu verwenden, klicken Sie mit der rechten Maustaste in der Design-Ansicht auf die lokale Vorlage und wählen Sie den Befehl **Globale Vorlage verwenden**. Bei Verwendung einer globalen Vorlage werden ihre Verarbeitungsanweisungen aufgerufen und von der lokalen Vorlage zur Laufzeit verwendet.

Immer wenn eine globale Vorlage im Design zum Einsatz kommt, kann an der globalen Vorlage ein XPath-Filter angewendet werden, um das Ziel-Nodeset zu filtern. Um einen solchen Filter zu erstellen, klicken Sie mit der rechten Maustaste im Design auf den Tag der globalen Vorlage und wählen Sie im Kontextmenü den Befehl **XPath-Filter bearbeiten** 240**.** Daraufhin wird das Dialogfeld "XPath-Filter bearbeiten" 440 angezeigt, in das Sie den gewünschten Ausdruck eingeben können.

Rekursive globale Vorlagen

Globale Vorlagen können rekursiv sein, das heißt, eine globale Vorlage kann sich selbst aufrufen. Um sich in Authentic aber gegen eine Endlosschleife zu schützen, kann eine Eigenschaft zur Einstellung der maximalen Aufruftiefe eingestellt werden. Diese Eigenschaft, die *Maximale Tiefe des Vorlagenaufrufs* steht auf dem Register "Authentic" im Dialogfeld "Eigenschaften" des SPS (**Datei | Eigenschaften** 489) zur Verfügung. Sie gibt die maximale Anzahl der Vorlagenaufrufe an, die rekursiv gemacht werden dürfen, wenn die Verarbeitung für die Authentic-Ausgabe läuft. Wenn die Anzahl der Vorlagenaufrufe die in der Eigenschaft *Maximale Tiefe des Vorlagenaufrufs* angegebene Zahl übersteigt, wird eine Fehlermeldung zurückgegeben.

Lokales Kopieren einer globalen Vorlage

Nachdem eine globale Vorlage erstellt wurde, können ihre Verarbeitungsanweisungen direkt in eine Vorlage mit demselben qualifizierten Namen in der Hauptvorlage kopiert werden. Klicken Sie dazu mit der rechten Maustaste auf die lokale Vorlage und wählen Sie den Befehl **Globale Vorlage lokal kopieren**. Der Unterschied zwischen dem lokalen Kopieren eine globalen Vorlage und der Verwendung der globalen Vorlage (zur Laufzeit) besteht darin, dass die Verarbeitungsanweisungen nur einmal kopiert werden. Die globale Vorlage hat keinen weiteren Einfluss auf die lokale Vorlage. Jeder der beiden Vorlagen bzw. beide können zu einem späteren Zeitpunkt unabhängig voneinander geändert werden, ohne dass dies einen Einfluss auf die andere hat. Wenn allerdings festgelegt wurde, dass eine globale Vorlage von einer lokalen Vorlagen (zur Laufzeit) *verwendet* werden soll, so wirken sich alle Änderungen an der globalen Vorlage zur Laufzeit in der lokalen Vorlage aus.

Aktivieren und Deaktivieren von globalen Vorlagen

Eine globale Vorlage kann in der Schema-Struktur durch Aktivieren des entsprechenden Eintrags in der Liste der globalen Vorlagen aktiviert werden. Durch erneutes Anklicken des Eintrags wird sie deaktiviert. Wenn eine globale Vorlage aktiviert wurde (dies ist die Standardeinstellung, wenn eine globale Vorlage erstellt wurde), wird sie im XSLT Stylesheet generiert. Wenn sie deaktiviert wurde, wird sie nicht im XSLT Stylesheet generiert, bleibt aber weiterhin im SPS Design gespeichert.

Alle lokalen Vorlagen, die eine deaktivierte globale Vorlage verwenden, greifen - da sie die fehlende globale Vorlage nicht referenzieren können -auf die Standardvorlagen von XSLT zurück, wodurch der Inhalt von untergeordneten Text-Nodes ausgegeben wird.

© 2019-2025 Altova GmbH

Die Vorteile der Aktivierung/Deaktivierung sind: (i) Globale Vorlagen müssen nicht gelöscht werden, wenn sie vorübergehend nicht benötigt werden; sie können zu einem späteren Zeitpunkt wieder aktiviert werden; (ii) Wenn es zu Namenskonflikten mit Vorlagen aus importierten Stylesheets kommt, so kann die nicht benötigte globale Vorlage vorübergehend deaktiviert werden.

Entfernen einer globalen Vorlage

Um eine globale Vorlage zu entfernen, klicken Sie mit der rechten Maustaste entweder in der Design-Ansicht oder in der Seitenleiste "Schema-Struktur" auf die gewünschte globale Vorlage und wählen Sie den Befehl **Als global definieren / Globale Vorlage löschen.**

Einfache globale Vorlagen und komplexe globale Vorlagen

Es gibt zwei Arten von globalen Vorlagen: einfache und komplexe globale Vorlagen. Komplexe globale Vorlagen stehen aus Gründen der Rückwärtskompatibilität zur Verfügung. Wenn eine globale Vorlage in einem SPS, das mit einer StyleVision Version vor Version 2006 erstellt wurde, eine Tabelle oder Liste enthält, so wird diese globale Vorlage normalerweise in StyleVision 2006 und späteren Versionen als komplexe globale Vorlage geöffnet.

Ein komplexe globale Vorlage unterscheidet sich von einer einfachen globalen Vorlage in der Art, wie der Node, für den die globale Vorlage erstellt wurde, verarbeitet wird. Sobald die erste Instanz des Node im Dokument gefunden wird, verarbeitet die komplexe globale Vorlage unmittelbar danach alle nachfolgenden Instanzen dieses Node. Eine einfache globale Vorlage verarbeitet dagegen jede Node-Instanz erst dort, wo diese Node-Instanz vorkommt.

Beachten Sie bitte: Eine einfache globale Vorlage wird automatisch in eine komplexe globale Vorlage konvertiert, wenn **rund um** den Element-Node, für den die globale Vorlage erstellt wurde, ein vordefiniertes Format[328] oder eine neue Zeile erstellt wird. Dies führt dazu, dass die Verarbeitung für komplexe globale Vorlagen (im vorhergehenden Listeneintrag beschrieben) durchgeführt wird. Um wieder eine einfache globale Vorlage zu verwenden, muss das vordefinierte Format[328] entfernt werden (indem Sie den Node aus dem vordefinierten Format herausziehen und das vordefinierte Format dann löschen), bzw. die neue Zeile muss entfernt werden (durch Löschen des Elements in der Seitenleiste Designstruktur[41]). Um die automatische Konvertierung von einfachen globalen Vorlagen in komplexe globale Vorlagen zu vermeiden, stellen Sie sicher, dass das vordefinierte Format[328] bzw. die neue Zeile innerhalb der Node Tags des Elements hinzugefügt wird, für welches die einfache globale Vorlage erstellt wurde.

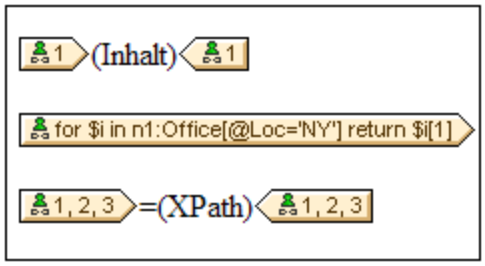
Globale Vorlagen in modularen SPSs

Wenn ein SPS Modul zu einem anderen SPS Modul hinzugefügt wird[216], stehen die globalen Vorlagen im hinzugefügten Modul dem referenzierenden SPS zur Verwendung zur Verfügung. Nähere Informationen zur Verwendung modularer SPSs finden Sie im Abschnitt Modulare SPSs[216].

6.5.3 Benutzerdefinierte Vorlagen

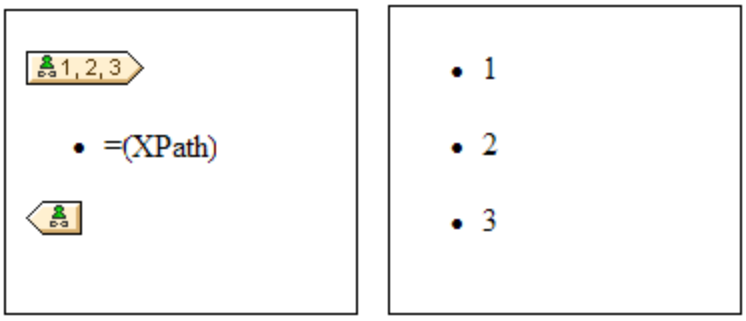
Benutzerdefinierte Vorlagen sind Vorlagen für Elemente, die durch einen von Ihnen definierten XPath-Ausdruck generiert wurden. Dabei kann es sich um atomare Werte oder Nodes handeln. Beachten Sie in der Abbildung unten, in der Sie drei benutzerdefinierte Vorlagen sehen, das Symbol für eine benutzerdefinierte Vorlage (grüne Figur) auf der linken Seite der Tags. Benutzerdefinierte Vorlagen sind sehr nützlich, da sie beim Erstellen von

Vorlagen besonders viel Flexibilität bieten.



Der XPath-Ausdruck in den drei in der obigen Abbildung gezeigten benutzerdefinierten Vorlagen bewirkt Folgendes:

- Ein Node in einem Quellschema wird ausgewählt. Mit Hilfe eines XPath-Ausdrucks kann jeder beliebige Node in jeder beliebigen Schemaquelle von jedem Kontext-Node aus ausgewählt werden. Wenn es einen eindeutigen Node gibt, der von StyleVision ausgewählt werden kann, so wird die Vorlage automatisch von einer benutzerdefinierten Vorlage in eine normale Vorlage umgewandelt. Wenn es sich um eine benutzerdefinierte Vorlage handelt, so wird dies durch ein grünes Symbol für eine benutzerdefinierte Vorlage am linken Rand des Vorlagen-Tags gekennzeichnet.
- Ein Node wird ausgewählt, der eine durch das `for` Konstrukt von XPath 2.0 und 3.0 definierte Bedingung erfüllt. Solche Vorlage können niemals zu normalen Vorlagen aufgelöst werden (sondern bleiben benutzerdefinierte Vorlagen), da das `for` Konstrukt es nicht zulässt, dass StyleVision das Ziel anhand der derzeit verfügbaren Schemainformationen eindeutig aufgelöst wird.
- Eine Sequenz atomarer Werte `{1, 2, 3}` wird ausgewählt. Es ist zwar zulässig, eine Vorlage für einen atomaren Wert zu erstellen, doch kann in einer solchen Vorlage der Platzhalter `Inhalt` nicht verwendet werden. Der Grund hierfür ist, dass die Anweisung `xsl:apply-templates` (also das, was der Platzhalter `Inhalt` generiert) nur auf Node-Elemente (nicht aber auf atomare Werte) angewendet werden kann. Sie könnten jedoch eine automatische Berechnung in Kombination mit einem Design-Element wie z.B. einer Liste, verwenden. So würde die benutzerdefinierte Vorlage in der Abbildung auf der linken Seite die Ausgabe auf der rechten Seite generieren.



Anmerkung: Wenn im SPS XSLT 1.0 verwendet wird, so muss der von Ihnen eingegebene XPath-Ausdruck ein Node-Set zurückgeben, da sonst ein Fehler gemeldet wird.

Vorteil der Verwendung von XPath zur Auswahl eines Vorlagen-Node

Der Vorteil der Auswahl eines Schema-Node mittels eines XPath-Ausdrucks (benutzerdefinierte Vorlagen) ist, dass man mit Hilfe des XPath-Selektors jeden beliebigen Node, bzw. jede beliebige Elementsequenz auswählen kann und Filter und Bedingungen bei der Node-Auswahl einsetzen kann. Dadurch können

 © 2019-2025 Altova GmbH

spezifische XML-Dokument-Nodes für jede angegebene Vorlage adressiert werden. So wählt z.B. der XPath-Ausdruck `//Office/Department[@Location="NY"]` nur diejenigen `Department` Nodes aus, die ein `Location` Attribut mit dem Wert `NY` haben. Siehe auch die anderen Beispiele in diesem Abschnitt.

Anmerkung: Wenn ein XPath-Ausdruck mehrere location path-Schritte enthält, dann macht es - v.a. beim Gruppieren und Sortieren - einen Unterschied, ob die verschiedenen location path-Schritte in Klammern gesetzt werden oder nicht. So wird der XPath-Ausdruck `/Org/Office/Dept` z.B. anders verarbeitet als `(/Org/Office/Dept)`. Beim ersten Ausdruck (ohne Klammern) führt der Prozessor eine Schleifenverarbeitung der einzelnen location-Schritte durch. Beim zweiten Ausdruck (mit Klammern) werden alle `Dept` Elemente aller `Office` Elemente in einem einzigen Nodeset zurückgegeben.

Klammern	Zugrunde liegende XSLT-Methode	Ergebnis

Nein	`<xsl:for-each select="Org">`
	`  <xsl:for-each select="Office">`
	`  <xsl:for-each select="Dept">`
	`  ...`
	`  </xsl:for-each>`
	`  </xsl:for-each>`
	`</xsl:for-each>`
		Jedes `Office` Element hat seine eigene `Dept` Population. Dadurch kann die Gruppierung und Sortierung innerhalb der einzelnen `Office`Elemente durchgeführt werden.

Ja	`<xsl:for-each`
	`select="/Org/Office/Dept">`
	`  ...`
	`</xsl:for-each>`
		Die `Dept` Population umfasst alle `Office` Elements und alle `Org` Elemente.

Dieser Unterschied in der Auswertung von XPath-Ausdrücken kann sich auf die Gruppierung und Sortierung auswirken.

Einfügen einer benutzerdefinierten Vorlage

Um eine benutzerdefinierte Vorlage einzufügen, gehen Sie folgendermaßen vor:

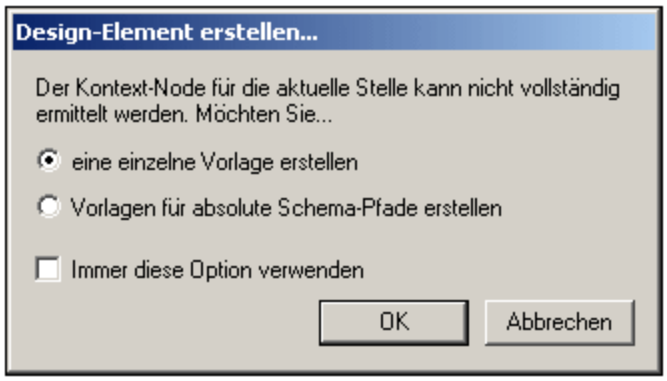
1. Klicken Sie in der Symbolleiste "Design-Elemente" auf die Schaltfläche **Benutzerdefinierte Vorlage einfügen** und klicken Sie anschließend an die Stelle im Design, an der die Vorlage eingefügt werden soll. Als Alternative können Sie auch mit der rechten Maustaste an die gewünschte Stelle im Design klicken und aus dem Kontextmenü anschließend den Befehl **Benutzerdefinierte Vorlage einfügen** auswählen.
2. Geben Sie im Dialogfeld XPath-Ausdruck bearbeiten 440, das nun angezeigt wird, den gewünschten XPath-Ausdruck ein und klicken Sie auf **OK**. Beachten Sie, dass der Kontext-Node des XPath-Ausdrucks derjenige Node sein wird, in den Sie hineingeklickt haben. Daraufhin wird eine leere Node-Vorlage erstellt. Manchmal wird ein verbundener Node erstellt. Wenn ein Node verbunden wird, werden die Ziel-Instanznodes so ausgewählt, als befänden sie sich auf einer einzigen Ebene, während die Node-Auswahl bei nicht verbundenen Nodes (d.h. wenn er auf mehrere hierarchische Ebenen aufgeteilt ist) durch Schleifenverarbeitung der einzelnen Instanz-Nodes auf den einzelnen Ebenen erfolgt. Das zurückgegebene Nodeset ist in beiden Fällen (bei verbundenen und getrennten Nodes) dasselbe, es sei denn ein Gruppierungs- oder Sortierkriterium wurde definiert. Nähere Informationen darüber, wie sich verbundene Nodes auf die Gruppierung und Sortierung auswirken, finden Sie unter Node-Vorlagen-Operationen 239.

Bearbeiten einer Vorlagenauswahl

Die Node-Auswahl jeder Node-Vorlage (benutzerdefiniert oder normal) kann über einen XPath-Ausdruck geändert werden, sodass ein neuer Suchausdruck ausgewählt wird. Um die Vorlagenauswahl einer Node-Vorlage zu bearbeiten, klicken Sie mit der rechten Maustaste auf die Node-Vorlage und wählen Sie den Befehl **Vorlagenauswahl ändern**. Daraufhin wird das Dialogfeld "XPath-Ausdruck bearbeiten" angezeigt, in dem Sie den XPath-Ausdruck zur Auswahl des neuen Node eingeben können. Klicken Sie anschließend auf **OK**.

Hinzufügen von Nodes zu benutzerdefinierten Vorlagen

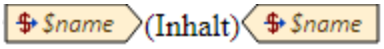
Wenn ein Node aus der Schemastruktur zu einer benutzerdefinierten Vorlage hinzugefügt wird, ist der Kontext für den neuen Node unbekannt, wenn die benutzerdefinierte Vorlage für einen Node oder eine Sequenz erstellt wurde, der/die nicht in den Kontext der Schemaquelle des SPS platziert werden kann. Sie werden daher gefragt (*Abbildung unten*), wie der neue Node referenziert werden soll: (i) durch seinen Namen (im Wesentlichen ein relativer Pfad) oder (ii) durch einen vollständigen Pfad von der Root der Schemaquelle aus.



Standardmäßig werden Sie gefragt, wie der Node referenziert werden soll. Sie können dieses Standardverhalten auf dem Register "Design" des Dialogfelds Extras | Optionen[552] ändern.

6.5.4 Variablenvorlagen

Eine **Variablenvorlage** ist eine Vorlage für eine Variable, die standardmäßig den Inhalt der Variablen ausgibt. Diese Vorlage wird mit dem Befehl **Einfügen | Variablenvorlage** oder **Einschließen in | Variable** eingefügt. Daraufhin wird an der Cursorposition eine Vorlage für eine im SPS definierte Variable eingefügt. Die Variablenvorlage (*Abbildung unten*) enthält standardmäßig einen Platzhalter `Inhalt` zur Ausgabe des Inhalts der Variablen. Sie können je nach Wunsch zusätzlichen Inhalt (sowohl statischen als auch dynamischen) in die Variablenvorlage einfügen oder die Vorlage wie jede andere Vorlage bearbeiten. Eine Variablenvorlage wird durch ein Dollarsymbol im Start- und End-Tag gekennzeichnet.

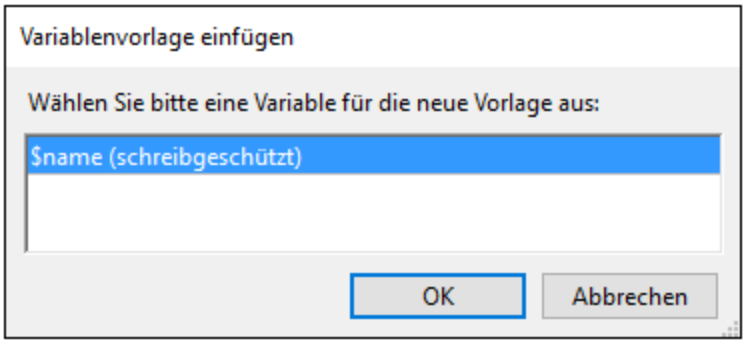


Um eine Variablenvorlage einzufügen, gehen Sie folgendermaßen vor:

1. Platzieren Sie den Cursor an die Stelle im Design, an der die Vorlage eingefügt werden soll.
2. Klicken Sie auf die rechten Maustaste und wählen Sie den Befehl **Einfügen | Variablenvorlage**.

© 2019-2025 Altova GmbH

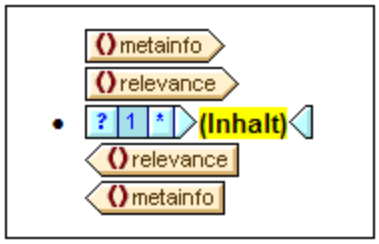
Daraufhin wird das Dialogfeld "Variablenvorlage einfügen" angezeigt (*Abbildung unten*).



3. Das Dialogfeld enthält eine Liste aller im SPS definierten benutzerdefinierten Parameter und Variablen[282]. Wählen Sie die Variable aus, für die Sie eine Variablenvorlage hinzufügen möchten.
4. Klicken Sie zum Fertigstellen auf **OK**.

6.5.5 Node-Vorlagenoperationen

Eine Node-Vorlage ist eine Vorlage im Design, in der die Verarbeitung für einen Node definiert ist. Node-Vorlagen werden im Design mit beigen Start- und End-Tags (*Abbildung unten*) angezeigt. Die Art des Node wird durch ein Symbol innerhalb der Tags angezeigt. (Z.B.: eckige Klammern für Element-Nodes und Ist-gleich-Zeichen für Attribut-Nodes). In der Abbildung unten sehen Sie zwei Node-Vorlagen für Elemente: `metainfo` und `relevance`. *Siehe auch* *Nodes im XML-Dokument*[435].



Die Operationen, die an einer Node-Vorlage ausgeführt werden können, können über das Kontextmenü dieser Node-Vorlage (durch Rechtsklick auf den Start- oder End-Tag einer Node-Vorlage) aufgerufen werden.

Die Befehle im Kontextmenü werden unten beschrieben:

- Globale Vorlagen[240]
- Vorlagenauswahl[240]
- XPath-Filter[240]
- Gruppieren nach, Sortieren nach, Variablen definieren, Vorlage dient als Ebene[241]
- Design Fragment erstellen[241]
- Nur Tag entfernen[241]
- Bearbeiten, Einschließen in, Ändern in, Authentic-Eigenschaften[242]

Diese Menübefehle sind im Folgenden beschrieben. Beachten Sie, dass einige Befehle für bestimmte Node-Vorlagen manchmal nicht zur Verfügung stehen; im Kontextmenü erscheinen diese ausgegraut.

Globale Vorlagen: definieren, verwenden, lokal kopieren

Eine Node-Vorlage in der Hauptvorlage kann über die folgenden Befehle geändert werden oder mit einer globalen Vorlage verknüpft werden:

- *Als globale Vorlage definieren:* Diese Option steht zur Verfügung, wenn die Node-Vorlage ein Element repräsentiert, das im Schema als globales Element definiert ist. Von der Node-Vorlage wird eine globale Vorlage erstellt. Die Node-Vorlage in der Hauptvorlage verwendet dann diese globale Vorlage und die Tags werden anschließend grau angezeigt (Kennzeichnung der Verwendung als globale Vorlage).
- *Globale Vorlage verwenden:* Wenn eine globale Vorlage desselben qualifizierten Namens wie die Node-Vorlage definiert wurde, verwendet die Node-Vorlage die Verarbeitung der globalen Vorlage. Die Tags der Node-Vorlage werden dann grau.
- *Globale Vorlage lokal kopieren:* Die Verarbeitungsanweisungen einer globalen Vorlage, die denselben qualifizierten Namen hat, wie die Node-Vorlage, werden physisch in die Node-Vorlage kopiert. Die Node-Vorlage ist unabhängig von der globalen Vorlage. Anschließend können beide - sowohl die Node-Vorlage als auch die globale Vorlage - unabhängig voneinander bearbeitet werden. Da die Node-Vorlage keine globale Vorlage referenziert, behält Sie ihre beige Farbe.

Nähere Informationen dazu finden Sie unter Globale Vorlagen 231.

Bearbeiten der Vorlagenauswahl

Mit Hilfe dieses Befehls kann ein Node, für den eine Vorlage erstellt wurde, geändert werden. Mit dem Befehl "Vorlagenauswahl ändern" wird das Dialogfeld "XPath-Ausdruck bearbeiten" 440 aufgerufen, in dem Sie einen XPath-Ausdruck eingeben können, der einen anderen Node im Schema auswählt. Sie können auch einen beliebigen XPath-Ausdruck eingeben, der die Vorlage in eine benutzerdefinierte Vorlage 120 ändert.

XPath-Filter bearbeiten/löschen

Mit Hilfe eines XPath-Filters können Sie das Nodeset filtern, auf das eine Node-Vorlage angewendet wird. XPath-Filter können auch an globalen Vorlagen 231 angewendet werden.

Standardmäßig wird eine Node-Vorlage auf Nodes (Elemente oder Attribute) angewendet, die dem Node entsprechen, für den die Node-Vorlage erstellt wurde (die denselben Namen haben und in der Schemahierarchie an derselben Stelle vorkommen). So würde beispielsweise eine Node-Vorlage für den Node `/Personnel/Office` alle `/Personnel/Office` Elemente auswählen. Wenn jetzt am `Office` Element (durch Rechtsklick auf das `Office` Element und Bearbeitung seines XPath-Filters) ein XPath-Filter mit dem Ausdruck `1` erstellt wird, so erhält das `Office` Element einen Prädikat-Ausdruck, so dass der gesamte XPath-Ausdruck `/Personnel/Office[1]` lauten würde. Mit diesem XPath-Ausdruck wird das erste `Office` Child des Elements `Personnel` ausgewählt, mit dem Resultat, dass die anderen `Office` Elemente herausgefiltert werden.

Ein Filter im Design kann zu jeder beliebigen Node-Vorlage und zu mehreren Node-Vorlagen hinzugefügt werden. Auf diese Art können Sie eine Auswahl treffen, die XPath-Ausdrücken wie z.B.: `/Personnel/Office[@country='US']/Person[Title='Manager']` entspricht, um alle Manager in den amerikanischen Büros des Unternehmens auszuwählen. In diesem Beispiel wurde jeweils ein Filter für die Node-Vorlage `Office` und `Person` erstellt.

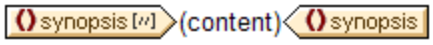
Überall dort, wo eine globale Vorlage verwendet, also aufgerufen wird, kann ein XPath-Filter auf diese Vorlage angewendet werden. So kann z.B. für jede Instanz einer verwendeten globalen Vorlage ein XPath-Filter auf die globale Vorlage angewendet werden, um das Ziel-Nodeset einzuschränken.

© 2019-2025 Altova GmbH

Um einen XPath-Filter zu einer Node-Vorlage hinzuzufügen, klicken Sie auf die Node-Vorlage und wählen Sie den Befehl **XPath-Filter bearbeiten**. Geben Sie den XPath-Filterausdruck ohne Anführungszeichen, eckige Klammern oder Trennzeichen ein. Es kann jeder beliebige gültige XPath-Ausdruck eingegeben werden, z.B:

- `1`
- `@country='US'`
- `Title='Manager'`

Nachdem für eine Node-Vorlage ein XPath-Filter erstellt wurde, wird dies im Start-Tag der Node-Vorlage durch ein Filtersymbol gekennzeichnet. In der Abbildung unten hat die Node-Vorlage `synopsis` einen Filter.



Anmerkung: Jede Node-Vorlage unterstützt einen XPath-Filter.

Gruppieren noch, Sortieren noch, Definieren von Variablen, Vorlage dient als Ebene

Die Funktionweise dieser Befehle wird in den entsprechenden Abschnitten näher beschrieben:

- Mit dem Befehl **Gruppieren nach** können Instanzen des durch die ausgewählte Node-Vorlage dargestellten Node gruppiert werden. Die Gruppierungsfunktion wird im Abschnitt Gruppieren[267] näher beschrieben..
- Mit dem Befehl **Sortieren nach** können Instanzen des durch die ausgewählte Node-Vorlage dargestellten Node sortiert werden. Die Sortierfunktion wird im Abschnitt Sortieren[276] näher beschrieben.
- Mit dem Befehl **Variablen definieren** können Sie Variablen definieren, die in der ausgewählten Node-Vorlage im Geltungsbereich sind. Eine Beschreibung zum Arbeiten mit Variablen finden Sie im Abschnitt Variablen[287].
- Der Befehl **Vorlage dient als Ebene** ist ein Ein-/Aus-Befehl, der eine Ebene in der Node-Vorlage erstellt/entfernt. Ebenen können auf verschiedenen Stufen erstellt werden, um das Dokument hierarchisch zu gliedern. Anhand dieser Struktur können anschließend ein Inhaltsverzeichnis, automatische Nummerierung und Textreferenzen generiert werden. Diese Funktionen werden im Abschnitt Inhaltsverzeichnis und Referenzierung[290] näher beschrieben.

Design Fragment erstellen

Erstellt anhand einer ausgewählten Vorlage eine Design Fragment-Vorlage. Die erzeugte Design Fragment-Vorlage wird am unteren Rand des Designs sowie in der Design-Struktur und der Schema-Struktur zu den Design-Fragment-Vorlagen hinzugefügt. Das Design Fragment wird auch an der Stelle im Design angewendet, an der es erstellt wurde.

Nur (Vorlagen- oder Formatierungs-) Tag entfernen

Mit diesem Befehl wird nur der Tag für die ausgewählte Vorlage oder Formatierung entfernt. Die untergeordneten Nodes oder Formatierungs-Tags werden nicht entfernt. Dieser Befehl eignet sich dazu, einen Formatierungstag oder den Tag eines übergeordneten Elements zu entfernen, ohne dabei den gesamten Inhalt des Tags zu entfernen (was bei Anwendung des Befehls **Löschen** an einem ausgewählten Tag der Fall wäre). Beachten Sie allerdings, dass bei Entfernung eines übergeordneten Elements unter Umständen dessen untergeordnete Nodes ungültig werden könnten. In diesem Fall werden ungültige Nodes rot durchgestrichen angezeigt.

Bearbeiten, Einschließen in, Ändern in

Diese Befehle werden im Folgenden beschrieben:

- *Bearbeiten:* Daraufhin wird ein Untermenü mit den bekannten Windows-Befehlen angezeigt: Ausschneiden, Kopieren, Einfügen und Löschen.
- *Einschließen in:* Die Node-Vorlage kann in die folgenden Design-Komponenten, die in separaten Abschnitten dieser Dokumentation beschrieben werden, eingeschlossen werden: Absatz 111, Spezieller Absatz. 113 Nummerierung und Aufzählungszeichen 147, Hyperlink 320, Bedingung 262, Inhaltsverz.-Markierung und Ebene 290.
- *Ändern in:* Mit Hilfe der Funktion "Ändern in" können Sie (i) den Node wechseln, auf den diese Vorlage angewendet wird oder (ii) ändern wie der Node im Design erstellt wird. Näheres dazu finden Sie im Abschnitt Die Funktion "Ändern in" 181

6.5.6 Design-Fragmente

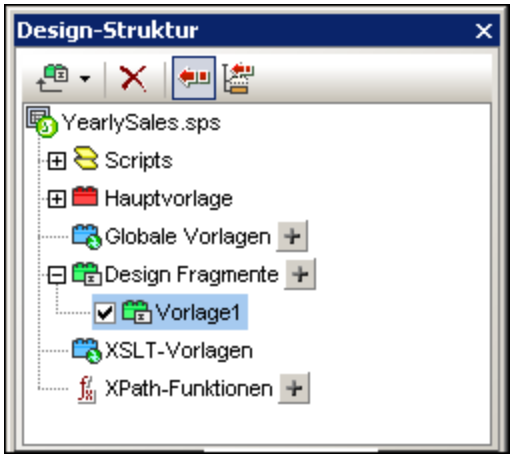
Design-Fragmente dienen zum Erstellen von Komponenten, die an verschiedenen Stellen im Dokument wiederverwendet werden können, ähnlich wie Funktionen, die wiederverwendet werden können. Die Vorgangsweise ist die folgende:

1. Erstellen Sie das Design-Fragment im Design 242.
2. Erstellen Sie den Inhalt des Design-Fragments 243.
3. Fügen Sie das Design-Fragment an der gewünschten Stelle in einer Vorlage ein 244

Erstellen eines Design-Fragments

So erstellen Sie ein Design-Fragment:

1. Klicken Sie in der Design-Struktur oder der Schema-Struktur auf die Schaltfläche "Neues Design-Fragment hinzufügen" , das sich in der Struktur rechts neben dem Element "Design-Fragmente" befindet (*siehe Abbildung unten*). Daraufhin wird in der Struktur in der Liste der Design-Fragmente ein Design-Fragment hinzugefügt (*Siehe auch Anmerkung unten*).

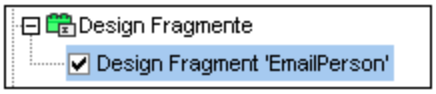


Beachten Sie, dass im SPS-Design eine Design-Fragmentvorlage erstellt wird. Diese Vorlage wird an die bereits im Design vorhandenen Vorlagen angehängt und mit einem grünen Symbol versehen.

© 2019-2025 Altova GmbH

(Wenn Sie nur die Design-Fragmente sehen möchten, die im Design enthalten sind, blenden Sie die Hauptvorlage und die globalen Vorlagen aus. Klicken Sie dazu in der Vorlagenfilter[464]-Symbolleiste von StyleVision auf die entsprechenden Schaltflächen zum Ein-/Ausblenden[464].) Außerdem werden die Design-Fragment-Vorlagen auch in der Schema-Struktur aufgelistet, um jederzeit aufgerufen werden zu können.

2. Doppelklicken Sie (entweder in der Design- oder der Schema-Struktur) auf den Design-Fragment-Eintrag, um den Namen zu bearbeiten. Benennen Sie das Design-Fragment wie gewünscht und drücken Sie die **Eingabetaste**. Der bearbeitete Name wird in die Design-Struktur (*Abbildung unten*) und in die Vorlage im Design eingegeben.



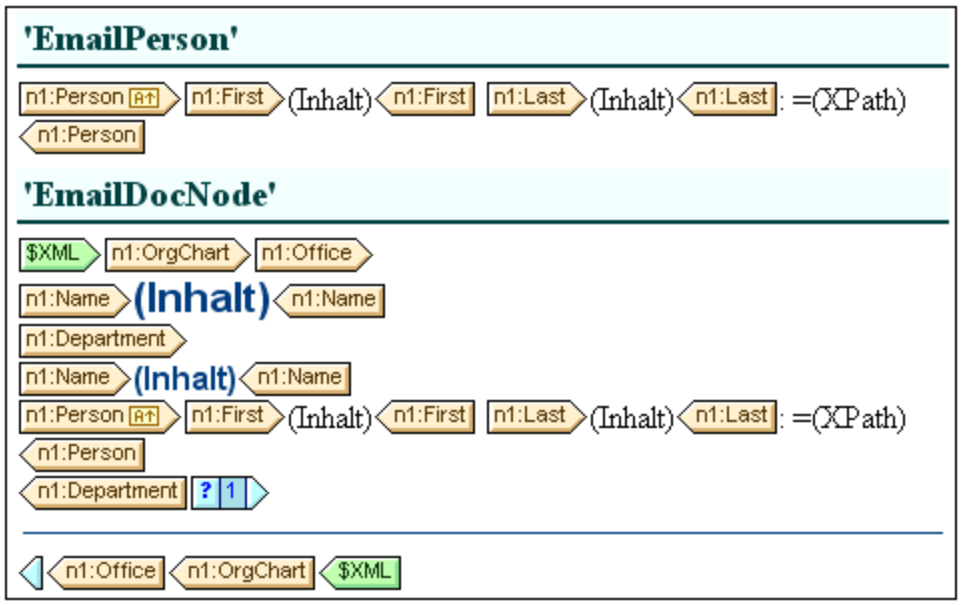
3. Erstellen Sie im Design den Inhalt des Design-Fragments, wie im nächsten Abschnitt beschrieben.

Anmerkung: Wenn Sie ein Design-Fragment anhand einer bestehenden Vorlage erstellen, klicken Sie mit der rechten Maustaste auf die Vorlage und wählen Sie im Kontextmenü den Befehl **Design Fragment erstellen.** Dadurch erstellen Sie anhand der ausgewählten Vorlage an dieser Stelle im Design eine Design-Fragment-Vorlage. Außerdem wird die Design-Fragment-Vorlage an die bestehenden Design-Fragment-Vorlagen am unteren Rand des Designs angehängt und zur Design-Struktur und der Schema-Struktur hinzugefügt. Wenn Sie auf diese Art ein Design-Fragment erstellen, wird es auch direkt an der Stelle, an der es erstellt wurde, angewendet und muss nicht aus der Design- oder Schema-Struktur eingefügt werden[242].

Erstellen des Inhalts eines Design-Fragments

Erstellen Sie im Design den Inhalt des Design-Fragments wie bei jeder globalen Vorlage[231]. Um statischen Inhalt einzufügen, platzieren Sie den Cursor in das die Vorlage für das Design-Fragment und fügen Sie den gewünschten statischen Inhalt ein. Um dynamischen Inhalt einzufügen, ziehen Sie den gewünschten Schema-Node in das Design-Fragment-Dokument.

Beim Ziehen eines Node aus der Schemaquelle können Sie den Node entweder aus der Struktur der globalen Elemente oder aus der Struktur der Root-Elemente ziehen. Der Unterschied besteht darin, dass ein Node, der aus der Struktur der globalen Elemente gezogen wurde, ohne seine übergeordneten Elemente erstellt wird (siehe das Design Fragment `EmailPerson` Abbildung unten) und daher bei Verwendung in einer Vorlage innerhalb des Kontexts seines übergeordneten Elements verwendet werden muss. Wenn ein Node dagegen aus der Struktur der Root-Elemente gezogen wird, wird er innerhalb einer Struktur erstellt, die mit dem Dokument-Node beginnt (siehe Design-Fragment `EmailDocNode` in der Abbildung unten) und kann daher an jeder beliebigen Stelle der Vorlage verwendet werden.



In der Abbildung oben sehen Sie zwei Design-Fragmente, die identische Ausgaben für das Element `Person` erzeugen. Im Design-Fragment `EmailPerson` wurde der `Person`-Node durch Ziehen des globalen Elements `Person` in die Vorlage Email erstellt. Im Design-Fragment `EmailDocNode` wurde der `Person`-Node durch Ziehen aus der Root-Elementestruktur erstellt und erhielt einen absoluten Pfad (aus `$XML`, dem Dokument-Node).

Wenn diese Design-Fragmente in der Hauptvorlage aus eingefügt werden, müssen Sie darauf achten, das das Design-Fragment `EmailPerson` von innerhalb eines Kontexts aufgerufen wird, bei dem es sich um den übergeordneten Node des `Person`-Node handelt. Sie können mit diesen Design-Fragmenten experimentieren. Sie befinden sich in der Beispieldatei `Email.sps`, im Ordner (Eigene) Dateien[25] `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2025\StyleVisionExamples\Tutorials\DesignFragments`.

Sie können zu einem Design-Fragment auch einen Parameter mit einem Standardwert definierten. Dem Parameter kann in jeder Instanz des Design-Fragments ein anderer Wert zugewiesen werden. Nähere Informationen dazu finden Sie unter Parameter für Design-Fragmente[284].

Beachten Sie, dass die Designkomponenten nach Fertigstellung des Designs in der Design-Struktur auch grafisch dargestellt werden.

Einfügen eines Design-Fragments in eine Vorlage

Um ein Design-Fragment einzufügen, ziehen Sie das Design-Fragment aus der Design-Struktur oder aus der Schema-Struktur an die gewünschte Stelle. Die Stelle, an die Sie das Design-Fragment gezogen haben, sollte so beschaffen sein, dass darin der korrekte Kontext für das Design-Fragment enthalten ist. Wenn der Inhalt des Design-Fragments anhand eines globalen Elements erstellt würde, wäre der korrekte Kontext in der Hauptvorlage der übergeordnete Node des in das Design-Fragment gezogenen Node. Siehe dazu den Abschnitt oben Erstellen des Inhalts eines Design-Fragments[243].

Alternativ dazu klicken Sie mit der rechten Maustaste an eine Stelle, an der das Design-Fragment eingefügt werden soll und wählen Sie im Kontextmenü den Befehl **Design-Fragment einfügen.**

Anmerkung: Wenn ein Design-Fragment in der Hauptvorlage referenziert wird und der Name des Design-Fragments später geändert wird, ist die Referenz in der Hauptvorlage später nicht mehr korrekt, was einen

© 2019-2025 Altova GmbH

XSLT-Fehler zur Folge hat. Um dies zu beheben, löschen Sie die ursprüngliche Referenz aus der Hauptvorlage und erstellen Sie eine neue Referenz zum neu benannten Design-Fragment.

Rekursive Design-Fragmente

Design-Fragmente können rekursiv sein, das heißt, ein Design-Fragment kann sich selbst aufrufen. Um sich in Authentic aber gegen eine Endlosschleife zu schützen, kann eine Eigenschaft zur Einstellung der maximalen Aufruftiefe eingestellt werden. Diese Eigenschaft, die *Maximale Tiefe des Vorlagenaufrufs* steht auf dem Register "Authentic" im Dialogfeld "Eigenschaften" des SPS (**Datei | Eigenschaften**) zur Verfügung. Sie gibt die maximale Anzahl der Vorlagenaufrufe an, die während der Verarbeitung für die Authentic-Ausgabe rekursiv gemacht werden dürfen. Wenn die Anzahl der Vorlagenaufrufe die in der Eigenschaft *Maximale Tiefe des Vorlagenaufrufs* angegebene Zahl übersteigt, wird eine Fehlermeldung zurückgegeben.

Löschen eines Design-Fragments

Um ein Design-Fragment zu löschen, wählen Sie es in der Design-Struktur aus und klicken Sie in der Design-Struktur auf die Symbolleistenschaltfläche "Entfernen" .

Design-Fragmente in modularen SPSs

Wenn ein SPS Modul zu einem anderen SPS Modul hinzugefügt wird[216], stehen die Design-Fragmente im hinzugefügten Modul dem referenzierenden SPS zur Verwendung zur Verfügung. Nähere Informationen zur Verwendung modularer SPSs finden Sie im Abschnitt Modulare SPSs[216].

Beispieldatei

Ein Beispiel-SPS finden Sie im Ordner (Eigene) Dateien[25], `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2025\StyleVisionExamples\Tutorials\DesignFragments`.

6.6 XSLT-Vorlagen

XSLT-Dateien können in ein SPS importiert werden und die XSLT-Vorlagen darin stehen im Stylesheet als globale Vorlagen zur Verfügung. Wenn bei der Verarbeitung des XML-Dokuments einer der XML-Nodes einem Node in einer importierten XSLT-Vorlage entspricht, so wird die importierte XSLT-Vorlage auf diesen Node angewendet. Wenn die importierte XSLT-Datei benannte Vorlagen enthält, stehen diese zum Einfügen in das Design zur Verfügung.

Beachten Sie die folgenden Punkte:

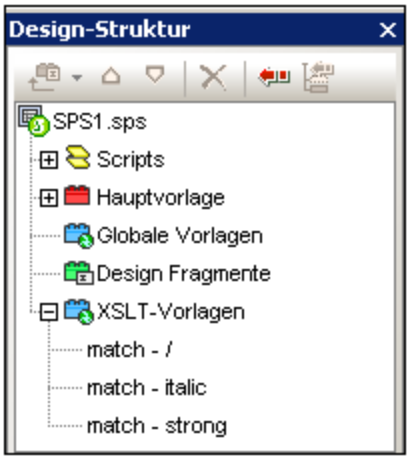
- Importierte XSLT-Vorlagen können in StyleVision nicht geändert werden.

Importieren der XSLT-Datei

Um eine XSLT-Datei zu importieren, gehen Sie folgendermaßen vor:

1. Klicken Sie in der Seitenleiste "Design-Übersicht" (*Abbildung unten*) auf den Link **Neue XSLT-Datei hinzufügen.**

2. Navigieren Sie im Öffnen-Dialogfeld, das daraufhin angezeigt wird, zur gewünschten XSLT-Datei, wählen Sie sie aus und klicken Sie auf **Öffnen.** Die XSLT-Datei wird importiert. Eine `xsl:import` Anweisung wird zum XSLT-Stylesheet hinzugefügt und in der Seitenleiste "Design-Struktur" (*Abbildung unten*) werden die in der importierten XSLT-Datei enthaltenen XSLT-Vorlagen unter der Überschrift "XSLT-Vorlagen" angezeigt.



© 2019-2025 Altova GmbH

Es gibt zwei Arten von importierten XSLT-Vorlagen: (i) Match-Vorlagen (die durch *Match* gekennzeichnet sind), und (ii) benannte Vorlagen (die durch *Name* gekennzeichnet sind). In der Design-Struktur sind diese beiden Typen mit (i) dem Wert des `select` Attributs von Match-Vorlagen bzw. (ii) dem Wert des `name` Attributs von benannten Vorlagen aufgelistet.

Match-Vorlagen

Match-Vorlagen werden verwendet, wenn eine Vorlage im Laufe der Verarbeitung Vorlagen auf einen Node in der XML-Dokumentinstanz anwendet und die Match-Vorlage dafür ausgewählt wird. Dies geschieht, wenn der qualifizierte Name des XML-Node mit dem qualifizierten Namen der importierten Match-Vorlage übereinstimmt. Wenn eine globale Vorlage im SPS erstellt wurde, die denselben qualifizierten Namen hat, so hat diese Vorrang vor einer importierten Vorlage und wird verwendet. Wenn es mehrere importierte XSLT-Dateien gibt, so hat die zuerst importierten Datei (die im XSLT-Code zuerst aufgelistet ist), die niedrigste Priorität, gefolgt von der als zweites importierten Datei, usw.

Benannte Vorlagen

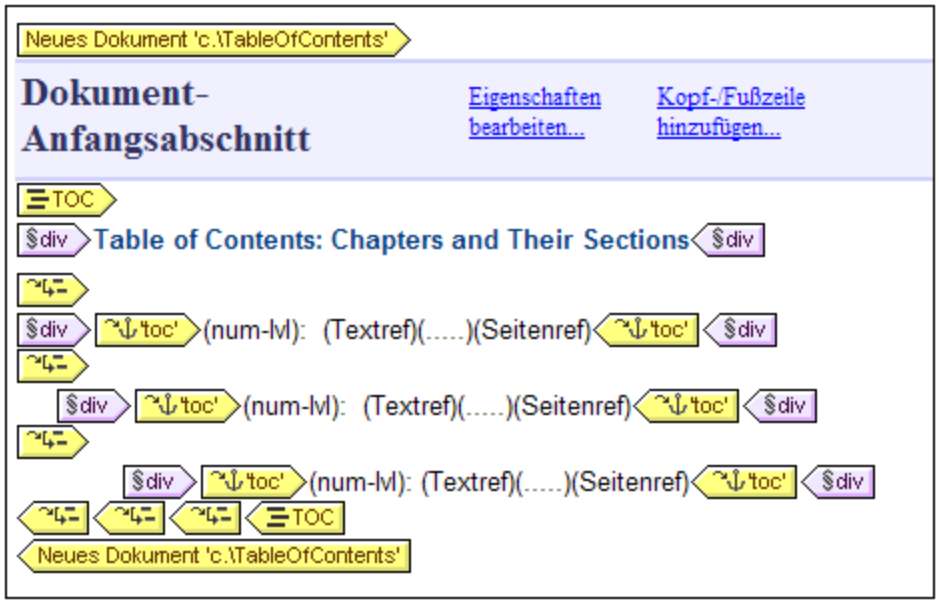
Eine benannte Vorlage kann aus der Design-Struktur an jede Stelle im Design gezogen werden. Sie wird an dieser Stelle als `xsl:call-template` Element*(Abbildung unten*) erstellt, das die benannte Vorlage aufruft.

```
<xsl:call-template name="BoldItalic" />
```

Das Resultat davon in der Ausgabe ist, dass die benannte Vorlage an dieser Stelle im Design implementiert wird. Dies eignet sich zum Einfügen von Inhalt, der unabhängig vom XML-Instanzdokument und dem XSLT-Stylesheet ist.

6.7 Mehrfach-Dokumentausgabe

Sie können ein SPS für die Ausgabe mehrerer Ausgabedokumente (ein Hauptausgabedokument und ein oder mehrere zusätzliche Dokumente) anlegen. Diese Methode eignet sich vor allem, wenn Sie Module für die Ausgabe erstellen möchten. Ausgabedokumente werden im Design erstellt, indem Sie eine Vorlage für ein neues Dokument einfügen (*siehe Abbildung unten*). Inhalt für die einzelnen Ausgabedokumente wird in die neue Dokumentvorlage dafür platziert.



Vorlagen für neue Dokumente können an jeder beliebigen Stelle im Design angelegt werden, sodass die Ausgabe auf jeder beliebigen Ebene modularisiert werden kann. So kann z.B. ein Bericht über die verschiedenen Niederlassungen eines internationalen Unternehmens auf jeder der folgenden Ebenen separate Ausgabedokumente haben: (i) Welt, (ii) Kontinent, (iii) Land, (iv) Bundesstaat und/oder (v) Niederlassung. So kann z.B. jede Niederlassung in einem separaten Ausgabedokument dargestellt werden oder es können alle Niederlassungen in einem Land zusammen in einem einzigen Bericht für das jeweilige Land behandelt werden. In Design müsste für jede der hierarchischen Ebenen, für die separate Ausgabedokumente benötigt werden, eine neue Dokumentvorlage angelegt werden. Eine Anleitung zum Erstellen der richtigen Dokumentstruktur finden Sie im Abschnitt Vorlagen für neue Dokumente und Design-Struktur 250.

Diese Beschreibung mehrerer Ausgabedokumente ist in die folgenden Unterabschnitte gegliedert:

- Einfügen einer Vorlage für ein neues Dokument 249
- Vorlagen für neue Dokumente und Design-Struktur 250
- URLs von Vorlagen für neue Dokumente 250
- Vorschau und Ausgabedokumentdateien 253
- Dokumenteigenschaften und Stile 255

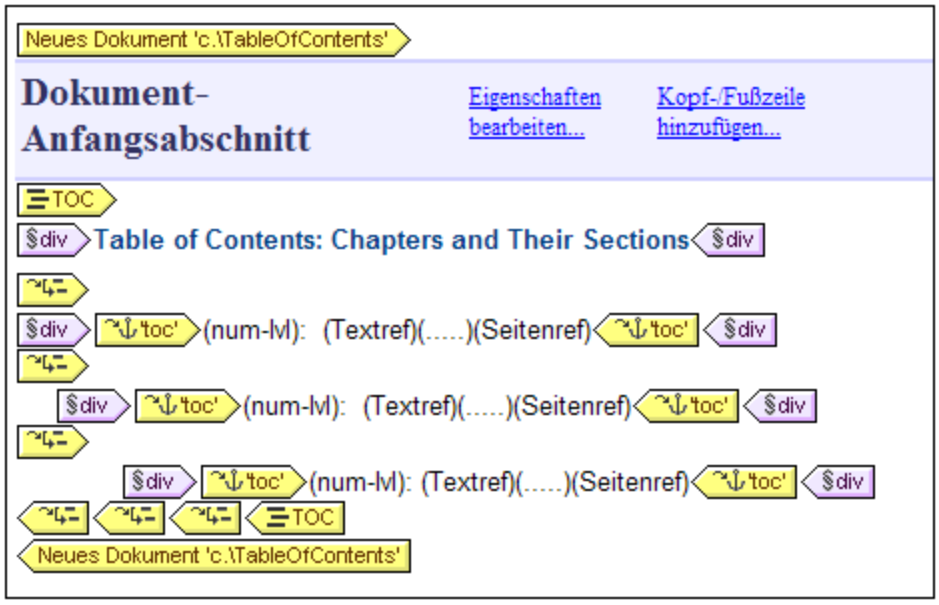
 © 2019-2025 Altova GmbH

6.7.1 Einfügen einer Vorlage für ein neues Dokument

Eine Vorlage für ein neues Dokument kann auf zwei Arten in ein SPS-Design eingefügt werden:

- Ein neues Ausgabe-Dokument kann an jeder beliebigen Stelle in das Design eingefügt werden. In diesem Fall wird der Inhalt des neuen Dokuments im Anschluss an die Einfügung zur Vorlage hinzugefügt. Um eine Vorlage für ein neues Dokument einzufügen, platzieren Sie den Cursor an die gewünschte Stelle im Design und wählen Sie den Befehl **Einfügen | Neues Dokument einfügen** oder klicken Sie mit der rechten Maustaste auf die gewünschte Stelle und wählen Sie im Kontextmenü, das daraufhin angezeigt wird, den Befehl **Neues Dokument einfügen.**
- Ein neues Ausgabe-Dokument kann in das Design platziert werden, indem **Inhalt** in die Vorlage für ein neues Dokument **eingeschlossen** wird. In diesem Fall enthält die Vorlage für das neue Dokument nach der Erstellung den eingeschlossenen Inhalt. Sie können diesen Inhalt im Design ändern oder Inhalt hinzufügen. Um eine Vorlage für ein neues Dokument so zu platzieren, dass der Inhalt darin eingeschlossen wird, markieren Sie den gewünschten Inhalt und wählen Sie anschließend den Befehl **Einschließen in | Neues Dokument**. Alternativ dazu können Sie den gewünschten Inhalt auch markieren, anschließend auf die rechte Maustaste klicken und aus dem Kontextmenü den Befehl **Einschließen in | Neues Dokument** auswählen.

In der Abbildung unten sehen Sie eine Vorlage für ein neues Dokument.



Beachten Sie in der Abbildung die folgenden Punkte:

1. Die "Neues Dokument"-Vorlagen-Tags enthalten die URL (Pfad und Namen) des Ausgabedokuments, das mit der Vorlage generiert wird. Die Dateierweiterung wird je nach Dateityp des Ausgabeformats automatisch generiert. So wird z.B. die Dateierweiterung für das HTML-Ausgabeformat `.html` an den Dateinamen in der URL angehängt. Informationen im Zusammenhang mit der Zuweisung von URLs finden Sie im Abschnitt URLs von Vorlagen für neue Dokumente 250.

6.7.2 Vorlagen für neue Dokumente und Design-Struktur

Bei Erstellung mehrerer Ausgabedokumente müssen Sie die verschiedenen Vorlagen für die neuen Dokumente in den entsprechenden Nodes des Quelldokuments erstellen. Daher müssen Sie beim Entwerfen mehrerer Ausgabedokumente sowohl die Ausgabestruktur[250] als auch die Struktur des Eingabedokuments (XML-Quelldokument)[250] berücksichtigen.

Hauptausgabedokument und zusätzliche Ausgabedokumente (Ausgabestruktur)

Wenn die erste Vorlage für ein neues Dokument zum Design hinzugefügt wird, wird der gesamte Inhalt des Designs außerhalb der Vorlage für das neue Dokument automatisch einem separaten Dokument zugewiesen. Dieses separate Dokument wird als das Hauptausgabedokument betrachtet und wird in den Ausgabevorschauen von StyleVision als Hauptausgabedokument bezeichnet.

Als Name des Hauptausgabedokuments wird in den (mit dem Befehl **Datei | Generierte Dateien speichern**) generierten Ausgabedokumenten der Name verwendet, den Sie diesem Dokument beim Generieren der Ausgabedokumentdateien mit Hilfe des Befehls **Datei | Generierte Dateien speichern** zuweisen. Als Namen für die zusätzlichen Ausgabedokumentdateien werden die Namen verwendet, die in den URLs der Vorlagen für das jeweilige neue Dokument zugewiesen wurden.

Vorlagen für neue Dokumente und Quelldokumentstruktur

Bei der Erstellung einer Vorlage für ein neues Dokument ist die hierarchische Struktur des Nodes, in der es erstellt wird, ausschlaggebend. Es gibt zwei Möglichkeiten:

1. Der Node, in dem die Vorlage für das neue Dokument angelegt wird, wird nur einmal verarbeitet. In diesem Fall wird auch die Vorlage für das neue Dokument nur einmal verarbeitet. Als Name in der URL-Eigenschaft der neuen Dokumentvorlage kann daher ein statischer Name verwendet werden.
2. Der Node, in dem die Vorlage für das neue Dokument angelegt wird, wird mehrmals verarbeitet. Folglich wird die neue Dokumentvorlage so oft verarbeitet wie der dazugehörige Node. Ein Beispiel dafür wäre die folgenden Situation: Ein `Office`-Element hat mehrere untergeordnete `Department`-Elemente (für die verschiedenen Abteilungen). Wenn im Design im Node `Department` eine Vorlage für ein neues Dokument angelegt wird und der `Department`-Node (für die einzelnen `Department`-Elemente im Element `Office`) mehrmals verarbeitet wird, so wird diese Vorlage ebenfalls mehrmals verarbeitet, nämlich einmal für jedes `Department`-Element im XML-Quelldokument. Der Dateiname in der URL-Eigenschaft der neuen Dokumentvorlage muss daher ein dynamischer Name sein. Andernfalls erhalten die für die Department-Elemente erstellten Ausgabedokumente alle denselben Dateinamen.

6.7.3 URLs von Vorlagen für neue Dokumente

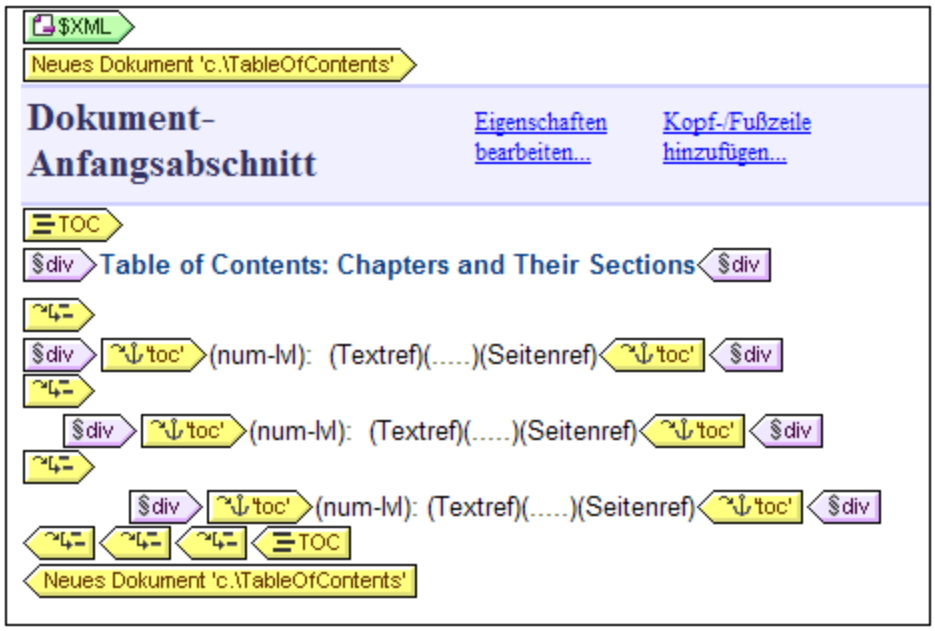
In diesem Abschnitt wird erläutert, wie die URLs von Vorlagen für neue Dokumente mit der Design-Struktur im Zusammenhang stehen[250], wie URLs bearbeitet werden[252] und wie mehrere Ausgabedokumente durch Links miteinander verknüpft werden können.[253]

URLs von Vorlagen für neue Dokumente

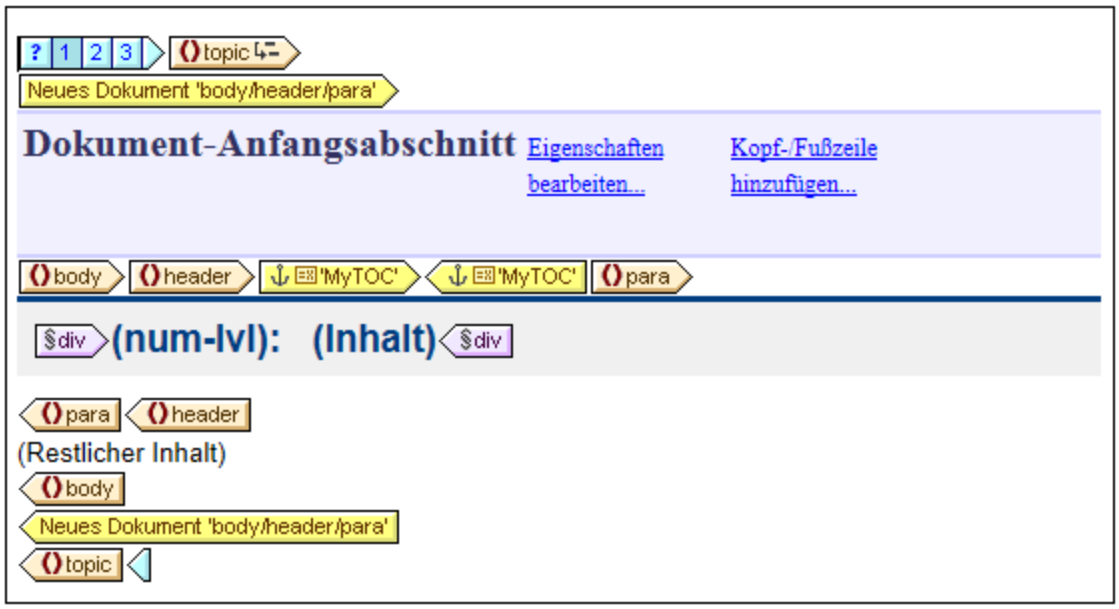
Wenn die Vorlage für das neue Dokument nur einmal verarbeitet wird (*siehe vorhergehender Abschnitt*[250]), dann kann als URL-Eigenschaft der Vorlage eine statische URL verwendet werden. Da sich die Vorlage für das neue Dokument in der Abbildung unten direkt innerhalb des Dokumentelements (`$XML`) befindet, wird sie nur einmal verarbeitet. Für die URL wurde ein statischer `TableOfContents`-Wert definiert. Dieser Wert wird daher

© 2019-2025 Altova GmbH

als Dateiname für das Ausgabedokument verwendet. Da dem Dateinamen kein Pfad vorangestellt wurde, wird die Datei im selben Verzeichnis wie die Hauptdokumentdatei generiert (nähere Informationen dazu *siehe* Mehrere Dokumentausgaben und Vorschauen[253]). Wenn die URL alternativ dazu einen Path enthält, so wird das Ausgabedokument unter dem definierten Pfad gespeichert.

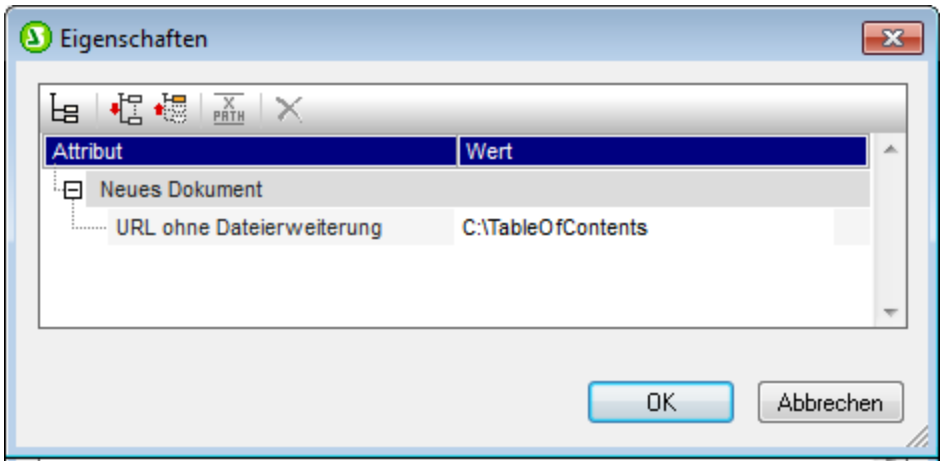


Wenn eine Vorlage für ein neues Dokument jedoch mehrmals verarbeitet wird, um mehrere Ausgabedokumente zu generieren (*siehe vorhergehender Abschnitt*[250]), so muss es sich bei der URL-Eigenschaft der Vorlage um eine dynamische URL handeln, die mit einem XPath-Ausdruck ermittelt wird. In der Abbildung unten ist die URL der Vorlage für das neue Dokument der XPath-Ausdruck: `body/header/para`. Die Vorlage für das neue Dokument befindet sich im Element `topic`, daher wird sie jedes Mal verarbeitet, wenn das Element `topic` verarbeitet wird. Bei jeder Iteration durch das `topic`-Element wird der Inhalt des `body/header/para`-Elements des `topic`-Elements der neuen Dokumentvorlage als URL zugewiesen. Dadurch wird für jedes `topic`-Element ein neues Dokument erstellt. Jedes dieser Dokumente hat einen anderen Namen, nämlich den seines `body/header/para`-Elements (also den Text der Kapitelüberschrift).



Bearbeiten der URL

Wenn dem Design eine Vorlage für ein neues Dokument hinzugefügt wird, wird diese mit einer Standard-URL erstellt. Dabei handelt es sich um einen statischen Text-String: `DocumentX` (wobei `X` eine Ganzzahl ist). Wenn Sie die URL bearbeiten möchten, klicken Sie mit der rechten Maustaste auf die Vorlage für das neue Dokument und wählen Sie den Befehl **URL bearbeiten.** Daraufhin wird das Dialogfeld "Eigenschaften" (*Abbildung unten*) aufgerufen, wo Sie das Wertefeld der Eigenschaft *URL ohne Dateierweiterung* bearbeiten können.



Wenn Sie eine statische URL eingeben möchten, bearbeiten Sie das Wertefeld, damit es den gewünschten URL-Text enthält. Wenn Sie eine dynamische URL eingeben möchten, klicken Sie in das Wertefeld, klicken Sie in der Symbolleiste des Dialogfelds "Eigenschaften" auf die Schaltfläche "XPath" und geben Sie den gewünschten XPath-Ausdruck ein. Beachten Sie bitte Folgendes: (i) Der Kontext-Node für den XPath-Ausdruck ist der Node, in den die Vorlage für das neue Dokument eingefügt wurde. (ii) Um dem XPath-Ausdruck-Ergebnis einen Pfad voranzustellen, verwenden Sie die `concat()` Funktion von XPath. Beispiel: `concat('C:\MyOutput\', body\header\para)`. Mit diesem Beispielausdruck wird der URL-String: `C:`

 © 2019-2025 Altova GmbH

`\MyOutput\filename` generiert. Die entsprechende Dateierweiterung wird automatisch für das jeweilige Ausgabeformat generiert.

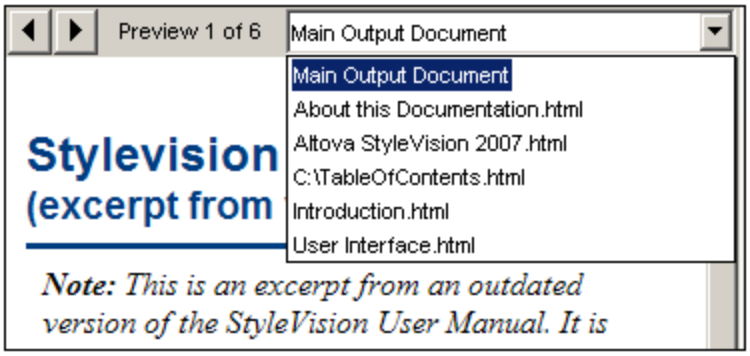
Verknüpfen der Dokuments mittels Links

Mehrere Ausgabedokumente können mit Hilfe von Lesezeichen und Hyperlinks[320] miteinander verknüpft werden. Ein Lesezeichen[320] kann am Anfang der Vorlage für das neue Dokument oder an einer beliebigen Stelle in der Vorlage für das neue Dokument platziert werden. Anschließend können in anderen Dokumenten Hyperlinks[322] erstellt werden, mit denen Sie wieder zum Lesezeichen zurücknavigieren können. Wenn Lesezeichen in einem Node erstellt werden müssen, der mehrmals verarbeitet wird, so stellen Sie sicher, dass der Name des Lesezeichens dynamisch generiert wird. Andernfalls (d.h. bei Erstellung eines statischen Lesezeichens) erhalten mehrere Nodes im Ausgabedokument denselben Lesezeichennamen.

Zum Verknüpfen von Dokumenten kann auch ein Inhaltsverzeichnis[290] verwendet werden. Das Inhaltsverzeichnis könnte sich in einem separaten Dokument (z.B. dem Hauptdokument) befinden und Links zu den verschiedenen Ausgabedokumenten enthalten, während man über Links in den Ausgabedokumenten wieder zurück zum Inhaltsverzeichnis gelangt.

6.7.4 Vorschaudateien und Ausgabedokumentdateien

In der Ausgabevorschau eines Design-Dokuments wird jedes der vielen Ausgabedokumente angezeigt, die im Design als separate Dokumente definiert wurden (*siehe Abbildung unten*).

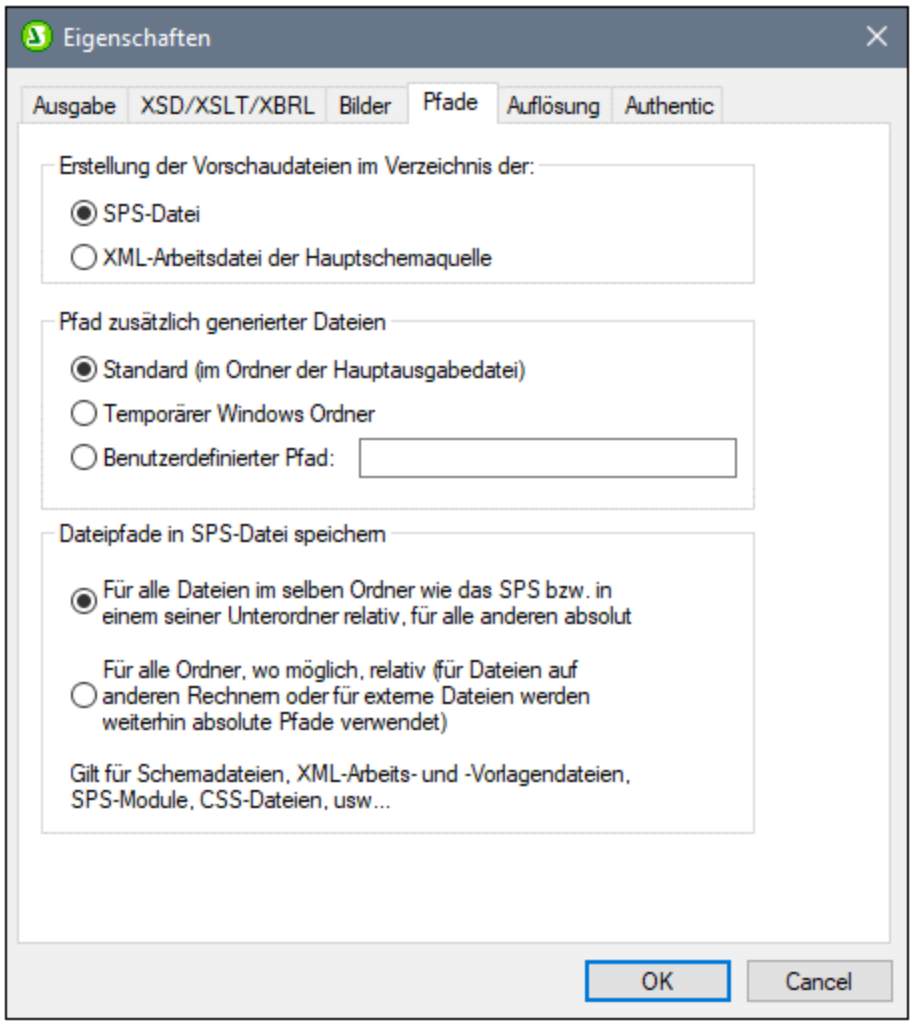


In der Abbildung oben sehen Sie die HTML-Vorschau eines SPS-Dokuments, das für die Generierung mehrerer Ausgabedokumente angelegt wurde. Die einzelnen Ausgabedokumente können auf folgende Weise im Vorschaufenster aufgerufen werden: (i) durch Navigation durch die verfügbaren Dokumente mit Hilfe der Pfeiltaste in der linken oberen Ecke oder (ii) durch Auswahl des gewünschten Dokuments aus der Dropdown-Liste der Auswahlliste (*siehe Abbildung oben*). Beachten Sie bitte, dass in den Einträgen der Dropdown-Liste die gesamte URL (Pfad plus Dateiname) angezeigt wird.

Pfad der Vorschaudateien

Die Vorschaudateien werden standardmäßig in dem Verzeichnis erstellt, in dem auch die SPS-Datei angelegt wurde. Diese Standardeinstellung kann im Dialogfeld "Eigenschaften" der SPS-Datei (*Abbildung unten*) auf dem Register "Pfade" (Aufruf mit dem Befehl **Datei | Eigenschaften**) geändert werden. Auf diesem Register können Sie als Alternative das Verzeichnis der XML-Arbeitsdatei definieren. Wenn die URL einer Vorlage für ein

neues Dokument einen Pfad enthält, so werden die Vorschaudateien im jeweils angegebenen Ordner angelegt. Wird der Ordner nicht gefunden, so wird ein Fehler zurückgegeben. Wenn Sie Ausgabedokumente erstellen, die mittels Links miteinander verknüpft sind, sollten Sie beachten, wo die Ausgabedokumente gespeichert werden.



Außerdem können Sie auf dem Register "Pfade" des Dialogfelds "Eigenschaften" (*siehe Abbildung oben*) festlegen, wo temporäre zusätzliche Dateien für Vorschauen wie z.B. Ausgabedokumentdateien, Bilder und Diagrammbilddateien gespeichert werden. Beachten Sie: Wenn die URL einer Vorlage für ein neues Dokument einen Pfad enthält, dann wird der in diesem Pfad angegebene Ort verwendet.

Generieren der Ausgabedokumente (Pfade usw.)

Um die Ausgabedokumentdateien zu generieren, gehen Sie folgendermaßen vor:

1. Platzieren Sie die Maus über den Menübefehl **Datei | Generierte Dateien speichern** und klicken Sie auf das gewünschte Ausgabeformat.
2. Navigieren Sie im Dialogfeld "Generierte Dateien speichern", das nun angezeigt wird, zu dem Ordner, in dem die generierte Datei gespeichert werden soll.
3. Geben Sie den Namen der Hauptdokumentdatei ein und klicken Sie auf **Speichern.**

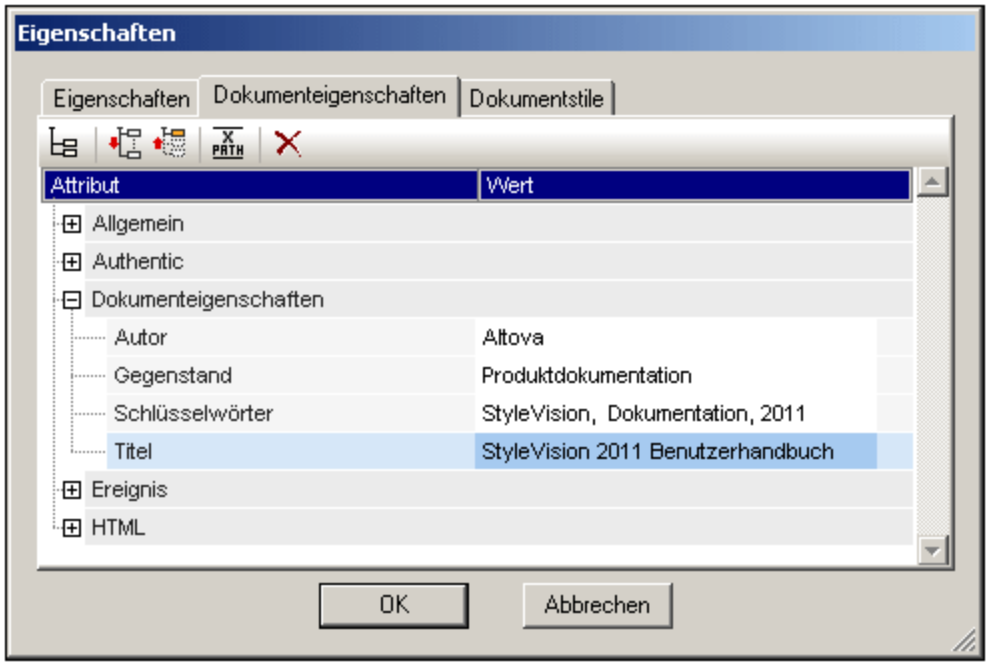
© 2019-2025 Altova GmbH

Der Speicherort aller Ausgabedokumentdateien sowie aller anderen zusätzlich generierten Dateien wie Bilddateien und Diagrammbilddateien wird zu Ihrer Information in einem Popup-Fenster angezeigt.

Die Hauptdokumentdatei wird in dem Ordner gespeichert, den Sie im Dialogfeld "Generierte Datei speichern" ausgewählt haben. Alle Ausgabedokumente, die mit Vorlagen für neue Dokumente erstellt wurden und deren URLs keine Pfadinformationen enthalten, werden im selben Ordner, wie die Hauptdokumentdatei gespeichert. Wenn der URL der Vorlage für das neue Dokument ein Pfad vorangestellt wurde, wird das Ausgabedokument unter dem in der URL definierten Pfad gespeichert. Wenn dieser Ordner nicht existiert, wird ein Fehler generiert.

6.7.5 Dokumenteigenschaften und Stile

Sie können die Ausgabe in einem SPS-Design auf mehrere Dokumente aufteilen. Jedem dieser Dokumente können separate Dokumenteigenschaften und -stile zugewiesen werden, die über den Dokumentanfangsabschnitt im Dialogfeld "Eigenschaften" auf den Registern *Dokumenteigenschaften* bzw. *Dokumentstile* (*siehe Abbildung unten*) definiert werden können. Um das Dialogfeld "Eigenschaften" aufzurufen, klicken Sie im gewünschten Dokument in der Titelleiste des Dokumentanfangsabschnitts auf den Link *Eigenschaften bearbeiten*. Die Dokumenteigenschaften und -stile gelten für das gesamte Ausgabedokument.



Auf dem Register "Dokumenteigenschaften" können Sie in der Eigenschaftsgruppe "Dokumenteigenschaften" Metainformationen für das Dokument eingeben. Diese Metainformationen werden im entsprechenden Ausgabedokument und unter den entsprechenden Eigenschaften für das jeweilige Ausgabeformat gespeichert. So werden z.B. die Eigenschaften im HTML-Ausgabeformat in den entsprechenden `META` Tags des Elements `HEAD` gespeichert.

Eine Beschreibung zu Dokumentstilen finden Sie im Abschnitt Definieren von CSS-Eigenschaftswerten 351.

7 Komplexe Funktionen

In den Abschnitten Inhalt der SPS-Datei[110] und Struktur der SPS-Datei[184] wird beschrieben, wie man die Grundstruktur und den Inhalt des SPS Designs erstellt. Oft müssen jedoch auch der Inhalt und/oder die Struktur von Quelldaten auf eine bestimmte Art bearbeitet werden. So müssen Sie eventuell eine Gruppe von Nodes - z.B. Nodes mit persönlichen Daten nach einem bestimmten Kriterium, z.B. dem Nachnamen der Angestellten in alphabetischer Reihenfolge sortieren. Oder Sie möchten alle Kunden in einer Datenbank nach Städten gruppieren. Oder Sie möchten die Umsätze für ein Produkt in einer bestimmten Stadt summieren. Für solche Zwecke stehen die in diesem Abschnitt beschriebenen komplexen Funktionen von StyleVision zur Verfügung.

Im Folgenden finden Sie eine Liste der komplexen SPS-Funktionen von StyleVision:

- Automatische Berechnungen[257]. Automatische Berechnungen sind eine leistungsfähige XPath-basierte Funktion zum Bearbeiten von Daten und (i) Darstellen der bearbeiteten Daten in der Ausgabe sowie (ii) zum Aktualisieren von Nodes im XML-Dokument mit dem Ergebnis der automatischen Berechnung.
- Bedingungen[262]. Die Verarbeitung und der Inhalt von Vorlagen kann von Datenstrukturen oder Werten im XML-Dokument oder dem Ergebnis eines XPath-Ausdrucks abhängig gemacht werden.
- Gruppieren[267]. Die Verarbeitung kann für eine Gruppe von Elementen, die mit einem XPath-Ausdruck ausgewählt werden, definiert werden.
- Sortieren[276]. Eine Gruppe von XML-Elementen kann nach mehreren Sortierschlüsseln sortiert werden.
- Parameter und Variablen[282]. Parameter werden auf globaler SPS-Ebene mit einem Standardwert deklariert. Diese Werte können anschließend während der Laufzeit durch Werte, die über die Befehlszeile an das Stylesheet übergeben werden, außer Kraft gesetzt werden. Variablen können im SPS definiert werden und können für die Verwendung im SPS referenziert werden.
- Inhaltsverzeichnis und Referenzierung[290]. Inhaltsverzeichnisse können an verschiedenen Stellen in der Dokumentausgabe und für alle Ausgabeformate erstellt werden. Dabei werden zuerst die im Inhaltsverzeichnis zu referenzierenden Einträge ausgewählt. Anschließend werden diese markierten Einträge im Inhaltsverzeichnis referenziert. Weitere Funktionen, für die die Referenzierung verwendet wird, sind: (i) Automatische Nummerierung[314] (sich wiederholende Nodes im Dokument können automatisch nummeriert werden und die Nummern werden formatiert; (ii) Textreferenzen[318] (Text im Dokument kann für die Referenzierung markiert werden und anschließend von einer anderen Stelle im Dokument referenziert werden); und (iii) Lesezeichen und Hyperlinks[320] (Mit Lesezeichen werden wichtige Stellen im Ausgabedokument markiert, die anschließend über Hyperlinks referenziert werden können. Mit Hyperlinks können auch externe Ressourcen mittels einer Reihe von Methoden zur Festlegen der Ziel-URI (statisch, dynamisch, einer Kombination der beiden Arten und ungeparste Entity-URIs) verlinkt werden. Alle diese Referenzierungsmechanismen werden in diesem Abschnitt beschrieben.

Siehe auch

- Inhalt der SPS-Datei[110]
- SPS-Datei: Struktur[184]
- SPS-Datei: Zusätzliche Funktionen[361]

© 2019-2025 Altova GmbH

7.1 Automatische Berechnungen

Mit Hilfe der Funktion **Automatische Berechnung** (i) kann das Ergebnis einer XPath-Auswertung an jeder beliebigen Stelle im Ausgabedokument angezeigt werden und (ii) optional dazu wird ein Node im XML-Hauptdokument (dem XML-Dokument, das in der Authentic-Ansicht bearbeitet wird) mit dem Ergebnis der XPath-Auswertung aktualisiert.

Die Funktion "Automatische Berechnung" eignet sich für folgende Zwecke:

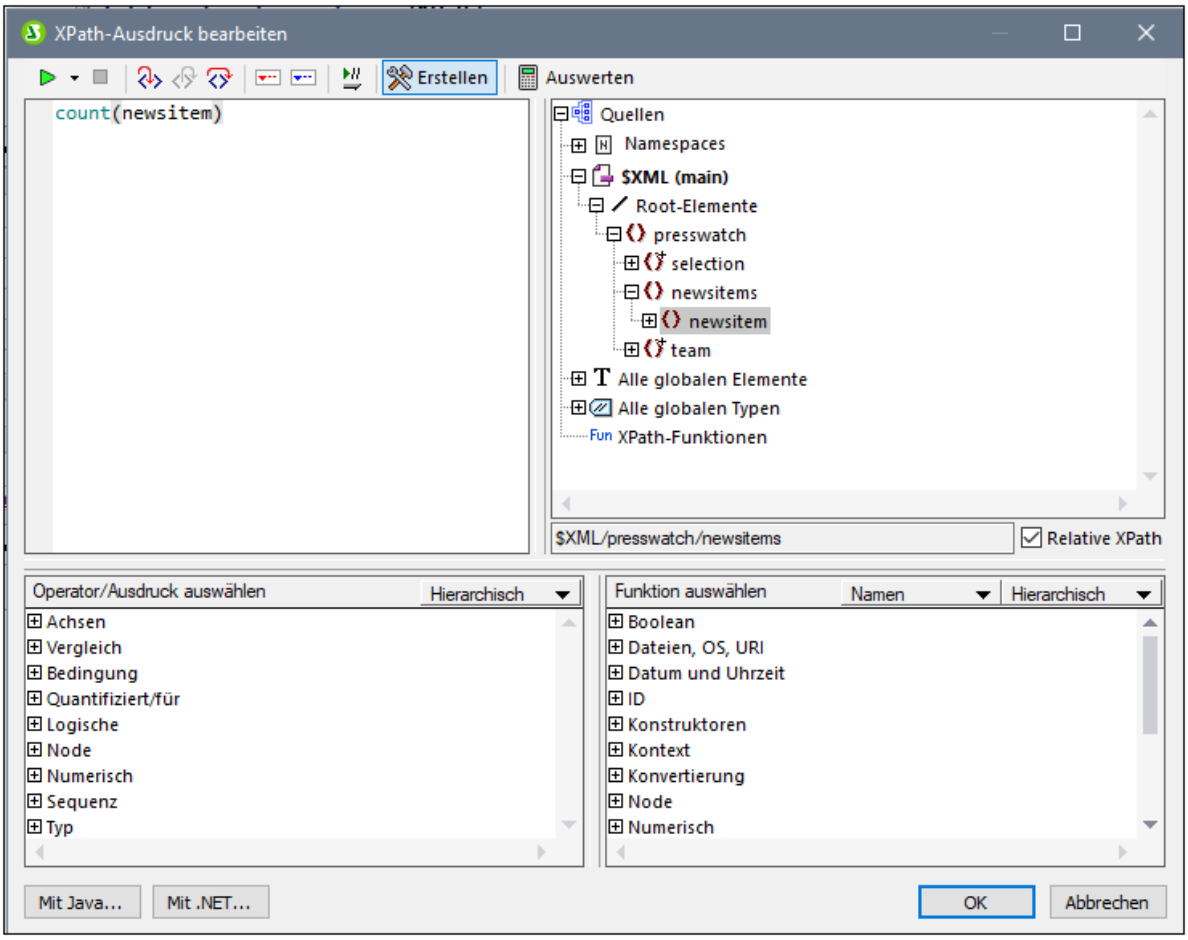
- um Berechnungen mit Operationen an dynamischen Datenwerten einzufügen. So können Sie z.B. die Anzahl der `Employee` Elemente in einem `Office` Element zählen (mit `count(Employee)`) oder die Werte aller `Price` Elemente in den einzelnen `Invoice` Elementen summieren (mit `sum(Price)`) oder die Elemente `FirstName` und `LastName` eines `Person` Elements verbinden (mit `concat(FirstName, ' ', LastName)`). Auf diese Art können Sie anhand von Daten im XML-Dokument, die sich dynamisch ändern, neue Daten generieren und die generierten Daten an die Ausgabe senden.
- um Informationen auf Basis der Struktur des Dokuments anzuzeigen. So können Sie z.B. mit Hilfe der XPath-Funktion `position()` Zeilennummern dynamisch in eine Tabelle einfügen oder die Abschnitte eines Dokuments dynamisch nummerieren. Dies hat den Vorteil, dass Informationen basierend auf Dokumentstrukturen, die sich dynamisch ändern, automatisch generiert werden.
- um Daten aus externen XML-Dokumenten einzufügen. Mit Hilfe der `doc()` Funktion von XPath 2.0 haben Sie Zugriff auf die Dokument-Root von externen XML-Dokumenten und können Node-Inhalt aus dem externen XML-Dokument in die Ausgabe einfügen.
- um den Inhalt eines Node an jeder beliebigen Stelle im Design anzuzeigen.

7.1.1 Bearbeiten und Verschieben von automatischen Berechnungen

Erstellen von automatischen Berechnungen

Um eine automatische Berechnung zu erstellen, gehen Sie folgendermaßen vor:

1. Platzieren Sie den Cursor als Einfügepunkt an die Stelle im Dokument, an der das Ergebnis der automatischen Berechnung angezeigt werden soll und klicken Sie auf **Einfügen | Automatische Berechnung**. Wählen Sie im Untermenü, das angezeigt wird, "Wert", wenn das Ergebnis als reiner Text angezeigt werden soll, wählen Sie die Option "Eingabefeld", wenn es in einem Eingabefeld (also einem Textfeld) angezeigt werden soll, oder wählen Sie "Mehrzeiliges Eingabefeld", wenn es in einem mehrzeiligen Textfeld angezeigt werden soll. (Beachten Sie, dass das Ergebnis der automatischen Berechnung als Wert oder in einem Eingabefeld angezeigt wird. Es handelt sich hierbei um eine Ausgabe in der Authentic-Ansicht und diese Ausgabe kann dort nicht bearbeitet werden.) Daraufhin erscheint das Dialogfeld "XPath-Ausdruck bearbeiten" (*Abbildung unten*).



2. Geben Sie im Bereich "Ausdruck" den XPath-Ausdruck für die automatische Berechnung über die Tastatur ein. Alternativ dazu können Sie den Ausdruck auch durch Doppelklick auf die Nodes, Operatoren und/oder Funktionen in den entsprechenden Bereichen des Dialogfelds zusammenstellen. Beachten Sie, in welchem Kontext-Node die Berechnung eingefügt wird; der Kontext-Node wird in der Schemaquellenstruktur markiert (in der Abbildung oben erscheint der Kontext-Node `newsitems` markiert). (Wenn Sie als XSLT-Sprache für Ihr SPS XSLT 1.0 gewählt haben, müssen Sie XPath 1.0-Ausdrücke verwenden; wenn Sie XSLT 2.0 oder XSLT 3.0 ausgewählt haben, müssen Sie XPath 2.0- bzw. XPath 3.0-Ausdrücke verwenden.) Eine ausführliche Beschreibung des Dialgofelds "XPath-Ausdruck bearbeiten" finden Sie im Abschnitt XPath-Ausdruck bearbeiten 440.

Klicken Sie zum Fertigstellen auf **OK.** Auf dem Register "Design" wird das Symbol für automatische Berechnungen angezeigt. Um das Ergebnis der automatischen Berechnung zu sehen, wechseln Sie in die HTML-Ansicht.

Bearbeiten von automatischen Berechnungen

Um den XPath-Ausdruck der automatischen Berechnung zu bearbeiten, wählen Sie die automatische Berechnung aus und klicken Sie in der Seitenleiste "Eigenschaften" auf die **Bearbeiten**-Schaltfläche der Eigenschaft `XPath` in der Eigenschaftsgruppe *XPath* (*Abbildung unten*). Daraufhin wird das Dialogfeld "XPath-Ausdruck bearbeiten 440" (*Abbildung unten*) angezeigt, in dem Sie den XPath-Ausdruck bearbeiten können.

© 2019-2025 Altova GmbH

Formatieren von automatischen Berechnungen

Sie können auf automatische Berechnungen genau wie bei normalem Text vordefinierte Formate und CSS-Stile anwenden: Wählen Sie die automatische Berechnung aus und wenden Sie die Formatierung an. Des Weiteren kann die Eingabeformatierung[333] einer automatischen Berechnung, bei der es sich um einen numerischen oder einen Datums-Datentyp handelt, im Fenster "Eigenschaften" in der Eigenschaftsgruppe "XPath" über die Eigenschaft "Eingabeformatierung" definiert werden.

Beachten Sie außerdem, dass Sie Zeilenschaltungen und Umbrüche (CR/LFs) in XPath-Ausdrücke einfügen können. Wenn die automatische Berechnung im Sonderabsatztyp pre eingeschlossen wird, wird in der Ausgabe eine neue Zeile erzeugt. Ein Beispiel dafür ist der XPath-Ausdruck:

```
translate('a;b;c', ';', codepoints-to-string(13))
```

Verschieben von automatischen Berechnungen

Durch Klicken auf die automatische Berechnung und Ziehen der Berechnung an eine andere Stelle, können Sie eine automatische Berechnung an eine andere Stelle verschieben. Zum Verschieben/Kopieren der automatischen Berechnung können Sie auch die Befehle Ausschneiden/Kopieren und Einfügen verwenden. Beachten Sie jedoch, dass der XPath-Ausdruck geändert werden muss, wenn der Kontext-Node an der neuen Stelle nicht der gleiche ist, wie vorher.

Zusammenfassung der wichtigsten Punkte

Beachten Sie die folgenden Punkte:

- Eine automatische Berechnung kann überall im Design-Dokument eingefügt werden.
- Die Stelle, an der Sie die automatische Berechnung einfügen, bestimmt den Kontext-Node für die XPath-Auswertung.

7.1.2 Beispiel: Eine Rechnung

Anhand des Beispiels `SimpleInvoice.sps` im Ordner (Eigene) Dateien[25], `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2025\StyleVisionExamples/Tutorials/Auto-`

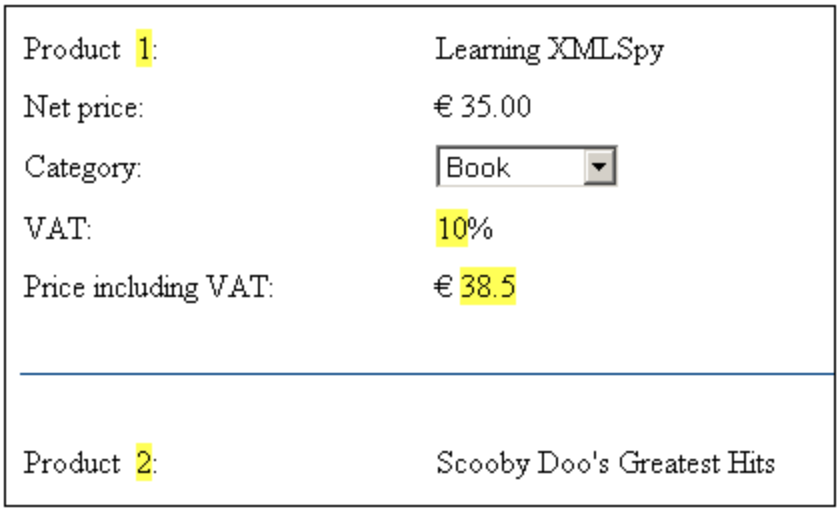
`Calculations/` wird demonstriert, wie automatische Berechnungen für die folgenden Zwecke verwendet werden können:

- Zählen von Nodes
- Auswahl eines Node auf Basis einer Eingabe des Benutzers in der Authentic-Ansicht
- Erstellen komplexer Berechnungen

In der Beispieldatei wurden die automatischen Berechnungen mit einer gelben Hintergrundfarbe markiert (*siehe Abbildung unten*).

Zählen von Nodes

Im Beispiel "SimpleInvoice" ist jedes Produkt in der Liste nach seiner Position in der Liste der Produkte nummeriert, die ein Kunde bestellt hat (`Product 1`, `Product 2`, usw.). Diese Nummerierung wird mit Hilfe einer automatischen Berechnung erstellt (*Abbildung unten*).



In diesem speziellen Fall genügt der XPath-Ausdruck `position()` um eine korrekte Nummerierung zu erhalten. Eine andere nützliche Methode zur Ermittlung der Position eines Node ist, die Anzahl der vorhergehenden gleichrangigen Elemente zu zählen und die Zahl Eins dazu zu addieren. Der XPath-Ausdruck dafür würde lauten: `count(preceding-sibling::Product)+1`. Die zweite Methode würde sich in solchen Fällen als nützlich erweisen, in denen die `position()` Funktion nur schwer oder gar nicht verwendet werden kann. Sie können diese automatische Berechnung in der Beispieldatei testen, indem Sie Produkte löschen und/oder neue Produkte hinzufügen und löschen.

Auswahl eines Node auf Basis einer Benutzereingabe

Im Beispiel "SimpleInvoice" wählt der Benutzer die Produktkategorie (`Book`, `CD`, `DVD` oder `Electronics`) über eine Auswahlliste aus. Diese Auswahl wird in den Node `//Product/Category` node im XML-Dokument eingegeben. Bei der automatischen Berechnung wird anhand dieses Werts eine "Nachschlagetabelle" im XML-Dokument referenziert und der Node, der den Mwst. Prozentsatz für diese Produktkategorie enthält, identifiziert. Der XPath-Ausdruck für diese automatische Berechnung lautet:

```
for $i in Category return /Invoice/Categories/Category[. = $i]/@rate.
```

Der Mehrwertsteuersatz wird in der Ausgabedatei an der Stelle angezeigt, an der die automatische Berechnung eingefügt wurde. Im Beispiel "Rechnung" ist die Nachschlagetabelle im selben XML-Dokument wie die

© 2019-2025 Altova GmbH

Rechnungsdaten gespeichert. Eine solche Tabelle kann aber auch in einem separaten Dokument gespeichert werden. In diesem Fall würde sie mit Hilfe der `doc()` Funktion von XPath 2.0 aufgerufen. Beachten Sie, dass der Mehrwersteuerwert von unterschiedlichen Produkten unterschiedlich ist (`Book=10%`; `CD=15%`; `DVD=15%`; `Electronics=20%`). Sie wurden mittels der automatischen Berechnung berechnet.

Erstellen einer komplexen automatischen Berechnung

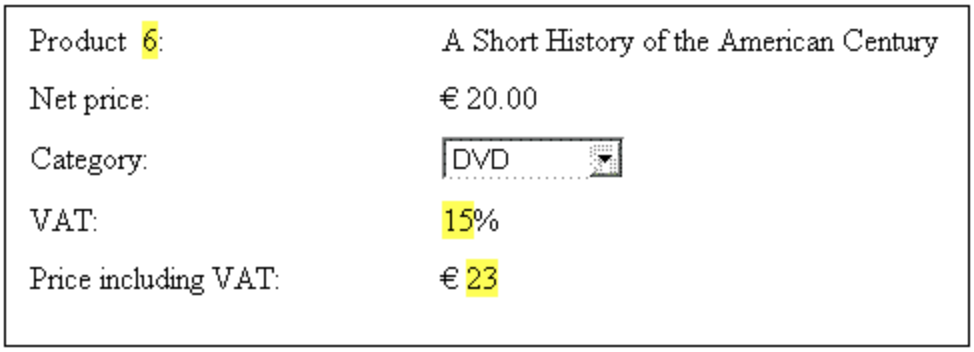
Der durch die oben beschriebene automatische Berechnung ermittelte Mehrwertsteuersatz wird benötigt, um den Bruttopreis (Nettopreis + Mwst.) der einzelnen Produkte zu ermitteln. Die Formel, die hierfür benötigt wird, würde folgendermaßen abgeleitet werden:

```
Bruttopreis = Nettopreis + Mwst-Betrag
Da Mwst-Bestrag = Nettopreis * Mwstsatz div 100
Bruttopreis = Nettopreis + (Nettopreis * Mwst-Betrag div 100)
```

Der Nettopreis eines Produkts wird aus dem Node `PriceNet` abgerufen. Der Mehrwertsteuersatz wird mittels der oben beschriebenen automatischen Berechnung ermittelt. Es ist in keinem Node enthalten. Da dieser Wert nicht direkt aus einem Node abgerufen werden kann, muss er in der Berechnung des Bruttopreises neu berechnet werden. Der XPath-Ausdruck dafür würde lauten:

```
for $i in Category return PriceNet + (PriceNet *(/Invoice/Categories/Category[. =
          $i]/@rate) div 100)
```

Der XPath-Ausdruck kann im Fenter "Eigenschaften" angezeigt und bearbeitet werden 257. Sie können die automatische Berechnung für den Bruttopreis testen, indem Sie in der XML-Datei entweder den Preis oder die Produktkategorie eines beliebigen Produkts ändern und das SPS anschließend neu laden. Beachten Sie, dass sich auch der Bruttopreis (Preis inkl. Mwst) des Produkts ändert.



7.2 Bedingungen

Sie können überall im Design Bedingungen einfügen, sowohl in der Hauptvorlage als auch in globalen Vorlagen. Eine Bedingung ist eine SPS Komponente, die aus einem oder mehreren Verzweigungen (Branches) besteht, wobei jede Verzweigung als XPath-Ausdruck definiert ist. Angenommen, Sie haben eine aus zwei Verzweigungen bestehende Bedingung. Der XPath-Ausdruck für die erste Verzweigung überprüft, ob der Wert im Attribut `Location` des Kontext-Node "`US`" ist. Der XPath-Ausdruck der zweiten Verzweigung überprüft, ob der Wert des Attributs `Location` "`EU`" ist. Jede Verzweigung enthält eine Vorlage, eine so genannte Bedingungsvorlage. Wenn ein Node mit einer Bedingung verarbeitet wird, so wird die erste Verzweigung, bei dessen Auswertung das Ergebnis "true" ergibt, ausgeführt, d.h. die Bedingungsvorlage wird verarbeitet und die Bedingung wird beendet; es werden keine weiteren Verzweigungen dieser Bedingung ausgewertet. Auf diese Art können Sie je nach Node-Wert unterschiedliche Vorlagen verwenden. Im oben angeführten Beispiel könnten Sie für die Standorte US und EU unterschiedliche Vorlagen verwenden.

Dieser Abschnitt enthält die folgenden Themen:

- Definieren der Bedingungen 262: Hier wird beschrieben, wie eine Bedingung und ihre Verzweigungen erstellt werden.
- Bearbeiten von Bedingungen 265: Hier erfahren Sie, wie XPath-Ausdrücke von Bedingungszweigen nach ihrer Erstellung bearbeitet werden.
- Bedingungen und automatische Berechnungen 266: Hier wird erklärt, wie Bedingungen und automatische Berechnungen gemeinsam verwendet werden.

7.2.1 Definieren der Bedingungen

Das Definieren der Bedingung erfolgt folgendermaßen:

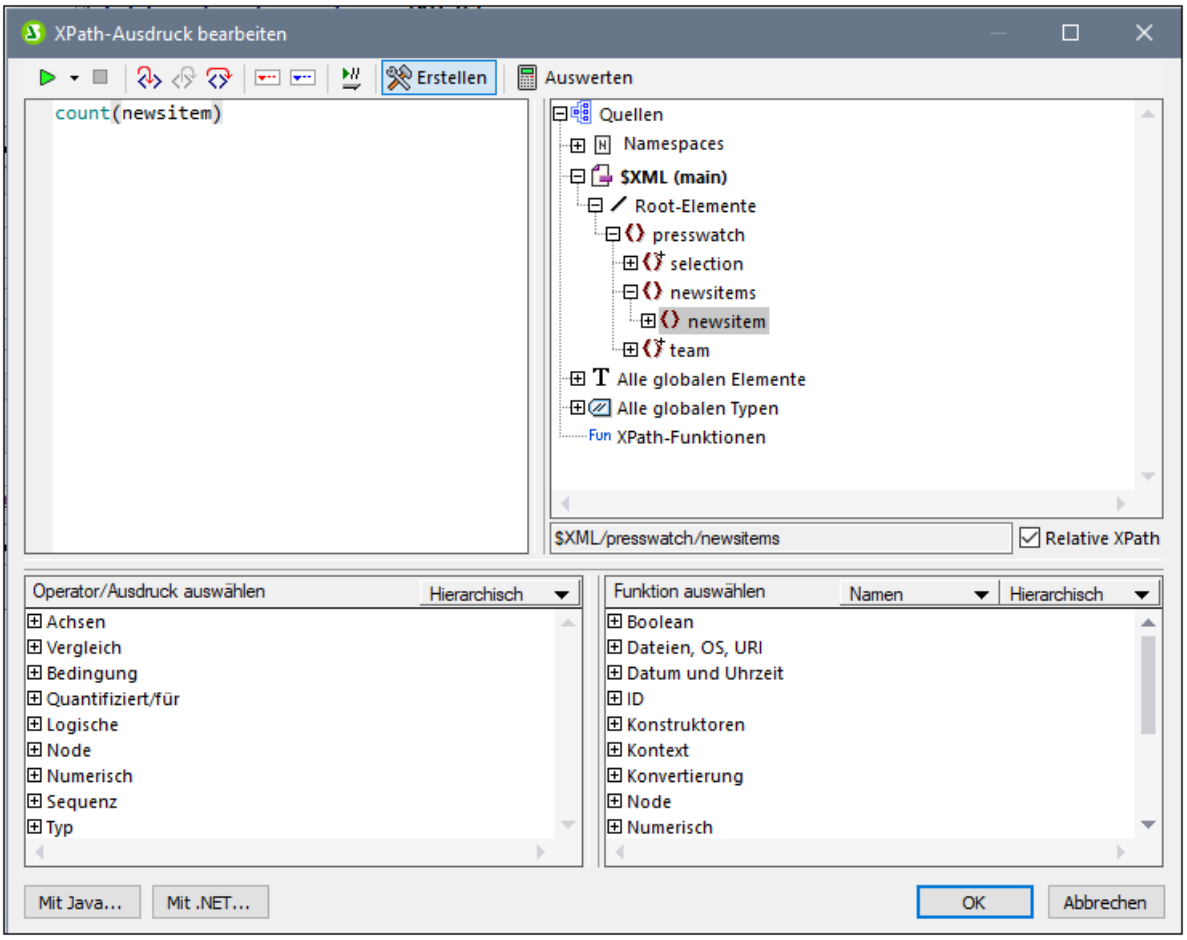
1. Erstellen Sie die Bedingung mit ihrer ersten Verzweigung.
2. Erstellen Sie zusätzliche Verzweigungen für eine alternative Verarbeitung.
3. Erstellen und bearbeiten Sie die Vorlagen in den verschiedenen Bedingungszweigen.

Erstellen der Bedingung mit ihrer ersten Verzweigung

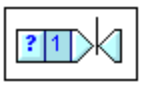
Gehen Sie beim Erstellen der Bedingung folgendermaßen vor:

1. Platzieren Sie den Cursor an eine beliebige Stelle im Design oder wählen Sie eine Komponente aus und anschließend den Menübefehl **Einfügen | Bedingung**. Daraufhin wird das Dialogfeld "XPath-Ausdruck bearbeiten 440" angezeigt (*Abbildung unten*).

© 2019-2025 Altova GmbH



2. Geben Sie im Bereich "Ausdruck" über die Tastatur den XPath-Ausdruck für den Bedingungszweig ein. Alternativ dazu können Sie den Ausdruck auch durch Doppelklick auf Nodes, Operatoren und/oder Funktionen in den Bereichen des Dialogfelds eingeben. Beachten Sie, in welchem Kontext-Node der Einfügepunkt ist; der Kontext-Node wird beim Anzeigen des Dialogfelds in der Struktur der Schemaquellen markiert angezeigt.
3. Klicken Sie zum Fertigstellen auf **OK**. Die Bedingung wird mit ihrer ersten Verzweigung erstellt; der XPath-Ausdruck, den Sie eingegeben haben, ist der XPath-Ausdruck der ersten Verzweigung. Wenn die Bedingung an einem Texteinfügepunkt eingefügt wurde, so ist die erste Verzweigung leer (es gibt darin keine Vorlage; *siehe Abbildung unten*). Wenn die Bedingung eingefügt wurde, während eine Komponente ausgewählt war, so wird die Bedingung rund um die Komponente erstellt und die Komponente wird zur Vorlage für die erste Verzweigung.

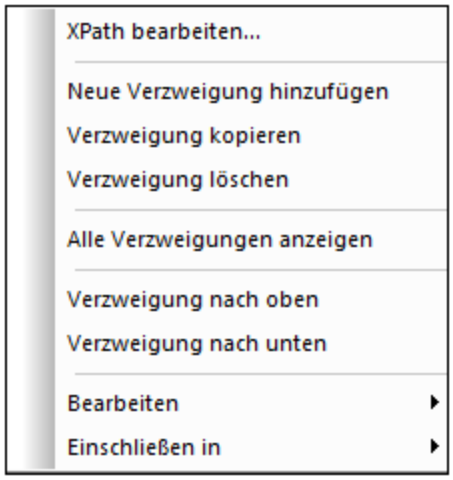


Um die gesamte Bedingung auszuwählen, klicken Sie auf die Zelle mit dem Fragezeichen. Um die erste Verzweigung auszuwählen, klicken Sie auf die Zelle mit der Nummer eins.

Nachdem Sie eine Bedingung mit einer Verzweigung erstellt haben (der eine Vorlage enthält oder auch nicht), können Sie so viele zusätzliche Verzweigungen wie nötig erstellen.

Erstellen zusätzlicher Verzweigungen

Zusätzliche Verzweigungen werden der Reihe nach erstellt. Eine zusätzliche Verzweigung wird über das Kontextmenü (*Abbildung unten*) erstellt und kann auf zwei Arten erstellt werden: (i) ohne eine Vorlage darin (**Neue Verzweigung hinzufügen**); und (ii) mit einer Kopie einer vorhandenen Vorlage in der neuen Verzweigung (**Verzweigung kopieren**).



Um eine neue Verzweigung zu erstellen, rechtsklicken Sie auf eine Verzweigung der Bedingung und wählen Sie im Kontextmenü den Befehl **Neue Verzweigung hinzufügen**. Daraufhin wird das Dialogfeld "XPath-Ausdruck bearbeiten[440]" angezeigt. Nachdem Sie einen XPath-Ausdruck eingegeben haben und auf OK geklickt haben, wird eine neue leere Verzweigung zur Bedingung hinzugefügt. Dies wird im Design dadurch angezeigt, dass zur Bedingung eine neue Zelle hinzugefügt wird; die neue Zelle wird im Anschluss an die letzte Verzweigung fortlaufend nummeriert.

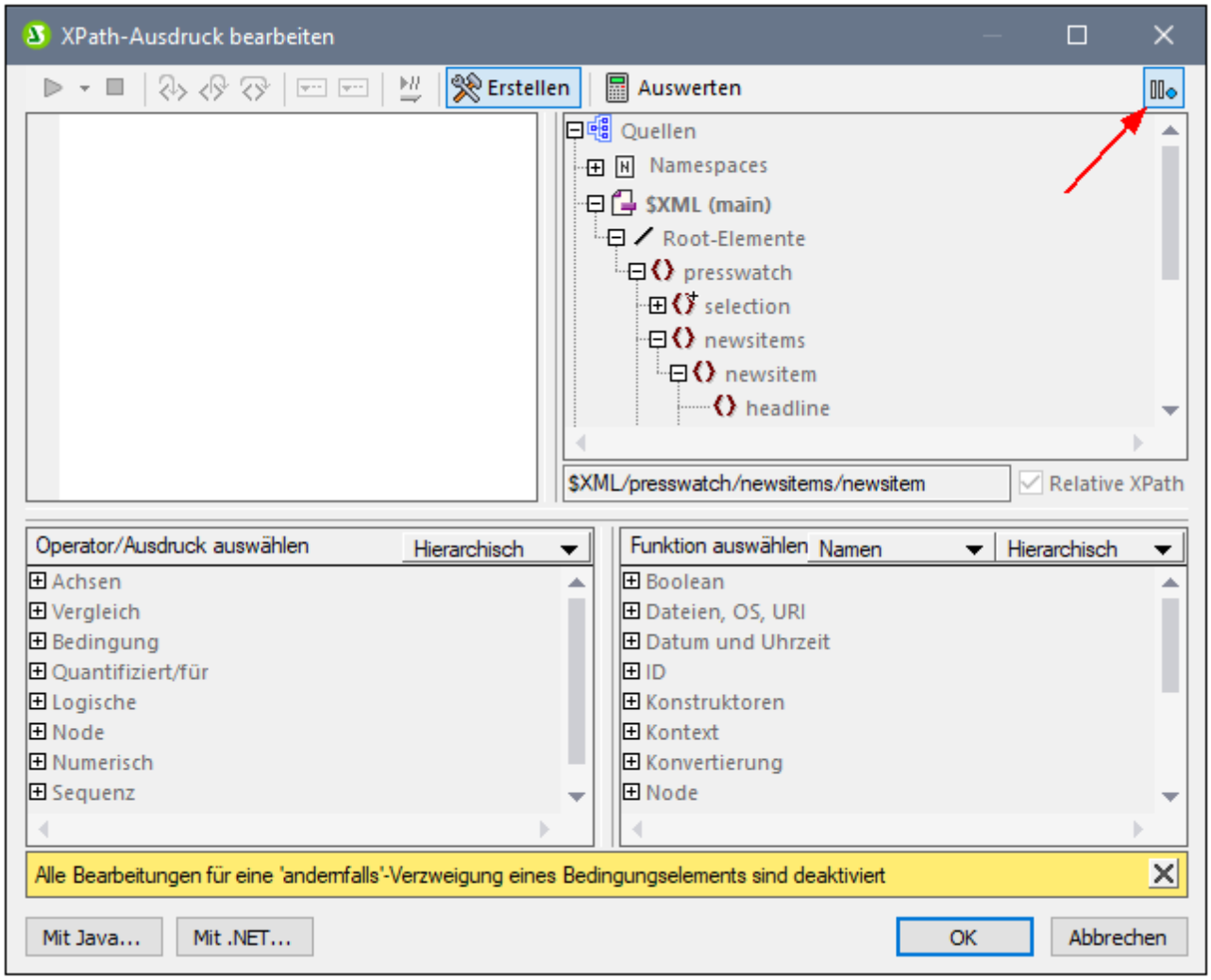
Um eine Kopie einer vorhandenen Verzweigung zu erstellen, rechtsklicken Sie auf die gewünschte Verzweigung der Bedingung und wählen Sie **Verzweigung kopieren**. Daraufhin wird das Dialogfeld "XPath-Ausdruck bearbeiten[440]" angezeigt. Es enthält den XPath-Ausdruck der gerade kopierten Verzweigung. Nachdem Sie den XPath-Ausdruck geändert und auf **OK** geklickt haben, wird eine neue Verzweigung zur Bedingung hinzugefügt. Die neue Verzweigung enthält eine Kopie der Vorlage der kopierten Verzweigung. Die neue Verzweigung wird im Design durch eine neue im Anschluss an die vorherigen Verzweigungen fortlaufend nummerierte Zelle dargestellt.

Die Verzweigung "Andernfalls"

Die Verzweigung `Andernfalls` ist eine alternative Verzweigung, in der eine bestimmte Art der Verarbeitung (Vorlage) definiert ist, die angewendet werden soll, wenn das Ergebnis bei keiner der definierten Verzweigungen "true" ist. Ohne die Verzweigung `Andernfalls` müssten Sie Verzweigungen für alle möglichen Eventualitäten erstellen, um nicht zu riskieren, dass die Bedingungsvorlage womöglich beendet wird, ohne dass eine Verzweigung ausgeführt wird.

Um eine `Andernfalls` Verzweigung einzufügen, verwenden Sie entweder den Befehl **Neue Verzweigung hinzufügen** oder **Verzweigung kopieren** wie oben beschrieben und aktivieren Sie im Dialogfeld "XPath-Ausdruck bearbeiten[440]" das Kontrollkästchen "Andernfalls" (*Abbildung unten*).

© 2019-2025 Altova GmbH



Verschieben von Verzweigungen nach oben oder unten

Es ist wichtig, in welcher Reihenfolge die Verzweigungen in der Bedingung angeordnet sind, da die erste Verzweigung, deren Ergebnis "true" ist, ausgeführt wird und die Bedingung anschließend beendet wird. Um Verzweigungen nach oben oder unten zu verschieben, wählen Sie die gewünschte Verzweigung aus, klicken Sie mit der rechten Maustaste darauf und wählen Sie **Verzweigung nach oben** oder **Verzweigung nach unten**.

Löschen einer Verzweigung

Um eine Verzweigung zu löschen, wählen Sie sie aus, rechtsklicken Sie und wählen Sie **Verzweigung löschen**.

7.2.2 Bearbeiten von Bedingungen

Um den XPath-Ausdruck einer Bedingungsverzweigung zu bearbeiten, gehen Sie folgendermaßen vor:

1. Wählen Sie die Verzweigung der Bedingung (nicht die Bedingung) aus.

2. Wählen Sie in der Seitenleiste "Eigenschaften" in der Spalte "Eigenschaften für" `Bedingungsverweigung` aus (*Abbildung unten*).

3. Klicken Sie in der Eigenschaftsgruppe *when* auf die **Bearbeiten**-Schaltfläche ... der Eigenschaft `XPath`. Daraufhin wird das Dialogfeld "XPath-Ausdruck bearbeiten 440" angezeigt. Hier können Sie den XPath-Ausdruck für diese Verzweigung bearbeiten.

7.2.3 Bedingungen und automatische Berechnungen

Wenn Sie Bedingungen und automatische Berechnungen gemeinsam verwenden, sind ein paar Dinge zu beachten. Vor allem die folgenden zwei Punkte sind von Bedeutung:

- Nur automatische Berechnungen in **sichtbaren Bedingungen -** also in der Verzweigung, die als "true" ausgewählt ist - werden ausgewertet.
- Automatische Berechnungen werden vor Bedingungen ausgewertet.

Hier einige Richtlinien zur Zusammenfassung dieser Punkte
.

1. Wenn ein Node mit Hilfe einer automatischen Berechnung aktualisiert wird und wenn dieser Node in einer Bedingung vorkommt (entweder, weil er im XPath-Ausdruck einer Verzweigung vorkommt oder im Inhalt einer Bedingungsvorlage), so achten Sie darauf, dass sich die automatische Berechnung wenn möglich außerhalb der Bedingung befindet. Damit stellen Sie sicher, dass die automatische Berechnung immer sichtbar ist, egal welche Verzweigung der Bedingung sichtbar ist. Befände sich die automatische Berechnung innerhalb einer Verzweigung, die nicht sichtbar ist, würde sie nicht ausgelöst.
2. Wenn eine automatische Berechnung in eine Bedingung platziert werden muss, stellen Sie sicher (i) dass sie in jede Verzweigung der Bedingung platziert wird und (ii) dass die verschiedenen Verzweigungen der Bedingung alle möglichen Bedingungen abdecken. Es sollte keine Eventualität geben, für die es keine Bedingung in der Bedingungsvorlage gibt; andernfalls (wenn sich die automatische Berechnung nicht in einer sichtbaren Vorlage befindet) besteht das Risiko, dass die automatische Berechnung nicht ausgelöst wird.
3. Wenn für verschiedene Bedingungen unterschiedliche automatische Berechnungen benötigt werden, stellen Sie sicher, dass alle möglichen Eventualitäten für jede automatische Berechnung abgedeckt sind.
4. Es ist wichtig, in welcher Reihenfolge Bedingungen in einer Bedingungsvorlage definiert sind. Die erste Bedingung, deren Auswertung "true" ergibt, wird ausgeführt. Die Bedingung `Andernfalls` ist eine praktische Methode, um für alle Eventualitäten ein Ergebnis zu definieren.

© 2019-2025 Altova GmbH

7.3 Gruppieren

Die Gruppierungsfunktion steht in **XSLT 2.0 und 3.0** SPSs und für HTML-Ausgabedokumente zur Verfügung.

Mit der Gruppierungsfunktion können Sie Elemente (im Allgemeinen Nodes) in Gruppen verarbeiten. Nehmen wir als Beispiel eine Autoinventarliste, in der die Details zu den einzelnen Autos unter dem Element `car` enthalten sind. Wenn das Element `car` z.B. ein Attribut `brand` hat, können die Autos nach Marken (brand) gruppiert werden. Dies kann aus verschiedensten Gründen hilfreich sein. z.B:

- Alle Autos einer einzigen Marke können unter der Überschrift des Markennamens in der Ausgabe angezeigt werden.
- Operationen können innerhalb einer Gruppe ausgeführt werden und die Ergebnisse dieser Operation können separat für jede einzelne Gruppe angezeigt werden. So kann z.B. die Anzahl der für die einzelnen Marken verfügbaren Modelle aufgelistet werden.

Außerdem kann eine Gruppe weiter in Untergruppen unterteilt und verarbeitet werden. So können Autos z.B. innerhalb der einzelnen Marken nach Modell und anschließend Jahr gruppiert werden.

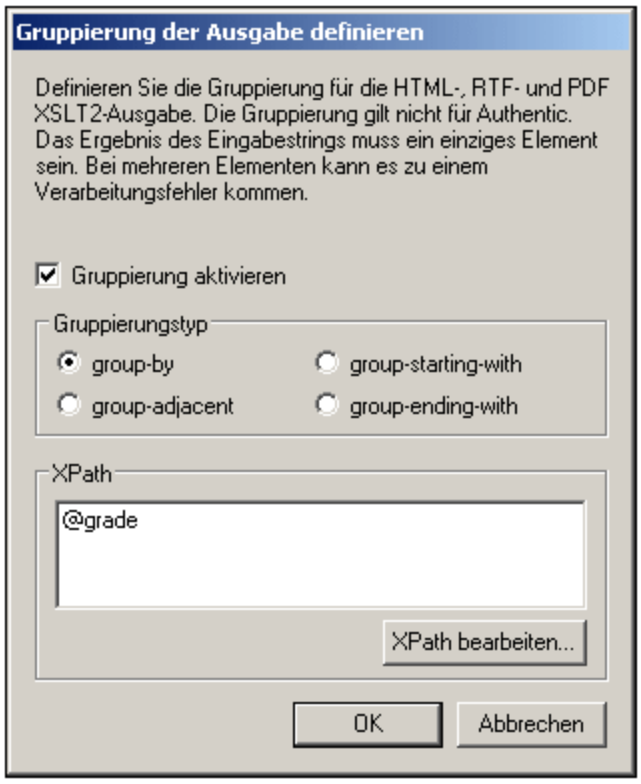
Gruppierungskriterien

Elemente können nach zwei allgemeinen Kriterien gruppiert werden: (i) einem Gruppierungsschlüssel, der normalerweise den Wert eines Node überprüft und (ii) der relativen Position des Elements. Es stehen die folgenden spezifischen Gruppierungskriterien zur Verfügung:

- ***group-by***: Gruppiert die Elemente auf Basis eines XPath-definierten Schlüssels. So können z.B. `car` Elemente nach ihren `brand` Attributen gruppiert werden. Die Gruppierung wird nach dem Element `car` vorgenommen. Ein XPath-Ausdruck wählt das `brand` Attribut aus.
- ***group-adjacent*** verwendet eine Kombination von Gruppierungsschlüsseln und Positionskriterien. Alle benachbarten Elemente, die denselben Wert für den Gruppierungsschlüssel haben, werden in eine Gruppe inkludiert. Wenn sich der Gruppierungsschlüsselwert eines Elements von dem des vorhergehenden Elements unterscheidet, wird mit diesem Element eine neue Gruppe begonnen.
- ***group-starting-with*** beginnt eine neue Gruppe, wenn ein Node einem definierten XPath-Muster entspricht. Wenn ein Node nicht mit dem definierten XPath-Muster übereinstimmt, wird er der aktuellen Gruppe zugewiesen.
- ***group-ending-with*** beendet eine Gruppe, wenn ein Node einem definierten XPath-Muster entspricht; der übereinstimmende Node ist der letzte in dieser Gruppe. Der nächste Node beginnt eine neue Gruppe. Wenn ein darauf folgender Node, der eine Gruppe beginnt, nicht dem definierten XPath-Muster entspricht, wird er der aktuellen Gruppe zugeordnet.

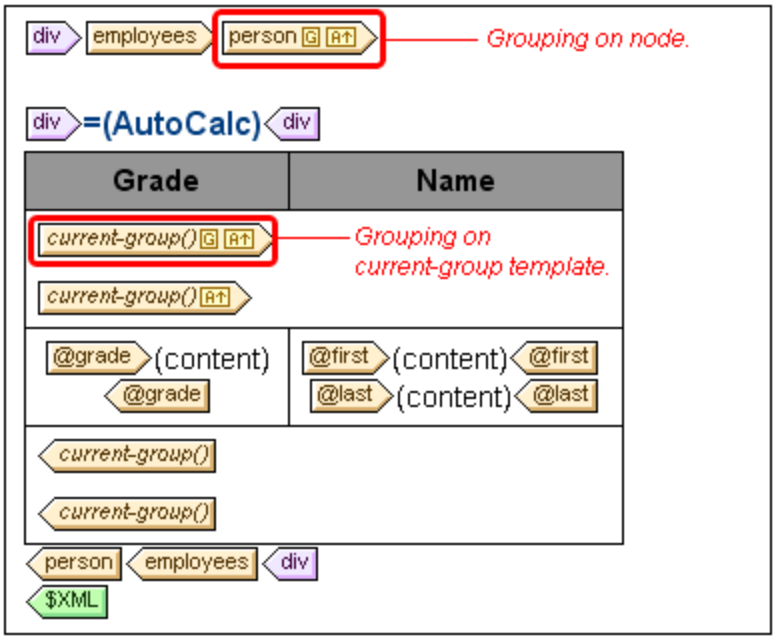
Erstellen von Gruppen

Gruppen können über das Kontextmenü entweder an einem Node oder einer Vorlage für eine aktuelle Gruppe erstellt werden. Klicken Sie zum Erstellen einer Gruppe mit der rechten Maustaste auf einen Node oder eine current-group-Vorlage und wählen Sie im daraufhin angezeigten Kontextmenü den Befehl **Gruppieren nach**. Daraufhin wird das Dialogfeld "Gruppierung der Ausgabe definieren" angezeigt (*Abbildung unten*).



Aktivieren Sie in dem Dialogfeld das Kontrollkästchen "Gruppierung aktivieren", wählen Sie anschließend den Gruppierungstyp aus und geben Sie im Textfeld "XPath" den XPath-Ausdruck ein, der den Gruppierungsschlüssel (für die Optionen *group-by* und *group-adjacent)* oder das gewünschte Muster (für die Optionen *group-starting-with* und *group-ending-with)* definiert. Wenn Sie auf **OK** klicken, wird ein Dialogfeld angezeigt, in dem Sie gefragt werden, ob die Gruppe alphabetisch (in aufsteigender Reihenfolge) sortiert werden soll. Sie können Gruppen auch später noch sortieren oder die Sortierung aufheben. In der Abbildung unten werden Nodes und current-group-Vorlagen angezeigt, zu denen eine Gruppierung hinzugefügt wurde.

© 2019-2025 Altova GmbH



In der Abbildung oben wurde der Node `person` gruppiert und die erzeugten Gruppen wurden sortiert. Wenn z.B. die `person` Elemente nach Abteilung gruppiert wurden, können die verschiedenen Abteilungen alphabetisch in aufsteigender Reihenfolge sortiert werden. Die dadurch erstellten Gruppen wurden in Untergruppen unterteilt, indem an der current-group()-Vorlage eine Gruppierung erstellt wurde. Auf diese Art können die `person` Elemente z.B. zuerst nach Abteilung und anschließend nach Anstellungsart gruppiert werden.

Sortieren von Gruppen

Nachdem Sie eine Gruppendefinition bestätigt haben, erscheint ein Fenster, in dem Sie gefragt werden, ob die Gruppen in aufsteigender Reihenfolge sortiert werden sollen. Sie können die Sortierung später jederzeit definieren, ändern oder aufheben.

Um die Sortierung später definieren, ändern oder aufheben zu können, klicken Sie mit der rechten Maustaste auf die Gruppierungsvorlage und wählen Sie den Befehl **Sortieren nach**. Daraufhin erscheint das Dialogfeld Sortierreihenfolge der Ausgabe definieren 276. Eine Beschreibung zur Verwendung dieses Dialogfelds finden Sie im Abschnitt Sortieren 276. Beachten Sie dabei: Um Gruppen anhand ihres Gruppierungsschlüssels zu sortieren, müssen Sie die XPath-Funktion `current-grouping-key()` als Sortierschlüssel auswählen. Als Beispiele dienen die in den folgenden Abschnitten beschriebenen Dateien.

Anzeigen und Bearbeiten von Gruppierungs- und Sortierungseinstellungen

Um die Gruppierung und Sortierung in einer Vorlage anzuzeigen und zu bearbeiten, klicken Sie mit der rechten Maustaste auf die Vorlage und wählen Sie "Gruppieren nach" bzw. "Sortieren nach". Daraufhin erscheint das entsprechende Dialogfeld, in dem Sie die Einstellungen anzeigen oder ändern können.

Benutzerdefinierte Vorlagen

Benutzerdefinierte Vorlagen 235, sind Vorlagen, die auf Elemente angewendet werden, die durch einen von Ihnen definierten XPath-Ausdruck ausgewählt werden. Die Nodes, die durch den XPath-Ausdruck einer benutzerdefinierten Vorlage ausgewählt werden, können auch gruppiert werden. In diesem Fall wird die Gruppierung an der benutzerdefinierten Vorlage angewendet.

7.3.1 Beispiel: Group-By (Persons.sps)

Im Beispiel `Persons.sps`, das auf dem Schema `Persons.xsd` basiert, wird als XML-Arbeitsdatei `Persons.xml` verwendet. Es befindet sich im Ordner (Eigene) Dateien[25], `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2025\StyleVisionExamples\Tutorials\Grouping\Persons\`. Die XML-Dokumentstruktur ist die folgende: Ein `employees` Dokumentelement kann unbeschränkt viele Angestellte `person` enthalten. Jeder Angestellte `person` ist nach dem folgenden Muster strukturiert:

```
<person first="Vernon" last="Callaby" department="Administration" grade="C"/>
```

Im Design werden die Angestellten nach Abteilung gruppiert. Für jede Abteilung gibt es eine separate Tabelle und die Abteilungen werden aufsteigend in alphabetischer Reihenfolge sortiert. Innerhalb der einzelnen Abteilungstabellen werden die Angestellten (person) nach Stufen (grade) (in aufsteigender Reihenfolge sortiert) gruppiert und innerhalb jeder Stufe werden die Angestellten in alphabetischer Reihenfolge nach Nachnamen aufgelistet.

Strategie

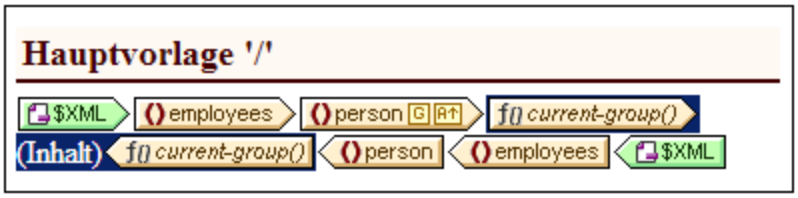
Die Strategie zum Erstellen von Gruppen ist die folgende. Die Gruppierung wird am Element "person" vorgenommen, wobei als Gruppierungsschlüssel das Attribut "department" verwendet wird. Dadurch werden die person-Elemente anhand des Werts im Attribut "department" in Gruppen eingeteilt. (Wird eine Sortierung definiert, können die Department-Gruppen in alphabetischer Reihenfolge aufgelistet werden, z.B. beginnend mit "Administration", usw.). Da für die Abteilungen separate Tabellen angelegt werden sollen, wird die aktuelle Gruppierung (current-grouping) (welche, auf dem Gruppierungsschlüssel "department" basiert) als Tabelle erstellt. Innerhalb dieser gruppierten Person-Elemente definieren wir nun, dass jede Gruppe noch weiter geordnet werden soll, und zwar mit dem Attribut "grade" als Gruppierungsschlüssel.

Erstellen des SPS

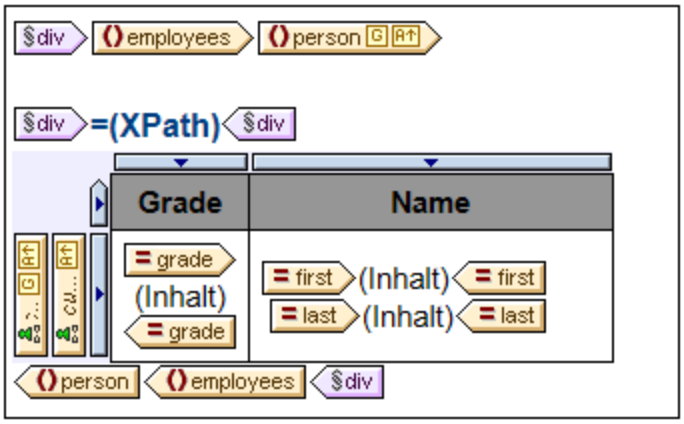
Das Design wurde folgendermaßen erstellt:

1. Ziehen Sie das Element `person` aus der Schemastruktur und erstellen Sie es als Inhalt.
2. Klicken Sie mit der rechten Maustaste auf den Element-Tag `person` und wählen Sie im Kontextmenü den Befehl "Gruppieren nach".
3. Aktivieren Sie im Dialogfeld "Gruppierung der Ausgabe definieren" die Option *group-by*, geben Sie als XPath-Ausdruck im Textfeld "XPath" `@department` ein und klicken Sie auf **OK**.
4. Es wird ein Dialogfeld angezeigt, in dem Sie gefragt werden, ob die Gruppen sortiert werden sollen. Da die Gruppen nach der Standardsortierung alphabetisch in aufsteigender Reihenfolge sortiert werden sollen, klicken Sie auf **Ja**. (Die Sortierung kann später jederzeit definiert, geändert oder aufgehoben werden.)
5. Da jede Gruppe (in diesem Fall die Abteilungen) in Form einer separaten Tabelle angelegt werden soll, erstellen Sie die aktuelle Gruppe als Tabelle. Klicken Sie dazu mit der rechten Maustaste auf den Tag `current-group()` (*siehe Abbildung unten*) und wählen Sie den Befehl **Ändern in | Tabelle**, wobei Sie die Child-Attribute `@last` und `@grade` als Spalten der Tabelle auswählen.

© 2019-2025 Altova GmbH



6. Strukturieren Sie den Inhalt der Spalten und Zellen der Tabelle so um, dass die erste Spalte `@grade` und die zweite Spalte die Nodes `@first` und `@last` enthält (*siehe Abbildung unten*).
7. Um innerhalb der aktuellen Gruppen, die nach "department" gruppiert ist, eine Gruppierung nach "grade" vorzunehmen, erstellen Sie in der `current-group()`-Vorlage eine Gruppierung für das Attribut `grade`. Bestätigen Sie die Standardsortierung. Daraufhin wird eine neue Vorlage `current-group()` erstellt (siehe Abbildung oben).
8. Sortieren Sie die aktuelle Gruppe (also die Untergruppe von Angestellten, die nach Stufe sortiert ist) nach dem Attribut `last`.



9. Definieren Sie die Formatierung für die Tabelle.
10. Definieren Sie oberhalb der Tabelle eine Überschrift für die Tabelle. Da die Tabellen einzelne Abteilungen repräsentieren, kann der Name der Abteilung durch eine automatische Berechnung mittels eines XPath-Ausdrucks, der die XPath 2.0/3.0-Funktion `current-grouping-key()` aufruft, dynamisch aus dem aktuellen Kontext ermittelt werden.
11. Wiederholen Sie den gesamten Prozess, um eine ähnliche Ausgabe zu erstellen, dieses Mal durch Gruppierung von Angestellten nach Stufe (grade) und erst dann nach Abteilung (department).

Um die Gruppierung oder Sortierung einer Vorlage anzuzeigen oder zu ändern, klicken Sie mit der rechten Maustaste auf die Vorlage und wählen Sie im Kontextmenü den Befehl **Gruppieren nach** oder **Sortieren nach.** Daraufhin wird das entsprechende Dialogfeld angezeigt, in dem die Einstellungen angezeigt oder geändert werden können.

7.3.2 Beispiel: Group-By (Scores.sps)

Im Beispiel `Scores.sps`, das auf dem Schema `Scores.xsd` basiert, wird als XML-Arbeitsdatei `Scores.xml` verwendet. Es befindet sich im Ordner (Eigene) Dateien[25], `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2025\StyleVisionExamples\Tutorials\Grouping\Scores\`. Die Struktur des XML-Dokuments ist die folgende: Ein Dokumentelement `results` enthält ein oder mehrere `group` Elemente

sowie ein oder mehrere `match` Elemente. Ein `group` Element enthält ein oder mehrere `team` Elemente und ein `match` Element ist wie im folgenden Beispiel strukturiert:

```
<match group="A" date="2007-10-12">
       <team name="Brazil" for="2" points="3"/>
       <team name="Germany" for="1" points="0"/>
</match>
```

Das Design besteht aus drei Teilen (*Abbildung unten*): (i) den Matchergebnissen, aufgelistet nach Tag (gruppiert nach `//match/@date`); (ii) den Matchergebnissen, aufgelistet nach Gruppe (gruppiert nach `//match/@group`); und (iii) Gruppentabellen mit einer Rangliste nach Gruppe (eine dynamische Tabelle des group-Elements, wobei die benötigten Daten mittels einer automatischen Berechnung ermittelt werden).

© 2019-2025 Altova GmbH

Match Results: Day-by-Day

2007-10-12

Brazil - Germany	2 - 1
Italy - Holland	2 - 2

2007-10-13

Argentina - France	2 - 0
England - Spain	0 - 0

Match Results: By Group

Group A

Brazil - Germany	2 - 1
Italy - Holland	2 - 2
Brazil - Italy	1 - 2
Germany - Holland	2 - 2
Brazil - Holland	1 - 0
Germany - Italy	1 - 1

Group Tables

Group A

Team	P	W	D	L	F	A	Pts
Brazil	3	2	0	1	4	3	6
Italy	3	1	2	0	5	4	5
Germany	3	0	2	1	4	5	2
Holland	3	0	2	1	4	5	2

Strategie

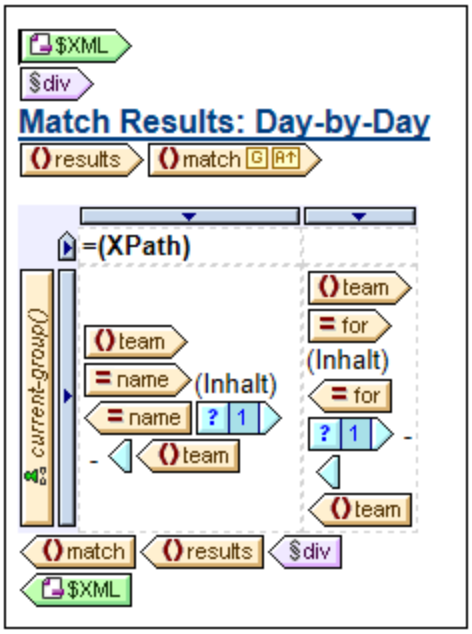
Für die beiden Abschnitte, die die Match-Ergebnisse enthalten, gruppieren wir Matches nach Datum und Turniergruppe. Für die Mitglieder der einzelnen Gruppen (Datum und Turniergruppe) erstellen wir randlose Tabellen (um den Inhalt ausrichten zu können). Auf diese Weise befinden sich Matches, die an einem bestimmten Datum stattfinden, in einer separaten Tabelle und alle Match-Ergebnisse einer einzelnen

Turniergruppe befinden sich ebenfalls in einer separaten Tabelle (z.B. Matches der Gruppe A). Für den Abschnitt der Gruppentabellen wird das `group` Element als dynamische Tabelle erstellt, wobei die benötigten Daten mittels einer automatischen Berechnung ermittelt werden.

Erstellen des SPS

Das Design wurde folgendermaßen erstellt:

1. Ziehen Sie das `/results/match` Element aus der Schemastruktur und erstellen Sie es als Inhalt.
2. Klicken Sie mit der rechten Maustaste auf den Element-Tag `match` und wählen Sie im Kontextmenü den Befehl **Gruppieren nach**.
3. Aktivieren Sie im Dialogfeld "Gruppierung der Ausgabe definieren" die Option *group-by*, setzen Sie den XPath-Ausdruck im Textfeld "XPath" auf `@date und` klicken Sie auf **OK**.
4. Daraufhin wird ein Dialogfeld angezeigt, in dem Sie gefragt werden, ob die Gruppen sortiert werden sollen. Da die Gruppen nach der Standardsortierung alphabetisch in aufsteigender Reihenfolge sortiert werden sollen, klicken Sie auf **Ja**. (Die Sortierung kann später jederzeit definiert, geändert oder aufgehoben werden.)
5. Da jede Gruppe (in diesem Fall das Datum) in Form einer separaten Tabelle angelegt werden soll, erstellen Sie die aktuelle Gruppe als Tabelle. Klicken Sie dazu mit der rechten Maustaste auf den Tag `current-group()` und wählen Sie den Befehl **Ändern in | Tabelle**, wobei Sie die untergeordneten Nodes `team/@name` und `team/@for` als Spalten der Tabelle auswählen (*siehe Abbildung unten*).



6. Setzen Sie einen Bindestrich in jede Zelle, die ausgegeben wird, wenn das Match nicht das letzte in der aktuellen Gruppe ist (Verwenden Sie dazu eine Bedingungsvorlage mit einer Bedingung `position() != last()`. Als Ergebnis erhalten Sie dann z.B.: `Brazil - Germany` oder `2 - 1`.
7. Fügen Sie in die Kopfzeile der Tabelle eine automatische Berechnung ein, die den aktuellen Gruppierungsschlüssel für die entsprechende Gruppe ausgibt (XPath-Ausdruck: `current-grouping-key()`).
8. Formatieren Sie die Tabelle wie gewünscht.
9. Um die Matches nach Turniergruppe zu gruppieren, wiederholen Sie den gesamten Vorgang, gruppieren Sie die Matches dieses Mal aber nach dem `group` Attribut von `match`.

© 2019-2025 Altova GmbH

10. Erstellen Sie für die Gruppentabellen (im dritten Abschnitt des Designs), die die Rangliste der Teams in der Gruppe enthalten, das `/results/group` Element als dynamische Tabelle. Fügen Sie wie erforderlich (mit Hilfe der Befehle **Tabelle | Spalte anhängen** oder **Tabelle | Spalte einfügen**) Spalten hinzu. Definieren Sie in jeder Spalte automatische Berechnungen, um die erforderliche Ausgabe zu berechnen (3 Punkte für einen Sieg; 1 Punkt für ein Unentschieden; 0 Punkte für eine Niederlage). Und sortieren Sie abschließend die Tabelle nach der Gesamtzahl der Punkte in absteigender Reihenfolge. Um den XPath-Ausdruck zur Berechnung dieser Ergebnisse zu sehen, klicken Sie mit der rechten Maustaste auf die automatische Berechnung bzw. die sortierte Vorlage und wählen Sie **XPath bearbeiten** bzw. **Sortieren nach**.

7.4 Sortieren

Die Sortierfunktion steht für die HTML-Ausgabe zur Verfügung.

Eine Gruppe von gleichrangigen Element-Nodes desselben qualifizierten Namens kann nach einem oder mehreren von Ihnen ausgewählten Sortierschlüsseln sortiert werden. So können z.B. alle `Person` Elemente (z.B. innerhalb von einem `Company` Element) nach dem Child-Element `LastName` des Elements `Person` sortiert werden. Der Sortierschlüssel muss ein Node im Dokument sein und ist normalerweise ein Nachkomme (Element oder Attribut) des sortierten Element-Nodes. Im erwähnten Beispiel ist `LastName` der Sortierschlüssel.

Wenn es in der für die Sortierung vorgesehenen Gruppe zwei Elemente gibt, die Sortierschlüsselnamen mit demselben Wert haben, dann könnte ein zusätzlicher Sortierschlüssel für die weitere Sortierung verwendet werden. So könnte im oben erwähnten Beispiel `Person` zusätzlich zu einem ersten Sortierschlüssel `LastName` ein zweiter Sortierschlüssel `FirstName` definiert werden. So könnte für `Person`-Elemente mit dem selben `LastName`-Wert eine zusätzliche Sortierung nach `FirstName` durchgeführt werden. Auf diese Art können in einem SPS für eine einzige Sortieraktion mehrere Sortieranweisungen (jede mit einem einzigen Sortierschlüssel) definiert werden.

Die Vorlage wird auf die sortierte Gruppe angewendet und die Ergebnisse werden in der sortierten Reihenfolge an die Ausgabe gesendet. Die Sortierfunktion wird in der HTML-Ausgabe unterstützt.

Benutzerdefinierte Vorlagen

Benutzerdefinierte Vorlagen[235] sind Vorlagen, die auf Elemente angewendet werden, die mit Hilfe eines von Ihnen definierten XPath-Ausdrucks ausgewählt werden. Die durch den XPath-Ausdruck einer benutzerdefinierte Vorlage ausgewählten Nodes können auch sortiert werden. In diesem Fall wird die Sortierung an der benutzerdefinierten Vorlage angewendet.

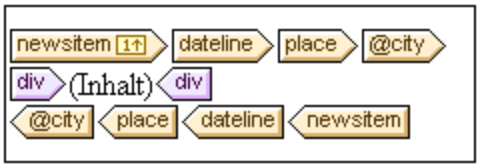
In diesem Abschnitt

- Beschreibung des Sortiermechanismus[276].
- Ein Beispiel[278] zur Beschreibung der Verwendung der Sortierfunktion.

7.4.1 Funktionsweise

Die Konfiguration eines Schema-Element-Nodes für die Sortierung erfolgt in zwei Schritten:

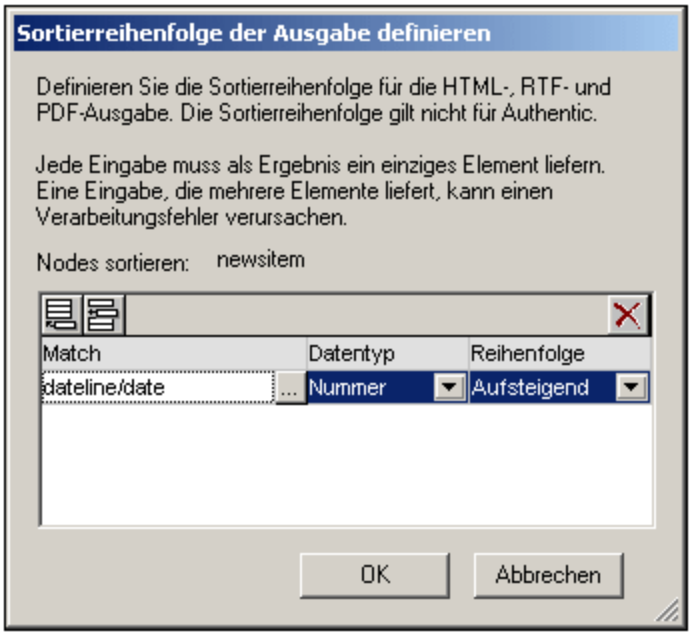
1. Wählen Sie in der Design-Ansicht den zu sortierenden Schema-Element-Node aus. Beachten Sie, dass im XML-Dokument die Instanzen **dieses** Elements sortiert werden. Oft ist es auf den ersten Blick nicht ersichtlich, welches Element sortiert werden soll. Werfen Sie dazu einen Blick auf die unten stehende Abbildung.



 © 2019-2025 Altova GmbH

Jedes `newsitem` hat eine `dateline`, die ein `place` Element mit einem `city` Attribut enthält. Die `@city` Nodes aller `newsitem` Elemente sollen in alphabetischer Reihenfolge ausgegeben werden. Soll im Design für die Sortierung der `@city` Node ausgewählt werden, oder eines der Elemente `place`, `dateline` oder `newsitem`? Bei Auswahl von `@city` wird nur der Node `city` sortiert. Bei Auswahl von `place` oder `dateline` wird wieder nur das jeweilige eine ausgewählte Element sortiert, da diese Elemente innerhalb ihrer Parent-Elemente nur einmal vorkommen. Bei Auswahl von `newsitem` können die `newsitem` Elemente jedoch nach dem Sortierschlüssel `dateline/place/@city` sortiert werden und es wird für jedes `newsitem` der `@city` Node ausgegeben. In diesem Fall ist es das `newsitem` Element, das sortiert werden sollte, da jeder Nachrichteneintrag (news item) ein dateline city enthält.

2. Nach Auswahl des zu sortierenden Elements klicken Sie im Kontextmenü, das Sie durch Rechtsklick auf die Elementauswahl aufrufen, auf den Befehl **Ausgabe sortieren**. Daraufhin wird das Dialogfeld "Sortierreihenfolge der Ausgabe definieren" (*Abbildung unten*) aufgerufen, in welchem Sie eine oder mehrere Sortieranweisungen einfügen oder anhängen können.



Jede Sortieranweisung enthält: (i) einen Sortierschlüssel (wird eingegeben in die Spalte "Match"); (ii) den Datentyp, der der Sortierschlüssel sein soll (Text oder Zahl); (iii) und die Sortierreihenfolge (aufsteigend oder absteigend). Die Reihenfolge, in der die Sortieranweisungen aufgelistet sind, ist von Bedeutung. Die Sortierung erfolgt, indem die Sortieranweisungen der Reihe nach, beginnend bei der ersten bis zur letzten ausgeführt werden. Es können beliebig viele Sortieranweisungen ausgeführt werden.

Ein Beispiel zur Verwendung der Sortierung finden Sie unter Beispiel: Sortieren nach mehreren Sortierschlüsseln 278.

Benutzerdefinierte Vorlagen

Benutzerdefinierte Vorlagen 235 sind Vorlagen, die auf Elemente angewendet werden, die mit Hilfe eines von Ihnen definierten XPath-Ausdrucks ausgewählt werden. Die durch den XPath-Ausdruck einer benutzerdefinierte Vorlage ausgewählten Nodes können auch sortiert werden. In diesem Fall wird die Sortierung an der benutzerdefinierten Vorlage angewendet.

Anmerkung zu Sortierschlüsseln

Die für den Sortierschlüssel eingegebenen XPath-Ausdrücke dürfen für jede Elementinstanz nur einen **einzigen Node** auswählen und kein Nodeset (XPath 1.0) und keine Elementsequenz (XPath 2.0 und XPath 3.0); der Schlüssel für jedes Element sollte sich als String oder Zahlenwert auflösen.

Wenn in einem **XSLT 2.0 oder 3.0** SPS der Sortierschlüssel eine Sequenz von Nodes zurückgibt,wird ein XSLT-Verarbeitungsfehler zurückgegeben. So würde z.B im oben erwähnten Person-Beispiel ein XPath-Ausdruck wie z.B. `../Person/LastName` bei einem Kontext-Node `Person` einen Fehler zurückgeben, da dieser Ausdruck alle im Parent von `Person` enthaltenen `LastName` Elemente zurückgeben würde (vorausgesetzt, es gibt mehr als ein `Person` Element). Der korrekte XPath-Ausdruck mit `Person` als Kontextnode wäre: `LastName` (da es nur einen `LastName` Node für jedes `Person` Element gibt).

Gemäß der **XSLT 1.0** Spezifikation muss bei Rückgabe eines Nodeset nach Sortierschlüsselreihenfolge der Textwert des ersten Node verwendet werden. StyleVision gibt daher keinen Fehler zurück, wenn der XPath-Ausdruck mehrere Nodes für den Sortierschlüssel auswählt; es wird der Text des ersten Node verwendet und die anderen Nodes werden ignoriert. Jedoch ist nicht immer der erste ausgewählte Node der gewünschte Sortierschlüssel. So würde z.B. der XPath-Ausdruck `../Person/LastName` des oben beschriebenen Beispiels keinen Fehler zurückgeben. Doch würde auch keine Sortierung durchgeführt, da es sich um denselben Wert für jedes Element in der Sortierschleife handelt (den Textwert des ersten `LastName` Node). Bei einem Ausdruck der Art: `location/@*` würde jedoch unter Verwendung des ersten Attributs des `location` Child-Elements als Sortierschlüssel eine Sortierung durchgeführt. Diese Art von Ausdruck sollte jedoch vermieden werden. Es wird empfohlen eine genauere Auswahl des Sortierschlüssels durchzuführen (und einen einzelnen Node auszuwählen).

7.4.2 Beispiel: Sortieren nach mehreren Sortierschlüsseln

Schritt-für-Schritt-Beispiel

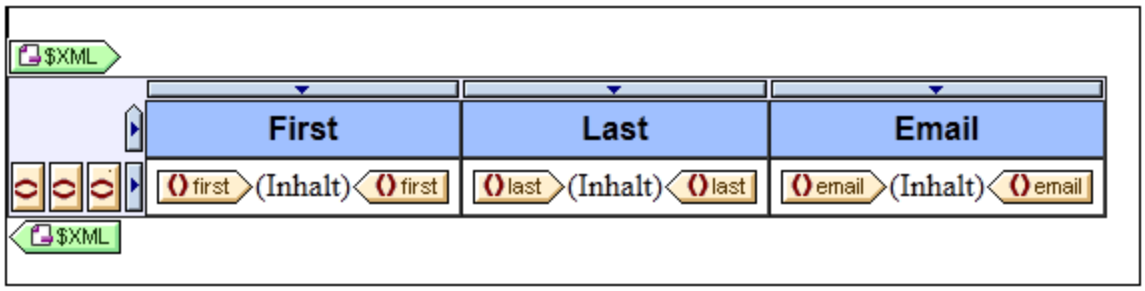
In dem einfachen Beispiel unten (zu finden im Ordner (Eigene) Dateien[25], `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2025\StyleVisionExamples\Tutorials\Sorting\SortingOnTwoTextKeys.sps`) werden die Mitglieder eines Teams in einer Tabelle aufgelistet. Jedes Mitglied wird mit dem Vornamen, Nachnamen und der E-Mail-Adresse in einer Tabellenzeile aufgelistet. Angenommen, wir wollen die Liste der Mitglieder in alphabetischer Reihenfolge sortieren, zuerst nach Nachnamen und anschließend nach Vornamen. Dies geschieht folgendermaßen:

Wenn die Liste nicht sortiert ist, entspricht die Reihenfolge der Ausgabe der Reihenfolge, in der die `member` Elemente im XML-Dokument aufgelistet sind (*Abbildung unten, welche die HTML-Ausgabe ist*).

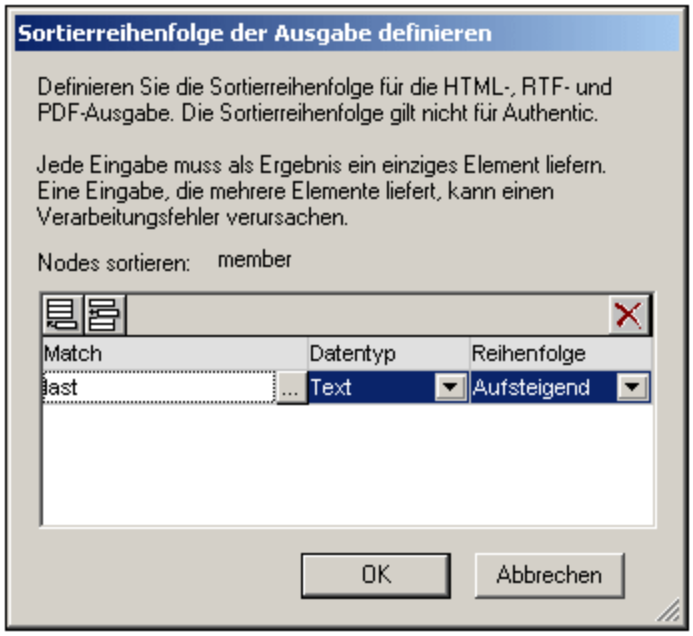
First	Last	Email
Andrew	Bentinck	a.bentinck@nanonull.com
Nadia	Edwards	n.edwards@nanonull.com
John	Edwards	j.edwards@nanonull.com
Janet	Ashe	j.ashe@nanonull.com

© 2019-2025 Altova GmbH

Klicken Sie in der Design-Ansicht mit der rechten Maustaste auf das `member` Element (*erscheint in der Abbildung unten blau markiert*) und wählen Sie im Kontextmenü den Befehl **Ausgabe sortieren.**.



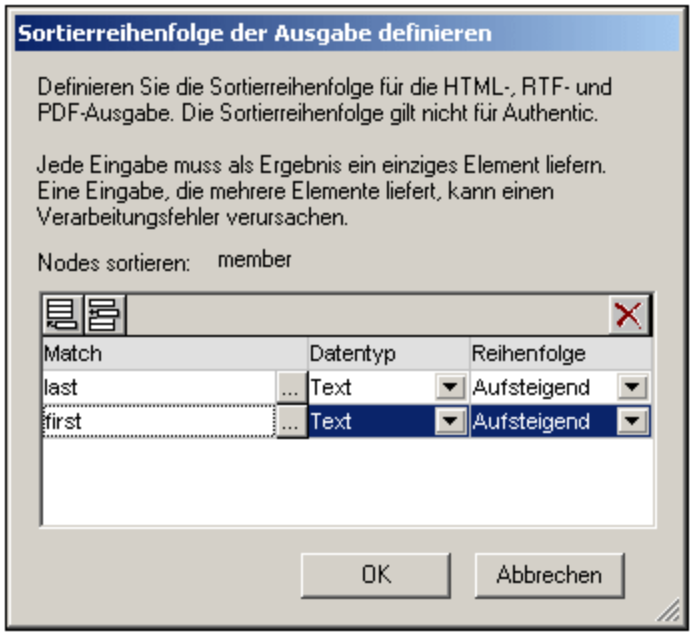
Daraufhin erscheint das Dialogfeld "Sortierreihenfolge der Ausgabe definieren" (*Abbildung unten*). Beachten Sie, dass das für die Sortierung ausgewählte Element `member` unter dem Eintrag "Nodes sortieren" angegeben ist. Dieser Node ist auch der Kontext-Node für XPath-Ausdrücke zur Auswahl des Sortierschlüssels. Klicken Sie auf die Schaltfläche "Zeile hinzufügen" (links in der Symbolleiste des Bereichs), um die erste Sortieranweisung hinzuzufügen. Geben Sie in der Zeile, die hinzugefügt wird, einen XPath-Ausdruck in die Spalte "Match" ein, um den Node `last` auszuwählen. Alternativ dazu können Sie auf die Erzeugen-Schaltfläche ... klicken, um den XPath-Ausdruck zu erzeugen. In der Spalte "Datentyp" können Sie auswählen, wie der Sortierschlüsselinhalt ausgewertet werden soll: als Text oder als Zahl. In der Spalte "Reihenfolge" wird die Sortierreihenfolge aufgelistet: Aufsteigend oder absteigend. Wählen Sie `Text` und `Aufsteigend`. Klicken Sie zum Fertigstelle auf **OK**.



In der Design-Ansicht enthält der `member` Tag ein Symbol, das anzeigt, dass er einen Sortierfilter enthält member . In der Abbildung unten sehen Sie die HTML-Ausgabe der Liste der Teammitglieder, sortiert nach Nachnamen. Beachten Sie, dass die beiden Edwards nicht alphabetisch sortiert sind.

First	Last	Email
Janet	Ashe	j.ashe@nanonull.com
Andrew	Bentinck	a.bentinck@nanonull.com
Nadia	Edwards	n.edwards@nanonull.com
John	Edwards	j.edwards@nanonull.com

Rechtsklicken Sie in der Design-Ansicht auf den `member` Tag und wählen Sie im Kontextmenü den Befehl **Ausgabe sortieren**. Daraufhin wird das Dialogfeld "Sortierreihenfolge der Ausgabe definieren" angezeigt, wobei die `last` Sortieranweisung aufgelistet ist. Um eine weitere Sortieranweisung hinzuzufügen, hängen Sie eine neue Zeile an und geben Sie das `first` Element als Sortierschlüssel ein (*Abbildung unten*). Klicken Sie zum Fertigstellen auf **OK**.



In der HTML-Ausgabe wird die Liste nun nach Nachnamen und anschließend Vornamen in alphabetischer Reihenfolge sortiert.

© 2019-2025 Altova GmbH

First	Last	Email
Janet	Ashe	j.ashe@nanonull.com
Andrew	Bentinck	a.bentinck@nanonull.com
John	Edwards	j.edwards@nanonull.com
Nadia	Edwards	n.edwards@nanonull.com

7.5 Parameter und Variablen

Parameter und Variablen können im SPS deklariert und referenziert werden. Der Unterschied zwischen den beiden Arten ist folgender: Während der Wert einer Variablen bei ihrer Deklaration definiert wird, kann der Wert bei einem Parameter (während der Laufzeit über die Befehlszeile) an den Parameter übergeben werden. Dieser Wert setzt den optionalen Standardwert, der dem Parameter bei seiner Deklaration zugewiesen wurde, außer Kraft.

In diesem Abschnitt werden die Funktionalitäten beschrieben, die für Parameter und Variablen zur Verfügung stehen:

- Vom Benutzer deklarierte Parameter 282: Hier wird die Verwendungsweise von benutzerdefinierten Parametern in einem SPS erklärt.
- Unter Parameter für Design-Fragmente 284 wird beschrieben, wie Parameter mit Design-Fragmenten verwendet werden können.
- SPS Parameter für Quelldokumente 286: Hierbei handelt es sich um eine spezielle Art von Parametern, die automatisch von StyleVision für Schemaquellen (insbesondere die XML-Arbeitsdateien von Schemas) definiert werden. Da dem Benutzer der Name und Wert eines solchen Parameters bekannt sind, kann der Parameter im SPS referenziert werden und ein Wert kann während der Laufzeit von der Befehlszeile an diesen Parameter übergeben werden.
- Variablen 287: Mit Hilfe von Variablen können Sie: (i) eine Variable mit einem bestimmten Geltungsbereich deklarieren und ihren Wert definieren. und (ii) den Wert von deklarierten Variablen referenzieren und in einem oder mehreren von der Variablen ausgewählten Nodes eine Vorlagen erstellen.

7.5.1 Vom Benutzer deklarierte Parameter

In einem SPS werden vom Benutzer deklarierte Parameter global mit einem Namen und einem Standardstringwert deklariert. Anschließend können sie an jeder Stelle im SPS in XPath-Ausdrücken verwendet werden.

Verwendung von Parametern

Vom Benutzer deklarierte Parameter können in den folgenden Situationen verwendet werden:

- Wenn Sie einen Wert an mehreren Stellen oder als Eingabewert für mehrere Berechnungen verwenden möchten. In diesem Fall können Sie den Wert als Parameterwert speichern und den Parameter an den gewünschten Stellen und in den gewünschten Berechnungen verwenden.
- Wenn Sie einen Wert während der Verarbeitung an das Stylesheet übergeben möchten. Im SPS (und Stylesheet) verwenden Sie einen Parameter mit einem Standardwert. Während der Verarbeitung wird der gewünschte Wert über StyleVision Server an den Parameter übergeben.

Funktionsweise

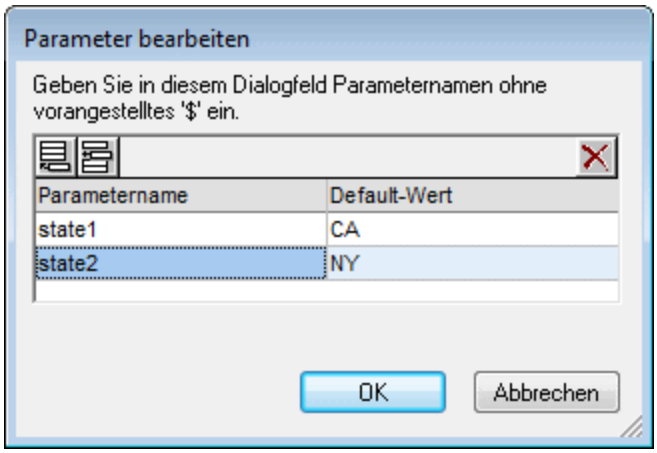
Die Verwendung von benutzerdeklarierten Parametern im SPS erfolgt in zwei Schritten:

1. Deklarieren des gewünschten Parameters 283
2. Referenzieren des deklarierten Parameters 283

© 2019-2025 Altova GmbH

Deklarieren von Parametern

Alle benutzerdefinierten Parameter werden im Dialogfeld "Parameter bearbeiten" (*Abbildung unten*) deklariert und bearbeitet. Das Dialogfeld "Parameter bearbeiten" wird folgendermaßen aufgerufen: über den Befehl **Bearbeiten | Stylesheet Parameter**[498],



Bei der Deklaration eines Parameters erhält der Parameter einen Namen und einen Stringwert - seinen Standardwert. Wenn kein Wert angegeben wird, ist der Standardwert ein leerer String..

Um einen Parameter zu deklarieren, gehen Sie folgendermaßen vor:

1. Fügen Sie im Dialogfeld "Parameter bearbeiten" einen neuen Parameter durch Klicken auf die Schaltfläche "Anhängen" oder "Einfügen" hinzu. Daraufhin wird eine neue Zeile hinzugefügt.
2. Geben Sie den Namen des Parameters ein. Parameternamen müssen mit einem Buchstaben beginnen und können die Zeichen `A` bis `Z`, `a` bis `z`, `0` bis `9` und den Unterstrich enthalten.
3. Geben Sie einen Standardwert für diesen Parameter ein. Der Wert wird als Textstring gespeichert.

Sie können beliebig viele Parameter einfügen und bestehende jederzeit ändern, während Sie das SPS bearbeiten.

Anmerkung:

- Das Dialogfeld "Parameter bearbeiten" enthält alle benutzerdefinierten Parameter in einem SPS.
- Parameter können auch in der Seitenleiste Design-Übersicht[36] deklariert werden.

Referenzieren von deklarierten Parametern

Parameter können in XPath-Ausdrücken durch Voranstellen eines `$` Zeichens vor den Parameternamen referenziert werden. So könnten Sie z.B. einen Parameter im XPath-Ausdruck einer automatischen Berechnung referenzieren (z.B. `concat('www.', $company, '.com')`).

Anmerkung: Die Referenzierung eines undeklarierten Parameters ergibt einen Fehler. Wird jedoch ein Parameter deklariert und nicht referenziert, ist dies kein Fehler.

7.5.2 Parameter für Design-Fragmente

Mit Hilfe von Parametern für Design-Fragmente können Sie in einem von Ihnen erstellten Design-Fragment einen Parameter definieren und dem Parameter einen Standardwert zuweisen. An jeder Stelle, an der das Design-Fragment im Design verwendet wird, können Sie einen anderen Parameterwert eingeben und dadurch die Ausgabe einzelner Design-Fragmente variieren.

So könnten Sie z.B. ein Design-Fragment mit dem Namen `EMailAddresses` und einem Parameter namens `Domain` erstellen und dem Parameter den Standardwert `altova.com` zuweisen. Angenommen, dieser Parameter wird nun in einer automatischen Berechnung im Design-Fragment verwendet, um die E-Mail-Adresse der Angestellten der Firma zu generieren. Für die EU-Adressen könnten wir das Design-Fragment `EmailAddresses` verwenden und den Wert des Parameters `Domain` auf `altova.eu` setzen. Auf dieselbe Art könnten wir den Wert des Parameters `Domain` in der Vorlage für die Angestellten in Japan auf den Wert `altova.jp` setzen. Bei den US-Angestellten der Firma könnten wir den Parameterwert von `Domain` unverändert lassen und so den Standardwert `altova.com`. generieren.

Die Verwendung von Parameters für Design-Fragmente erfolgt in zwei Schritten:

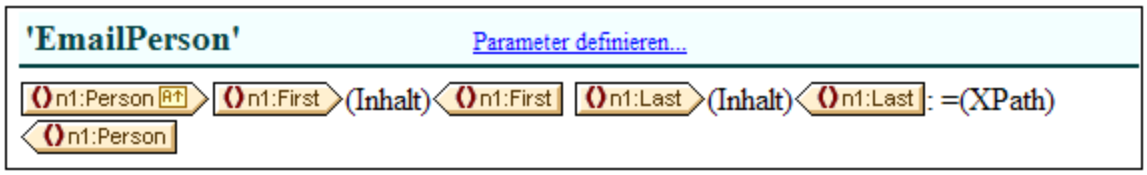
1. Definition des Parameters 284 mit einem Standardwert für das Design-Fragment, für das er erstellt wird.
2. Bearbeitung des Parameterwerts 285 an den Stellen, an denen das Design-Fragment verwendet wird.

Im Folgenden werden diese Schritte näher beschrieben.

Anmerkung: Parameter für Design-Fragmente werden in der Authentic-Ansicht nur in der Enterprise Edition von Altova-Produkten unterstützt.

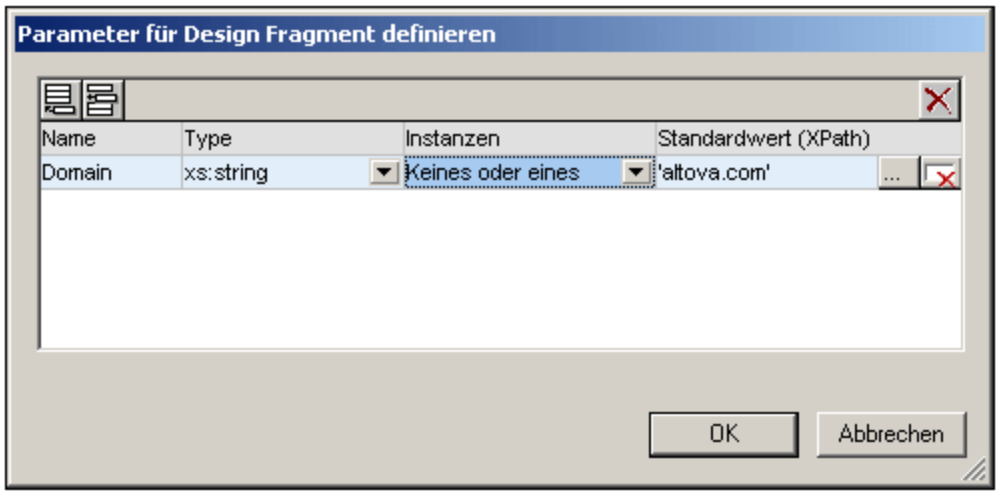
Definition des Parameters

Jedem Design-Fragment kann eine ganze Reihe von Parametern zugewiesen werden. Kicken Sie dazu in der Titelleiste des Design-Fragments (*siehe Abbildung unten*) auf den Link "Parameter definieren".



Daraufhin wird das Dialogfeld "Parameter für Design Fragment definieren" (*Abbildung unten*) aufgerufen. Klicken Sie auf das **Anhängen** oder das **Einfügen** Symbol links oben, um eine Parametereingabezeile hinzuzufügen. Geben Sie den Namen, den Datentyp, die Anzahl der Instanzen und den Standardwert für den Parameter ein, bzw. wählen Sie diese aus. Mit dem Parameterattribut *Instanzen* definieren Sie, wieviele Elemente bei Auswertung des als Standardwert des Parameters definierten XPath-Ausdrucks zurückgegeben werden sollen. Das Attribut *Instanzen* ist optional und standardmäßig Keines oder eines. Sie können beliebig viele Parameter hinzufügen.

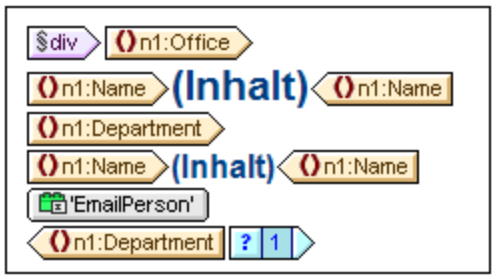
© 2019-2025 Altova GmbH



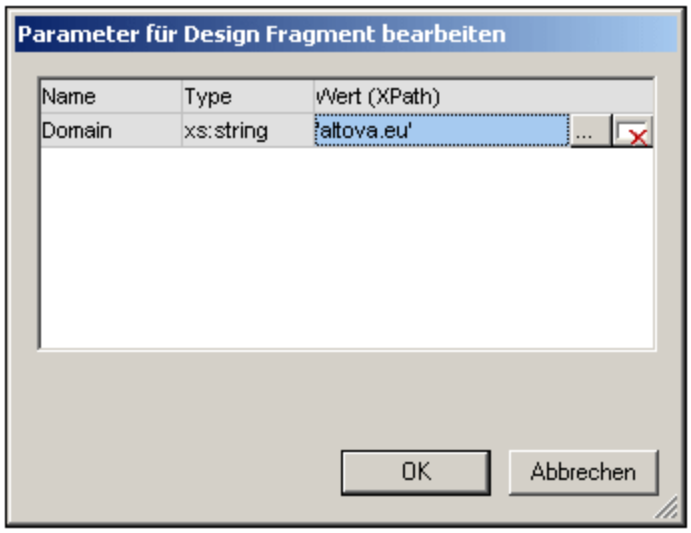
Es gibt zwei Arten von **Löschsymbolen.** Mit dem Löschsymbol rechts von den Parametereinträgen löschen Sie den Standardwert dieses Parameters. Mit dem **Löschsymbol** im rechten oberen Eck des Fensters löschen Sie den gerade markierten Parameter.

Verwendung des Parameters

Nachdem Sie ein Design-Fragment erstellt haben, können Sie es (durch Ziehen aus der Design- oder der Schema-Struktur) an mehreren Stellen im Design einfügen. In der Abbildung unten sehen Sie das Design-Fragment `EmailPerson`, das nach dem Element `n1:Name` eingefügt wurde.



Wenn für dieses Design-Fragment ein Parameter definiert wurde, kann sein Wert für diese bestimmte Verwendungsinstanz des Design-Fragments bearbeitetet werden. Klicken Sie dazu mit der rechten Maustaste auf das Design-Fragment und wählen Sie den Befehl **Parameter bearbeiten**. Daraufhin wird das Dialogfeld "Parameter für Design-Fragment bearbeiten" aufgerufen (*Abbildung unten).*



In diesem Dialogfeld können Sie den Wert des Parameters bearbeiten. Klicken Sie zum Fertigstellen auf **OK**. In dieser Verwendungsinstanz des Design-Fragments wird der neue Parameterwert verwendet. Wenn der Parameterwert nicht bearbeitet wird, wird der ursprüngliche (Standard)-Wert des Parameters verwendet.

Anmerkung: Wenn XSLT 1.0 verwendet wird, muss der XPath-Ausdruck ein Node-Set zurückgeben, da sonst ein Fehler gemeldet wird.

7.5.3 SPS Parameter für Quelldokumente

Ein SPS kann mehrere Schemaquellen haben, wobei es sich dabei um eine DTD oder ein XML-Schema handeln kann, auf der/dem ein XML-Dokument basiert, oder um ein XML-Schema, das anhand einer Datenbank generiert wurde und auf dem die Datenbank basiert.

In jedem SPS gibt es ein Hauptschema und optional dazu ein oder mehrere zusätzliche Schemas. Wenn Sie eine neue Schemaquelle hinzufügen, deklariert StyleVision automatisch einen Parameter für dieses Schema und weist dem Parameter einen Wert zu, der die URI für die XML-Arbeitsdatei darstellt, die Sie diesem Schema zuweisen. Im Fall von Datenbanken generiert StyleVision eine temporäre XML-Datei anhand einer Datenbank und definiert die Parameter als Zielparameter für den Dokument-Node dieser temporären XML-Datei.

Referenzieren von Parametern für Quellen

Jeder SPS Parameter für eine Schemaquelle verweist auf den Dokument-Node einer XML-Datei, die diesem Schema entspricht. In StyleVision handelt es sich bei der XML-Datei für die einzelnen Schemas um die XML-Arbeitsdatei oder die anhand einer Datenbank generierte XML-Datei. SPS Parameter für Quellen können daher auf zwei Arten verwendet werden:

1. In XPath-Ausdrücken im SPS, um Nodes in verschiedenen Dokumenten zu identifizieren. Der Parameter dient zum Identifizieren des Dokuments und der gewünschte Node im Dokument wird durch einige auf den Parameter folgende Pfadangaben im XPath-Ausdruck identifiziert. Der Ausdruck: `count($XML2//Department/Employee)` gibt z.B. die Anzahl der `Employee` Elemente in allen `Department` Elementen in dem XML-Dokument zurück, das für die als `$XML2` designierte Schemaquelle als XML-Arbeitsdatei dient.
2. In der Befehlszeile kann die URI einer anderen XML-Datei als Wert eines SPS Parameters für Quellen übergeben werden. Natürlich müsste die neue XML-Datei auf dem Schema basieren, für das dieser

 © 2019-2025 Altova GmbH

Parameter steht. Wenn z.B. `FileA.xml` und `FileB.xml` nach demselben Schema gültig sind, und `FileA.xml` die XML-Arbeitsdatei ist, die einem Schema `$XML3` zugewiesen ist, das in einem SPS verwendet wird, so kann bei Aufruf einer XSLT-Transformation für dieses SPS über die Befehlszeile die Datei `FileA.xml` mit Hilfe des Parameters `$XML3="FileB.xml"` durch die Datei `FileB.xml` ersetzt werden. Beachten Sie außerdem, dass in der Befehlszeile für alle SPS Parameter für Quellen Werte eingegeben werden sollten mit Ausnahme des Parameters für das Hauptschema. Die XML-Datei für das Hauptschema dient als Ausgangspunkt für das XSLT Stylesheet und ist daher die XML-Datei, an der die Transformation ausgeführt wird.

7.5.4 Variablen

Die Verwendung von Variablen erfolgt in zwei Schritten: (d) Deklarieren einer Variablen 287 und (ii) Verwenden der Variablen 289.

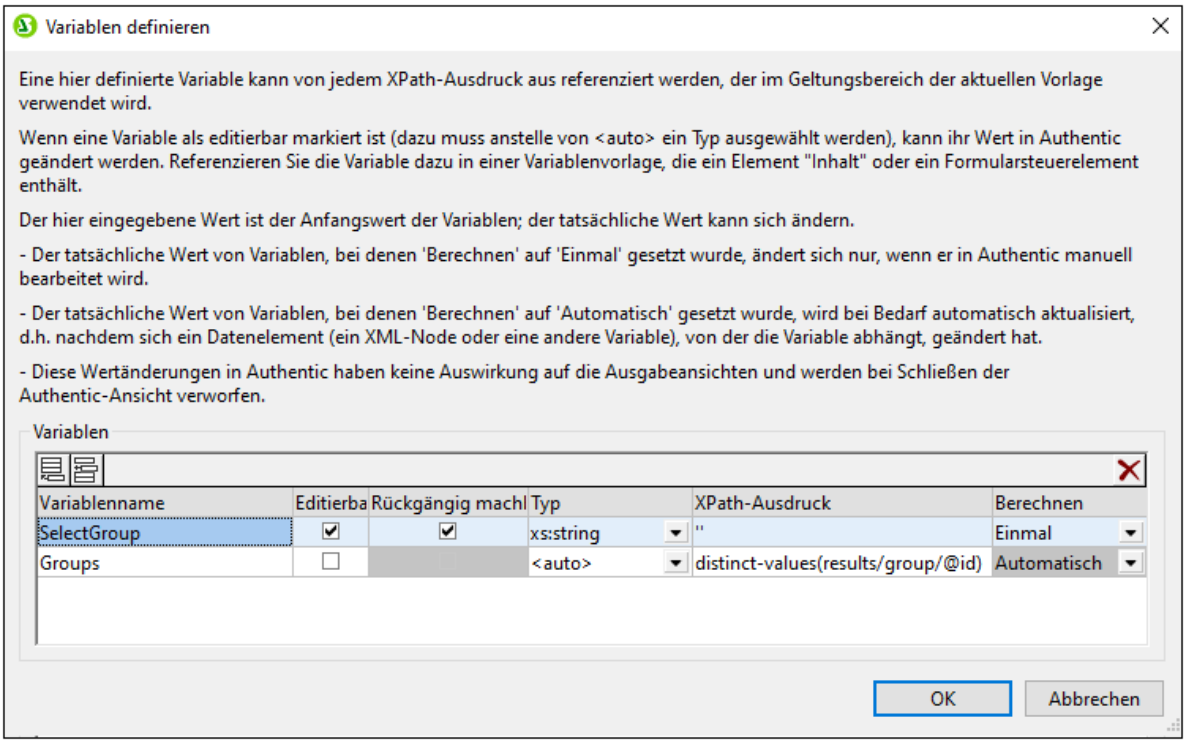
Anmerkung: Variablen werden in der Authentic-Ansicht nur in der Enterprise Edition von Altova-Produkten unterstützt.

Deklarieren einer Variablen

Eine Variable kann mit einem Namen in jeder in das Design inkludierten Vorlagen deklariert werden. Die Variable erhält einen Namen, eine Datentyp und einen Wert zugewiesen. Zusätzlich dazu können Sie festlegen, ob sie in der Enterprise Edition in der Authentic-Ansicht auch bearbeitet werden kann. Die Variable ist dann in dieser Vorlage im Geltungsbereich und kann dann innerhalb dieses Geltungsbereichs verwendet werden. Um eine Variable so zu deklarieren, dass sie sich für das gesamte Dokument im Geltungsbereich befindet, deklarieren Sie die Variable für die Root-Vorlage. Ein großer Vorteil der Deklaration einer Variablen nur für die Vorlage, in der sie benötigt wird, ist, ist dass XPath-Ausdrücke zur Auswahl eines untergeordneten Node einfacher sind.

Um eine Variable zu deklarieren, gehen Sie folgendermaßen vor:

1. Klicken Sie mit der rechten Maustaste auf die Node-Vorlage, in der die Variable erstellt werden soll und wählen Sie den Befehl **Variablen definieren** aus.
2. Klicken Sie im Dialogfeld "Variablen definieren", das nun angezeigt wird (*Abbildung unten*) in der linken oberen Ecke des Fensters "Variablen" auf die Schaltfläche **Variable anhängen** und geben Sie anschließend den Variablennamen an. Der Wert der Variablen wird durch einen XPath-Ausdruck angegeben. Wenn Sie einen String als Wert der Variablen angeben möchten (wie in der ersten Variablen in der Abbildung unten), so setzen Sie den String in Anführungszeichen. In der Abbildung unten ist der Wert der Variablen `SelectGroup` der leere String. Andernfalls wird der Text als Node-Name oder Funktionsaufruf gelesen.



3. Wenn Sie eine Variable editierbar machen (indem Sie das Kontrollkästchen *Editierbar* aktivieren), kann die Variable in der Authentic-Ansicht (die nur in der Enterprise und Professional Edition von StyleVision als Vorschau zur Verfügung steht) bearbeitet werden. In diesem Fall müssen Sie auch den Wert des Datentyps auf den richtigen Typ setzen, z.B. `xs:string`. Wenn eine Variable editierbar ist, kann der ursprünglich vom SPS-Designer definierte Wert bearbeitet werden, wenn der Benutzer in der Authentic-Ansicht Änderungen am Dokument in der Authentic-Ansicht vornimmt. Bei diesen Änderungen kann es sich um eine explizite Bearbeitung der Variablen handeln (z.B. wenn der Variablenwert als editierbarer `(Inhalt)` erstellt wurde oder ein editierbares Textfeld erstellt wurde und dieses vom Benutzer der Authentic-Ansicht bearbeitet wird), oder wenn ein im XPath-Ausdruck der Variablen verwendeter Node oder Wert durch den Benutzer der Authentic-Ansicht geändert wird.
4. Wenn die Variable auf "Editierbar" gesetzt wurde, so werden zwei weitere für die Authentic-Ansicht relevante Optionen aktiv: *Rückgängig machbar* und *Berechnen*. Bei Aktivierung der Option *Rückgängig machbar* wird für jede Änderung, die an der Variablen vorgenommen wird, ein Rückgängig-Schritt generiert. Dadurch kann der Benutzer der Authentic-Ansicht die einzelnen Schritte rückgängig machen, um einen früheren Wert der Variablen abzurufen. Der *Berechnen*-Wert kann entweder `Einmal` oder `Automatisch` sein. Wenn diese Option auf `Einmal` gesetzt ist, wird der Variablenwert einmal bei der Auswertung der Vorlage, die die Variable enthält, berechnet. Der Wert kann nur geändert werden, wenn der Benutzer die Variable explizit bearbeitet (z.B. wenn die Variable als editierbar `(Inhalt)` oder als editierbares Textfeld erstellt wurde). Wenn diese Option dagegen auf `Automatisch` gesetzt wurde, wird die Variable jedes Mal, wenn ein im XPath-Ausdruck der Variablen verwendeter Node oder Wert geändert wurde, neu berechnet.
5. Sie können so viele Variablen, wie Sie möchten, hinzufügen, doch darf der Name der Variablen nicht mit dem Namen einer bereits deklarierten Variablen, die sich im Geltungsbereich befindet, identisch sein. Um eine Variable zu löschen, klicken Sie in der rechten oberen Ecke des Fenster auf die Schaltfläche **Löschen.**
6. Klicken Sie anschließend auf **OK**. Der Vorlagen-Tag weist nun ein $-Symbol auf, um anzuzeigen, dass eine oder mehrere Variablen darin deklariert wurden.

© 2019-2025 Altova GmbH

Auf diese Art kann für jede im Design vorhandene Node-Vorlage eine Variable erstellt werden. Jede dieser Variablen erhält einen Namen und einen Wert und ist innerhalb der Vorlage, in der sie erstellt wurde, im Geltungsbereich. Um eine Variable später zu bearbeiten, klicken Sie mit der rechten Maustaste auf die Node-Vorlage, in der die Variable erstellt wurde, und wählen Sie den Befehl **Variablen definieren** um das gleichnamige Dialogfeld aufzurufen.

Verwenden einer Variablen

Damit eine Variable an jeder beliebigen Stelle verwendet werden kann, muss Sie sich an dieser Stelle innerhalb des Geltungsbereichs befinden, d.h. eine Variable kann nur innerhalb der Vorlage verwendet werden, in der sie definiert wurde. Variablen können auch in Authentic bearbeitet werden, sodass der Benutzer festlegen kann, was angezeigt wird. Die bearbeitete Variable wird verworfen, wenn das SPS geschlossen wird.

Eine Variable kann innerhalb eines jeden XPath-Ausdrucks verwendet werden. Die Variable wird referenziert, indem man ihrem Namen ein `$` Symbol voranstellt. So wählt z.B. der XPath-Ausdruck `$VarName/Name` das `Name` Child-Element des Node aus, der durch die Variable mit dem Namen `VarName` ausgewählt wird.

Wenn Sie im Dialogfeld "XPath-Ausdruck bearbeiten[440]" einen XPath-Ausdruck eingeben, werden die Variablen im Geltungsbereich in einem Popup-Fenster angezeigt (*siehe Abbildung oben*). Wenn Sie eine Variable im Popup-Fenster auswählen und die Eingabetaste drücken, wird die Variablenreferenz in den Ausdruck eingefügt.

7.6 Inhaltsverzeichnis, Referenzierung, Lesezeichen

Mit den Inhaltsverzeichnis- und anderen Referenzierungsfunktionen werden Markierungen an den gewünschten Punkten im Design gesetzt und diese werden vom Inhaltsverzeichnis, von Textreferenzen, automatischen Nummerierungssequenzen und Hyperlinks aus referenziert.

Zum besseren Verständnis erläutern wir hier zuerst kurz das Prinzip der Inhaltsverzeichnis-Markierung und befassen uns anschließend mit der allgemeinen Funktionsweise von Inhaltsverzeichnissen.

Das Prinzip der Inhaltsverzeichnis-Markierung

Es kommen zwei Arten von Inhaltsverzeichnis-Markierungen zum Einsatz: einfache und komplexe. Zur Erstellung von Inhaltsverzeichnissen werden komplexe Inhaltsverzeichnis-Markierungen verwendet.

- An einer bestimmten Stelle im Design-Dokument wird eine einfache Markierung erstellt. Die Markierung erhält einen eindeutigen Namen. Dieser Name wird als Ziel von Links verwendet, die darauf verweisen. Dieser einfache Mechanismus wird für die Lesezeichen- und Hyperlinkfunktion 320 verwendet. (Beachten Sie, dass Hyperlinks zusätzlich auf URLs außerhalb des Dokuments verweisen können).
- Bei komplexeren Referenzierungen wie z.B. Inhaltsverzeichnissen und der automatischen Nummerierung von Dokumentabschnitten erfolgt die Erstellung der Markierung in zwei Schritten.

  1. Zuerst wird für das Design-Dokument die für das Inhaltsverzeichnis benötigte Hierarchie angelegt. Diese Ebenen werden als Inhaltsverzeichnis-Ebenen bezeichnet. Dazu werden verschiedenen Stellen in der Dokumentstruktur Ebenen zugewiesen. Um dem Dokument eine hierarchische Struktur zu geben, können Inhaltsverzeichnis-Ebenen innerhalb von anderen Inhaltsverzeichnis-Ebenen verschachtelt werden. (So kann einem Buchkapitel z.B. eine Inhaltsverzeichnis-Ebene zugewiesen werden und eine weitere Inhaltsverzeichnis-Ebene kann den einzelnen Kapitelabschnitten innerhalb dieser Ebene zugewiesen werden.)
  2. Innerhalb der verschiedenen Inhaltsverzeichnis-Ebenen werden Inhaltsverzeichnis-Markierungen erstellt. Mit diesen Inhaltsverzeichnis-Markierungen werden die Dokumentabschnitte auf den verschiedenen Ebenen gekennzeichnet, die in das Inhaltsverzeichnis aufgenommen werden sollen. Zusätzlich dazu müssen die einzelnen Inhaltsverzeichnis-Markierungen definiert werden, die den Text enthalten sollen, der in der referenzierenden Komponente angezeigt werden soll.

Nach Definition der Inhaltsverzeichnis-Ebenen und des Referenztextes für die Inhaltsverzeichnis-Markierungen, kann die Inhaltsverzeichnis-Vorlage, die die referenzierenden Komponenten enthält, erstellt werden.

Weiter unten finden Sie unter Die Funktionsweise beim Erstellen von Inhaltsverzeichnissen 290 einen Überblick über die Funktionsweise von Inhaltsverzeichnissen. Die verschiedenen Referenzierungsfunktionen werden im restlichen Teil dieses Abschnitts näher beschrieben.

Die Funktionsweise beim Erstellen von Inhaltsverzeichnissen

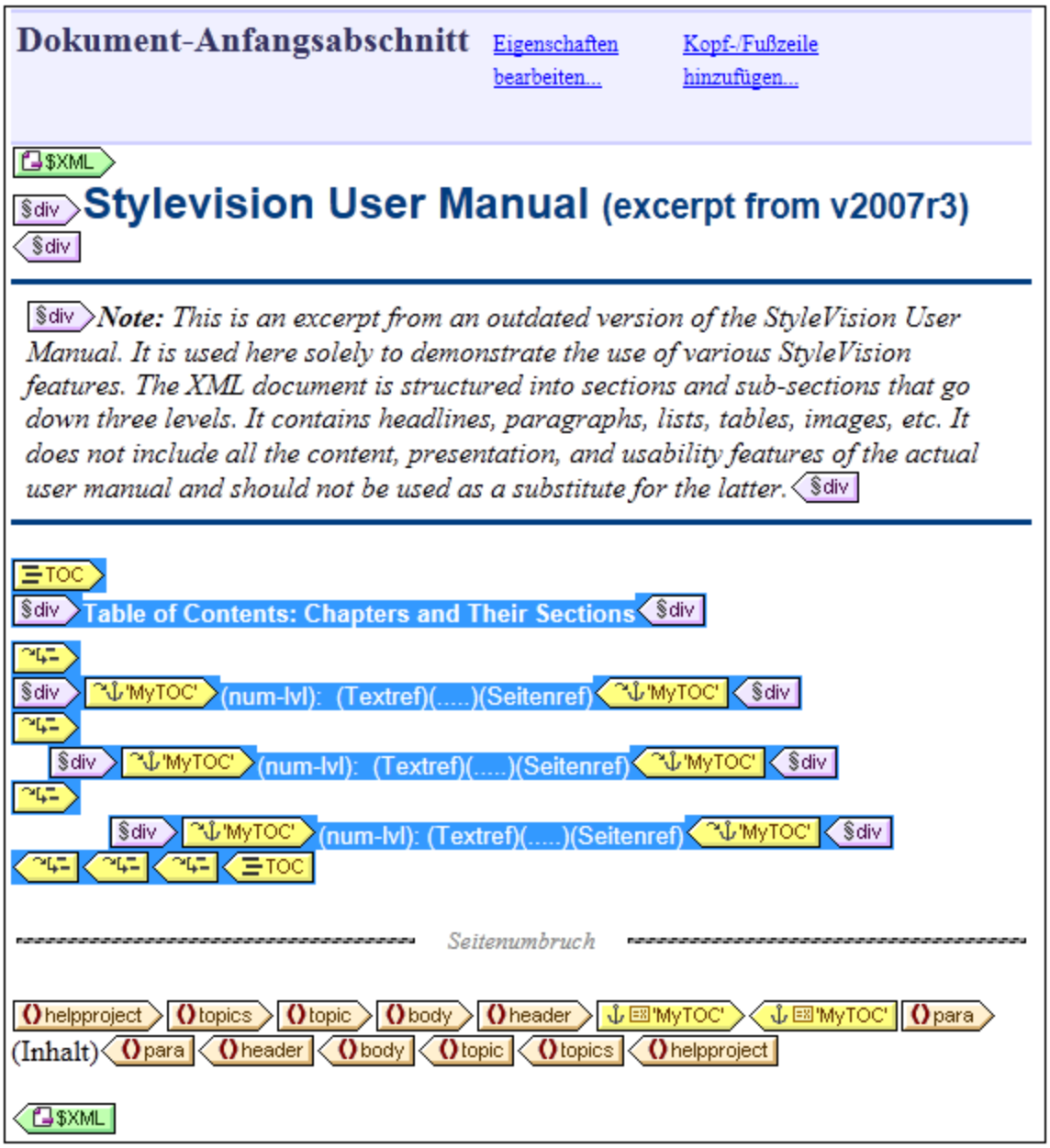
Wenn Sie die Option **XSLT 2.0 oder XSLT 3.0** 100 (nicht XSLT 1.0) als XSLT-Version für Ihr SPS ausgewählt haben, können Sie an jeder beliebigen Stelle in Ihrem Design ein Inhaltsverzeichnis - im Grunde eine Vorlage für das Inhaltsverzeichnis - erstellen.

- Die Elemente im Design, die im Inhaltsverzeichnis enthalten sein sollen bzw. auf die im Inhaltsverzeichnis verwiesen werden soll, zuerst im Design markiert 293 werden. Dabei kann es sich um

© 2019-2025 Altova GmbH

statischen oder dynamischen Inhalt handeln. In der unteren Hälfte der Abbildung unten, wird das Element `header` durch gelbe Inhaltsverzeichnis-Tags 'MyTOC' 'MyTOC' für das Einschließen in die Inhaltsverzeichnis-Vorlage markiert.

- Für das Inhaltsverzeichnis wird eine Vorlage erstellt 301 (*in der Abbildung unten markiert*). Die Inhaltsverzeichnis-Vorlage enthält das Design des Inhaltsverzeichnisses; sie kann sich an jeder beliebigen Stelle des Dokuments befinden. Im Beispiel in der Abbildung unten befindet sich die Inhaltsverzeichnis-Vorlage im oberen Bereich des Dokuments.



Anmerkung: Es spielt keine Rolle, welcher der beiden Teile zuerst erstellt wird, es können auch beide Teile gleichzeitig erstellt werden. Es wird jedoch empfohlen, die Inhaltsverzeichnis-Markierungen noch vor der Inhaltsverzeichnis-Vorlage zu erstellen.

Das Inhaltsverzeichnis wird in der HTML-Ausgabe angezeigt. Beachten Sie außerdem die folgenden Features: (i) Inhaltsverzeichnisse können mit flacher oder hierarchischer Struktur (mit der entsprechenden Nummerierung) erstellt werden und (ii) es können innerhalb eines Designs mehrere Inhaltsverzeichnisse angelegt werden. Dadurch kann man in einem Stylesheet ein Dokument z.B. mit einem (hierarchischen) Inhaltsverzeichnis auf Buchebene und anderen (ebenfalls hierarchischen) auf Kapitelebene sowie (flache) Listen mit Abbildungen und Tabellen erstellen.

Verfahren zum Erstellen von Inhaltsverzeichnissen

Im Folgenden finden Sie eine schrittweise Anleitung zum Erstellen eines Inhaltsverzeichnisses. Dabei werden zuerst Elemente markiert, die in das Inhaltsverzeichnis aufgenommen werden sollen, und anschließend wird die Inhaltsverzeichnis-Vorlage erstellt. (Alternativ dazu können Sie auch zuerst die Inhaltsverzeichnis-Vorlage erstellen und die zu inkludierenden Elemente erst dann markieren; oder Sie erstellen die Inhaltsverzeichnis-Vorlage und wählen parallel dazu die gewünschten Elemente aus.)

1. Stellen Sie sicher, dass als XSLT-Version XSLT 2.0 [100] ausgewählt ist.
2. *Strukturieren Sie das Dokument in Inhaltsverzeichnis-Ebenen* [295]. Wenn das Inhaltsverzeichnis mehrere Ebenen haben soll, strukturieren Sie das Dokument-Design in einer Hierarchie verschachtelter Inhaltsverzeichnis-Ebenen. Wenn das Inhaltsverzeichnis eine flache Struktur haben soll (also nur eine Ebene), erstellen Sie (im Dokument-Design) zumindest eine Inhaltsverzeichnis-Ebene, die die Inhaltsverzeichnismarkierungen enthalten soll.
3. *Erstellen Sie ein oder mehrere Inhaltsverzeichnismarkierungen* [298] *innerhalb der einzelnen Ebenen im Dokument-Design.* Mit den Inhaltsverzeichnismarkierungen werden die Komponenten innerhalb der einzelnen Ebenen gekennzeichnet, die im Inhaltsverzeichnis zu sehen sein sollen.
4. *Erstellen Sie eine Inhaltsverzeichnis-Vorlage, die die Inhaltsverzeichnis-Ebenenreferenzen (levelrefs) enthält* [301]. Die Inhaltsverzeichnis-Vorlage sollte die erforderliche Anzahl von Inhaltsverzeichnis-Ebenenreferenzen (levelrefs) haben. Im Fall eines Inhaltsverzeichnisses mit mehreren Ebenen sollten die Ebenenreferenzen verschachtelt sein (*siehe Abbildung oben).*
5. *Erstellen Sie in der Inhaltsverzeichnis-Vorlage Inhaltsverzeichnisreferenzen (TOCrefs)* [304]. Definieren Sie in der Inhaltsverzeichnis-Vorlage für jede Ebene eine Inhaltsverzeichnis-Ebenenreferenz. Jede Inhaltsverzeichnis-Ebenenreferenz referenziert die entsprechenden Inhaltsverzeichnismarkierungen innerhalb dieser Dokumentebene anhand ihres Namens. Alternativ dazu kann die Inhaltsverzeichnis-Ebenenreferenz auch Inhaltsverzeichnismarkierungen auf anderen Ebenen referenzieren.
6. *Formatieren Sie die Einträge des Inhaltsverzeichnisses* [305]. Jeder Texteintrag in der Inhaltsverzeichnisausgabe wird durch eine Inhaltsverzeichnis-Referenz in der Inhaltsverzeichnis-Vorlage generiert. Die Inhaltsverzeichnisreferenz-Definitionen können eine Nummerierung (auch eine hierarchische), den Text des Inhaltsverzeichniseintrags, eine punktierte Linie und bei paginierten Medien eine Seitennummer enthalten. Jede Inhaltsverzeichnisreferenz sowie ihre Bestandteile können je nach Wunsch separat formatiert werden. (Beachten Sie: Die automatische Nummerierung kann auch in einer Inhaltsverzeichnis-Markierung im Hauptteil des Dokuments definiert werden. Siehe dazu der Abschnitt Automatische Nummerierung [314].

Aktualisierung der Seitenreferenzen im Inhaltsverzeichnis in DOCX- und RTF-Dokumenten

Wenn ein Benutzer ein DOCX- oder RTF-Ausgabedokument in MS Word auf eine Weise bearbeitet, dass sich die Seitenzahl ändert, kann es vorkommen, dass das Inhaltsverzeichnis nicht mit den neuen Seitennummern aktualisiert wird. Dies ist ein MS Word-Problem. Um die Seitenreferenzen im Inhaltsverzeichnis zu aktualisieren, drücken Sie **Strg+A**, um alles auszuwählen und anschließend F9. Nähere Informationen dazu finden Sie hier.

© 2019-2025 Altova GmbH

Terminologie

In der Tabelle unten finden Sie die Namen der auf der Benutzeroberfläche verwendeten Hauptkomponenten für Inhaltsverzeichnisse. Die Komponenten sind je nachdem, wo sie verwendet werden, in zwei unterschiedlichen Spalten aufgelistet: Im **Dokumenthauptteil** oder in der **Inhaltsverzeichnisvorlage** (dies ist die Vorlage, in der das Design des eigentliche Inhaltsverzeichnisses definiert ist und das normalerweise am Beginn des Dokuments steht).

- Als ***Inhaltsverzeichnis-Komponenten im Dokumenthauptteil*** werden Elemente markiert, die in der Inhaltsverzeichnisvorlage verwendet werden sollen.
- Die ***Inhaltsverzeichnis-Komponenten in der Inhaltsverzeichnis-Vorlage*** referenzieren die markierten Elemente im Dokumenthauptteil. Komponenten in der Inhaltsverzeichnisvorlage haben das Word *'Referenz'* in ihrem Namen.

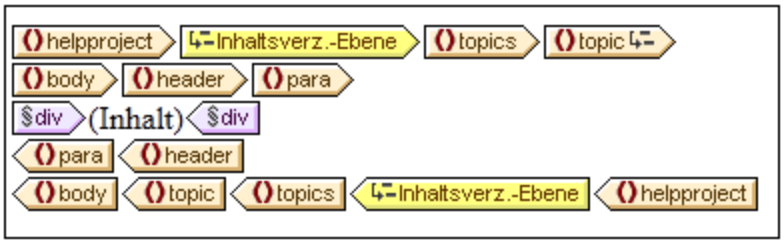
Dokumenthauptteil	Inhaltsverzeichnis-Vorlage
Inhaltsverz.-Ebene: Mit Hilfe der Inhaltsverzeichnis-Ebenen wird das Dokument in Form einer verschachtelten Hierarchie gegliedert.	**Ebenenreferenzen (levelrefs):** Entspricht der im Dokumenthauptteil definierten Inhaltsverzeichnis-Ebenenstruktur. Dadurch können Inhaltsverzeichnis-Referenzen (TOCrefs) innerhalb einer bestimmten Ebene auf Inhaltsverzeichnis-Markierungen auf der entsprechenden Ebene verweisen.
Inhaltsverz.-Markierung: Hat einen Namen, mit dem es einen Node im Dokument als Inhaltsverzeichnis-Eintrag kennzeichnet.	**Inhaltsverz.-Referenzen (TOCrefs):** Referenziert eine Inhaltsverzeichnis-Markierung anhand ihres Namens.

7.6.1 Markieren von Elementen für das Inhaltsverzeichnis

Die Markierung eines Elements im Design für die Aufnahme in das Inhaltsverzeichnis erfolgt in zwei Schritten, welche in beliebiger Reihenfolge ausgeführt werden können:

1. Strukturierung des Design-Dokuments in einer Hierarchie verschachtelter Inhaltsverzeichnis-Ebenen 295 . Eine Inhaltsverzeichnis-Ebene wird im Design entweder in einer Vorlage oder rund um eine Design-Komponente erstellt. In der Abbildung unten wurde in der Vorlage `topic` eine Inhaltsverzeichnis-Ebene erstellt

.



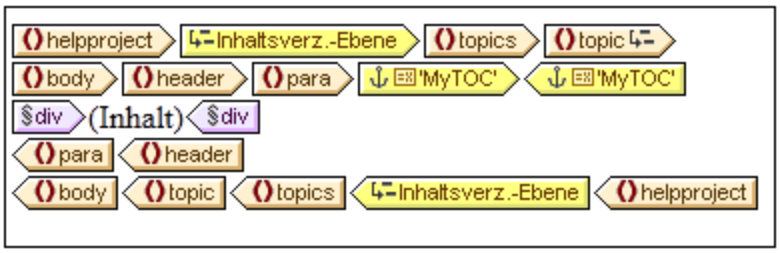
Wenn eine Ebene in einer Vorlage erstellt wird, wird dies durch das Ebenensymbol im Start-Tag der Vorlage gekennzeichnet, z.B.

. Wenn rund um eine Komponente eine Ebene erstellt wird,

wird dies durch Inhaltsverzeichnis-Ebenen-Tags gekennzeichnet

. In der Abbildung oben befindet sich die Vorlage topics innerhalb einer Ebene. Der Unterschied zwischen den beiden Arten der Ebenenmarkierung wird im Abschnitt Strukturieren des Designs in Ebenen[295] erläutert. Eine-Inhaltsverzeichnis-Vorlage[301] muss bei der Erstellung in Form einer Hierarchie von Ebenen gegliedert sein, wobei die Ebenen in der Inhaltsverzeichnis-Vorlage den im Design erstellten Ebenen entsprechen. Auch bei Inhaltsverzeichnissen mit flacher Struktur (eine Ebene) muss das Design über eine entsprechende Ebene verfügen.

2. Erstellen einer Inhaltsverzeichnismarkierung[298] im Design mit einem Namen und Text für den Inhaltsverzeichniseintrag. Die Inhaltsverzeichnismarkierung kann entweder eine Design-Komponente umschließen oder keine Design-Komponente umschließen; im zweiten Fall ist die Markierung leer. In der Abbildung unten enthält die Inhaltsverzeichnismarkierung keine Design-Komponente.



Die Inhaltsverzeichnismarkierung dient als "Anker" im Dokument. In der Abbildung oben befindet sich die Inhaltsverzeichnismarkierung (also der Anker) am Beginn von Instanzen des Elements `para`. Die Inhaltsverzeichnismarkierung hat zwei Attribute: (i) einen Namen, der zum Referenzieren der Inhaltsverzeichnismarkierung dient, wenn in der Inhaltsverzeichnis-Vorlage ein Inhaltsverzeichniseintrag erstellt wird, und (ii) einen Textstring, der als Text für den entsprechenden Inhaltsverzeichniseintrag verwendet wird. Eine Anleitung dazu, wie diese beiden Attribute zugewiesen werden, finden Sie im Abschnitt, Erstellen von Inhaltsverzeichnismarkierungen[298].

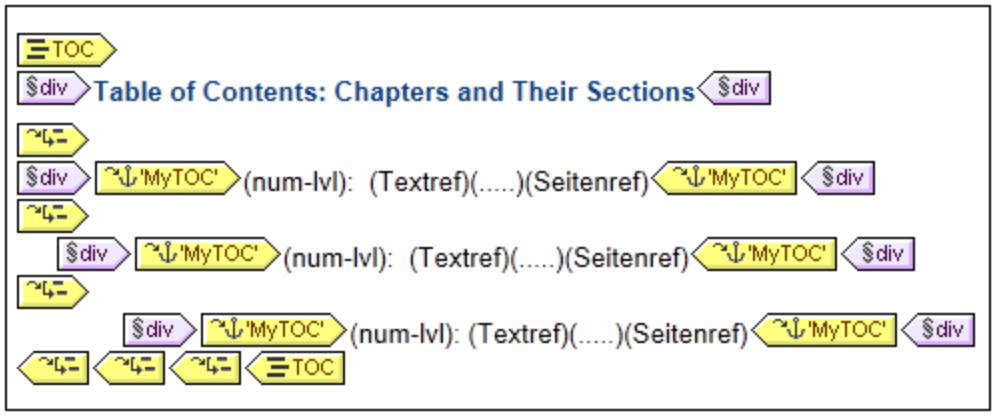
Wie werden markierte Einträge in der Inhaltsverzeichnis-Vorlage referenziert?

Die Inhaltsverzeichnis-Vorlage[301] ist in hierarchisch verschachtelte Ebenen gegliedert (in so genannte Ebenenreferenzen (Ebenenref), um diese von den im Hauptteil der Design-Vorlage erstellten Ebenen zu unterscheiden). Innerhalb jeder Ebenenreferenz wird eine Inhaltsverzeichnis-Referenz (TOCref)

eingefügt (*siehe Abbildung unten*). Die Inhaltsverzeichnis-Referenz innerhalb einer Ebenenreferenz referenziert Inhaltsverzeichnismarkierungen anhand des Namens der Inhaltsverzeichnis-Markierung. Jede Inhaltsverzeichnismarkierung mit diesem Namen, die sich im XML-Dokument innerhalb der entsprechenden Ebene befindet, wird im Inhaltsverzeichnis auf dieser Ebene als Inhaltsverzeichniseintrag angelegt. So referenziert z.B. die Inhaltsverzeichnis-Referenz, die durch den Tag

gekennzeichnet ist, alle Inhaltsverzeichnis-Markierungen, die in der entsprechenden Ebene im XML-Dokument die Bezeichnung `Kapitel` haben (wenn als Geltungsbereich der Inhaltsverzeichnis-Referenz `aktuelle Ebene` gewählt wurde). Das Textattribut der entsprechenden instantiierten Inhaltsverzeichnismarkierungen wird als Text des Inhaltsverzeichniseintrags ausgegeben.

© 2019-2025 Altova GmbH



In der obigen Abbildung einer Inhaltsverzeichnis-Vorlage gibt es drei verschachtelte Ebenenreferenzen. Jede dieser Referenzen enthält eine Inhaltsverzeichnis-Referenz, die die Vorlage für den Inhaltsverzeichniseintrag dieser Ebene enthält. So befindet sich z.B. in der ersten Ebene eine Inhaltsverzeichnisreferenz, die Inhaltsverzeichnismarkierungen referenziert, die den Namen `MyTOC` 'MyTOC' haben. Dadurch werden alle Inhaltsverzeichnismarkierungen auf der ersten Ebene (entsprechend der Struktur des Designs) mit dem Namen `MyTOC` für die Ausgabe auf dieser Ebene des Inhaltsverzeichnisses aufgerufen. Die Inhaltsverzeichnisreferenz auf der zweiten Ebene referenziert ebenfalls Inhaltsverzeichnismarkierungen mit dem Namen `MyTOC`. Dadurch werden alle Inhaltsverzeichnismarkierungen auf der zweiten Ebene (entsprechend der Struktur des Designs) mit dem Namen `MyTOC` für Elemente auf der zweiten Ebene des Inhaltsverzeichnisses verwendet. Die dritte Ebene funktioniert auf dieselbe Art: Inhaltsverzeichnismarkierungen mit dem Namen `MyTOC`, die auf der dritten Ebene des Dokuments vorkommen, werden für Elemente auf der dritten Ebene des Inhaltsverzeichnisses referenziert.

In den Unterabschnitten dieses Abschnitts finden Sie Anleitungen zu folgenden Themen: (i) Strukturierung des Designs in Ebenen 295 und (ii) Erstellung von Inhaltsverzeichnismarkierungen 298. Eine Beschreibung zum Erstellen der Inhaltsverzeichnis-Vorlage 301 finden Sie im Abschnitt Erstellen der Inhaltsverzeichnis-Vorlage 301.

7.6.1.1 Strukturieren des Designs in Inhaltsverzeichnis-Ebenen

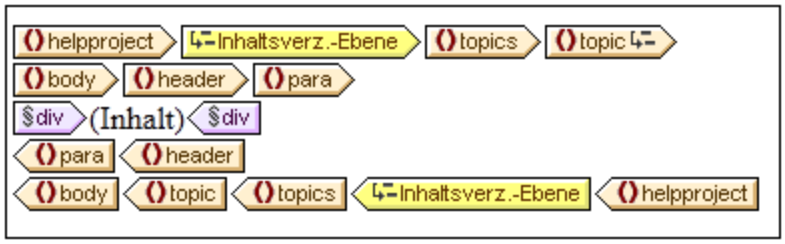
Die hierarchische Struktur, die für das Inhaltsverzeichnis erstellt werden soll, wird als eine Reihe **verschachtelter Ebenen** definiert. Es handelt sich hierbei um eine hierarchische Struktur, die zwar in Beziehung zur Struktur des XML-Dokuments steht, jedoch separat davon erstellt wird. Diese Struktur wird im Dokument-Design des SPS definiert. In der Inhaltsverzeichnis-Vorlage, die Sie dabei erstellen, wird eine Struktur angelegt, die dieser hierarchischen Struktur entspricht. Im Falle eines Inhaltsverzeichnisses mit einer flachen Struktur (nur eine Ebene) muss das Design-Dokument mindestens eine Ebene haben; bei mehr als einer Ebene im Dokument kann ein flaches Inhaltsverzeichnis für jede dieser Ebenen oder für mehrere (zu einer Ebene aggregierte) Ebenen im Dokument-Design erstellt werden.

Ebenen können in im Design der Hauptvorlage, in globalen Vorlagen oder in einer Kombination von Hauptvorlage und globalen Vorlagen erstellt werden. Wichtig ist, dass diese Ebenen - egal wo sie erstellt werden - gemeinsam eine klare hierarchische Struktur definieren müssen.

Erstellen von Ebenen

Jede Ebene im Design wird separat erstellt. Ebenen können in einer Vorlage oder rund um eine Komponente erstellt werden. In der Abbildung unten wurde eine Ebene in der Vorlage `topic` erstellt (gekennzeichnet durch topic) und eine weitere Ebene rund um das Element `topics` (gekennzeichnet durch

TOC level TOC level). Der entscheidende Unterschied zwischen diesen beiden Arten der Ebenenerstellung ist, dass mit der Option zum Einschließen innerhalb einer Ebene TOC level TOC level Ebenen rund um Komponenten erstellt werden können, bei denen es sich nicht um Vorlagen handelt.

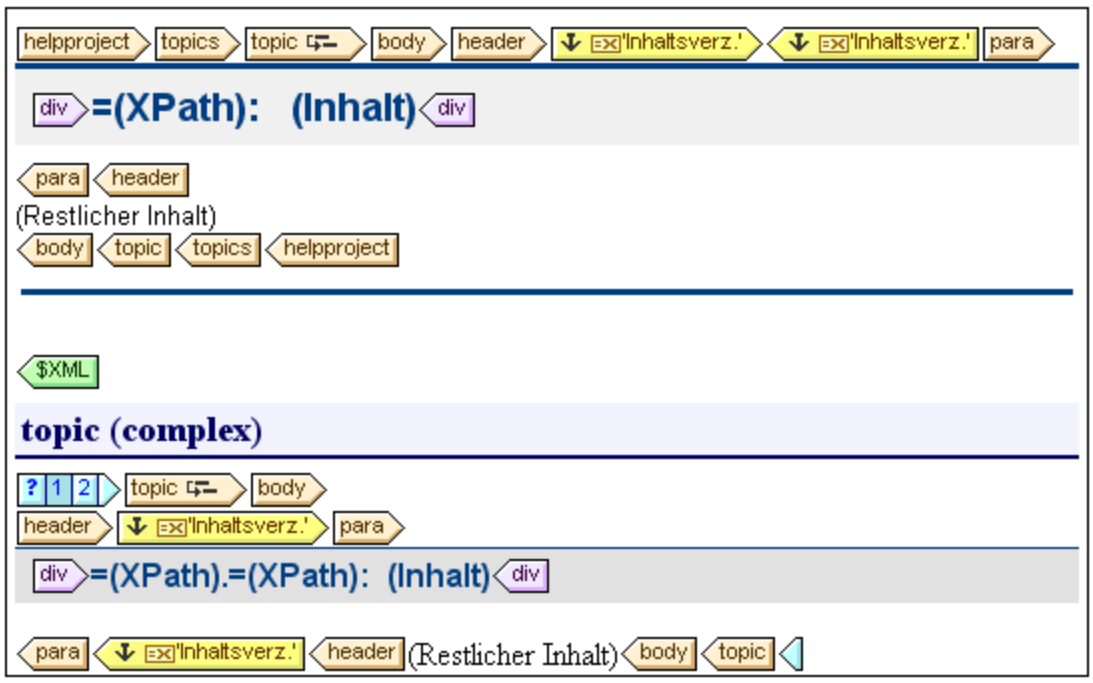


So erstellen Sie eine Ebene:

1. Wählen Sie die Komponente aus (Vorlage oder andere Komponente).
2. Klicken Sie auf die rechte Maustaste und wählen Sie im Kontextmenü entweder die Option **Vorlage dient als Ebene** (aktiv, wenn eine Vorlage ausgewählt wurde) oder die Option **Einschließen in | Inhaltsverz.-Ebene**. Beide Optionen stehen auch im Menü **Einfügen | Inhaltsverzeichnis einfügen** zur Verfügung: **Inhaltsverz.-Ebene** oder **Vorlage dient als Ebene**.

Ebenen in globalen Vorlagen

Ebenen können auch in globalen Vorlagen definiert werden. In diesem Fall müssen Sie darauf achten, dass die in verschiedenen globalen Vorlagen und in der Hauptvorlage erstellten Ebenen **gemeinsam** eine hierarchische Struktur bilden, wenn das SPS ausgeführt wird. In der Abbildung unten sehen Sie zwei Ebenen, eine in der Hauptvorlage (in der Vorlage `topic`) und eine in der globalen Vorlage für `topic` (in der Vorlage `topic`).



Im Content Model, das in der Abbildung oben dargestellt wird, ist `topic` ein rekursives Element, d.h. ein `topic`-Element kann selbst ein untergeordnetes `topic`-Element enthalten. In der Hauptvorlage (das Ende der

 © 2019-2025 Altova GmbH

Hauptvorlage wird durch den Tag $XML gekennzeichnet) wurde eine Ebene auf der ersten Ebene von `topic` topic angelegt. Die Anweisung `restlicher Inhalt` in der Hauptvorlage legt fest, dass für alle Child-Elemente von `topic/body` mit Ausnahme von `header`Vorlagen angewendet werden. Das bedeutet, dass die globale Vorlage für `topic` Child-Elemente von `topic/body` verarbeitet wird.

In der globalen Vorlage für topic wurde eine Ebene in der Vorlage `topic` definiert (gekennzeichnet durch topic). Diese zweite Ebene der Inhaltsverzeichnishierarchie, welche auf der zweiten Ebene von `topic` Elementen definiert ist, ist in die erste Ebene der Inhaltsverzeichnishierarchie eingebettet. Da diese globale Vorlage ebenfalls über eine Anweisung `restlicher Inhalt` verfügt, wird die globale Vorlage für `topic` auf alle rekursiven `topic` Elemente angewendet, wodurch in der Inhaltsverzeichnishierarchie zusätzliche verschachtelte Ebenen (dritte Ebene, vierte Ebene usw.) erstellt werden.

Beim Erstellen des Designs sollten Sie wissen, wie viele Ebenen im Design angelegt werden, da Sie bei Erstellung der Inhaltsverzeichnis-Vorlage explizit angeben müssen, wie Inhaltsverzeichniseinträge für die einzelnen Ebenen ausgewählt und formatiert werden.

Ebenen in flachen Inhaltsverzeichissen

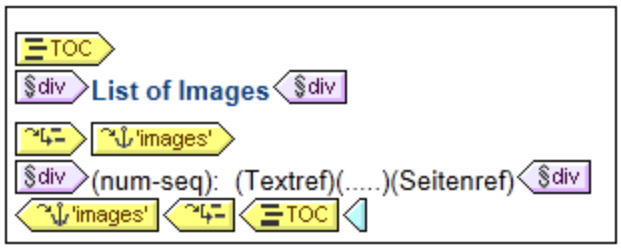
In einer flachen Inhaltsverzeichnishierarchie werden die Inhaltsverzeichniseinträge auf einer einzigen Ebene ausgegeben; z.B. in Form einer einfachen Liste der Bilder in einem Dokument.

Eine flache Hierarchie kann auf verschiedene Arten erzielt werden:

- Das Design-Dokument kann mit nur einer einzigen Inhaltsverzeichnis-Ebene gegliedert werden. In diesem Fall hat die Inhaltsverzeichnis-Vorlage eine einzige Ebenenreferenz mit einer einzigen Inhaltsverzeichnis-Referenz (TOCref) darin.
- Wenn das Design-Dokument mehrere Inhaltsverzeichnis-Ebenen enthält, könnte die Inhaltsverzeichnis-Vorlage mehrere Ebenenreferenzen enthalten, die der jeweiligen Position der referenzierten Inhaltsverzeichnis-Ebene entsprechen. Die Ebenenreferenz, die der gewünschten Inhaltsverzeichnis-Ebene entspricht, enthält dann die einzige Ebenenreferenz in der Inhaltsverzeichnis-Vorlage.
- Wenn das Design-Dokument mehrere Inhaltsverzeichnis-Ebenen enthält, muss die einzige Inhaltsverzeichnisreferenz in der Inhaltsverzeichnis-Vorlage einen Geltungsbereich haben, der alle gewünschten Dokumentebenen umfasst. Alle diese Ebenen werden dann zu einer einzigen Ebene nivelliert.

Angenommen, wir möchten alle Bilder in einem Dokument in einem einzigen Inhaltsverzeichnis mit flacher Hierarchie auflisten. Das Dokument-Design muss daher mindestens eine Ebene enthalten und diese Ebene muss alle erforderlichen Inhaltsverzeichnis-Markierungen enthalten. Die aufzulistenden Bilder werden in der Inhaltsverzeichnis-Vorlage auf die übliche Art referenziert: (i) durch Erstellung einer entsprechenden Anzahl von Ebenenreferenzen; und (ii) durch Erstellung einer Inhaltsverzeichnisreferenz innerhalb der Ebenenreferenzen, die der Zielebene des Inhaltsverzeichnisses entspricht. Die Inhaltsverzeichnisreferenz erhält den Namen der Inhaltsverzeichnis-Markierung auf der Zielebene des Inhaltsverzeichnisses.

In der unten gezeigten Inhaltsverzeichnis-Vorlage gibt es eine Ebenenreferenz, die eine Inhaltsverzeichnisreferenz enthält, die Inhaltsverzeichnis-Markierungen mit dem Namen `images` referenziert. Der Geltungsbereich der Inhaltsverzeichnisreferenz wurde auf *Aktuelle Ebene und unterhalb* gesetzt. Infolgedessen werden alle Inhaltsverzeichnis-Markierungen auf der ersten Ebene und unterhalb davon, die den Namen `images` haben, (also alle Inhaltsverzeichnis-Markierungen im gesamten Dokument) referenziert.



Wenn das Design mehrere Ebenen enthält und ein flaches Inhaltsverzeichnis, z.B. für Elemente auf der ersten Ebene, benötigt wird, könnte die Inhaltsverzeichnis-Vorlage zwei Ebenenreferenzen haben, wobei nur auf der zweiten Ebene eine Inhaltsverzeichnisreferenz vorhanden ist (und keine auf der ersten Ebene). Alternativ dazu können Sie mit Hilfe des Attributs "Geltungsbereich" von Inhaltsverzeichnis-Referenzen definieren, welche Ebene(n) im Design-Dokument nach Markierungen mit einem bestimmten Namen durchsucht werden sollen.

7.6.1.2 Erstellen von Inhaltsverzeichnismarkierungen

Inhaltsverzeichnismarkierungen werden im Dokument-Design innerhalb einer Inhaltsverzeichnisebene[295] erstellt. Sie können in der Hauptvorlage und/oder in globalen Vorlagen erstellt werden. Eine Inhaltsverzeichnismarkierung dient zwei Zwecken:

- Sie markiert eine (statische oder dynamische) Komponente im Design mit einem statischen Namen, den Sie ihr zuweisen. Die Markierung kann entweder eine Design-Komponenten umschließen oder nicht umschließen; im zweiten Fall sind die Tags leer. In der Ausgabe wird die Inhaltsverzeichnismarkierung als Anker, der durch seinen Namen identifiziert wird, instantiiert. Dieser Anker kann über seinen Namen von Elementen in der Inhaltsverzeichnis-Vorlage referenziert werden.
- Eine Inhaltsverzeichnis-Markierung definiert auch den Textstring, der als Text für die Einträge des Inhaltsverzeichnisses verwendet wird. Dabei kann es sich um den Inhalt von Child-Elemente des Node handeln, in dem die Markierung gesetzt wurde, oder um das Ergebnis eines XPath-Ausdrucks.

Sie können eine Inhaltsverzeichnismarkierung auf zwei Arten erstellen:

- mit Hilfe des Assistenten "Inhaltsverzeichnismarkierung erstellen"[298], mit dem Sie den Namen der Inhaltsverzeichnismarkierung definieren können, ihren Texteintrag, ob eine automatische Nummerierung verwendet werden soll und auf welcher Ebene sie aufscheinen soll.
- durch Einfügen einer leeren Inhaltsverzeichnismarkierung[300], deren Eigenschaften Sie erst später definieren.

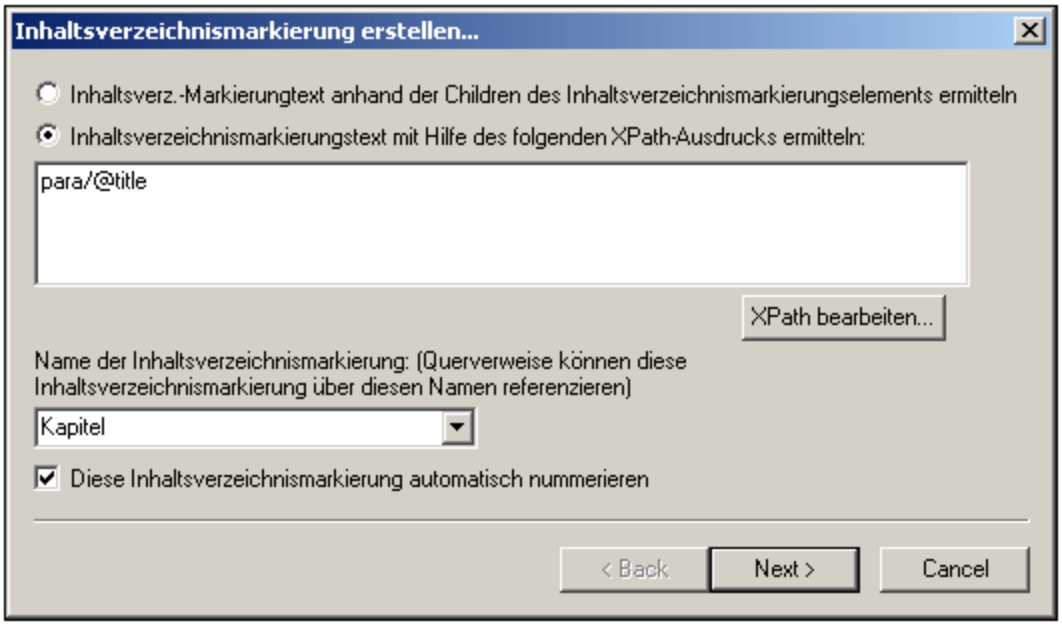
Erstellen von Inhaltsverzeichnismarkierungen mit Hilfe des Assistenten "Inhaltsverzeichnismarkierung erstellen"

So erstellen Sie eine Inhaltsverzeichnismarkierung mit Hilfe des entsprechenden Assistenten:

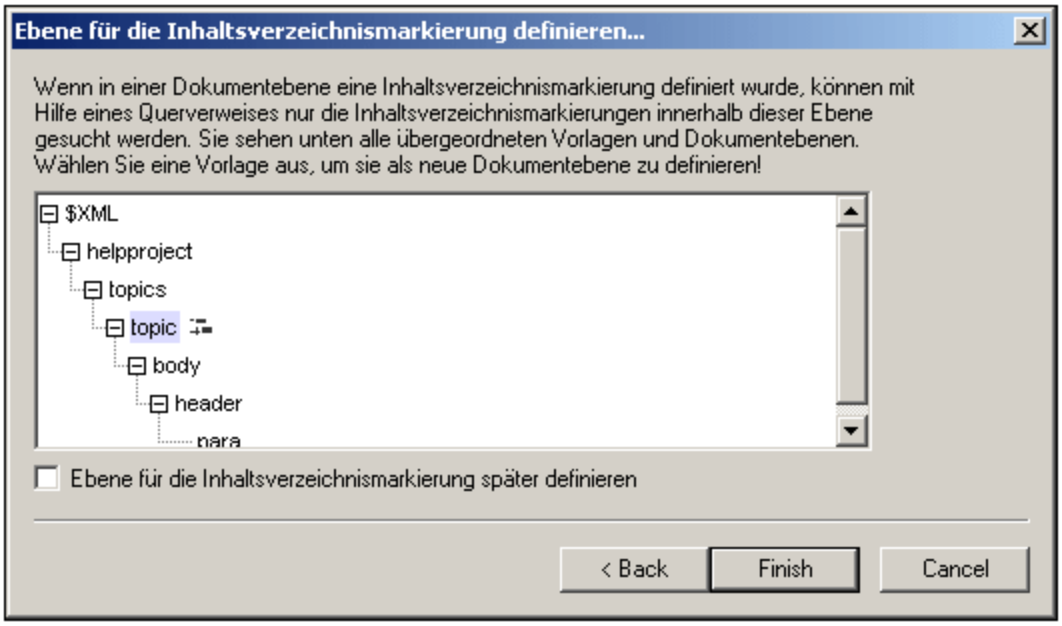
1. Platzieren Sie den Cursor an die Stelle im Design, an der die Inhaltsverzeichnismarkierung eingefügt werden soll oder wählen Sie alternativ dazu eine Design-Komponente, rund um die die Inhaltsverzeichnismarkierung eingefügt werden soll.
2. Wählen Sie über das Kontextmenü (Aufruf durch Rechtsklick) oder im Menü **Einfügen** die Option **Inhaltsverzeichnis einfügen | Inhaltsverz.-Markierung (Assistent)**. Um einen Node in eine Inhaltsverzeichnismarkierung einzuschließen, verwenden Sie den Befehl **Einschließen in |**

 © 2019-2025 Altova GmbH

Inhaltsverz.-Markierung (**Assistent**). Daraufhin wird der Assistent zum Erstellen der Inhaltsverzeichnismarkierung aufgerufen (*Abbildung unten*).



3. Im ersten Fenster des Assistenten (*Abbildung oben*) definieren Sie (i) den Text für den Inhaltsverzeichniseintrag; (ii) den Namen der Inhaltsverzeichnismarkierung; und, (iii) ob diese Markierung in der Ausgabe nummeriert werden soll. Beim Texteintrag können Sie auswählen, ob der Text von Child-Elementen oder ein XPath-Ausdrucks verwendet werden soll. Beim Namen der Inhaltsverzeichnis-Markierung können Sie Text direkt eingeben oder aus einer Dropdown-Liste mit den Namen der bereits definierten Markierungsnamen auswählen. Klicken Sie anschließend auf **Weiter**.
4. Im zweiten Fenster des Assistenten (*Abbildung unten*) können Sie bei Bedarf eine Inhaltsverzeichnis-Ebene in einer Vorlage erstellen. Dem Einfügepunkt übergeordnete Vorlagen werden in Form einer Baumstruktur angezeigt. Wenn eine Vorlage bereits als Inhaltsverzeichnis-Ebene erstellt wurde, wird dies durch ein Symbol gekennzeichnet. In der Abbildung unten zeigt das Symbol neben der Vorlage `topic` an, dass diese bereits als Ebene erstellt wurde. Wenn Sie in einer der übergeordneten Vorlagen eine zusätzliche Ebene anlegen möchten, wählen Sie diese Vorlage aus. Alternativ dazu können Sie die Ebene auch später definieren, indem Sie das Kontrollkästchen "Ebene später definieren" aktivieren. Klicken Sie anschließend auf **Fertig stellen.** (Beachten Sie: Wenn bereits eine Inhaltsverzeichnis-Ebene in einer Vorlage vorhanden ist, wird bei Auswahl dieser Vorlage und Klicken auf **Fertig stellen** keine neue Inhaltsverzeichnis-Ebene erstellt).



Wenn Sie auf **Fertig stellen** klicken, wird am Einfügepunkt eine Inhaltsverzeichnismarkierung erstellt. Falls im zweiten Bildschirm des Assistenten definiert, wird in einer Vorlage eine Inhaltsverzeichnis-Ebene erstellt. Die Inhaltsverzeichnismarkierung befindet sich dann auf der Inhaltsverzeichnis-Ebene, die diese unmittelbar enthält. Wenn diese Inhaltsverzeichnisebene z.B. die dritte Inhaltsverzeichnisebene in der Inhaltsverzeichnisebenenhierarchie ist, so befindet sich die eingefügte Inhaltsverzeichnismarkierung auf der dritten Inhaltsverzeichnisebene.

Erstellen einer Inhaltsverzeichnismarkierung

So erstellen Sie eine Inhaltsverzeichnismarkierung ohne Attribute:

1. Platzieren Sie den Cursor an die Stelle im Design-Dokument, an der die Inhaltsverzeichnismarkierung eingefügt werden soll oder wählen Sie die Design-Komponente aus, um die herum die Inhaltsverzeichnismarkierung erstellt werden soll.
2. Wählen Sie im Kontextmenü (Aufruf durch Rechtsklick) oder im Menü **Einfügen** die Option **Inhaltsverzeichnis einfügen | Inhaltsverz.-Markierung**. Daraufhin wird eine Inhaltsverzeichnismarkierung eingefügt. Diese Markierung hat weder einen Namen noch einen Texteintrag. Sie können diese Attribute später mit Hilfe der Bearbeitungsbefehle[301] (*siehe unten*) definieren.

Einfügen einer hierarchischen oder sequenziellen Nummerierung für eine Komponente

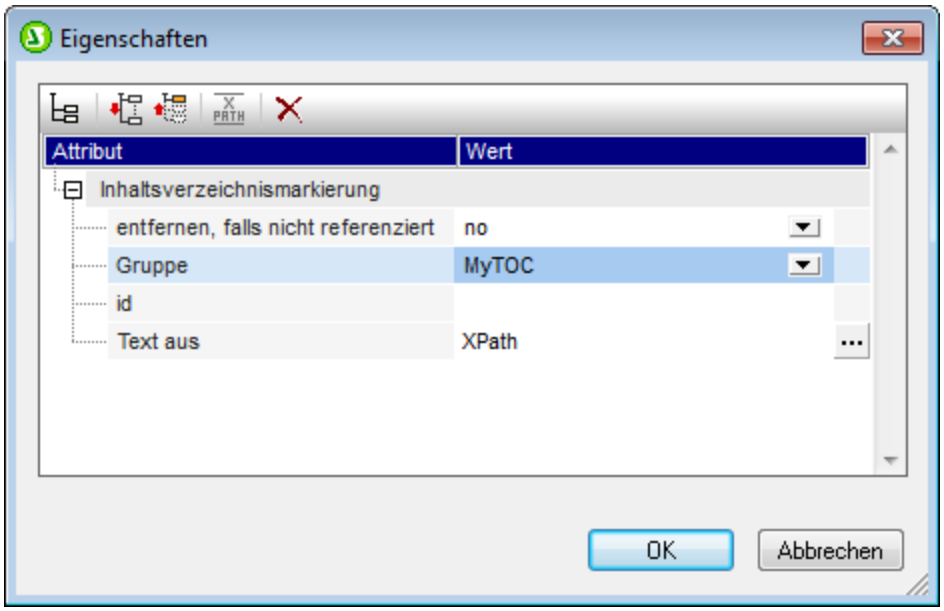
Eine hierarchische oder sequenzielle Nummerierung im Hauptteil des Ausgabedokuments (nicht innerhalb des Inhaltsverzeichnisses) kann innerhalb (aber auch außerhalb) der Tags einer Inhaltsverzeichznismarkierung eingefügt werden. Rechtsklicken Sie an die Stelle, an der Sie die Nummerierung einfügen möchten, und wählen Sie den Befehl **Inhaltsverzeichnis einfügen | Hierarchische Nummerierung / Sequenzielle Nummerierung**. So wird z.B. mit einer Inhaltsverzeichnismarkierung mit automatischer Nummerierung, die rund um die Kapitelüberschriftsvorlage platziert wird, eine Nummerierung für alle von der Kapitelüberschriftsvorlage generierten Kapitelüberschriften generiert.

© 2019-2025 Altova GmbH

Beachten Sie, dass die Nummerierung auf der Gliederung der Inhaltsverzeichnisebenen basiert. Wenn sich z.B. ein Kapitelüberschriftselement auf der ersten Inhaltsverzeichnisebene befindet, so erhält die vierte Kapitelüberschrift die Nummer 4, da es sich um die vierte Instanz einer Kapitelüberschrift auf der ersten Inhaltsverzeichnisebene handelt. Wenn sich die Abschnitte eines Kapitels auf der zweiten Inhaltsverzeichnisebene befinden, so erhält der dritte Abschnitt des vierten Kapitels die Nummer `4.3`, da es sich um die vierte Instanz eines Kapitels auf der ersten (Kapitel) Inhaltsverzeichnisebene und um die dritte Instanz eines Abschnitts auf der zweiten Inhaltsverzeichnisebene (Abschnitt) (des vierten Kapitels) handelt.

Bearbeiten des Names und des Texteintrags einer Inhaltsverzeichnismarkierung

Der Name und der Texteintrag einer Inhaltsverzeichnismarkierung können im Fenster "Eigenschaften" bearbeitet werden (*Abbildung unten*). Wählen Sie dazu die Inhaltsverzeichnismarkierung aus und bearbeiten Sie die Eigenschaft entweder direkt im Fenster "Eigenschaften" 49 oder rechtsklicken Sie auf die Inhaltsverzeichnismarkierung und wählen Sie die Eigenschaft aus, die Sie bearbeiten möchten.



Die Inhaltsverzeichnismarkierung hat die folgenden Eigenschaften: (i) den Namen derr Inhaltsverzeichnismarkierungsgruppe (*Gruppe*); (ii) eine eindeutige ID; (iii) ein Option, die Markierung zu entfernen, falls sie nicht referenziert wird und (iii) eine Option (*Text aus*) zur Definition des Texteintrags, der entweder aus dem Inhalt der Inhaltsverzeichnismarkierung oder aus einem XPath-Ausdruck stammen kann.

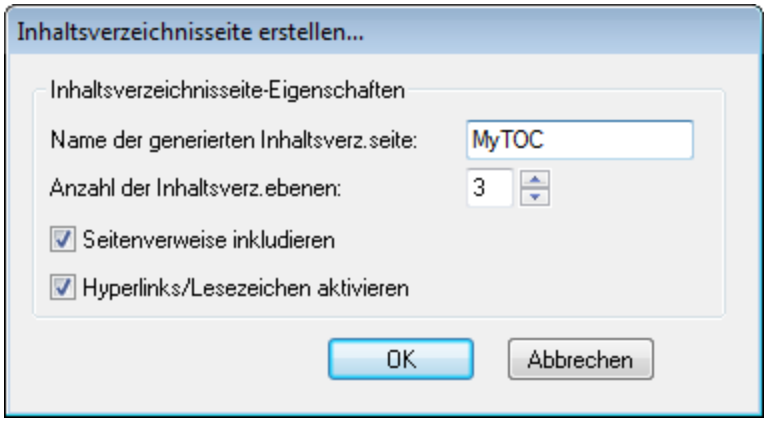
7.6.2 Erstellen der Inhaltsverzeichnis-Vorlage

Die Inhaltsverzeichnis-Vorlage ist die Vorlage, anhand der in der Ausgabe das Inhaltsverzeichnis erstellt wird. Sie kann an jeder beliebigen Stelle im SPS-Design angelegt werden. Es können in einem einzigen SPS-Design auch mehrere Inhaltsverzeichnis-Vorlagen erstellt werden.

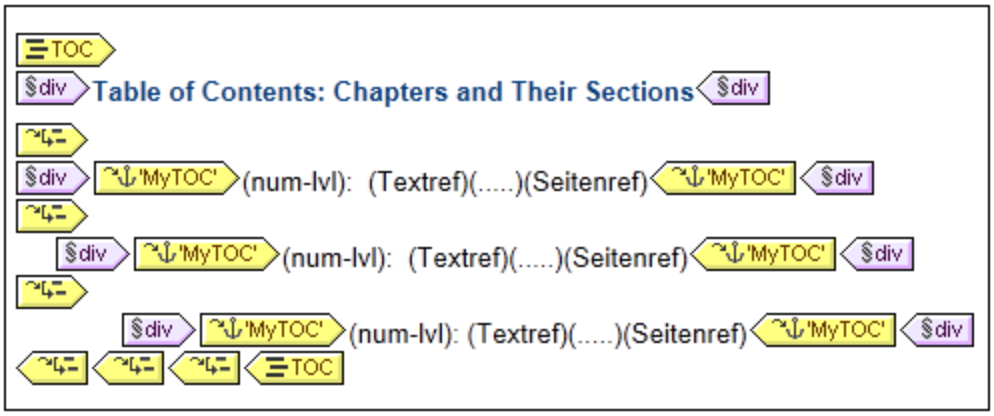
So erstellen Sie eine Inhaltsverzeichnis-Vorlage:

1. Platzieren Sie den Cursor an die Stelle, an der die Inhaltsverzeichnis-Vorlage eingefügt werden soll.

2. Klicken Sie auf den Menübefehl **Einfügen | Inhaltsverzeichnis einfügen | Inhaltsverzeichnis**. Daraufhin wird das Dialogfeld "Inhaltsverzeichnisseite erstellen" (*Abbildung unten*) angezeigt. (Alternativ dazu kann dieser Befehl auch über das Kontextmenü aufgerufen werden, welches durch einen Rechtsklick aufgerufen werden kann.)



3. Geben Sie die erforderlichen Informationen in das Dialogfeld ein: (i) Der Namen der generierten Inhaltsverzeichnisseite ist der (Inhaltsverzref)-Name, der dazu verwendet wird, um die Inhaltsverzeichnismarkierung[298] im Design-Dokument zu referenzieren. Wenn Sie mehrere Ebenen für das Inhaltsverzeichnis (genauer gesagt Ebenenreferenzen; nächste Option) auswählen, wird derselbe Inhaltsverzeichnisreferenzname auf allen Ebenenreferenzen verwendet (obwohl einzelne Inhaltsverzref-Namen später bearbeitet werden können[304]). (ii) Die Anzahl der Ebenenreferenzen des Inhaltsverzeichnisses[295] gibt an, wie viele Ebenen das Inhaltsverzeichnis haben soll. (iii) Bei gedruckten Medien steht die Option zur Ausgabe von Seitenreferenzen (d.h. Seitennummern) zur Verfügung. (iv) Die Texteinträge im Inhaltsverzeichnis können als Links zu Inhaltsverzeichnismarkierungen verwendet werden.
4. Klicken Sie zum Fertigstellen auf **OK**. Die Inhaltsverzeichnis-Vorlage wird mit der angegebenen Anzahl an Ebenenreferenzen erstellt (*Abbildung unten; die Formatierung der Inhaltsverzeichnis-Vorlage wurde geändert und unterscheidet sich von der anfangs erstellten Vorlage*).



Innerhalb jeder Ebenenreferenz befindet sich eine Inhaltsverzeichnisreferenz mit einem Namen, der Inhaltsverzeichnismarkierungen identifiziert, die für diese Ebenenreferenz des Inhaltsverzeichnisses als Inhaltsverzeichniseinträge verwendet werden sollen. Innerhalb jeder Inhaltsverzeichnisreferenz befindet sich eine Standardvorlage für den Inhaltsverzeichniseintrag, die Sie jederzeit bearbeiten können[305].

 © 2019-2025 Altova GmbH

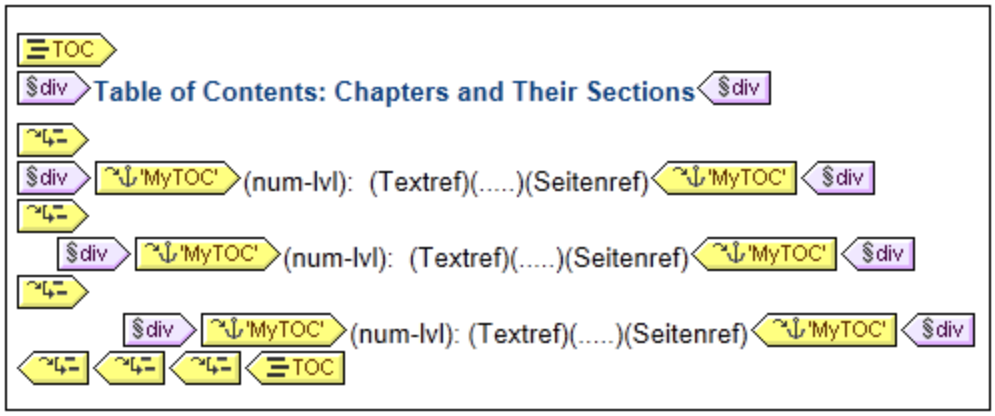
Bearbeiten der Inhaltsverzeichnis-Vorlage

Es stehen die folgenden Bearbeitungsoptionen zur Verfügung:

- Die Inhaltsverzeichnis-Vorlage kann an eine andere Stelle im SPS gezogen werden. Beachten Sie jedoch, dass sich eine Änderung des Kontext Node auf XPath-Ausdrücke innerhalb der Inhaltsverzeichnis-Vorlage auswirken könnte.
- Ebenenreferenzen[303] können zur Struktur der Inhaltsverzeichnis-Vorlage hinzugefügt oder daraus gelöscht werden.
- Die Eigenschaften einzelner Inhaltsverzeichnisreferenzen[304] (Inhaltsverzref) können bearbeitet werden. Der Name und Geltungsbereich einer Inhaltsverzref können geändert werden und Sie können auswählen, ob der Inhaltsverzeichniseintrag, der der Inhaltsverzref entspricht als Hyperlink erstellt werden soll oder nicht.
- Inhaltsverzeichnisreferenzen[304] können zu jeder Ebenenreferenz in der Inhaltsverzeichnis-Vorlage hinzugefügt bzw. daraus gelöscht werden.
- Der Inhaltsverzeichniseintrag[305] innerhalb einer Inhaltsverzref kann unter Verwendung der Standard StyleVision Verfahren[342] mit CSS-Eigenschaften formatiert werden.
- Standard SPS Features (wie z.B. Bilder, automatische Berechnungen und Blockformatierung können an einer beliebigen Stellen in der Inhaltsverzeichnis-Vorlage eingefügt werden.

7.6.2.1 Ebenenreferenzen in der Inhaltsverzeichnis-Vorlage

Die Inhaltsverzeichnis-Vorlage[301] ist in **Ebenenreferenzen (oder Ebenenref)** gegliedert; *siehe Abbildung unten*. Diese Ebenen werden anfangs bei Erstellung der Inhaltsverzeichnis-Vorlage erstellt. Die Anzahl der Ebenenreferenzen entspricht der Anzahl, die Sie im Dialogfeld "Inhaltsverzeichnisseite erstellen"[301] definiert haben.



Beachten Sie, dass die Ebenenreferenzen verschachtelt sind. Beim Design des Inhaltsverzeichnisses gibt es eine Eins-zu-Eins-Entsprechung zwischen den Ebenenreferenzen in der Inhaltsverzeichnis-Vorlage und den Ebenen im SPS Design. D.h. die erste Ebenenreferenz der Inhaltsverzeichnis-Vorlage entspricht der ersten Ebene im SPS Design, die zweite Ebenenreferenz in der Inhaltsverzeichnis-Vorlage entspricht der zweiten Ebene im SPS Design usw. Die Inhaltsverzeichnisreferenzen innerhalb einer bestimmten Ebenenreferenz der Inhaltsverzeichnis-Vorlage verweisen auf Inhaltsverzeichnismarkierungen[298] innerhalb eines festgelegten Geltungsbereiches[304] im SPS Design. So kann z.B. mit einer Ebenenreferenz definiert werden, dass die Inhaltsverzeichnisreferenz auf Inhaltsverzeichnismarkierungen auf der entsprechenden Dokumentebene, auf

Inhaltsverzeichnismarkierungen auf allen Dokumentebenen oder auf solche auf der aktuellen Dokumentebene und diesen untergeordnete Ebenen verweist.

Einfügen und Entfernen von Ebenenreferenzen

Ebenenreferenzen können in die Inhaltsverzeichnis-Vorlage eingefügt oder daraus gelöscht werden, nachdem diese erstellt wurde.

Um eine Ebenenreferenz rund um Inhalt einzufügen, wählen Sie in der Inhaltsverzeichnis-Vorlage den Inhalt aus, um den herum eine Ebenenreferenz erstellt werden soll und wählen Sie anschließend im Kontextmenü oder über die Menüleiste den Befehl **Einschließen in | Inhaltsverz.-Ebenenreferenz.** Eine Ebenenreferenz kann auch über den Menübefehl **Einfügen | Inhaltsverzeichnis einfügen | Inhaltsverz.-Ebenenreferenz** an der Cursorposition in die Inhaltsverzeichnis-Vorlage eingefügt werden (Dieser Befehl steht auch über das Kontextmenü zur Verfügung).

Um eine Ebenenreferenz aus der Inhaltsverzeichnis-Vorlage zu löschen, wählen Sie die gewünschte Referenz aus und drücken Sie entweder die **Entf**-Taste oder wählen Sie im Kontextmenü den Befehl **Entfernen**. Beachten Sie, dass nur die Ebenenreferenz entfernt wird - nicht aber ihr Inhalt.

7.6.2.2 Inhaltsverzeichnisreferenz: Name, Geltungsbereich, Hyperlink

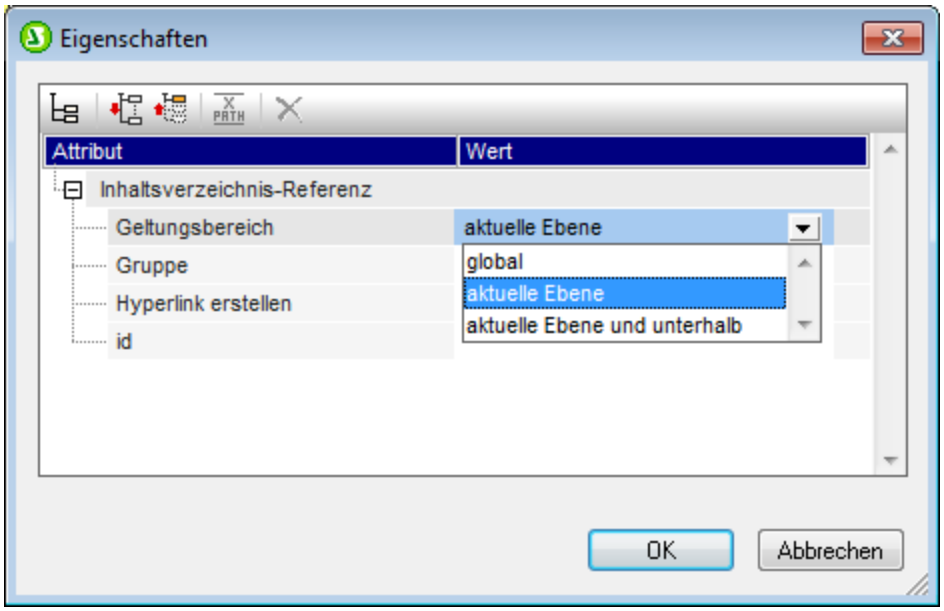
Inhaltsverzeichnisreferenzen (Inhaltsverzref) befinden sich innerhalb von Ebenenreferenzen (Ebenenref) und haben vier Eigenschaften (*siehe Abbildung unten*):

- eine Eigenschaft ***Hyperlink,*** die zwischen `ja` und `nein` hin- und hergeschaltet werden kann, um festzulegen, ob die entsprechenden Inhaltsverzeichniseinträge als Hyperlinks erstellt werden sollen oder nicht.
- eine Eigenschaft ***Gruppe,*** die den Namen der Inhaltsverzeichnisreferenz angibt und Inhaltsverzeichnismarkierungen desselben Namens, die innerhalb des angegebenen Geltungsbereichs vorkommen, identifiziert (*siehe unten*). Die damit ermittelten Inhaltsverzeichnismarkierungen liefern die Einträge, die auf dieser Ebenenreferenz des Inhaltsverzeicnisses inkludiert werden sollen.
- eine ***id,*** mit der die Inhaltsverzeichnisreferenz eindeutig identifiziert werden kann.
- einen ***Geltungsbereich***, mit dem festgelegt wird, für welche Ebenen im SPS Design die Inhaltsverzeichnisreferenz gilt. Es stehen drei Optionen zur Verfügung: (i) global, (ii) aktuelle Ebene, (ii) aktuelle Ebene und die Ebenen unterhalb davon (*siehe Abbildung unten*).

Um eine Inhaltsverzeichnisreferenz einzufügen, platzieren Sie den Cursor in einen Ebenenreferenz und wählen Sie im Menü **Einfügen** oder im Kontextmenü den Befehl **Inhaltsverzeichnis einfügen | Inhaltsverz.-Referenz**.

Um die Eigenschaft einer Inhaltsverz.-Referenz zu bearbeiten, klicken Sie mit der rechten Maustaste in der Inhaltsverzeichnis-Vorlage auf den Inhaltsverzeichnisref.-Tag und wählen Sie die gewünschte Eigenschaft (*Hyperlink erstellen*, *ID bearbeiten, Gruppe bearbeiten* oder *Geltungsbereich bearbeiten*) aus. Daraufhin wird das Fenster "Eigenschaften" angezeigt, in dem die jeweilige Eigenschaft zur Bearbeitung ausgewählt ist (*Abbildung unten*)

 © 2019-2025 Altova GmbH

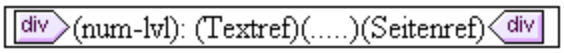


Alternativ dazu können Sie auch direkt zur gewünschten Eigenschaft im Fenster "Eigenschaften" gehen, wenn der Inhaltsverzref-Tag ausgewählt ist (Eigenschaftsgruppe *Inhaltsverz-Referenz*).

7.6.2.3 Formatieren von Inhaltsverzeichniseinträgen

Der Inhaltsverzeichniseintrag kann bis zu vier Standardkomponenten enthalten sowie optionalen durch den Benutzer definierten Inhalt. Die vier Standardkomponenten sind (*siehe Abbildung unten*):

- der Texteintrag des Inhaltsverzeichniseintrags, der in der Inhaltsverzeichnis-Vorlage durch `(Textref)` gekennzeichnet ist
- die punktierte Linie zwischen dem Texteintrag und der Seitennummer (bei Ausgabe in paginierten Medien), gekennzeichnet durch `(.....)`
- die Seitenreferenz des Inhaltsverzeichniseintrags (bei Ausgabe in paginierten Medien), gekennzeichnet durch `(Seitenref)`
- eine hierarchische oder sequenzielle Nummerierung, gekennzeichnet durch `(num-lvl)` bzw. `(num-seq)`



Wenn die Inhaltsverzeichnis-Vorlage anfangs erstellt wird, wird der Texteintrag automatisch innerhalb von Inhaltsverzeichnisreferenzen eingefügt. Wenn die Option "Seitenreferenz inkludieren" aktiviert war, werden auch die punktierte Linie und die Seitenreferenzkomponenten inkludiert. Komponenten können anschließend zum Inhaltsverzeichniseintrag hinzugefügt oder daraus gelöscht werden. Um eine Komponente einzufügen, platzieren Sie den Cursor an der gewünschten Position innerhalb des Inhaltsverzeichniseintrags und wählen Sie im Kontextmenü den Befehl **Inhaltsverzeichnis einfügen | Inhaltsverz.-Referenz | Eingabetext / Punktierte Linie / Seitenverweis** oder **Inhaltsverzeichnis einfügen | Hierarchische Nummerierung / Sequenzielle Nummerierung**. (Eine hierarchische Nummerierung sollte eingefügt werden, wenn das Design in verschachtelte Ebenen gegliedert ist, eine sequentielle Nummerierung, wenn keine Hierarchie vorhanden ist, d.h., wenn es nur eine, flache Inhaltsverzeichnis-Ebene gibt. Siehe Anmerkung unten zu flachen

Inhaltsverzeichnis-Ebenen). Um eine Komponente zu löschen, wählen Sie sie aus und drücken Sie die **Entf**-Taste.

Zusätzlich dazu können Sie statischen Inhalt (z.B. Text) und dynamischen Inhalt (z.B. automatische Berechnungen) in den Inhaltsverzeichniseintrag einfügen.

Ebenen in flachen Inhaltsverzeichnissen

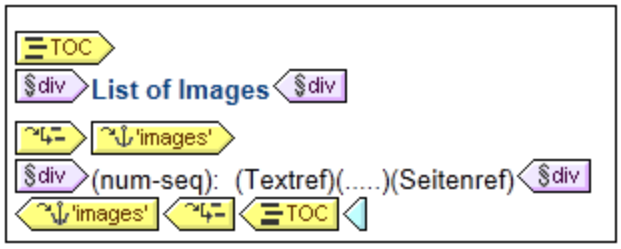
In einer flachen Inhaltsverzeichnishierarchie werden die Inhaltsverzeichniseinträge auf einer einzigen Ebene ausgegeben; z.B. in Form einer einfachen Liste der Bilder in einem Dokument.

Eine flache Hierarchie kann auf verschiedene Arten erzielt werden:

- Das Design-Dokument kann mit nur einer einzigen Inhaltsverzeichnis-Ebene gegliedert werden. In diesem Fall hat die Inhaltsverzeichnis-Vorlage eine einzige Ebenenreferenz mit einer einzigen Inhaltsverzeichnis-Referenz (TOCref) darin.
- Wenn das Design-Dokument mehrere Inhaltsverzeichnis-Ebenen enthält, könnte die Inhaltsverzeichnis-Vorlage mehrere Ebenenreferenzen enthalten, die der jeweiligen Position der referenzierten Inhaltsverzeichnis-Ebene entsprechen. Die Ebenenreferenz, die der gewünschten Inhaltsverzeichnis-Ebene entspricht, enthält dann die einzige Ebenenreferenz in der Inhaltsverzeichnis-Vorlage.
- Wenn das Design-Dokument mehrere Inhaltsverzeichnis-Ebenen enthält, muss die einzige Inhaltsverzeichnisreferenz in der Inhaltsverzeichnis-Vorlage einen Geltungsbereich haben, der alle gewünschten Dokumentebenen umfasst. Alle diese Ebenen werden dann zu einer einzigen Ebene nivelliert.

Angenommen, wir möchten alle Bilder in einem Dokument in einem einzigen Inhaltsverzeichnis mit flacher Hierarchie auflisten. Das Dokument-Design muss daher mindestens eine Ebene enthalten und diese Ebene muss alle erforderlichen Inhaltsverzeichnis-Markierungen enthalten. Die aufzulistenden Bilder werden in der Inhaltsverzeichnis-Vorlage auf die übliche Art referenziert: (i) durch Erstellung einer entsprechenden Anzahl von Ebenenreferenzen; und (ii) durch Erstellung einer Inhaltsverzeichnisreferenz innerhalb der Ebenenreferenzen, die der Zielebene des Inhaltsverzeichnisses entspricht. Die Inhaltsverzeichnisreferenz erhält den Namen der Inhaltsverzeichnis-Markierung auf der Zielebene des Inhaltsverzeichnisses.

In der unten gezeigten Inhaltsverzeichnis-Vorlage gibt es eine Ebenenreferenz, die eine Inhaltsverzeichnisreferenz enthält, die Inhaltsverzeichnis-Markierungen mit dem Namen `images` referenziert. Der Geltungsbereich der Inhaltsverzeichnisreferenz wurde auf *Aktuelle Ebene und unterhalb* gesetzt. Infolgedessen werden alle Inhaltsverzeichnis-Markierungen auf der ersten Ebene und unterhalb davon, die den Namen `images` haben, (also alle Inhaltsverzeichnis-Markierungen im gesamten Dokument) referenziert.



Wenn das Design mehrere Ebenen enthält und ein flaches Inhaltsverzeichnis, z.B. für Elemente auf der ersten Ebene, benötigt wird, könnte die Inhaltsverzeichnis-Vorlage zwei Ebenenreferenzen haben, wobei nur auf der zweiten Ebene eine Inhaltsverzeichnisreferenz vorhanden ist (und keine auf der ersten Ebene). Alternativ dazu können Sie mit Hilfe des Attributs "Geltungsbereich" von Inhaltsverzeichnis-Referenzen definieren, welche Ebene(n) im Design-Dokument nach Markierungen mit einem bestimmten Namen durchsucht werden sollen.

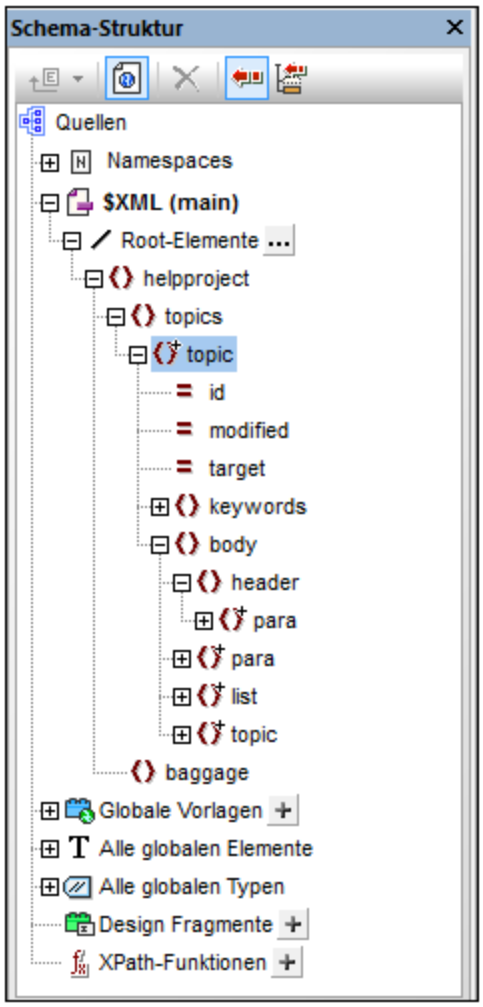
 © 2019-2025 Altova GmbH

Formatieren des Inhaltsverzeichniseintrags

Der Inhaltsverzeichniseintrag kann über die Seitenleiste "Stile" 348 mit CSS-Stilen 342 formatiert werden. Einzelne Inhaltsverzeichniseintrag-Komponenten können separat formatiert werden. Wählen Sie dazu die Komponente aus und weisen Sie ihr in der Seitenleiste "Stile" Stileigenschaften 348 zu.

7.6.3 Beispiel: Einfaches Inhaltsverzeichnis

Eine Beispiel-SPS-Datei mit dem Namen `ChaptersSimple.sps`, in der die Grundlagen der Verwendung von Inhaltsverzeichnissen gezeigt werden, befindet sich im Ordner Eigene Dateien 25, `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2025\StyleVisionExamples\Tutorial\TOC`. Dieses SPS basiert auf einem Schema, in dem das Inhaltsmodell eines großen in Kapitel unterteilten Dokuments definiert ist. In der Abbildung unten sehen Sie die Struktur des Schemas. Wenn Sie die Datei `ChaptersSimple.sps` öffnen, sehen Sie die Struktur im Schema-Struktur-Fenster von StyleVision. (Ein komplexeres Beispiel für ein Inhaltsverzeichnis basierend auf demselben Schema ist im nächsten Abschnitt dieser Dokumentation, unter Beispiel: Hierarchische und sequenzielle Inhaltsverzeichnisse 310 beschrieben.)



Das Dokumentelement ist `helpproject`. Es enthält ein Child-Element `topics`. Das Element `topics` kann unbeschränkt viele `topic` Elemente enthalten, von denen jedes wiederum untergeordnete `topic` Elemente enthalten kann. Die erste Ebene der `topic` Elemente sind die Kapitel des Dokuments, die untergeordneten `topic` Elemente sind Abschnitte, Unterabschnitte usw.

Mit diesem SPS wird im obersten Bereich des Dokuments ein Inhaltsverzeichnis erstellt, das die Namen der einzelnen Kapitel (der "topics" auf der obersten Ebene) enthält. Die Erstellung des Inhaltsverzeichnisses erfolgt in drei Schritten:

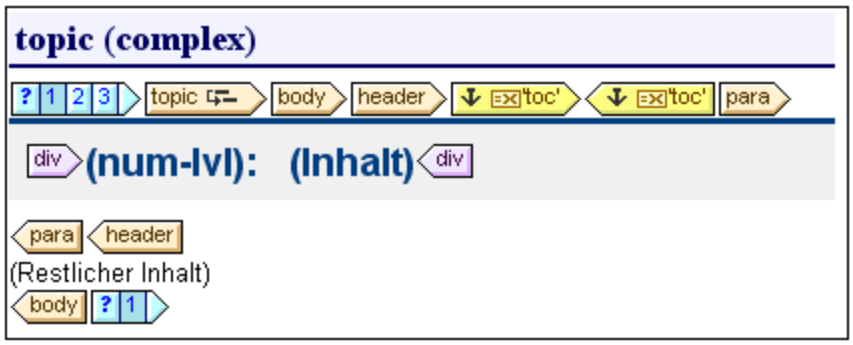
1. Gliedern des Designs in Inhaltsverzeichnis-Ebenen[295]: In das Design-Dokument werden eine oder mehrere Ebenen eingefügt, um das Ausgabedokument hierarchisch zu gliedern. Diese hierarchische Struktur entspricht der des Inhaltsverzeichnisses. In unserem aktuellen Beispiel wurde zum besseren Verständnis nur eine einzige Ebene erstellt - in der Vorlage `Topic`. Infolgedessen gibt es im Design nur eine Ebene und die **Inhaltsverzeichnisvorlage** kann - sobald sie einmal erstellt wurde - in der Gliederung nur eine Ebene (d.h. eine Ebenenreferenz) enthalten.
2. Erstellen von Inhaltsverzeichnis-Markierungen[298]: Innerhalb der in Schritt 1 (im Design-Dokument) erstellten Inhaltsverzeichnis-Ebene wird eine **Inhaltsverzeichnis-Markierung** angelegt. Mit Hilfe dieser Markierung können Inhaltsverzeichnis-Referenzen in der im nächsten Schritt erstellten Inhaltsverzeichnis-Vorlage auf diese Inhaltsverzeichnis-Markierung verweisen. Die Inhaltsverzeichnis-Markierung definiert auch den Text, der im Inhaltsverzeichnis-Eintrag, der darauf verweist, angezeigt wird.
3. Erstellen der Inhaltsverzeichnis-Vorlage[301]: Dies ist die Vorlage, mit der das Inhaltsverzeichnis im Dokument erstellt wird. Sie ist in Ebenen-Referenzen gegliedert, die der Struktur der Inhaltsverzeichnis-Ebenen im Design-Dokument entsprechen müssen. Wenn die Inhaltsverzeichnisvorlage z.B. drei verschachtelte Ebenenreferenzen enthält, so muss das Design-Dokument mindestens drei Ebenen haben. In diesem Beispiel haben wir eine einzige Ebenenreferenz, die der einzigen Inhaltsverzeichnis-Ebene im Design-Dokument entspricht. Sie befindet sich innerhalb des Ebenenreferenz, in die die **Inhaltsverzeichnis-Referenz** platziert wurde. Anhand dieser Inhaltsverzeichnis-Referenz werden die Einträge des Inhaltsverzeichnisses für diese Ebene des Inhaltsverzeichnisses generiert.

SPS-Struktur und -Ebenen

Werfen Sie einen Blick auf die Struktur des Designs im SPS. Wie Sie sehen, enthält die Hauptvorlage (mit den grünen `$XML` Tags) das Inhaltsverzeichnis. Im Rest der Hauptvorlage wird mittels einer `restlicher Inhalt`-Anweisung festgelegt, dass globale und Standardvorlagen anzuwenden sind. Das restliche SPS-Design, außerhalb der Hauptvorlage und dahinter besteht aus globalen Vorlagen.

Die Inhaltsverzeichnis-Definitionen (Inhaltsverzeichnis-Ebenen und Inhaltsverzeichnis-Markierungen im Design) befinden sich in der globalen Vorlage für `topic` (*Abbildung unten*). In diese globale Vorlage wurde eine Bedingung eingefügt, um `topic` Elemente anhand der Anzahl ihrer übergeordneten `topic` Elemente zu trennen, sodass Kapitel, Abschnitte und Unterabschnitte (innerhalb unterschiedlicher Bedingungsverzweigungen) unterschiedlich verarbeitet werden.

© 2019-2025 Altova GmbH

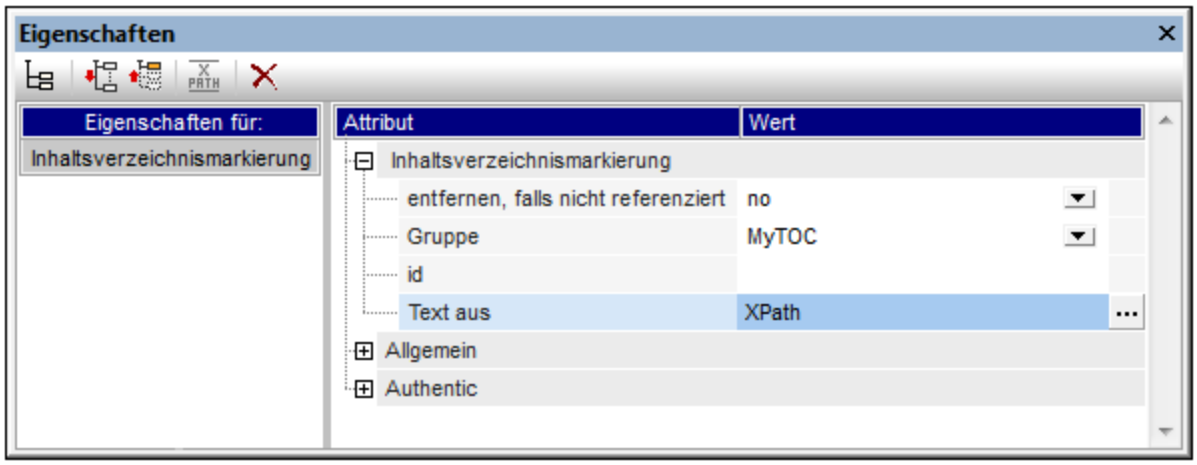


In der Abbildung oben sehen Sie den Inhalt der ersten Bedingungsverzweigung für `topic` Elemente der ersten Ebene, vom Typ Kapitel. Beachten Sie, dass im Start-Tag dieses `topic` Elements eine Inhaltsverzeichnis-Ebene erstellt wurde. In den beiden anderen Bedingungsverzweigungen wurden in der topic-Vorlage keine Inhaltsverzeichnis-Ebenen erstellt. Als Ergebnis davon wurde dem Dokument nur eine Inhaltsverzeichnis-Ebene zugewiesen und zwar auf der Ebene des obersten `topic` Elements (vom Typ "Kapitel").

Erstellen der Inhaltsverzeichnis-Markierung

Im untergeordneten Element `header` des Elements `topic` (aber außerhalb des `para` Elements) wurde eine Inhaltsverzeichnis-Markierung (*in der Abbildung unten gelbe Tags*) erstellt. Diese Inhaltsverzeichnis-Markierung dient als Anker für die obersten `topic` Elemente vom Typ "Kapitel".

Die Eigenschaften der Inhaltsverzeichnis-Markierung können in der Seitenleiste "Eigenschaften" (*Abbildung unten*) bearbeitet werden.

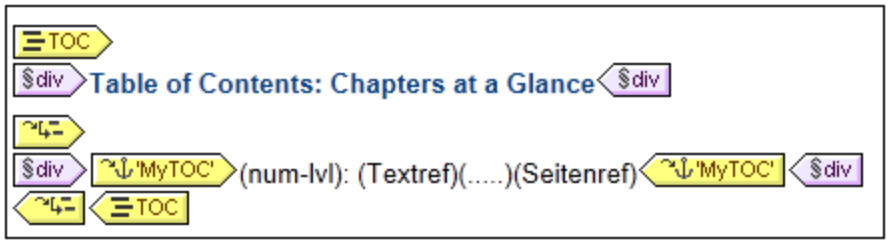


Mit der Eigenschaft *Gruppe* wird die Inhaltsverzeichnismarkierungsgruppe definiert (Diese Eigenschaft ist auch gleichzeitig der Name der Gruppe). In unserem Beispiel haben wir für diese Eigenschaft den Wert `MyTOC` definiert. Die Markierungsgruppe wird bei Erstellung des Inhaltsverzeichnisses referenziert. Mit Hilfe dieser Eigenschaft können unterschiedliche Inhaltsverzeichnisgruppen für dieselbe Ebene definiert werden. Mit Hilfe der Eigenschaft *ID* können für die einzelnen erstellten Markierungsinstanzen eindeutige IDs definiert werden. Die Eigenschaft *entfernen, falls nicht referenziert* ist eine Option, die dazu dient, die Markierung zu entfernen, falls sie nicht referenziert wird. In der Eigenschaft *Text aus* wird der Text definiert, der als Text des Inhaltsverzeichniseintrags im Inhaltsverzeichnis verwendet werden soll. Der Text kann aus dem Inhalt der

Inhaltsverzeichnis-Markierung (dem Inhalt zwischen dem Start- und dem End-Tag der Markierung im Design) oder aus einem XPath-Ausdruck stammen. In unserem Beispiel wird ein XPath-Ausdruck verwendet, der den Text der Überschrift bzw. den der jeweiligen `topic` Elemente der ersten Ebene zurückgibt.

Inhaltsverzeichnisvorlage

In die Inhaltsverzeichnis-Vorlage (*Abbildung unten*) wurde eine einzige Ebenenreferenz (levelref) eingefügt. Dieses Ebenenreferenz entspricht der Inhaltsverzeichnis-Ebene, die dem Kapitel-`topic` Element der ersten Ebene im Design zugewiesen wurde (*siehe weiter oben 'SPS-Struktur und -Ebenen*).



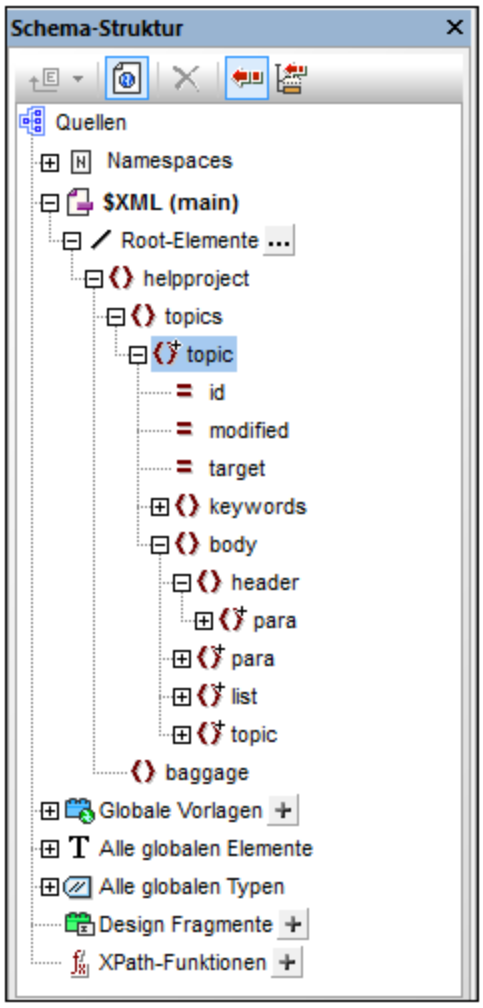
In diese Ebenenreferenz wurde eine Inhaltsverzeichnis-Referenz (TOCref) 'MyTOC' eingefügt. Diese TOCref dient zum Auswählen von Markierungen (i) die sich in der Markierungsgruppe `MyTOC` (*siehe 'Erstellen der Inhaltsverzeichnis-Markierung' weiter oben*) befinden und (ii) sich ausschließlich im Geltungsbereich der aktuellen Ebene befinden. Diese Einstellungen können entweder in der Seitenleiste "Eigenschaften" vorgenommen werden, wenn die TOCref ausgewählt ist, oder durch Rechtsklick auf die TOCref im Design und Auswahl des entsprechenden Bearbeitungsbefehls aus dem Kontextmenü..

Das Aussehen des Inhaltsverzeichniseintrags wird innerhalb der TOCref Tags des Inhaltsverzeichnisses definert. Das Nummerierungsformat, der Text, die Überschrift und die Seitenreferenz können durch Rechtsklick innerhalb der TOCref-Tags und Auswahl der einzufügenden Komponente aus dem Kontextmenü eingefügt werden. Jede dieser Komponenten kann durch Auswahl im Design und Ändern ihrer Eigenschaften in der Seitenleiste "Eigenschaften" bearbeitet werden.

7.6.4 Beispiel: Hierarchische und sequenzielle Inhaltsverzeichnisse

Ein Beispiel SPS zur Verdeutlichung der Verwendung von Inhaltsverzeichnissen finden Sie unter dem Namen `Chapters.sps` im Ordner (Eigene) Dateien[25], `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2025\StyleVisionExamples\Tutorial\TOC`. Dieses SPS basiert auf einem Schema, das das Content Model eines großen in Kapitel gegliederten Dokuments definiert. Die Schemastruktur wird in der Abbildung unten dargestellt und kann im Fenster "Schema-Struktur" von StyleVision angezeigt werden, wenn Sie `Chapters.sps` öffnen.

© 2019-2025 Altova GmbH



Das Dokumentelement ist `helpproject`. Es enthält ein untergeordnetes `topics` Element. Das `topics` Element kann eine unbegrenzte Anzahl an `topic` Elementen enthalten, welche ihrerseits wiederum untergeordnete `topic` Elemente enthalten können. Die erste Ebene der `topic` Elemente bildet sozusagen die Kapitel des Dokuments, die untergeordneten `topic` Elemente stehen für Abschnitte, Unterabschnitte usw.

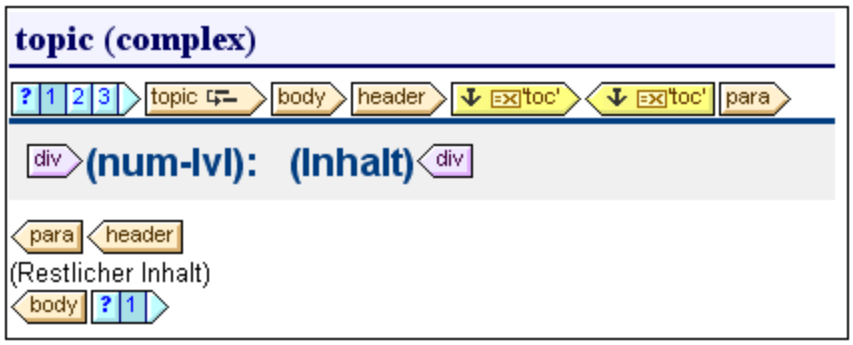
Das SPS enthält drei Inhaltsverzeichnisse, die sich auf der obersten Ebene des Dokuments befinden. Sie haben die folgende Reihenfolge:

1. *Kapitelübersicht*[312] *(Chapters at a Glance):* Listet die Namen der einzelnen Kapitel auf (die Kapitel auf der obersten Ebene).
2. *Kapitel und ihre Abschnitte*[313] *(Chapters and Their Sections):* Listet die einzelnen Kapitel mit ihren untergeordneten Abschnitten auf (Kapitel auf oberster Ebene plus die Hierarchie der einzelnen Unterkapitel des Kapitels bis hinunter zur untersten Ebene, welche im dazugehörigen XML-Dokument, `chapters.xml`, das Kapitel auf der dritten Ebene ist)
3. *Liste der Bilder*[314] *(List of Images):* Eine flache Liste aller Bilder im Dokument (mit Ausnahme des ersten), aufgelistet nach Dateiname.

SPS-Struktur

Bevor wir uns den Inhaltsverzeichnissen im Detail widmen, werfen Sie einen Blick auf die Struktur des Designs. Beachten Sie, dass die Hauptvorlage (mit den grünen `$XML` Tags) die Inhaltsverzeichnisse enthält. Der Rest der Hauptvorlage definiert über die Anweisung `Restlicher Inhalt`, dass globale und Standardvorlagen angewendet werden sollen.

Die Inhaltsverzeichnisdefinitionen befinden sich in den globalen Vorlagen für `topic` und `image`. In der globalen Vorlage für `topic` (*Abbildung unten*) wurde im Element `topic` eine Inhaltsverzeichnis-Ebene erstellt und innerhalb des Child-Elements `header` (aber außerhalb des Elements `para`) wurde eine Inhaltsverzeichnis-Markierung erstellt.



Da das Element `topic` rekursiv ist, sind die Inhaltsverzeichnis-Ebenen und -Markierungen ebenfalls rekursiv, d.h. es werden bei der ersten Rekursion eine neue hierarchisch untergeordnete Inhaltsverzeichnis-Ebene und eine neue Inhaltsverzeichnis-Markierung erstellt. Dies wird für jedes untergeordnete topic-Element fortgesetzt, wodurch eine hierarchische Struktur von verschachtelten Inhaltsverzeichnis-Ebenen entsteht, von denen jede eine entsprechende Inhaltsverzeichnis-Markierung enthält. Da die Formatierung der Überschrift (der Titel des Kapitels) für die einzelnen Inhaltsverzeichnis-Ebenen unterschiedlich sein soll, haben wir die einzelnen Ebenen in eine separate Verzweigung einer Bedingung mit drei Verzweigungen eingeschlossen. Jede Verzweigung überprüft, auf welcher Inhaltsverzeichnis-Ebene ein Kapitel vorkommt: der ersten, der zweiten oder der dritten.

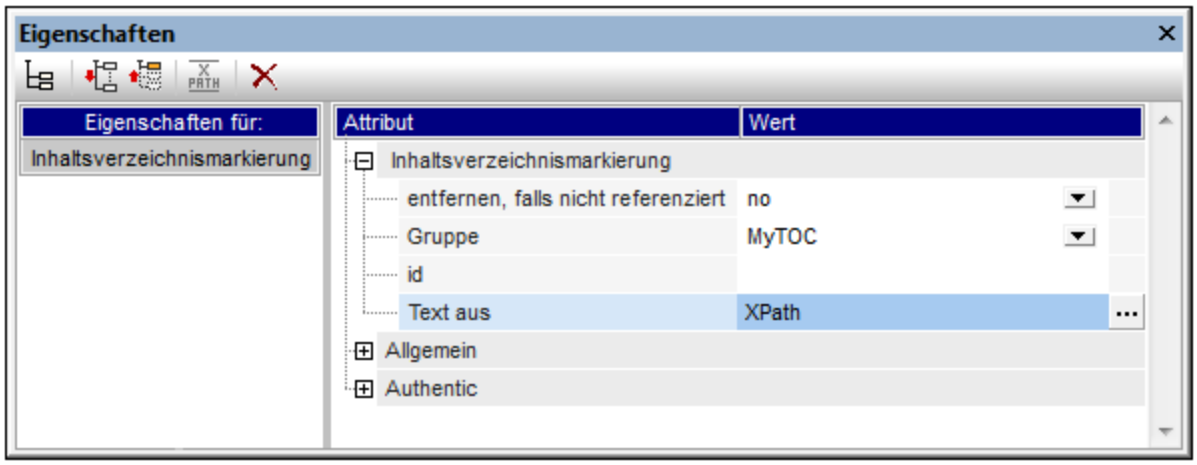
Beachten Sie, dass innerhalb der Ebenen eine hierarchische Nummerierung `(num-lvl)` eingefügt wurde. Dies erfolgt durch Rechtsklick an die gewünschte Stelle und Auswahl des Befehls **Inhaltsverzeichnis einfügen | Hierarchische Nummerierung**. Dadurch wird vor dem Titel jedes einzelnen Kapitels im **Textfluss des Dokuments** die korrekte hierarchische Nummerierung eingefügt, z.B. `3.1` oder `4.2.3`.

Inhaltsverzeichnisbeschreibungen

Im Folgenden finden Sie eine kurze Beschreibung der einzelnen Inhaltsverzeichnisse und die dabei zu beachtenden Punkte.

Kapitelübersicht: Wählen Sie die Inhaltsverzeichnis-Markierung in der globalen Vorlage für `topic` aus. Beachten Sie in der Seitenleiste "Eigenschaften" (*Abbildung unten*), dass der Eingabetext laut Definition mit Hilfe eines XPath-Ausdrucks erstellt wird. Wen Sie im Wertefeld der Eigenschaft *Text aus* auf die Schaltfläche "Bearbeiten" klicken, sehen Sie, dass der XPath-Ausdruck als `para` definiert wurde. Dies bedeutet, dass der Inhalt des Child-Elements `para` des Elements `header` als Text für den Inhaltsverzeichniseintrag verwendet wird (da die Inhaltsverzeichnis-Markierung innerhalb des Elements `header` eingefügt wurde).

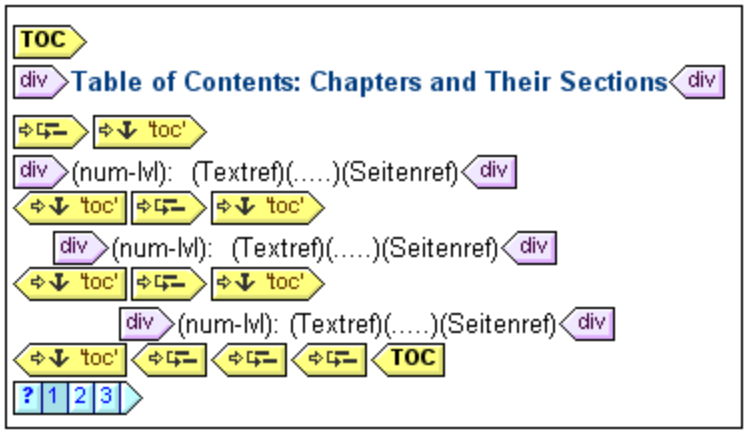
© 2019-2025 Altova GmbH



Die Inhaltsverzeichnisvorlage selbst (*Abbildung unten*) enthält eine Ebenenreferenz und die Inhaltsverzeichnisreferenz innerhalb dieser Ebenenreferenz 'MyTOC' wurde so konfiguriert, dass sie Inhaltsverzeichnis-Markierungen mit dem Namen `MyTOC` nur innerhalb des Geltungsbereichs der aktuellen Ebene - also der ersten Ebene - auswählt. Folglich werden Inhaltsverzeichniseinträge nur für Kapitel der ersten Ebene erstellt.

Beachten Sie außerdem, dass die Nummerierung als hierarchische Nummerierung definiert wurde.

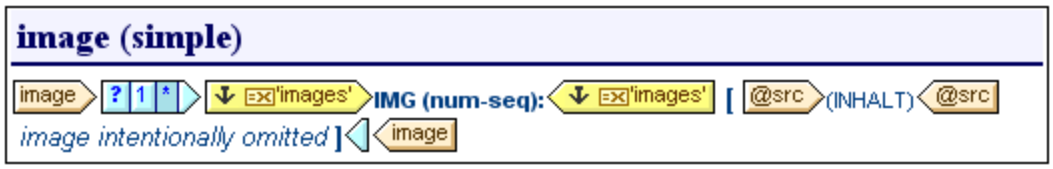
Kapitel und ihre Abschnitte: In diesem Inhaltsverzeichnis (*Abbildung unten*) wurden drei verschachtelte Ebenenreferenzen definiert, von denen jede eine Inhaltsverzeichnisreferenz enthält, für die der Geltungsbereich die aktuelle Ebene ist.



Da jeder Inhaltsverzeichniseintrag sich in einem `div` Block befindet, können Formatierungseigenschaften (einschließlich Einrückungen) für den Block gesetzt werden.

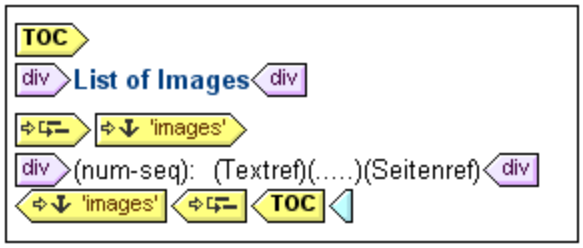
Liste der Bilder: Bei der Liste der Bilder handelt es sich um eine flache Liste. Überlegen Sie zuerst, innerhalb welcher Ebenen Bilder im instantiierten Dokument vorkommen werden. Das `image` Element ist ein Subelement des `para` Elements. Da in `topic` Elementen Ebenen erstellt wurden, werden `image` Elemente innerhalb der ersten, zweiten und/oder dritten Ebene des Dokuments vorkommen. Daher müssen keine Ebenen für das `image` Element erstellt werden.

Aufgrund der Bedingung in der globalen Vorlage für `image` (*siehe Abbildung unten*) können (i) das erste Bild (welches Sie in diesem Beispiel sehen) und (ii) die anderen Bilder (die in diesem Beispiel nicht dargestellt werden) separat verarbeitet werden.



Beachten Sie, dass die Inhaltsverzeichnis-Markierung nur innerhalb der zweiten Verzweigung der Bedingung platziert wird; daher werden Bilder, die in der ersten Verzweigung ausgewählt werden, nicht mit einer Markierung versehen. Auch die sequenzielle Nummerierung der Bilder (`num-seq`), die mit dem Befehl **Inhaltsverzeichnis einfügen | Sequenzielle Nummerierung** eingefügt wird, beginnt beim zweiten Bild (da das erste Bild in der ersten Verzweigung der Bedingung ausgewählt wird). Beachten Sie außerdem, dass die Nummerierung - wie hier in diesem Beispiel - formatiert werden kann. Um die Formatierung zu sehen, klicken Sie mit der rechten Maustaste auf `(num-seq)` und wählen Sie den Befehl **Format bearbeiten**. In dem Dialogfeld, das daraufhin angezeigt wird, sehen Sie, dass die Formatierung auf `01`, gesetzt wurde, d.h. es wird eine `0` vor einstelligen Zahlen eingefügt.

In der Inhaltsverzeichnisvorlage für Bilder (*Abbildung unten*) gibt es eine einzige Inhaltsverzeichnisreferenz, die Markierungen mit dem Namen `images`, identifiziert. Diese Inhaltsverzeichnisreferenz befindet sich innerhalb einer einzigen Ebenenreferenz. Der Geltungsbereich der Inhaltsverzeichnisreferenz (kann im Fenster "Eigenschaften" bearbeitet werden, wenn die Inhaltsverzeichnisreferenz ausgewählt ist) wurde auf `aktuelle Ebene und unterhalb` gesetzt. Die aktuelle Ebene, die durch die Ebenenreferenz festgelegt wird, ist die erste Ebene. Die Ebenen unterhalb davon sind also die zweite, dritte usw. Auf diese Art werden alle Bilder von der ersten Ebene an abwärts als Einträge im Inhaltsverzeichnis ausgewählt.



Da als Nummerierung eine sequenzielle Nummerierung ausgewählt wurde, werden die Bilder in einer flachen Liste sequenziell nummeriert.

7.6.5 Automatische Nummerierung im Hauptteil der Dokuments

Sich wiederholende Node-Instanzen können im Hauptteil des Dokuments mit Hilfe der Funktion "Automatische Nummerierung" automatisch nummeriert werden. So kann z.B. in einem Element `Book`, das mehrere `Chapter`-

© 2019-2025 Altova GmbH

Elementen enthält, jedes `Chapter`-Element mit Hilfe dieser Funktion automatisch nummeriert werden. Dies ist eine einfache Methode, um auf Basis der Struktur des XML-Dokuments eine Nummerierung einzufügen.

Anmerkung: Die Funktion zur automatischen Nummerierung bezieht sich auf die Nummerierung im Hauptteil des Dokuments, **nicht aber** auf die Nummerierung innerhalb von Inhaltsverzeichnissen, wo die Nummerierung als Eigenschaft des Inhaltsverzeichniseintrags betrachtet wird.

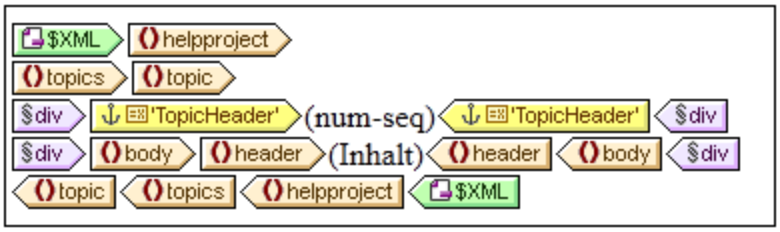
Die automatische Nummerierung kann entweder sequenziell (flach) oder hierarchisch erfolgen. Bei der sequenziellen Nummerierung wird eine normale Nummerierung auf einer Ebene vorgenommen. Die hierarchische Nummerierung basiert auf der im Dokument erstellten Inhaltsverzeichnis-Ebenenhierarchie und nummeriert die Einträge nach ihrer Position in der Hierarchie der Inhaltsverzeichnis-Ebenen.

Für die Nummerierung stehen die verschiedensten Formatierungsoptionen zur Verfügung. Bei der hierarchischen Nummerierung können einzelne Zahlensymbole separat formatiert werden. So kann z.B. eine aus drei Zahlensymbolen bestehende Nummerierung das Format: `A.1.i.`erhalten, wobei jedes der drei Symbole ein anderes Zahlenformat haben kann. Die Formatierung für die Nummerierung wird der sequenziellen und hierarchischen Nummerierung auf unterschiedliche Art zugewiesen. Eine Beschreibung dazu finden Sie in den nachstehenden Abschnitten.

Sequenzielle Nummerierung (num-seq)

Eine sequenzielle (oder flache) Nummerierung kann in eine Inhaltsverzeichnismarkierung[298] im Dokument-Design eingefügt werden (*siehe Abbildung unten*). Gehen Sie dazu folgendermaßen vor:

1. Platzieren Sie den Cursor in den zu nummerierenden Node und erstellen Sie eine Inhaltsverzeichnismarkierung (Klicken Sie auf die rechte Maustaste und wählen Sie den Befehl **Inhaltsverzeichnis einfügen | Inhaltsverz.-Markierung**). Die Inhaltsverzeichnismarkierung wird daraufhin erstellt. In der Abbildung unten soll das Element `topic` nummeriert werden, daher wurde die Inhaltsverzeichnis-Markierung innerhalb des Elements `topic` erstellt. Wo genau die Markierung im Element `topic` platziert wird, hängt davon ab, wo im Layout die Nummerierung angezeigt werden soll. (In der Abbildung unten wurde die Nummerierung direkt links von der Kapitelüberschrift platziert.
2. Platzieren Sie den Cursor innerhalb der Tags der Inhaltsverzeichnismarkierung, klicken Sie auf die rechte Maustaste und wählen Sie **Inhaltsverzeichnis einfügen | Sequenzielle Nummerierung**. Daraufhin wird der Platzhalter für die automatische sequenzielle Nummerierung `(num-seq)` eingefügt (*in der Abbildung unten in der Inhaltsverz.-Markierung 'TopicHeader' markiert*).



3. Wenn die Inhaltsverzeichnis-Markierung von einer Inhaltsverzeichnis-Vorlage aus referenziert werden soll, können Sie die Eigenschaften der Inhaltsverzeichnis-Markierung wie gewünscht eingeben. Wenn die Inhaltsverzeichnismarkierung nur für die sequenzielle Nummerierung verwendet wird, wird kein Name dafür benötigt. Wenn Sie ihr einen Namen geben möchten, klicken Sie mit der rechten Maustaste auf die Inhaltsverzeichnismarkierung und wählen Sie den Befehl **Gruppe bearbeiten**).

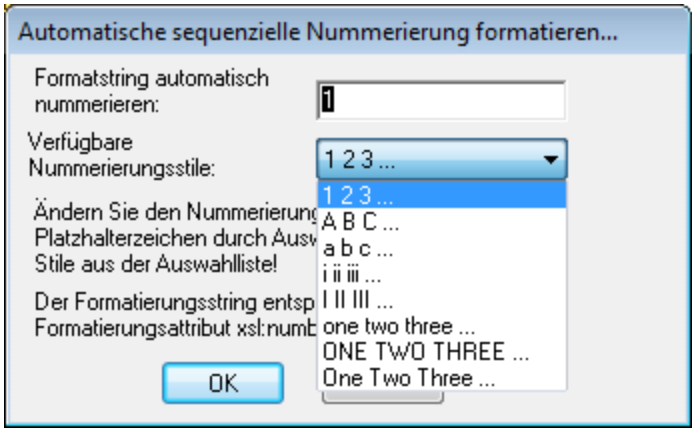
In dem Beispiel in der Abbildung oben wurde eine sequenzielle Nummerierung für den Node `topic` definiert. Das Ergebnis ist, dass jedes `topic` Element mit einer sequenziellen Nummer versehen wird, wie in der

Abbildung unten gezeigt. Beachten Sie, dass die Nummerierung im Grunde die Position des jeweiligen `topic` Elements innerhalb der Sequenz aller gleichrangigen `topic` Elements auf dieser Gliederungsebene des XML-Dokuments wiedergibt.

1: Altova StyleVision 2007
2: About this Documentation
3: Introduction
4: User Interface

Anmerkung: Wenn die sequenzielle Nummerierung an einer anderen Gruppe von Nodes fortgesetzt werden soll, verwenden Sie für beide Node-Gruppe eine Inhaltsverzeichnismarkierung mit dem selben Namen.

Um die sequenzielle Nummerierung zu formatieren, klicken Sie mit der rechten Maustaste auf den Platzhalter `num-seq` und wählen Sie den Befehl **Format bearbeiten**. Daraufhin wird das Dialogfeld "Automatische sequenzielle Nummerierung formatieren" angezeigt (*Abbildung unten*).



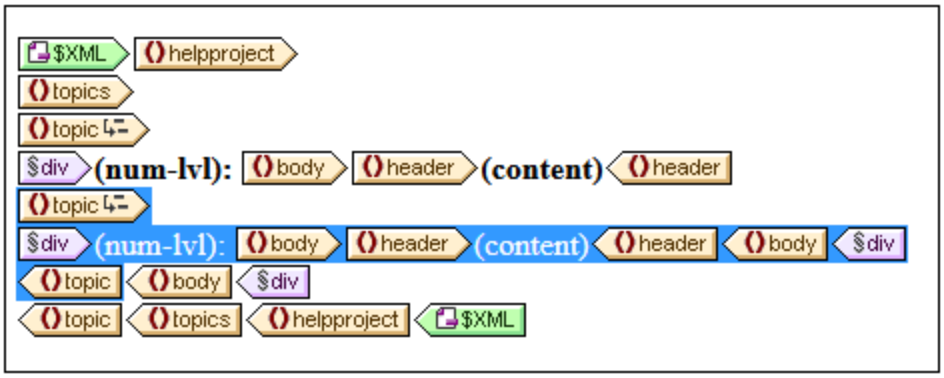
Wählen Sie das Format aus der Dropdown-Liste der Auswahlliste *Verfügbare Nummerierungsstile* aus (*siehe Abbildung oben*) und klicken Sie auf OK, um das ausgewählte Format anzuwenden.

Hierarchische Nummerierung (num-lvl)

Eine hierarchische Nummerierung kann innerhalb einer Inhaltsverzeichnis-Ebene im Design[295] eingefügt werden. Um eine hierarchische Nummerierung in einem Dokument zu erstellen, müssen Sie das Dokument zuerst in Inhaltsverzeichnis-Ebenen strukturieren (siehe Abschnitt Strukturieren des Designs in Ebenen)[295]. Beachten Sie dabei die folgenden Punkte:

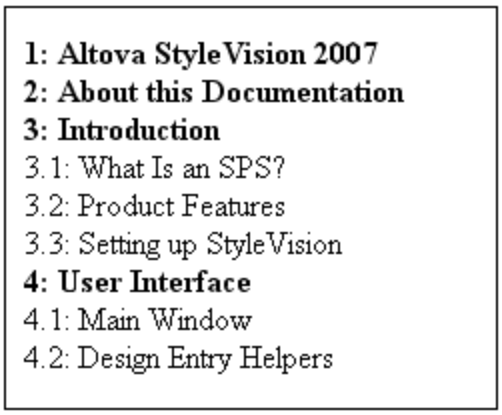
- Ebenen müssen entweder auf dem zu nummerierenden Node oder innerhalb dieses Nodes erstellt werden.
- Die Ebenen müssen entsprechend der Hierarchie der erforderlichen Nummerierung verschachtelt werden (*siehe Abbildung unten*).
- Der Platzhalter für die hierarchische Nummerierung muss innerhalb der entsprechenden Ebene im Design eingefügt werden (*siehe Abbildung unten*).

© 2019-2025 Altova GmbH



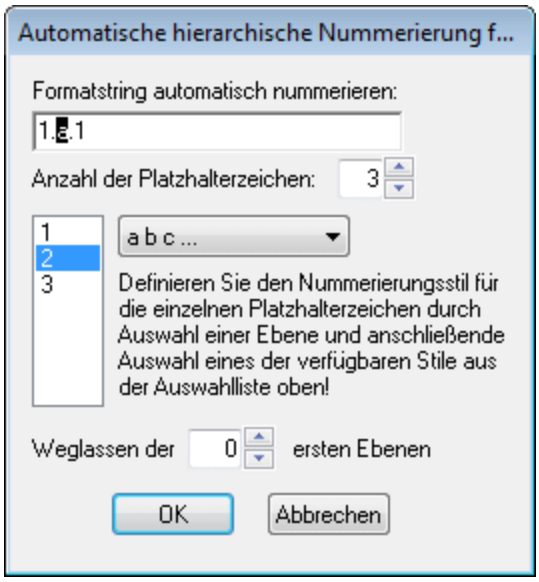
In der Abbildung oben gibt es zwei Ebenen. Das Element `topic` ist rekursiv und wurde auf zwei `topic` Elementen erstellt (durch Rechtsklick auf den Node Tag und Auswahl des Befehls **Vorlage dient als Ebene**). Ein `topic` Element (*in der Abbildung oben markiert*) ist in das andere verschachtelt. Infolgedessen sind auch die Ebenen verschachtelt. Innerhalb der einzelnen Ebenen wurde ein hierarchischer Platzhalter `(num-lvl)` eingefügt (durch Rechtsklick auf die Ebene und Auswahl des Befehls **Inhaltsverzeichnis einfügen | Hierarchische Nummerierung**).

Des Ergebnis des Designs in der Abbildung oben sieht folgendermaßen aus.



Die erste Ebene wird fett gedruckt, die zweite normal.

Zur Formatierung der automatischen Nummerierung klicken Sie mit der rechten Maustaste auf den Platzhalter `num-lvl` und wählen Sie den Befehl **Format bearbeiten**. Daraufhin wird das Dialogfeld "Automatische hierarchische Nummerierung formatieren" angezeigt (*Abbildungen unten*),



Wählen Sie zuerst in der Auswahlliste "Anzahl der Platzhalterzeichen" die Anzahl der Platzhalterzeichen aus. Diese Anzahl sollte der Anzahl der Inhaltsverzeichnisebenen im Dokument entsprechen. Jedes Platzhalterzeichen kann anschließend separat formatiert werden. Wählen Sie das zu formatierende Zeichen im unteren der beiden Anzeigefelder aus. (In der Abbildung oben wurde das zweite Platzhalterzeichen ausgewählt). Wählen Sie als nächstes in der Formatierungsauswahlliste den gewünschten Formatierungsstil aus. In der Abbildung oben wurde für das zweite Platzhalterzeichen eine Formatierung in Kleinbuchstaben gewählt. Dies wird auch im Anzeigefeld im oberen Bereich des Dialogfelds angezeigt. Außerdem können durch Eingabe der gewünschten Anzahl an Ebenen im Feld "Weglassen der....ersten Ebenen" Ebenen ausgelassen werden.

Beachten Sie, dass die Formatierung bei der hierarchischen Nummerierung immer nur für eine Ebene gleichzeitig definiert werden kann. Daher muss der Platzhalter `num-lvl` für die hierarchische Nummerierung für jede Ebene separat formatiert werden.

Klicken Sie auf **OK,** wenn Sie damit fertig sind.

7.6.6 Querverweise

Ein Querverweis ist ein Verweis auf einen anderen Teil des Dokuments. In einem SPS wird ein Querverweis in zwei Schritten erstellt: Zuerst wird das Ziel des Querverweises definiert. Im zweiten Schritt wird die Verknüpfung zum Ziel definiert.Bei der Definition des Ziels des Querverweises wird eine Inhaltsverzeichnis-Markierung innerhalb einer Inhaltsverzeichnis-Ebene definiert. Die Verknüpfung zum Ziel ist eine Textreferenz innerhalb einer Inhaltsverzeichnis-Referenz. Die Textreferenz generiert den Ausgabetext und dient als Link. Die Erstellung eines Querverweises erfolgt daher in den folgenden drei Schritten.

Schritt 1: Ebenen

Das Dokument ist, wie im Abschnitt Strukturieren des Designs in Ebenen 295 beschrieben, in Inhaltsverzeichnis-Ebenen gegliedert. Die Inhaltsverzeichnis-Ebenen dienen bei der Referenzierung dazu, den Geltungsbereich der Referenzierung zu definieren. Nur Inhaltsverzeichnis-Markierungen mit dem definierten

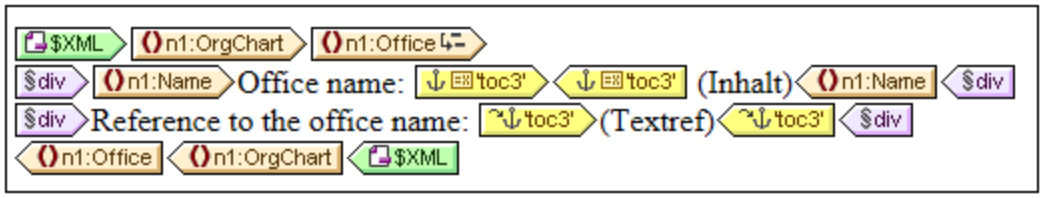
 © 2019-2025 Altova GmbH

Namen, die sich innerhalb des definierten Geltungsbereichs befinden, werden verwendet. In der Abbildung unten wurde auf dem Element `n1:Office` eine Ebene erstellt.

Schritt 2: Erstellen von Inhaltsverzeichnismarkierungen

Innerhalb einer Ebene wird eine Inhaltsverzeichnismarkierung durch Setzen des Cursors an die gewünschte Stelle, Rechtsklick und Auswahl des Befehls **Inhaltsverzeichnis einfügen | Inhaltsverz.-Markierung** erstellt. Die Inhaltsverzeichnis-Markierung erhält einen Namen und einen XPath-Ausdruck, der den Ausgabetext generiert. Normalerweise definiert der XPath-Ausdruck einen Node im Dokument, dessen Inhalt der gewünschte Text ist.

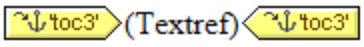
In der Abbildung unten hat die Inhaltsverzeichnismarkierung im Element `n1:Name` 'toc3' 'toc3' den Namen `toc3` und einen XPath-Ausdruck, mit dem der aktuelle Node gesucht wird. D.h. dass es sich beim Ausgabetext um den Inhalt des Node `n1:Name` handelt.



Bei der Verarbeitung des XML-Dokuments wird für jedes `n1:Name` -Element ein Anker erstellt. Dieser Anker hat eine Textreferenz (den Text des Querverweises), die der Wert des Elements `n1:Name` ist.

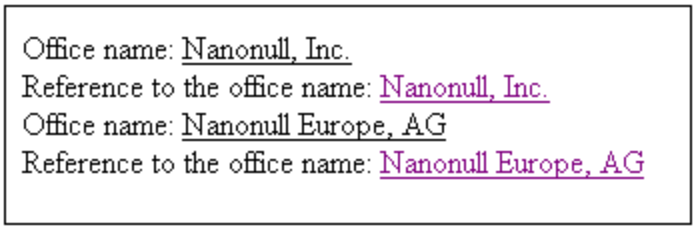
Schritt 3: Erstellen von Inhaltsverzeichnisreferenzen

Eine Inhaltsverzeichnisreferenz wird eingefügt (Kontextmenü, **Inhaltsverzeichnis einfügen | Inhaltsverz.-Referenz**), um eine Verknüpfung zu den Ankern herzustellen, die mit einer Inhaltsverzeichnismarkierung generiert wurden.



In der Abbildung oben befindet sich die Inhaltsverzeichnis-Referenz namens `toc3` (*Abbildung oben)* innerhalb derselben Inhaltsverzeichnis-Ebene (in der Ebene `Office`) wie die Inhaltsverzeichnismarkierung, die sie referenziert. Außerdem müssen Sie auch den Geltungsbereich der Inhaltsverzeichnisreferenz definieren. Mit dem Geltungsbereich wird definiert, welche Inhaltsverzeichnis-Ebenen nach Inhaltsverzeichnis-Markierungen desselben Namens wie die Inhaltsverzeichnis-Referenz durchsucht werden müssen. Im Beispiel oben ist der Geltungsbereich die aktuelle Ebene. Das bedeutet, dass diese Referenz Inhaltsverzeichnismarkierungen innerhalb der aktuellen Ebene, die den Namen `toc3` haben, referenziert.

In der Abbildung oben sehen Sie eine `n1:Office` Vorlage. Wenn ein `n1:Office` Node verarbeitet wird, wird ein Anker mit Ausgabetext erstellt, der der Inhalt des `n1:Name` Node ist. Der Grund dafür ist, dass mit der Inhaltsverzeichnis-Markierung (über die Eigenschaft der Inhaltsverzeichnis-Markierung*Text aus*) ein XPath-Ausdruck definiert wird, demzufolge der Inhalt dieses Node als Ausgabetext verwendet wird. Die Inhaltsverzeichnisreferenz in der nächsten Zeile identifiziert den Anker mit dem Namen `toc3` und die Textreferenzkomponente generiert den Ausgabetext des Links zum Anker (*in der Abbildung unten der violette Text*). Die Ausgabe würde etwa folgendermaßen aussehen:



Im obigen Beispiel wurde als Geltungsbereich die aktuelle Ebene definiert. Es gibt zwei weitere Möglichkeiten für den Geltungsbereich: (i) ein globaler Geltungsbereich, (ii) ein Bereich für die aktuelle Ebene und die darunter liegenden. Mit diesen Optionen können auch Inhaltsverzeichnismarkierungen in anderen Ebenen des Designs referenziert werden.

7.6.7 Lesezeichen und Hyperlinks

Sie können im SPS-Dokument überall im Design Lesezeichen einfügen. Diese Lesezeichen werden in der Ausgabe in Anker transformiert, zu denen über Hyperlinks eine Verknüpfung erstellt werden kann. Mit Hyperlinks können nicht nur zu Lesezeichen, sondern auch zu externen Ressourcen wie Webseiten Verknüpfungen erstellt werden. StyleVision bietet viel Flexibilität beim Erstellen von Ziel-URIs für Hyperlinks.

In diesem Abschnitt wird beschrieben:

- wie Lesezeichen 320 in das SPS eingefügt werden können.
- wie Hyperlinks 322 in das SPS eingefügt werden können und wie diese eine Verknüpfung zu den gewünschten Seiten bilden.

Anmerkung: Zwar werden in der FO-Spezifikation Links zu externen Dokumenten unterstützt, doch ist das nicht bei jedem FO-Prozessor der Fall. Überprüfen Sie zuerst, ob Ihr FO-Prozessor diese Funktion unterstützt, wenn Sie vorhaben, diese Funktion zu verwenden.

7.6.7.1 Einfügen von Lesezeichen

Ein Lesezeichen (oder Anker) kann überall im SPS eingefügt werden - an einem Cursor-Einfügepunkt oder rund um eine SPS-Komponente.

Lesezeichen im SPS werden über das Dialogfeld "Lesezeichen einfügen" (*Abbildung unten*) angelegt. In diesem Dialogfeld wird der Name des Lesezeichens definiert. Dabei kann es sich um einen statischen Namen oder einen dynamischen Namen, der (i) aus dem Inhalt des XML-Dokuments stammt oder (ii) mit Hilfe eines XPath-Ausdrucks erstellt wurde, handeln.

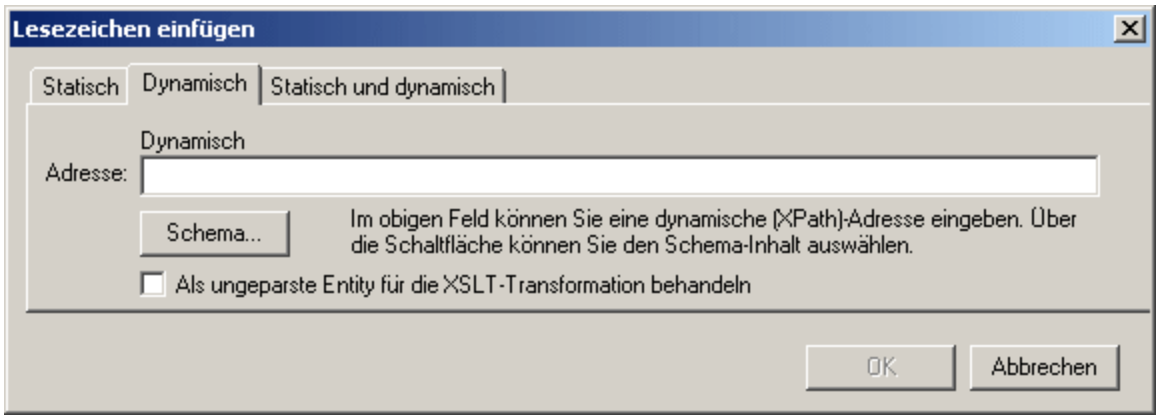
Erstellen eines Lesezeichens

Um ein Lesezeichen einzufügen, gehen Sie folgendermaßen vor:

1. Platzieren Sie den Cursor an die Stelle, an der Sie das Lesezeichen erstellen möchten.
2. Wählen Sie den Menübefehl **Einfügen | Lesezeichen** 515 oder klicken Sie auf die rechte Maustaste und wählen Sie **Einfügen | Lesezeichen**.

 © 2019-2025 Altova GmbH

3. Wählen Sie im Dialogfeld "Lesezeichen einfügen" (*Abbildung unten*) ein Register aus, je nachdem ob Sie einen statischen (Register "Statisch"), einen anhand des XML-Dokuments erstellten dynamischen (Register "Dynamisch") oder einen nach Belieben durch einen XPath-Ausdruck generierten Namen oder einen aus statischen und dynamischen Komponenten zusammengesetzten Namen ("Statisch und dynamisch") einfügen möchten. In der Abbildung unten wird ein dynamisches Lesezeichen erstellt, das als Namen eine eindeutige ID für jedes `Name` Child-Element des Kontextnode hat.



4. Klicken Sie auf **OK**. Das Lesezeichen wird definiert.

Nach Erstellung des Lesezeichens kann man über einen Hyperlink eine Verknüpfung dazu erstellen[322]

Hinweis:

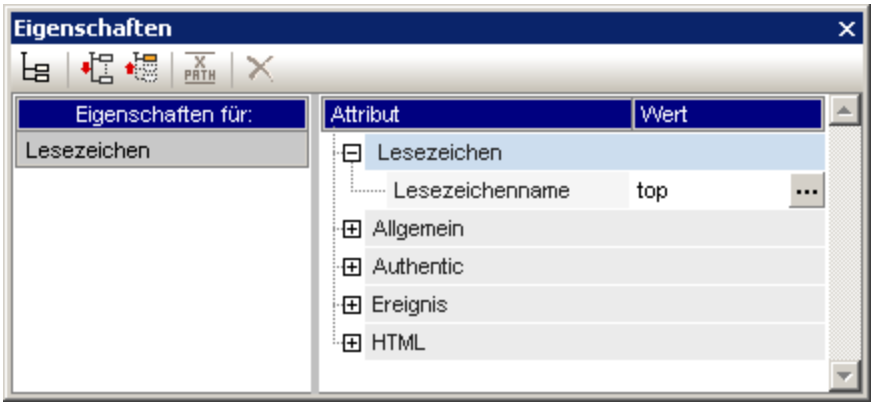
Ein Lesezeichen wird für eine einzelne fixe Stelle im Design angelegt. Wenn sich das Element, in dem sich das Lesenzeichen befindet, wiederholt, so wird in jeder Instanz des sich wiederholenden Elements ein Lesezeichen erstellt. Wenn Sie dem Lesenzeichen einen statischen Namen geben, so erhält jedes dieser Lesezeichen denselben Namen. Daher empfiehlt es sich, dem Lesenzeichen in solchen Fällen (bei sich wiederholenden Elementen) einen dynamischen Namen zu geben. Dabei kann es sich z.B. um den Namen eines Child-Elements des Kontextnode (also des Elements in dem das Lesezeichen erstellt wird) handeln. Wenn der für den dynamischen Namen ausgewählte Node eventuell in mehreren Instanzen denselben Namen haben könnte, können Sie mit Hilfe der `generate-id( )`-Funktion zur Generierung des Namens (*siehe Abbildung oben*) sicherstellen, dass das Lesezeichen einen eindeutigen Namen hat. Zur Referenzierung eines solchen Lesezeichens kann dieselbe ID auch als `href`-Wert eines Hyperlink[325] generiert werden. Verwenden Sie in diesem Fall unbedingt den Fragment-Identifier `#` vor der `generate-id()` Funktion. Der XPath-Ausdruck würde lauten: `concat('#', generate-id(nodeXXX))`.

Ändern eines Lesezeichens

Nachdem Sie das Lesezeichen erstellt haben, können Sie seinen Namen über das Dialogfeld "Lesezeichen bearbeiten" ändern. Dieses Dialogfeld wird folgendermaßen aufgerufen:

1. Wählen Sie das Lesezeichen im Design aus.
2. Klicken Sie in der Seitenleiste "Eigenschaften" in der Eigenschaftsgruppe *Lesezeichen* auf die **Bearbeiten**-Schaltfläche der Eigenschaft `Lesezeichenname` (*Abbildung unten*). Daraufhin wird

dasselbe Dialogfeld "Lesezeichen bearbeiten" (siehe *Abbildung oben*) angezeigt, das oben beschrieben wurde.

1. 

3. Bearbeiten Sie den Namen des Lesezeichens im Dialogfeld "Lesezeichen bearbeiten" auf einem der Register "Statisch", "Dynamisch" oder "Statisch und dynamisch".

Löschen eines Lesezeichens

Um ein Lesezeichen zu löschen, wählen Sie es im Design aus und drücken Sie die **Entf**-Taste.

7.6.7.2 Definieren von Hyperlinks

Hyperlinks können rund um SPS Komponenten wie z.B. Text oder Bilder erstellt werden. Hyperlinks können auf: (i) Lesezeichen im SPS Design oder (ii) externe Ressourcen wie z.B. Webseiten oder E-Mail-Nachrichten verweisen. Dieser Abschnitt enthält zuerst Informationen zum Inhalt von Hyperlinks (Text, Bilder, usw.) und anschließend zum Ziel von Hyperlinks.

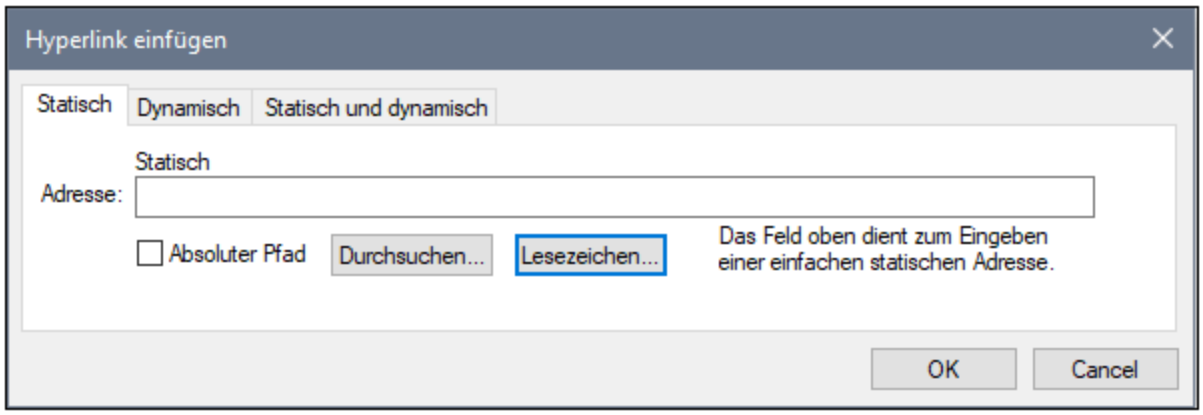
Erstellen von Hyperlinks

Ein Hyperlink kann auf folgende Arten erstellt werden:

- Rund um Text (statischen oder dynamischen), Nodes, Bilder, Bedingungsvorlagen, automatische Berechnungen und Content- oder Node-Blöcke. Er kann nicht rund um ein Dateneingabeelement wie z.B. ein Eingabefeld oder eine Auswahlliste erstellt werden. Allerdings kann der Hyperlink rund um einen Node oder eine Bedingungsvorlage erstellt werden, in dem bzw. in der sich das Dateneingabeelement befindet. Dieses ist der Inhalt des Links, über den Sie durch Anklicken zum Ziel des Links navigieren. Um einen Hyperlink rund um eine Komponente im SPS zu erstellen, wählen Sie die Komponente aus und wählen Sie den Menübefehl **Einschließen in | Hyperlink**.
- Ein neuer Hyperlink kann mit dem Menübefehl **Einfügen | Hyperlink** eingefügt werden. Der Inhalt des Links muss anschließend innerhalb der Tags des neu erstellen Hyperlinks eingefügt werden.

Definieren des Ziels des Hyperlinks

Das Ziel eines Hyperlinks wird im Dialogfeld "Hyperlink einfügen" (*Abbildung unten*), das über die Befehle **Einschließen in | Hyperlink** 528 oder **Einfügen | Hyperlink** 517 aufgerufen wird, definiert.



Ein Hyperlink kann

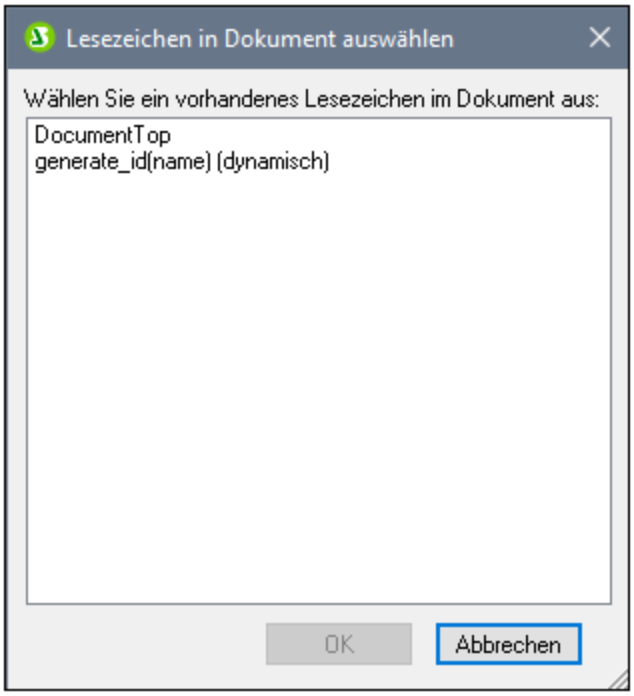
- auf ein Lesezeichen[322] im selben SPS Design verweisen (in diesem Fall muss die Ziel-URI ein Fragment Identifier sein),
- dynamisch generiert werden[325], um Lesezeichen-Markierungen zu finden (bei diesen URIs handelt es sich ebenfalls um Fragment Identifier),
- eine externe Ressource[325] sein; die URI kann statisch (direkt eingegeben), dynamisch (aus einem Node in einem XML-Dokument stammend), eine Kombination aus statischen und dynamischen Teilen oder der Wert einer ungeparsten Entity sein.

Weiter unten wird erklärt, wie diese beiden Ziele definiert werden. Nachdem Sie die URI im Dialogfeld "Hyperlink einfügen/bearbeiten" definiert haben, klicken Sie zum Fertigstellen auf **OK**.

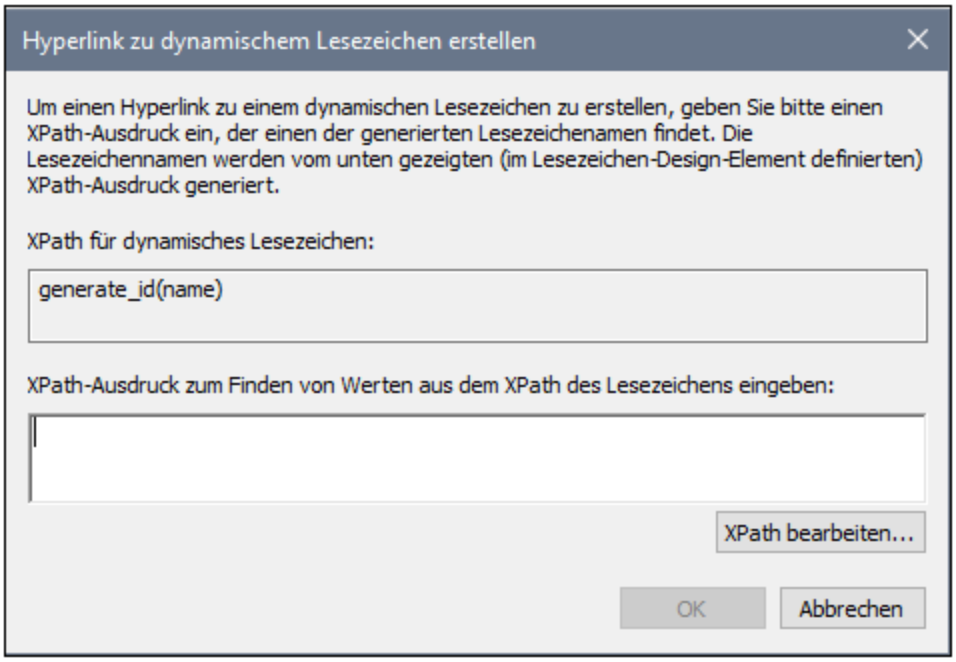
Anlegen eines Link zu einem Lesezeichen

Um einen Link zu einem Lesezeichen anzulegen, gehen Sie folgendermaßen vor:

1. Klicken Sie im Dialogfeld "Hyperlink einfügen" auf dem Register "Statisch" auf die Schaltfläche **Lesezeichen**. Daraufhin wird das Dialogfeld "Lesezeichen in Dokument auswählen" angezeigt (*Abbildung unten*). In der Abbildung unten sehen Sie zwei Lesezeichen, ein statisches und ein dynamisches.



2. Um als Ziel-URI ein statisches Lesezeichen auszuwählen, doppelklicken Sie auf das statische Lesezeichen und klicken Sie anschließend auf **OK.** Wenn Sie auf ein dynamisches Lesezeichen doppelklicken, werden Sie aufgefordert, einen XPath-Ausdruck einzugeben, der den Pfad des ausgewählten dynamischen Lesezeichens auswählt (*siehe Lesezeichen unten*).



Beim dynamischen Lesezeichen[320] handelt es sich eigentlich um einen XPath-Ausdruck, der den Namen des Lesezeichens generiert; er ist nicht selbst der Name des Lesezeichens. Im Dialogfeld "Hyperlink zu dynamischem Lesezeichen erstellen" wird der XPath-Ausdruck des dynamischen

 © 2019-2025 Altova GmbH

Lesezeichens angezeigt. Hier können Sie einen XPath-Ausdruck erstellen, der einen Namen generiert, der mit dem des gewünschten Lesezeichens übereinstimmt. Klicken Sie auf OK, wenn Sie fertig sind.

Anlegen eines Link zu dynamisch generierten ID-Lesezeichen

Lesezeichen können dynamisch generierte ID-Markierungen haben[320]. Wenn wir einen Link zu einem derartigen Lesezeichen erstellen möchten, stellt sich folgendes Problem: Da die Namen von dynamisch generierten Markierungen zur Laufzeit generiert werden und daher zum Zeitpunkt der Design-Erstellung nicht bekannt sind, wie soll man den `href` Wert eines Hyperlink[517] definieren, der auf eine solche Markierung verweist? Die Lösung des Problems ist, wieder die `generate-id()` Funktion zu verwenden, diesmal mit dem `href` Wert des Hyperlink[517]. Warum dies funktioniert, liegt an einer Eigenschaft der `generate-id()` Funktion. Jedes Mal, wenn die `generate-id()` Funktion während einer einzelnen Transformation für einen bestimmten Node ausgewertet wird, generiert sie immer dieselbe ID. Aus diesem Grund sind die im Lesezeichen und im Hyperlink generierten IDs identisch.

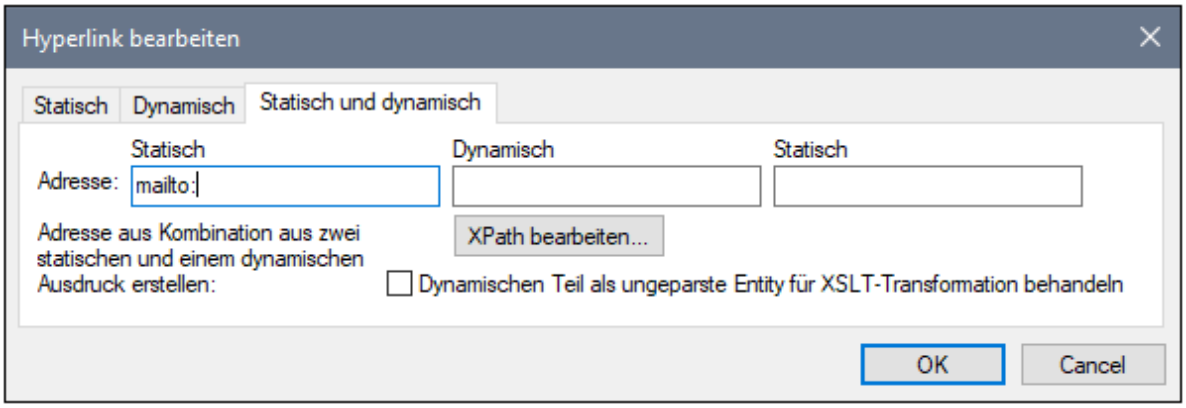
Beachten Sie dabei zwei Punkte:

- Da die `generate-id()` Funktion als XPath-Ausdruck ausgewertet werden muss, verwenden Sie zum Definieren des Ziels des Hyperlink das Register "Dynamisch" des Dialogfelds "Hyperlink" (*siehe Abbildung unten)*.
- Der ausgewertete Wert des `href` Attributs muss mit `#` (dem Fragment Identifiert) beginnen. Folglich lautet der XPath-Ausdruck: `concat('#', generate-id(nodeXXX))`. Alternativ dazu können Sie auf dem Register "Statisch und dynamisch" im statischen Teil der Adresse das Zeichen `#` und im dyamischen Teil `generate-id(nodeXXX)` eingeben.

Anlegen eines Link zu einer externen Ressource

URIs für externe Ressourcen können auf die folgende Art erstellt werden:

- Durch Eingabe der URI direkt in das Dialogfeld "Hyperlink einfügen" in das Register "Statisch". So kann z.B. ein Link zur Altova Homepage (`http://www.altova.com`) auf dem Register "Statisch" direkt in das Eingabefeld "Adresse" eingegeben werden.
- Durch Auswahl eines Node in der XML-Dokumentenquelle im Dialogfeld "Hyperlink einfügen" auf dem Register "Dynamisch". Der Node in der XML-Quelle kann einen Textstring bereitstellen, bei dem es sich entweder (i) um die gewünschte Ziel-URI handelt oder (ii) den Namen einer ungeparsten Entity[377], welche die erforderliche URI als Wert hat. Die Adresse der Altova Homepage kann z.B. als Textstring in einem Node enthalten sein.
- Durch Erstellen einer URI mit sowohl statischen als auch dynamischen Teilen im Dialogfeld "Hyperlink einfügen" auf dem Register "Statisch und dynamisch". Dies eignet sich um z.B. statische Präfixe (z.B. ein Protokoll) oder Suffixe (z.B. einen Domain-Namen) hinzuzufügen. So können z.B. E-Mail-Adressen unter Verwendung eines statischen Teils `mailto:` und eines dynamischen Teils für den String-Inhalt des Node `//Contact/@email` erstellt werden (*Abbildung unten*). (*In der Abbildung unten wird im Platzhalter "Inhalt" des Node* `//Contact/@email` *ein Link erstellt. Aus diesem Grund wurde der abgekürzte self::node() Selektor verwendet*). Über die Schaltfläche **XPath bearbeiten** gelangen Sie zum Dialogfeld "XPath-Ausdruck bearbeiten"[452], in dem Sie den dynamischen Teil des Hyperlink erstellen können..



Eine Beschreibung zur Verwendung ungeparster Entities finden Sie im Abschnitt URIs von ungeparsten Entities 377.

Bearbeiten von Hyperlink-Eigenschaften

Um einen Hyperlink zu bearbeiten, klicken Sie mit der rechten Maustaste auf den Start- oder End-Tag des Hyperlink (`A`) und wählen Sie im Kontextmenü den Befehl "**URL bearbeiten**". Daraufhin wird das Dialogfeld "Hyperlink bearbeiten" (*Abbildung oben*) angezeigt. Das Dialogfeld "Hyperlink bearbeiten" kann auch über die Eigenschaft `URL` der Eigenschaftsgruppe *Hyperlink* im Fenster "Eigenschaften" aufgerufen werden.

Entfernen und Löschen von Hyperlinks

Um einen Hyperlink zu entfernen, wählen Sie ihn durch Klicken auf den Start- oder End-Tag des Hyperlink (`A`) aus und drücken Sie die **Entf**-Taste. Der Hyperlink und sein Inhalt werden gelöscht.

© 2019-2025 Altova GmbH

8 Darstellung und Ausgabe

Im SPS Design ist für Komponenten eine einzige Gruppe von Stileigenschaften definiert. Diese Stile werden in den entsprechenden Ausgabedokumenten (*Authentic-Ansicht, HTML, RTF, PDF, Word 2007+ und Text in der Enterprise Edition; Authentic-Ansicht, HTML, RTF und Text in der Professional Edition; HTML in der Basic Edition*) in die entsprechenden Stil-Markup-Tags konvertiert.

Anmerkung: Die Ausgabe von Druckformaten wie z.B. RTF, PDF, Word 2007+ und Text wird (im Gegensatz zu HTML) in der Basic Edition von StyleVision nicht unterstützt. Folglich sind auch alle Funktionen im Zusammenhang damit (z.B. das Hinzufügen von Kopf- und Fußzeilen) in der Basic Edition deaktiviert. Um diese Funktionen nutzen zu können, müssen Sie eine Lizenz für die Professional Edition (inkludiert RTF- und Textausgabe) oder die Enterprise Edition (inkludiert RTF, PDF, Word 2007+ und Text) erwerben. Nähere Informationen finden Sie auf der Altova Website auf der Seite StyleVision Editionsvergleich.

Stile für SPS Komponenten

Die Definition von Stilen für SPS Komponenten erfolgt unter Verwendung von CSS2-Prinzipien und Syntax. Stile können in externen Stylesheets, global für das SPS und lokal für eine Komponente definiert werden. Die stufenförmige Reihenfolge von CSS2 wird auf das SPS angewendet und bietet beträchtliche Flexibilität bei der Stilerstellung. Die Arbeit mit CSS-Stilen wird in diesem Abschnitt im Unterabschnitt Arbeiten mit CSS-Stilen 342 näher beschrieben.

Die Werte von Stileigenschaften können direkt in die Seitenleiste "Stil" oder die Seitenleiste "Eigenschaften" eingegeben werden oder sie können über XPath-Ausdrücke 353 festgelegt werden. Die Vorteile der Verwendung von XPath-Ausdrücken sind: (i) dass der Eigenschaftswert von einer XML-Datei genommen werden kann und (ii) dass ein Eigenschaftswert auf Basis einer Bedingung eingefügt werden kann, wobei mit Hilfe eines XPath-Ausdrucks überprüft wird, ob die Bedingung zutrifft.

Zusätzlich dazu stehen im SPS Design bestimmte HTML-Elemente als Markup-Tags für SPS Komponenten zur Verfügung. Diese vordefinierten Formate 328 werden an die HTML-Ausgabe übergeben. Die in diesen Markup-Tags enthaltene Formatierung dient also, SPS Komponenten mit Stilen zu versehen. Wenn CSS-Stile auf vordefinierte Formate angewendet werden, haben die CSS-Stile Vorrang vor den Stilen des vordefinierten Formats. Eine Beschreibung zu vordefinierten Formaten finden Sie in diesem Abschnitt im Unterabschnitt Vordefinierte Formate 328.

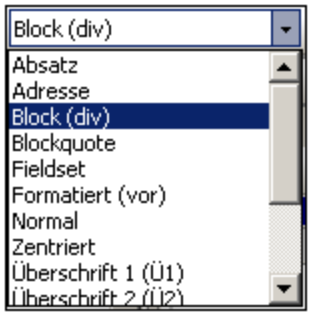
8.1 Vordefinierte Formate

StyleVision bietet eine Reihe von vordefinierten Formaten, wobei jedes davon einem HTML-Format entspricht (*Abbildung unten).* Wenn Sie einer Komponente im Design ein vordefiniertes Format zuweisen, wird diese Komponente mit Markups versehen, die der entsprechenden HTML-Semantik entsprechen. Dies hat zwei Auswirkungen:

- Es wird eine Formatierung angewendet, die im ausgewählten vordefinierten Format enthalten ist.
- Die Komponente ist im Komponententyp *Absatz* enthalten. Auf diese Art können ihr lokale Stile nach Komponententyp zugewiesen werden348.

Zuweisen von vordefinierten Formaten

Vordefinierte Formate können durch Klicken auf **Einfügen | Spezieller Absatz** und Auswahl des gewünschten Formats zugewiesen werden oder durch Auswahl des gewünschten Formats aus der Drowdown-Liste in der Symbolleiste (siehe Abbildung unten).



Enthaltene Stile

Die in StyleVision verwendeten vordefinierten Formate haben entweder eine oder beide der folgenden beiden Stil-Komponenten:

- eine Komponente für Textstile
- eine Komponente für Abstände

Das vordefinierte Format `para (p)` hat z.B nur eine Abstandskomponente; es setzt die ausgewählte Komponente zwischen vertikale Abstände und wendet keine Textformatierung an. Das vordefinierte Format `Überschrift 1 (Ü1)` dagegen hat sowohl eine Textstil-Komponente als auch eine Abstandskomponente.

Beachten Sie die folgenden Punkte zu vordefinierten Formaten:

- Die Abstandskomponente eines vordefinierten Formats gilt für jede Art von SPS Komponente, der Textstil gilt jedoch nur, wenn er angewendet werden kann. Wenn Sie z.B. ein Bild auswählen und das vordefinierte Format `Überschrift 1 (Ü1)` darauf anwenden, wird zwar die Abstandskomponente angewendet, nicht aber die Textstilkomponente.
- Die Textstilkomponente von vordefinierten Formaten wird nicht auf Dateneingabeelemente angewendet.
- Auf eine Komponente kann immer nur ein vordefiniertes Format angewendet werden.
- Das Format `Vorformatiert` wendet eine Formatierung an, die der vom HTML-Register `pre` entspricht: Zeilenumbrüche und Abstände im Text werden beibehalten und zur Anzeige wird eine Schriftart mit festen Leerzeichen (wie z.B. Courier) verwendet. Bei fortlaufendem Text ohne Zeilenumbrüche wie z.B.

 © 2019-2025 Altova GmbH

in einem Textabsatz, werden beim vordefinierten Format `Vorformatiert` Textzeilen ohne Zeilenumbrüche angezeigt. Wenn Sie Zeilenumbrüche verwenden wollen, verwenden Sie das vordefinierte Format `Vorformatiert, Zeilenumbruch`.

Definieren zusätzlicher Stile für ein vordefiniertes Format

Sie können für ein vordefiniertes Format zusätzlich zu den darin enthaltenen Stilen weitere Stile definieren. Wählen Sie es dazu aus und wenden Sie über die Seitenleiste "Stil" einen lokalen Stil an 348.

8.2 Output Escaping

Als Output Escaping wird bezeichnet, wenn ein Zeichen in einem Text als Zeichenreferenz oder Entity-Referenz dargestellt wird. Beide Typen (Zeichen- und Entity-Referenzen) werden durch ein kaufmännisches Und am Beginn und ein Semikolon am Ende der Zeichenfolge gekennzeichnet. Beispiel:

- Die hexadezimale (oder Unicode) Zeichenreferenz des Zeichens `A` ist `A`
- Die Dezimalzeichenreferenz des Zeichens `A` ist `A`
- Die HTML- (und XML) Entity-Referenz des Zeichens `&` ist `&`
- Die hexadezimale (oder Unicode) Zeichenreferenz des Zeichens `&` ist `&`
- Die Dezimalzeichenreferenz des Zeichens `&` ist `&`
- Die HTML- (und XML) Entity-Referenz des Zeichens `<` ist `<`

Output Escaping

Als "Output Escaping" wird die Art bezeichnet, wie Zeichen, **die im Input mit einem Escape-Zeichen versehen sind**, in der Ausgabe dargestellt werden. Ein Zeichen gilt als "output-escaped", wenn es in der Ausgabe als Zeichen- oder Entity-Referenz dargestellt wird. Beachten Sie, dass ein Zeichen nur dann "output-escaped" sein kann, wenn es im Input mit einem Escape-Zeichen versehen ist (*Beispiele dazu siehe Tabelle*). In einem SPS kann das Output Escaping für folgende Fälle aktiviert oder deaktiviert werden:

- Fragmente und statischen Text,
- den Platzhalter `Inhalt` und
- automatische Berechnungen

Dies erfolgt über das `Output Escaping deaktivieren` Attribut der Eigenschaftsgruppe *Text*. Der Standardwert dieser Eigenschaft ist `nein`, d.h. das Output Escaping ist nicht deaktiviert. D.h. Zeichen, die im Input mit Output Escaping versehen sind, werden in der Ausgabe standardmäßig ebenfalls mit Escape versehen (*Beispiele siehe Tabelle unten*):

So deaktivieren Sie das Output Escaping:

1. Wählen Sie (i) den statischen Text oder (ii) das Fragment des statischen Texts oder (iii) die automatische Berechnung, für die das Output Escaping deaktiviert werden soll, aus.
2. Wählen Sie in der Seitenleiste "Eigenschaften" die Eigenschaftsgruppe *Text* für das *Text*-Element aus und setzen Sie das `Output Escaping deaktivieren` Attribut für die verschiedenen Ausgaben einzeln oder für alle Ausgabe auf `yes`. Die verfügbaren Werte sind:

   - `Für HTML` (um `Output Escaping deaktivieren` für die HTML-Ausgabe auf `yes` zu setzen);
   - Für Authentic (um `Output Escaping deaktivieren` für die Authentic-Ausgabe auf `yes` zu setzen). Beachten Sie bitte, dass die Deaktivierung des Output Escaping für die Authentic-Ansicht **nur in der Enterprise Edition der Authentic-Ansicht möglich ist** (d.h. in der Enterprise Edition von StyleVision, Authentic Desktop, Authentic Browser und XMLSpy).
   - `Für alle` (um `Output Escaping deaktivieren` für alle Ausgaben außer Text `yes` zu setzen).

Wenn das Output Escaping für ein bestimmtes Ausgabeformat (z.B. für die HTML-Ausgabe) deaktiviert ist, wird der ausgewählte Text in diesem Ausgabeformat nicht mit Escape versehen, in den anderen Ausgabeformaten jedoch schon.

In der nachstehenden Tabelle sehen Sie einige Beispiele für Text, bei dem das Output Escaping deaktiviert und/oder aktiviert wurde.

© 2019-2025 Altova GmbH

Statischer Text	disable-output-escaping	Ausgabetext
&	no	&
&	yes	&
&	no	&
&	yes	&
<	no	<
<	yes	<
A	no	A
A	yes	A
<	no	<
<	yes	<
<	yes	<
&<	yes	&<

Anmerkung: `Output Escaping deaktivieren` wird in der Authentic-Ansicht nur in der Enterprise Edition von Altova-Produkten unterstützt.

Verwendung von "Output Escaping deaktivieren" in verschiedenen Formaten

Wenn das Output Escaping deaktiviert ist, kann der Textstring in einem Ausgabeformat eine Bedeutung haben, in einem anderen aber überhaupt keine. Betrachten Sie als Beispiel dazu den folgenden Eingabetext, der (hervorgehobene) Escape-Zeichen enthält:

```
&lt;b&gt;Dieser Text ist fett.&lt;/b&gt;
```

Wenn das Output Escaping deaktiviert ist, wird dieser Text folgendermaßen ausgegeben:

```
<b>Dieser Text ist fett.</b>
```

Wenn das Output Escaping für die HTML-Ausgabe deaktiviert ist und diese Ausgabe in einem Browser (und nicht in einem Text-Editor) angezeigt wird, erkennt der HTML-Browser die Markup-Tags und der Text wird fett angezeigt:

Dieser Text ist fett.

Wenn der Text allerdings in einem anderen Ausgabeformat wie z.B. PDF angezeigt wird, hat der Markup-Code, der in HTML eine Bedeutung hatte, in diesem anderen Format nicht unbedingt eine Bedeutung. Der Text aus dem oben gezeigten Beispiel wird als nicht mit Escape versehener Text (also wenn das Output Escaping deaktiviert wird) im PDF-Format folgendermaßen angezeigt:

```
<b>Dieser Text ist fett.</b>
```

Wie im obigen Beispiel gezeigt, kann der durch Deaktivierung des Output Escaping erzeugte Text in einem Ausgabeformat als Code interpretiert werden und im anderen aber nicht. Vergegenwärtigen Sie sich dies, wenn Sie die Eigenschaft "Disable-Output-Escaping" verwenden.

© 2019-2025 Altova GmbH

8.3 Werteformatierung (Formatieren von numerischen Datentypen)

Mit Hilfe der Werteformatierung kann der Inhalt von numerischen Datentyp-Nodes nach dem XML-Schema (*siehe Liste unten*[333]) in einem Format angezeigt werden, das nicht der lexikalischen Darstellung dieses Datentyps entspricht. (So lautet z.B. die lexikalische Darstellung des Datentyp-Node `xs:date` `JJJJ-MM-TT`, mit einer optionalen Zeitzonenkomponente wie z.B. `+02:00`.) Die Werteformatierung wird in der HTML- Ausgabe zur Verfügung stehen. Die Werteformatierung kann auch verwendet werden, um das Ergebnis einer automatischen Berechnung zu formatieren, wenn es im lexikalischen Format eines der numerischen Datentypen gespeichert ist (*siehe Liste unten*[333]), für die eine Werteformatierung verfügbar ist.

In den Unterabschnitten dieses Abschnitts wird Folgendes beschrieben:

- wie die Werteformatierungsmechanismen funktionieren[333], und
- die Syntax[336] zum Definieren der Werteformatierung.

Anmerkung: Die Werteformatierung ändert das Format, in dem die Daten im XML-Dokument gespeichert sind, nicht. Im gültigen XML-Dokument sind die Daten immer in dem lexikalischen Format des jeweiligen Node-Datentyps gespeichert. Die Werteformatierung wird auf die Anzeige in der Ausgabe (falls vorhanden) angewendet.

Numerische Datentypen, für die eine Werteformatierung verfügbar ist

Die Werteformatierung steht für die folgenden Datentypen zur Verfügung:

- `xs:decimal`; `xs:integer`; die 12 vordefinierte Typen, die von `xs:integer` abgeleitet sind
- `xs:double` und `xs:float`, für Werte von 0,000001 bis 1.000.000. Werte außerhalb dieses Bereichs werden in Exponentialschreibweise (z.B.: `1.0E7`) angezeigt. Es kann darauf keine Werteformatierung angewendet werden.
- `xs:date`; `xs:dateTime`: `xs:duration`
- `xs:gYear`; `xs:gYearMonth`; `xs:gMonth`; `xs:gMonthDay`; `xs:gDay`

Anmerkung: Nicht alle Formate sind in der Basic Edition verfügbar, da die Authentic-Ansicht in der Basic Edition nicht unterstützt wird.

8.3.1 Funktionsweise der Werteformatierung

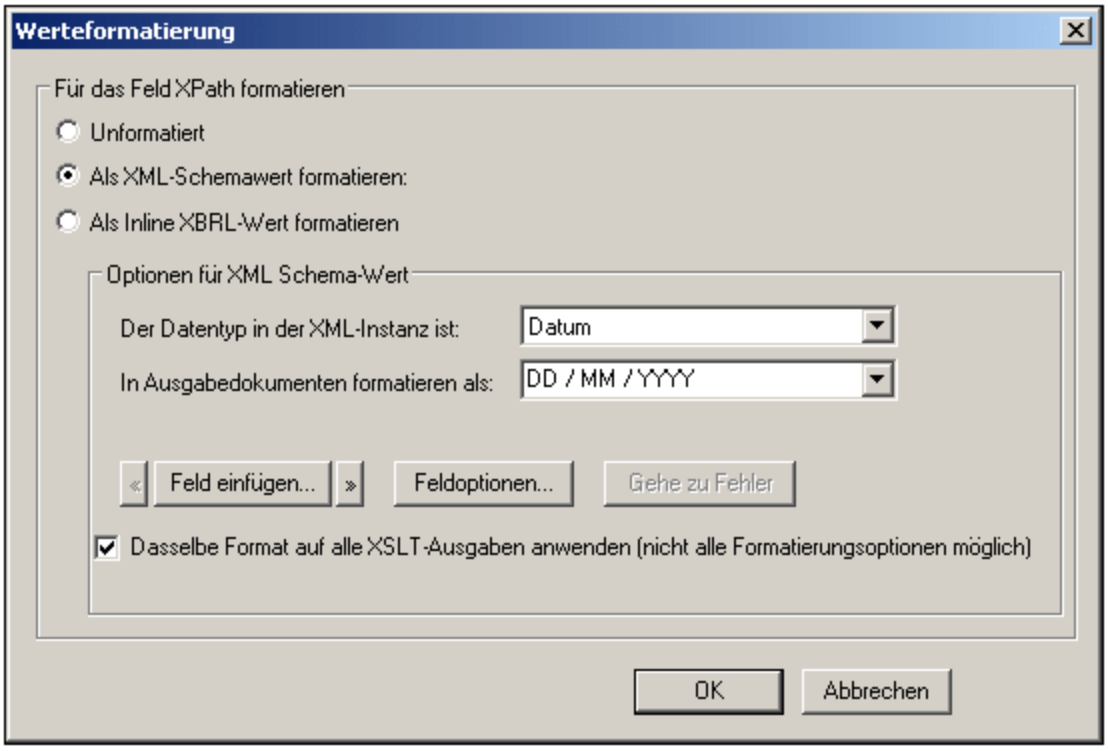
Werteformatierung kann auf folgende Datentypen angewendet werden:

- einen numerischen Datentyp-Node[333] wie z.B. `xs:decimal` oder `xs:date`, welcher im SPS **als Inhalt oder Eingabefeld vorhanden ist**.
- eine automatische Berechnung, deren Ergebnis ein Wert ist, der das lexikalische Format eines numerischen Datentyps[333] hat.

Definieren der Werteformatierung

Um die Werteformatierung für einen Node oder eine automatische Berechnung im SPS zu definieren, gehen Sie folgendermaßen vor:

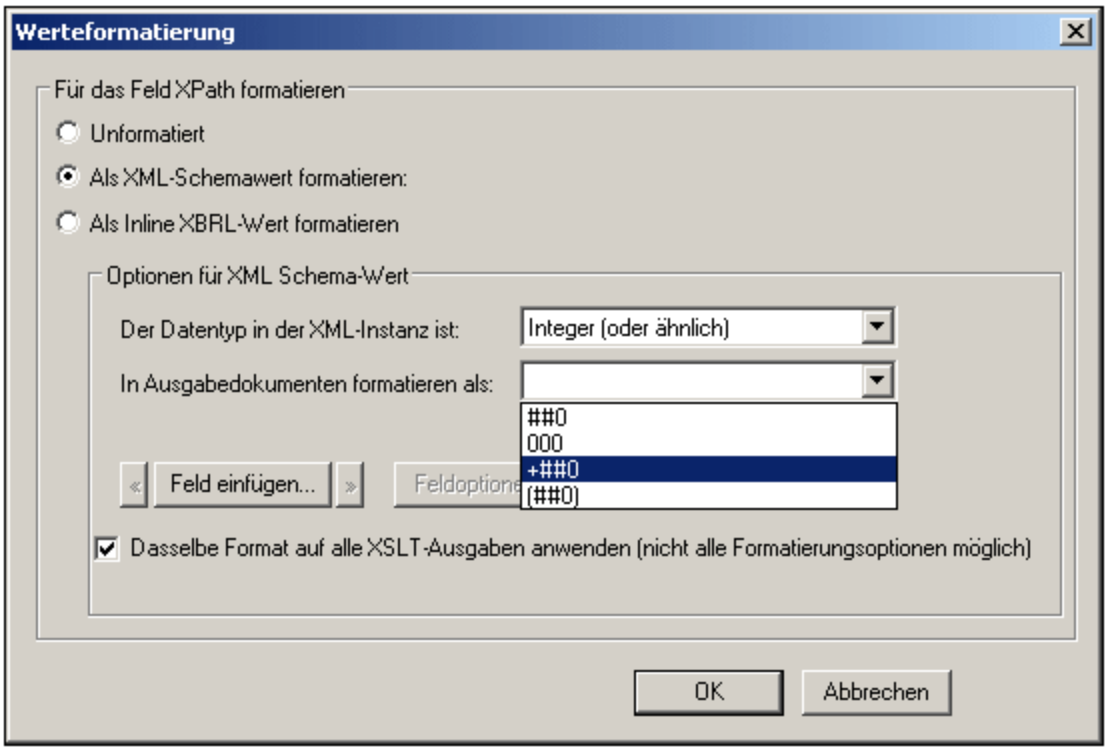
1. Wählen Sie den Platzhalter `Inhalt` oder das Eingabefeld des Node oder die automatische Berechnung aus.
2. Wählen Sie in der Seitenleiste "Eigenschaften" den Eintrag `Inhalt` und anschließend die Eigenschaftsgruppe *Inhalt* (oder `XPath`.). Klicken Sie anschließend auf die Bearbeiten-Schaltfläche ... der Eigenschaft `Werteformatierung`. Klicken Sie alternativ dazu auf die rechte Maustaste und wählen Sie im Kontextmenü den Befehl **Werteformatierung bearbeiten**. Daraufhin wird das Dialogfeld "Werteformatierung" angezeigt (*Abbildung unten*). Je nachdem, ob es sich bei der ausgewählten Komponente um einen Node oder eine automatische Berechnung handelt, sieht dieses Dialogfeld unterschiedlich aus. Wenn die ausgewählte Komponente ein Node ist, wird das unten stehende Dialogfeld angezeigt. In der Abbildung unten sehen Sie den Node vom Datentyp `xs:date`.



Beachten Sie, dass die Abbildung oben die Zeile: `Formate für Typ 'date'` enthält und dass neben dem Optionsfeld "*Unformatiert*" das Standardformat für den Datentyp `xs:date` angegeben wird. Bei einem Node eines anderen Datentyps, würden entsprechend andere Informationen angezeigt.

Wenn es sich bei der ausgewählten Komponente um eine automatische Berechnung handelt, erscheint das folgende Dialogfeld.

© 2019-2025 Altova GmbH



3. Sie geben nun an, ob die Anzeige des Komponentenwerts unformatiert oder formatiert sein soll. Wenn die Ausgabe unformatiert bleiben soll, aktivieren Sie das Optionsfeld *Unformatiert*. Aktivieren Sie andernfalls das Optionsfeld *Als XML-Schemawert formatieren*. (Wenn der Wert unformatiert ist, hat die Ausgabe die Standardformatierung für den Datentyp des ausgewählten Node oder den Datentyp des Ergebnisses der automatischen Berechnung. Wenn Sie *Als XML-Schemawert formatieren* für eine automatische Berechnung definieren, müssen Sie außerdem den Datentyp des erwarteten Ergebnisses aus einer Dropdown-Liste auswählen.
4. Geben Sie die Definition der Werteformatierung ein. Diese Definition kann auf drei Arten eingegeben werden: (i) durch Auswahl der Formatierung aus einer Dropdown-Liste von Optionen für diesen Datentyp (*siehe Eingabefeld "In Ausgabedokumenten formatieren als:" in der Abbildung unten*); (ii) durch Eingabe der Definition direkt in das Eingabefeld; und (iii) durch Verwendung der Schaltflächen **Feld einfügen** und **Feldoptionen** zur korrekten Eingabe der Definition. Eine vollständige Beschreibung der verschiedenen Formatierungsoptionen finden Sie unter Syntax der Werteformatierung 336.

Fehler in der Syntax

Bei einem Fehler in der Syntax geschieht Folgendes:

- Die Definition wird rot angezeigt.
- Unterhalb des Eingabefelds wird eine - ebenfalls rote - Fehlermeldung angezeigt.
- Die **OK**-Schaltfläche im Dialogfeld "Werteformatierung" ist deaktiviert.
- Die Schaltfläche **Gehe zu Fehler** im Dialogfeld "Werteformatierung" ist aktiviert. Wenn Sie darauf klicken, wird der Cursor an den Punkt in der Formatdefinition platziert, an dem der Syntaxfehler ist.

Daten und Datentyp stimmen nicht überein

Wenn die in den XML-Node eingegebenen Daten nicht mit dem lexikalischen Format des Datentyps für diesen Node übereinstimmen oder wenn das Ergebnis einer automatischen Berechnung nicht dem lexikalischen Format des erwarteten Datentyps entspricht, ist die Formatierung undefiniert und die Daten werden in der Ausgabe nicht korrekt dargestellt.

Anwendung der Werteformatierung auf die Ausgabe

Die von Ihnen definierte Werteformatierung wird auf die Authentic-Ansicht angewendet, die in der Enterprise und der Professional Edition zur Verfügung stehen.

Einige Werteformatierungsdefinitionen - nicht alle - können zusätzlich auch auf die HTML-Ausgabe angewendet werden. Aktivieren Sie dazu das Kontrollkästchen "Dasselbe Format auf alle XSLT-Ausgabe anwenden". Wenn diese Option nicht aktiviert ist oder nicht zur Verfügung steht, wird die Werteformatierung nur in der Authentic-Ansicht angezeigt, während in der Ausgabe der Wert im Standardformat für den Datentyp dieser Komponente angezeigt wird (das lexikalische Format).

8.3.2 Syntax der Werteformatierung

Die Syntax für die Werteformatierung lautet:

```
([präfix-zeichen]feld[suffix-zeichen][{feld-option1,feld-option2,...}])+
```

wobei **präfix-zeichen** und **suffix-zeichen** optionale Definitionen sind, um die Ausrichtung und die Anzeige positiver/negativer Symbole zu steuern;
feld jede Datentyp-spezifische Formatierung und jeder Text sein kann und
{feld-option(en)} ein optionaler Qualifier ist, mit dem zusätzliche Formatierungsoptionen aktiviert werden können.

Erklärung der Definitionssyntax

Die Werteformatierungsdefinition ist folgendermaßen zusammengesetzt:

- Die Definition besteht aus einem oder mehreren Feldern. So hat die Definition `TT Monat JJJJ` drei Felder.
- Felder können direkt hintereinander angezeigt werden oder durch die folgenden Zeichen getrennt werden: Leerzeichen, Bindestrich, Komma, Doppelpunkt, Punkt oder einen Textstring innerhalb von doppelten Anführungszeichen. So sind z.B. die Felder `TT` und `Monat` in der Definition: `TT-Monat' im Jahr 'JJJJ` durch einen Bindestrich getrennt und die Felder `Monat` und `JJJJ` sind durch einen in einfache Anführungszeichen gesetzten Textstring getrennt.
- Ein Feld kann optionale Präfix- und/oder Suffix-Zeichen haben. Z.B: `<+###,##0.00`.
- Ein Feld kann eine oder mehrere optionale Feldoptionen haben. Die Feldoption(en) für jedes Feld muss/müssen innerhalb einer einzigen geschwungenen Klammer stehen und darf/dürfen nicht durch ein Leerzeichen vom Feld getrennt sein. Mehrere Feldoptionen für ein einziges Feld werden durch ein "," (Komma) getrennt. In der Definition: `TT Monat{uc,ro} JJJJ` sind die Buchstaben `uc` und `ro` innerhalb der geschwungenen Klammer z.B. Feldoptionen für das Feld `Monat`.

© 2019-2025 Altova GmbH

Beispiele

Beispiel für eine Werteformatierung für einen `xs:decimal` Datentyp:

```
"$"(##0.00)
```

Beispiele für die Ausgabe wären:

```
$ 25.00
$ 25.42
$267.56
```

Beispiel für die Werteformatierung für einen `xs:date` Datentyp:

```
TT Monat{uc,ro} JJJJ
```

wobei `uc` und `ro` Feldoptionen sind, damit das Feld `Monat` in Großbuchstaben angezeigt wird und schreibgeschützt ist.

Ein Beispiel für die Ausgabe wäre:

```
24 SEPTEMBER 2003
```

Feldtypen

Ein Feldtyp stellt eine Komponente des Datums dar und gibt an, wie diese Komponente zu formatieren ist. Die Formatierung des Feldtyps kann mittels Präfixen und Suffixen sowie Feldoptionen geändert werden. In der folgenden Tabelle sind die verfügbaren Feldtypen aufgelistet. Beachten Sie, dass es im Dropdown-Menü des Dialogfelds "Werteformatierung" typspezifische und nur-Feld-spezifische Werteformatierungsdefinitionen gibt. Sie können eine dieser Definitionen auswählen und sie wie gewünscht durch Hinzufügen von Präfixen, Suffixen und/oder Feldoptionen ändern.

Feldtyp	Erklärung
#	Leerzeichen, wenn an dieser Stelle keine Ziffer steht
0	Null, wenn an dieser Stelle keine Ziffer steht
.	Dezimaltrennzeichen
,	Tausendertrennzeichen
Y	Jahr
y	Jahr (Basis = 1930); siehe Anmerkung unten
MM	Monat muss eine Länge 2 haben
DD	Tag muss eine Länge 2 haben

W	Wochennummer
d	Wochentagnummer (1 bis 7)
i	Tag im Jahr (1 bis 366)
hh	Stunde (0 bis 23), muss eine Länge 2 haben
HH	Stunde (1 bis 12), muss eine Länge 2 haben
mm	Minute, muss eine Länge 2 haben
ss	Sekunde, muss eine Länge 2 haben
AM	AM oder PM
am	am oder pm
AD	AD (n.Chr.) oder BC (v.Chr.)
ad	ad (n.Chr.) oder bc (v.Chr.)
CE	CE oder BCE
ce	ce oder bce

Feldtyp	Erklärung
Weekday	Wochentag (Sunday, Monday...)
WEEKDAY	Wochentag (SUNDAY, MONDAY...)
weekday	Wochentag (sunday, monday...)
Wkd	Wochentag (Sun, Mon...)
WKD	Wochentag (SUN, MON...)
wkd	Wochentag (sun, mon...)
Month	Monat (January, February...)
MONTH	Monat (JANUARY, FEBRUARY...)
month	Monat (january, february...)
Mon	Monat (Jan, Feb...)
MON	Monat (JAN, FEB...)
mon	Monat (jan, feb...)

© 2019-2025 Altova GmbH

Anmerkungen zur Feldlänge und der Länge des eingegebenen Strings

Beachten Sie bitte die folgenden Punkte zur Länge von Datenkomponenten:

Länge von Datumsfeldern: Bei Verwendung von Feldern wie z.B. `MM`, `DD`, `HH`, `hh`, `mm`, und `ss` müssen diese in der Definition die Länge 2 haben. Bei Verwendung der Felder `y` oder `Y` wird die Länge der Ausgabe durch die Anzahl der `y` oder `Y` Zeichen in der Definition festgelegt. Wenn Sie z.B. `YYY` angeben, wäre die Ausgabe für einen Wert `2006` `006`; bei einer Definition `YYYYYY` wäre die Ausgabe `002006`. Siehe auch Basisjahr weiter unten.

Erweitern der Feldlänge: Das `*` Symbol (Sternchen) dient dazu, die Länge eines nicht-semantischen numerischen Felds (Ganzzahlen, Dezimalzahlen, usw.) zu erweitern. Bei Dezimalzahlen kann es auf beiden Seiten des Dezimalpunkts (bzw. Kommas) stehen. So wird z.B. mit der Werteformatierung `*0.00*` sichergestellt, dass wie in der Formatierung definiert, an den leeren Stellen Nullen für eine Zahl angezeigt werden sowie eine beliebige Anzahl an Stellen links und rechts vom Dezimalpunkt bzw. -komma.

Anmerkung: Wenn in einem Feld kein Text dargestellt wird, kann dies an Ihren Regionseinstellungen in Windows liegen. Windows gibt einen leeren String für das Feld AM/PM zurück, wenn die Sprache auf Deutsch gesetzt ist.

Präfixe und Suffixe

Mit Hilfe von Präfixen und Suffixen können Sie die Textausrichtung und die positive/negative Darstellung von Feldern ändern. In der folgenden Tabelle sehen Sie eine Liste dieser Präfixe und Suffixe.

Präfix	Suffix	Erklärung
<		Ausrichtung linksbündig; Standard für Text. Bei Zahlen, die standardmäßig rechtsbündig ausgerichtet sind, ist dies von Bedeutung, wenn es eine feste Anzahl vorangestellter Leerzeichen gibt.
>		Ausrichtung rechtsbündig; Standardeinstellung bei Zahlen.
?		Minuszeichen neben einer Zahl, wenn es sich um eine negative Zahl handelt; andernfalls kein Präfix. Dies ist die Standardeinstellung bei Zahlen.
<?		Minuszeichen linksbündig ausgerichtet bei negativen Zahlen; andernfalls kein Präfix. Minuszeichen, gefolgt von linksbündig ausgerichteter Zahl.
<?>		Minuszeichen linksbündig ausgerichtet bei negativen Zahlen; andernfalls kein Präfix. Rechtsbündig ausgerichtete Zahl.
-	-	Minuszeichen neben der Zahl bei negativen Zahlen; andernfalls Leerzeichen; wird vor der Zahl angezeigt (Präfix); wird nach der Zahl angezeigt (Suffix).
<-	>-	Minuszeichen bei negativen Zahlen; andernfalls Leerzeichen. Zahl und Zeichen nebeneinander. Links ausgerichtet (Präfix); rechts ausgerichtet (Suffix).

<->		Minuszeichen linksbündig ausgerichtet bei negativen Zahlen; andernfalls Leerzeichen. Zahl ist rechtsbündig ausgerichtet.
+	+	Neben der Zahl wird immer ein Plus- oder Minuszeichen angezeigt; vor der Zahl (Präfix), nach der Zahl (Suffix).
<+	>+	Neben der Zahl wird immer ein Plus- oder Minuszeichen angezeigt, linksbündig ausgerichtet (Präfix), rechtsbündig ausgerichtet (Suffix).
<+>		Plus- oder Minuszeichen wird immer linksbündig, Zahl rechtsbündig angezeigt.
(	)	Klammern bei negativen Zahlen; andernfalls Leerzeichen. Neben der Zahl.
<(		Klammern bei negativen Zahlen; andernfalls Leerzeichen. Neben der Zahl. Linksbündig.
<(>		Klammern bei negativen Zahlen; andernfalls Leerzeichen. Linke Klammer linksbündig; Zahl und recht Klammer nebeneinander und rechtsbündig.
[	]	Klammern bei negativen Zahlen; andernfalls nichts. Neben der Zahl.
*	*	Erweiterbare Anzahl von Stellen auf der linken Seite (Präfix) oder auf der rechten Seite (Suffix)
_	_	Leerzeichen
^	^	Füllzeichen (in den Optionen definiert)
	th	Ordnungszahl in Englisch (st, nd, rd oder th)
	TH	Ordnungszahl in Englisch (ST, ND, RD oder TH)

Feldoptionen

Mit Hilfe von Feldoptionen können Sie komplexe Änderungen an Feldern vornehmen. Die folgenden Optionen stehen zur Verfügung:

Option	Erklärung
uc	In Großbuchstaben schreiben
lc	In Kleinbuchstaben schreiben
left	Linksbündig
right	Rechtsbündig
ro	Schreibgeschützt (XML); Bearbeitung ist nicht zulässig

© 2019-2025 Altova GmbH

edit	Das Feld kann bearbeitet werden (standardmäßig aktiv)
dec=<char>	Geben Sie ein Zeichen für den Dezimalpunkt ein (Standardeinstellung Punkt)
sep=<char>	Geben Sie ein Zeichen für das Stellentrennzeichen ein (Standardeinstellung Komma)
fill=<char>	Füllzeichen definieren
base=<year>	Basisjahr für Jahr-Felder (*siehe Anmerkung unten*)
pos	Nur positive Zahlen anzeigen; Eingabe von negativen Zahlen ist zulässig

Mit Hilfe der Feldoptionen wird die Zahlenformatierung für europäische Sprachen generiert, in denen Punkte und Kommas anders als im Englischen verwendet werden, z.B. `123.456,75` statt wie im Englischen `123,456.75`.

Um diese Formatierung zu definieren wäre die Werteformatierung: `###,###.##{dec=,,sep=.}`

Beachten Sie, dass im Feld die englische Formatierung beibehalten wird, während mit den Feldoptionen `dec` und `sep` das Dezimalsymbol und das Kommastellentrennzeichen definiert werden. Wenn kein Dezimalsymbol und Kommastellentrennzeichen definiert ist werden die regionalen Einstellungen des Windows-Betriebssystems (**Systemsteuerung | All Items | Region | Format**) verwendet.If the decimal symbol and digit separator are not specified, these characters will default to decimal symbol and digit separator of the regional settings of the Windows OS (**Control Panel | Alle Elemente | Region | Format**).

8.4 Arbeiten mit CSS-Stilen

Die Stile des SPS Design-Dokuments werden mit Hilfe von CSS-Regeln definiert. Stilregeln können folgendermaßen definiert werden:

- In externen CSS Stylesheets[343]. Externe Stylesheets können über die Seitenleiste Design-Übersicht[36] und die Seitenleiste Globale Stile[45] hinzugefügt werden.
- In globalen Stilen[346] für das SPS, die entweder im SPS oder an seinem Anfang definiert werden können. (In der HTML-Ausgabe werden diese globalen Stile im Child-Element `style` des Elements `head` definiert.) Globale Stile werden in der Seitenleiste Globale Steile[45] definiert.
- Lokal[348] für einzelne Komponenten des Dokuments. In der HTML-Ausgabe werden solche Regeln im `style`-Attribut der einzelnen HTML-Elemente definiert. Lokale Stile werden in der Seitenleiste Stil[47] definiert.

Jede der oben aufgelisteten Methoden zur Erstellung von Stilen wird in den Unterabschnitten dieses Abschnitts näher beschrieben. (*Links oben*).

Terminologie

Ein CSS Stylesheet besteht aus einer oder mehreren Stilregeln. Eine Regel sieht folgendermaßen aus.

```
H1 { color: blue }
```

oder

```
H1 { color: blue;
     margin-top: 16px; }
```

Jede Regel hat einen Selektor (im Beispiel oben `H1`) und eine Deklaration (`color: blue`). Die Deklaration ist eine Liste von Eigenschaften (z.B. `color`) mit Werten (`blue`). Jedes Eigenschaft-Wert-Paar wird als Stildefinition bezeichnet. In StyleVision können CSS-Stile in der Seitenleiste Stil[47] (lokale Stile) und in der Seitenleiste Gobale Stile[45] (globale Stile) definiert werden.

Stufenförmige Reihenfolge

Bei der Anwendung von CSS-Stilen gilt das stufenförmige Prinzip. D.h. welche Regel Vorrang hat, wird auf folgender Basis ermittelt:

1. ***Ursprung.*** Externe Stylesheets sind globalen Stilen untergeordnet, globale Stile sind lokalen Stilen untergeordnet. Externe Stylesheets werden als importiert betrachtet. Die Reihenfolge des Imports ist von Bedeutung. Das zweite von zwei importierten Stylesheets hat Vorrang.
2. ***Spezifizität.*** Wenn zwei Regeln für dasselbe Element gelten, hat die Regel mit dem spezifischeren Selektor Vorrang.
3. ***Reihenfolge.*** Wenn zwei Regeln denselben Ursprung und dieselbe Spezifizität haben, hat die Regel im Stylesheet Vorrang, die weiter unten vorkommt. Importierte Stylesheets werden in der Reihenfolge als vor dem importierenden Stylesheet kommend betrachtet.

CSS-Stile in modularen SPSs

Wenn ein SPS Modul zu einem anderen SPS hinzugefügt wird, haben die CSS-Stile im referenzierenden SPS Vorrang vor jenen im hinzugefügten Modul. Wenn mehrere Module hinzugefügt werden, haben die CSS-Stile in

© 2019-2025 Altova GmbH

den Modulen, die in der Modulliste weiter unten gereiht sind, Vorrang. Nähere Informationen zu modularen SPSs finden Sie im Abschnitt Modulare SPSs 216.

CSS-Unterstützung in Internet Explorer

Internet Explorer (IE)-Versionen vor IE 6.0 interpretieren bestimmte CSS-Regeln anders als IE 6.0 und höher. Daher muss man beim Design beachten, für welche Version von IE das Design erstellt wird. IE 6.0 und höher unterstützt sowohl die alte als auch neuere Interpretationen, daher können Sie auch die ältere Interpretation in den neueren Versionen (ab IE 6.0) verwenden. Welche Interpretation von IE 6.0 und höher verwendet wird, wird durch einen Switch im HTML-Code festgelegt. In einem SPS können Sie angeben 489, ob für das HTML-Ausgabedokument die Stile entsprechend der älteren oder neueren Interpretation von Internet Explorer 101 verwendet werden sollen. Definieren Sie anschließend die CSS-Stile entsprechend der gewählten Interpretation. Nähere Informationen finden Sie unter Eigenschaften: CSS-Unterstützung 490.

Anmerkung: Nähere Informationen zur CSS-Spezifikation finden Sie unter http://www.w3.org/TR/REC-CSS2/.

8.4.1 Externe Stylesheets

In diesem Abschnitt wird beschrieben, wie Sie externe CSS Stylesheets über die StyleVision-Benutzeroberfläche verwalten können. Der Abschnitt ist in die folgenden Unterabschnitte gegliedert:

- Hinzufügen eines externen CSS Stylesheet zum SPS 343
- Anzeigen und Ändern der Struktur der externen CSS Stylesheets 344
- Ändern der Vorrangigkeit der externen CSS Stylesheets 344
- Wechseln zwischen der dem vollständigen CSS Stylesheet-Satz und einem einzigen CSS Stylesheet 345

Externe CSS Stylesheets können über zwei Seitenleisten verwaltet werden: die Seitenleiste "Globale Stile 45 und die Seitenleiste "Design-Übersicht" 36. Wenn ein Aspekt der externen Stylesheets in beiden Seitenleisten angezeigt werden kann (z.B. die relative Vorrangigkeit mehrerer Stylesheets), so werden Änderungen, die in einer Seitenleiste vorgenommen wurden, automatisch auch in der anderen Seitenleiste angezeigt.

Hinzufügen eines externen CSS Stylesheet zum SPS

Um dem SPS ein externes CSS Stylesheet zuzuweisen, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Design-Ansicht im Fenster "Gobale Stile" den Eintrag "Extern" aus (*Abbildung unten*).

2. Klicken Sie links oben in der Symbolleiste des Fensters Gobale Stile auf die Schaltfläche **Hinzufügen** (*siehe Abbildung oben*).

3. Suchen Sie im Dialogfeld "Öffnen" die gewünschte CSS-Datei, wählen Sie sie aus und klicken Sie auf **Öffnen**. Die CSS-Datei wird in der Baumstruktur zum Eintrag "Extern" hinzugefügt (*siehe Liste der Struktur und Abbildung unten*).
4. Um ein weiteres externes CSS Stylesheet hinzuzufügen, wiederholen Sie die Schritte 1 bis 3. Das neue CSS Stylesheet wird unterhalb von allen zuvor hinzugefügten externen CSS Stylesheets zur externen Struktur hinzugefügt.

Anmerkung: Sie können ein externes Stylesheet auch über die Seitenleiste Design-Übersicht 36 hinzufügen.

Anzeigen und Ändern der Struktur der externen CSS Stylesheets

Die Struktur der externen CSS Stylesheets ist folgendermaßen gegliedert (*siehe auch Abbildung unten*):

```
- CSS-1.css  (File location appears on mouseover)
  - Media (can be defined in Style Repository window)
  - Rules (non-editable; must be edited in CSS file)
    - Selector-1
      - Property-1
      - ...
      - Property-N
    - ...
    - Selector-N
+ ...
+ CSS-N.css
```

Das Medium, auf das dieses jeweilige Stylesheet angewendet werden kann, kann im Fenster "Gobale Stile" bearbeitet werden; klicken Sie dazu auf den Nach-unten-Pfeil rechts vom Eintrag und wählen Sie in der Dropdown-Liste das gewünschte Medium aus. Die im externen CSS Stylesheet definierten Regeln werden im Fenster "Gobale Stile" angezeigt, können aber nicht bearbeitet werden. Das Stylesheet, die Regeln und die einzelnen Selektor-Einträge in der Struktur können durch Klicken auf das + und - Symbol links von jedem Eintrag erweitert und reduziert werden (*siehe Abbildung unten*).

Um ein externes Stylesheet zu löschen, wählen Sie das Stylesheet aus und klicken Sie in der Symbolleiste "Gobale Stile" auf die Schaltfläche **Zurücksetzen**.

Ändern der Vorrangigkeit der externen CSS Stylesheets

Die im Fenster "Gobale Stile" zugewiesenen externen CSS Stylesheets werden mittels der Anweisung `@import` in die HTML-Ausgabedatei importiert. Dies würde in der HTML-Datei in etwa folgendermaßen aussehen:

```
<html>
   <head>
      <style>
         <!--
         @import url("ExternalCSS-1.css");
         @import url("ExternalCSS-2.css")screen;
         @import url("ExternalCSS-3.css")print;
         -->
      </style>
   </head>
   <body/>
</html>
```

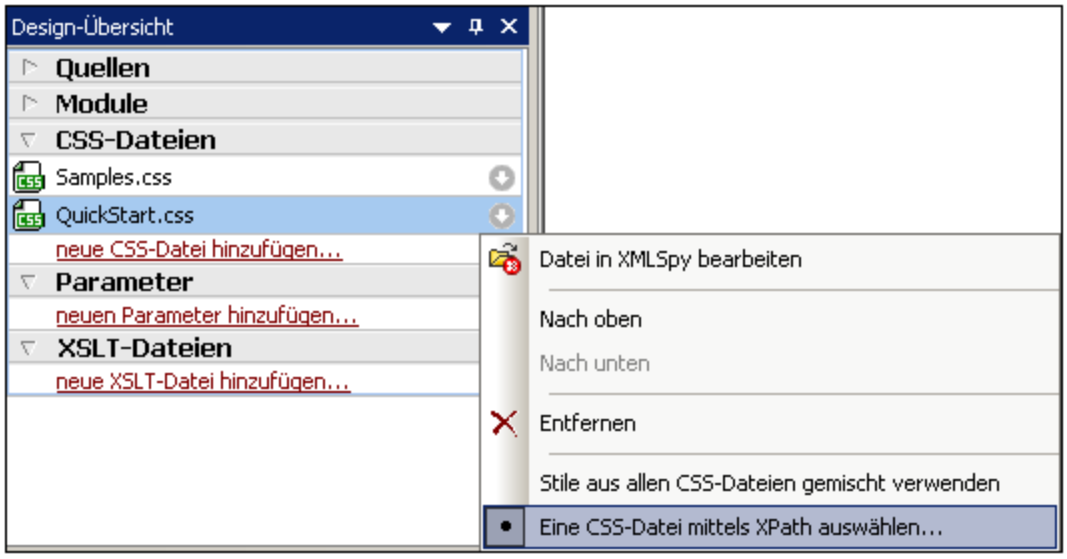
 © 2019-2025 Altova GmbH

Die Reihenfolge, in der die Dateien in der HTML-Datei aufgelistet sind, entsprechen der Reihenfolge, in der sie im Fenster "Gobale Stile" und in der Seitenleiste "Design-Übersicht" in der CSS-Dateistruktur unter "Extern" aufgelistet sind. Um die Reihenfolge der CSS Stylesheets im der Seitenleiste "Globale Stile" unter "Extern" zu ändern, wählen Sie das Stylesheet aus, für das sie die Priorität ändern möchten. Verschieben Sie das Stylesheet anschließend mit Hilfe der **Nach oben** oder **Nach unten** Schaltflächen der Symbolleiste der Seitenleiste "Gobale Stile" an die gewünschte Stelle in der Struktur. In der Seitenleiste "Design-Übersicht" müssen Sie dazu auf die Bearbeiten-Schaltfläche eines CSS Stylesheet klicken und je nach Bedarf den Befehl **Nach oben** oder **Nach unten** auswählen.

Wichtig: Beachten Sie, dass das niedrigste Stylesheet die höchste Import-Priorität hat und dass die Import-Priorität mit jedem Stylesheet, das weiter oben in der Liste steht, abnimmt. Die Import-Prioritäten in der Liste oben sind: (i) `ExternalCSS-3.css`; (ii) `ExternalCSS-2.css`; (iii) `ExternalCSS-1.css`. Wenn in zwei CSS-Regeln in zwei verschiedenen Stylesheets derselbe Selektor verwendet wird, wird die Regel im Stylesheet mit der höheren Import-Priorität angewendet.

Wechseln zwischen allen CSS-Dateien und einer einzigen CSS-Datei

Sie können auswählen, ob entweder (i) die Regeln aus allen CSS-Dateien angewendet werden sollen, wobei die Regeln für kaskadierende Stylesheets die Vorrangigkeit festlegen, oder (ii) ob nur die Regeln aus einer einzigen ausgewählten CSS-Datei angewendet werden sollen. Die gewünschte Option können Sie in der Seitenleiste "Design-Übersicht" (*siehe Abbildung unten*) auswählen. Klicken Sie auf die Bearbeitungsschaltfläche einer der aufgelisteten CSS-Dateien und wählen Sie entweder den Befehl **Stile ... gemischt verwenden** oder **Eine CSS-.Datei... auswählen**. Diese Option steht auch in den Globalen Stilen (oder jedem der externen Stylesheets) zur Verfügung.



Bei Auswahl des Befehls **Eine CSS-Datei mittels XPath auswählen** erscheint ein Dialogfeld, in dem Sie den XPath-Ausdruck eingeben können (*Abbildung unten*). Das Ergebnis des XPath-Ausdrucks muss der Namen einer der CSS-Dateien im SPS sein, und zwar genau in der Form, wie sie im oberen Bereich des Dialogfelds aufgelistet sind. Wenn Sie den Dateinamen als String eingeben, so muss der String wie alle Strings in XPath-Ausdrücken innerhalb von einfachen Anführungszeichen stehen.

- *Wenn Stile gemischt werden und aus allen CSS-Dateien verwendet werden:* In der Authentic- und der HTML-Ausgabe werden alle Regeln aus allen CSS-Dateien angewendet und für alle Design-

Komponenten unterstützt. Konflikte werden auf Basis des Vorrangs der CSS-Datei gelöst. In der werden nur alle anderen als Klassen-Selektor-Regeln angewendet, wobei Konflikte auf Basis des Vorrangs der CSS-Datei gelöst werden.

8.4.2 Globale Stile

Globale Stile werden im Fenster Gobale Stile für das gesamte SPS Design definiert und im Fenster Gobale Stile unter dem Eintrag "Global" aufgelistet. Sie werden an das HTML-Ausgabedokument als CSS-Regeln übergeben. Im HTML-Dokument werden diese CSS-Regeln in das Element `/html/head/style` geschrieben.

Im Fenster Gobale Stile ist ein globaler Stil eine einzige CSS-Regel, die aus einem Selektor und CSS-Eigenschaften für diesen Selektor besteht. Die Erstellung eines globalen Stils erfolgt daher in zwei Schritten:

- Hinzufügen eines neuen Stils und Deklarieren des CSS-Selektors dafür
- Definieren der CSS-Eigenschaften für den Selektor

Unterstützte Selektoren

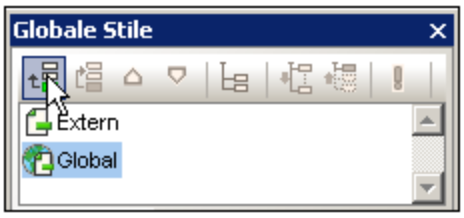
Die folgenden Selektoren werden unterstützt:

- *Universaler Selektor:* wird als `*` angegeben
- *Typ-Selektoren:* Elementnamen, wie z.B. `h1`
- *Attribut-Selektoren:* z.B., `[class=maindoc]`
- *Klassen-Selektoren:* z.B., `.maindoc`
- *ID-Selektoren:* z.B., `#header`

Hinzufügen eines globalen Stils

Um einen globalen Stil zum SPS Design hinzuzufügen, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Design-Ansicht im Fenster "Gobale Stile" den Eintrag "Global" aus (*Abbildung unten*).



2. Klicken Sie auf die Schaltfläche **Hinzufügen** links oben in der Symbolleiste des Fensters Gobale Stile (*siehe Abbildung oben*). Daraufhin wird unter dem Eintrag "Global" ein globaler Stil mit einem ***** Selektor eingefügt (der Selektor wählt alle HTML-Elemente aus); der Stern-Selektor ist der Standardselektor für alle neu eingefügten globalen Stile.
3. Um den Selektor vom Standardselektor zu ändern, doppelklicken Sie auf den Selektor und bearbeiten Sie ihn.

© 2019-2025 Altova GmbH

4. Definieren Sie nun die CSS-Eigenschaftswerte für den Selektor. Eine Anleitung dazu finden Sie im Abschnitt Definieren von Stilwerten[351].
5. Um einen weiteren globalen Stil hinzuzufügen, wiederholen Sie die Schritte 1 bis 4. Der neue globale Stil wird in der Baumstruktur unter dem Eintrag "Global" unterhalb aller zuvor hinzugefügten globalen Stile hinzugefügt.

Anmerkung:

- Sie können globale Stile unter dem Eintrag "Global" auch durch Klicken auf die **Einfügen-**Schaltfläche im Fenster "Gobale Stile" vor einem ausgewählten globalen Stil einfügen. Die Schaltflächen **Hinzufügen** und **Einfügen** stehen auch über das Kontextmenü zur Verfügung, das aufgerufen wird, wenn Sie mit der rechten Maustaste auf einen globalen Stil klicken.
- Ein globaler Stil mit einem Selektor, der ein HTML-Element ist, kann durch Rechtsklick auf einen Eintrag unter "Global" und Auswählen von **Selektor hinzufügen | HTML |** ***HTMLElementName*** hinzugefügt werden.

Bearbeiten und Löschen von globalen Stilen

Sowohl der Stilselektor als auch seine Eigenschaften können im Fenster "Gobale Stile" bearbeitet werden.

- Um einen Selektor zu bearbeiten, doppelklicken Sie auf den Selektornamen, platzieren Sie den Cursor in das Textfeld und bearbeiten Sie ihn.
- Informationen zum Definieren und Bearbeiten der Eigenschaftswerte eines Stils finden Sie unter Definieren von Stilwerten[351]. (Die Stileigenschaften können in drei möglichen Ansichten angezeigt werden. Diese Ansichten, bzw. wie man zwischen diesen wechselt, sind im Abschnitt Ansichten von Eigenschaftsdefinitionen[49] näher beschrieben.)

Bearbeiten und Löschen von globalen Stilen

Sie können im Fenster "Globale Stile" sowohl den Stil-Selektor als auch seine Eigenschaften bearbeiten.

- Um einen Selektor zu editieren, doppelklicken Sie auf den Selektornamen und platzieren Sie anschließend den Cursor in das Textfeld und editieren Sie es.
- Informationen zum Definieren und Editieren der Eigenschaftswerte eines Stils finden Sie unter Definieren von Stilwerten[351]. (Die Stileigenschaften können in drei möglichen Ansichten angezeigt werden. Eine Beschreibung dieser Ansichten und wie Sie zwischen diesen Ansichten wechseln, finden Sie unter Ansichten von Definitionen[49].

Um einen globalen Stil zu löschen, wählen Sie den Stil aus und klicken Sie in der Symbolleiste der globalen Stile auf **Zurücksetzen.**

Ändern der Priorität von globalen Stilen

Globale Stile, die im Fenster "Gobale Stile" zugewiesen werden, werden als CSS-Regeln in das Element `/html/head/style` platziert. In der HTML-Datei würden Sie ungefähr folgendermaßen aussehen:

```
<html>
   <head>
      <style>
         <!--
         h1    { color:blue;
                 font-size:16pt;
               }
         h2    { color:blue;
                 font-size:14pt;
               }
         .red  { color:red;}
         .green   { color:green;}
         .green   { color:lime;}
         -->
      </style>
   </head>
   <body/>
</html>
```

Die Reihenfolge, in der die globalen Stile in der Authentic-Ansicht und im HTML-Dokument aufgelistet sind, entspricht der Reihenfolge, in der sie im Fenster Gobale Stile unter "Global" aufgelistet sind. Die Reihenfolge in der Authentic -Ansicht und im HTML-Dokument ist von Bedeutung. Wenn von zwei Selektoren derselbe Node ausgewählt wird, so hat derjenige Selektor, der weiter unten in der Liste der globalen Stile aufgelistet ist, Vorrang. Angenommen Sie haben ein HTML-Dokument mit der oben gezeigten Liste. Gäbe es darin ein Element `<h1 class="green">`, so würden diesem Element drei globale Stile entsprechen: der Stil mit dem Selektor `h1` und die beiden Selektoren für die Klasse `.green`. Es wird die Farbeigenschaft des `.green` Selektors mit der Farbe `lime` verwendet, weil dieser nach dem `.green` Selektor mit der Farbe `green` aufgelistet ist und daher Vorrang vor dem Selektor `h1` hat. (Klassenselektoren haben immer Vorrang vor Node-Selektoren, daher haben beide `.green` Selektoren Vorrang vor dem `h1` Selektor, unabhängig von ihrer jeweiligen Position im Verhältnis zum `h1` Selektor.) Auf das `<h1>` Element wird jedoch die Schriftgröße des `h1` Stils angewendet, weil es keinen Selektor mit einer höheren Priorität gibt, der für das `<h1>` Element definiert ist und für den eine Schriftgröße definiert wurde.

Um die Priorität eines globalen Stils zu ändern, wählen Sie den Stil aus und verschieben Sie den globalen Stil in der Struktur der globalen Stile mit Hilfe der Schaltflächen **Nach oben** und **Nach unten**. Wenn Sie z.B. den globalen Stil `.green` an eine Stelle vor den globalen Stil .red verschieben, hätte die Farbeigenschaft des Stils .red Vorrang vor der des Stils `.green`.

Beachten Sie, dass Klassenselektoren immer Vorrang vor Typselektoren haben. Wenn Sie z.B. die Reihenfolge der Seletoren in `.red. .green h1 h2` ändern würden, wären `h1` und `h2` dennoch grün.

8.4.3 Lokale Stile

Wenn Stile lokal definiert werden, werden die Stilregeln direkt in der Komponente definiert. Diese lokalen Stilregeln haben Vorrang vor globalen Stilregeln und Stilregeln in externen CSS Stylesheets, die diese

 © 2019-2025 Altova GmbH

Komponente auswählen. Lokal definierte Stile sind CSS-Stile und werden in entweder über die Symbolleiste "Format" 459 oder in der Seitenleiste Stil 47 definiert (im Gegensatz zu globalen Stilen, die in der Seitenleiste Gobale Stile 45 definiert werden).

Definition lokaler Stile über die Symbolleiste "Format"

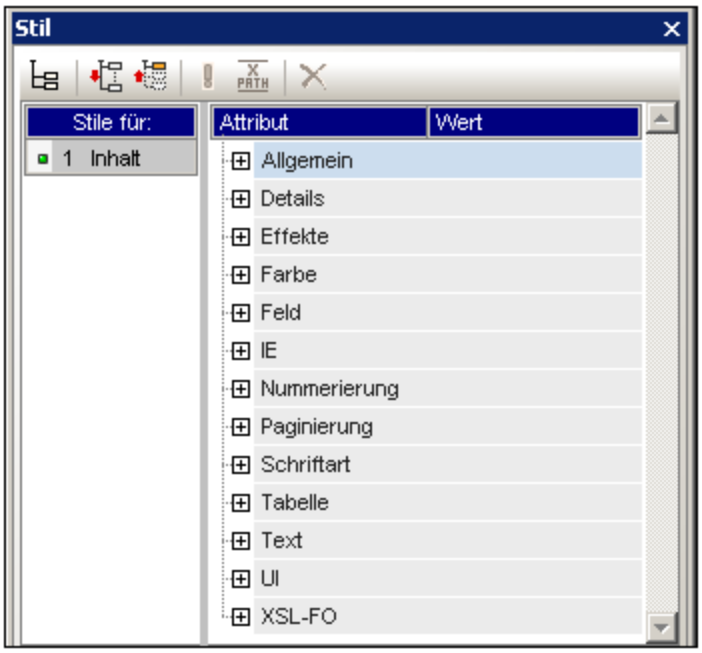
Sie können Inhalt im Design auswählen und über die Symbolleiste "Format" (*siehe Abbildung unten*) lokale Stile darauf anwenden.

Sie können vordefinierte HTML-Formatierungen wie z.B. `div`, `Ü1`, `vorformatiert`, usw.), Textstile, Hintergrundfarben, die Textausrichtung, Listen und Hyperlinks anwenden. Siehe dazu Abschnitt Format 459.

Zuweisung lokaler Stile über die Seitenleiste "Stil"

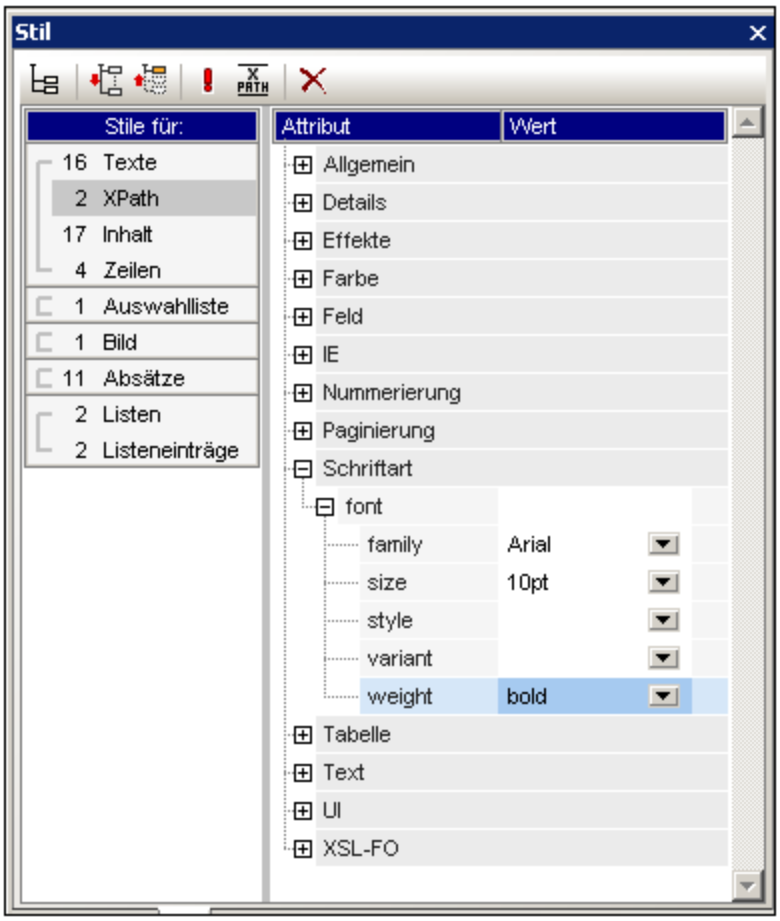
Die lokale Definition eines Stils über die Seitenleiste "Stil" erfolgt in drei Schritten:

1. Die Komponente, für die Stile definiert werden sollen, wirde in der Design-Ansicht ausgewählt. Stile können für jede Komponente im Design mit Ausnahme von Node-Tags definiert werden. Die in der Design-Ansicht ausgewählte Komponente wird in der Seitenleiste "Stil" in der Spalte *Stile für* angezeigt (*siehe Abbildung unten*). In der Abbildung unten wurde eine `Inhalt`-Komponente in der Design-Ansicht ausgewählt, die nun in der Spalte *Stile für* angezeigt wird..



Oft enthält die in der Design-Ansicht ausgewählte Komponente andere Komponenten. In diesem Fall werden alle Komponenten in der Auswahl geordnet nach Komponententyp in der Seitenleiste "Stil" in der Spalte *Stile für* angezeigt. In der Abbildung unten sehen Sie die verschiedenen Komponententypen, die die Auswahl in der Design-Ansicht enthält. Links von jeder Komponente wird die Anzahl der

Instanzen dieser Komponente innerhalb der Auswahl angezeigt. So gibt es z.B. in der Abbildung unten in der Design-Ansicht unter anderem 16-Textkomponenten und zwei XPath-Komponenten. Sie können auch einen Komponentenbereich auswählen, indem Sie die **Umschalttaste** gedrückt halten, während Sie die zweite Komponente am Ende des Bereichs auswählen.

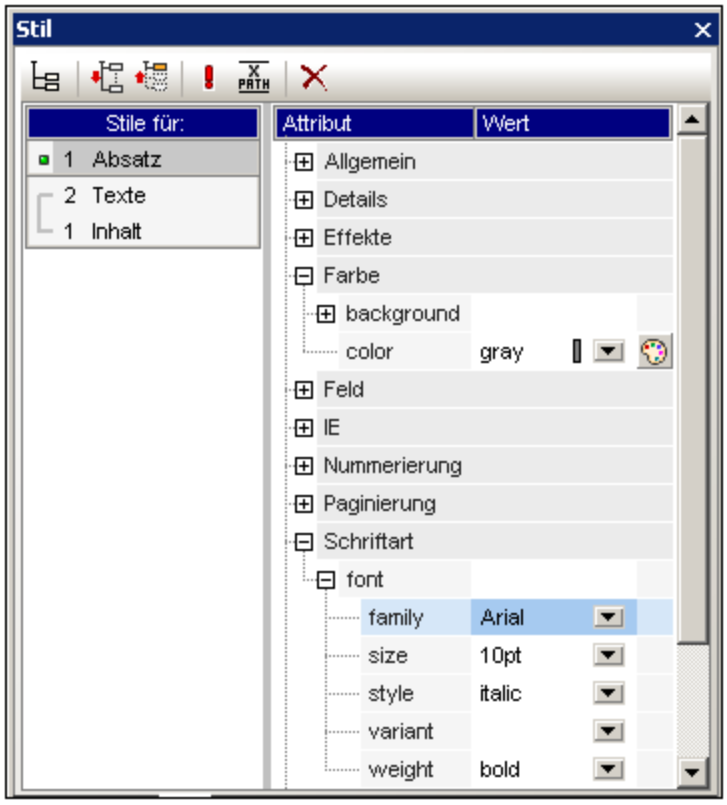


2. Nachdem Sie in der Design-Ansicht eine Auswahl getroffen haben, wählen Sie in der Spalte *Stile für* den **Komponententyp** aus, für den Sie einen Stil definieren möchten. Wenn es von diesem Komponententyp mehr als eine Komponente gibt, werden die Stile auf alle diese Komponenteninstanzen angewendet. Wenn Sie z.B. den Eintrag *16-Texte* in der Abbildung oben auswählen, so werden die von Ihnen definierten Stile (*siehe Schritt 3 unten*) auf alle 16 Textkomponenten angewendet. Wenn Sie vier dieser Textkomponenten andere Stile zuweisen möchten, so müssen Sie jede einzelne dieser Komponenten separat auswählen und mit Stilen versehen. Wenn zwei Komponenten desselben Komponententyps unterschiedliche Stile haben und beide in der Design-Ansicht ausgewählt sind, so werden die Stile beider Instanzen im Bereich *Stildefinitionen* angezeigt. So sehen Sie z.B. in der Abbildung oben, dass eine automatische Berechnung (XPath) eine normale Schriftstärke (font-weight) hat, während die andere fett (bold) formatiert ist. Wenn in der Eintrag *2-XPath* in der Spalte *Stile für* ausgewählt ist, so werden beide Schriftstärken angezeigt.
3. Nachdem Sie den Komponententyp, für den Stile definiert werden sollen, im Bereich *Stile für* ausgewählt haben, werden die Stile für diese Auswahl in der Spalte für die Stildefinitionen[47] definiert. Eine Anleitung dazu finden Sie im Abschnitt Definieren von Stilwerten[351].

 © 2019-2025 Altova GmbH

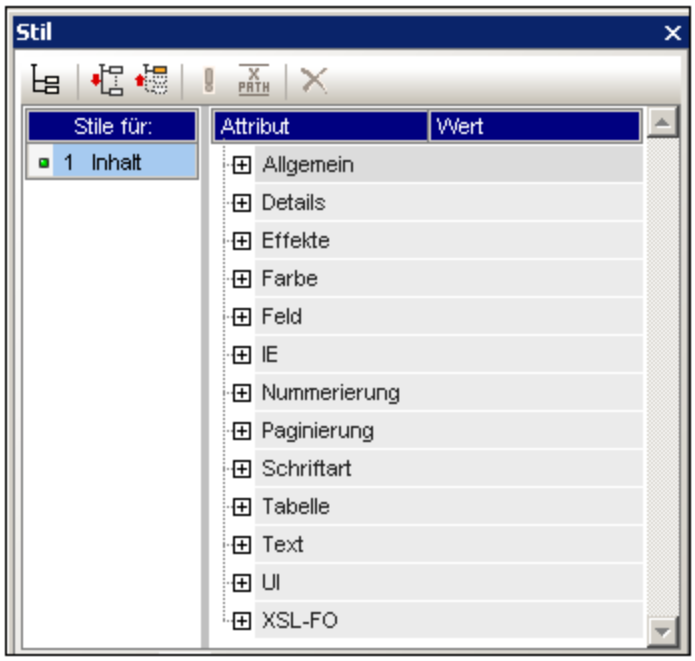
8.4.4 Definieren von Stilwerten

Die Stileigenschaften für den in der Spalte *Stile für* ausgewählten Komponententyp werden in der Seitenleiste "Stil" in der Spalte für die Stildefinitionen[47] (*Abbildung unten*) definiert. Sie können in der Spalte *Stile für* bei Bedarf auch mehr als einen Komponententyp auswählen. Wählen Sie dabei die zusätzlichen Komponententypen aus, während Sie die **Strg**-Taste gedrückt halten oder wählen Sie einen Komponentenbereich in der Spalte *Stile für* aus, wobei Sie die **Umschalttaste** gedrückt halten. Wenn mehrere Komponententypen ausgewählt sind, wird jeder im Bereich *Stildefinitionen* definierte Stilwert auf alle Instanzen der auswählten Komponententypen angewendet.



Stileigenschaftsgruppen

Die in der Spalte *Stildefinitionen* verfügbaren Eigenschaften sind in Gruppen gegliedert (*siehe Abbildung unten*).



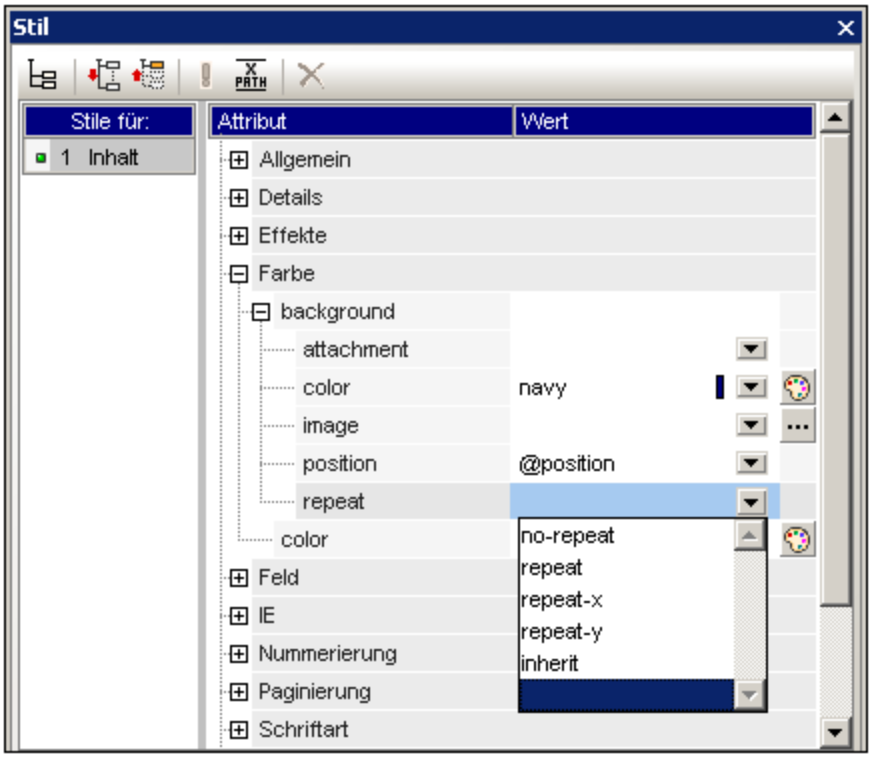
Die Anzeige der Eigenschaften kann mit Hilfe der Symbolleisten-Schaltflächen **Nicht leere auflisten** 47, 47 **Alle erweitern** 47 und 47 **Alle reduzieren** 47 konfiguriert werden. Die einzelnen Eigenschaftsgruppen können erweitert werden, um Zugriff auf die Untergruppen und Stileigenschaften zu erhalten (*siehe Abbildung unten*).

Eingabe von Stilwerten

Werte für Stileigenschaften (oder kurz: Stilwerte) können auf die folgenden Arten eingegeben werden (siehe Abbildung unten):

- durch Eingabe direkt in die Wertespalte. Wählen Sie dazu die Eigenschaft aus, doppelklicken Sie in ihre Wertespalte, geben Sie den Wert über die Tastatur ein und drücken Sie die **Eingabetaste** oder klicken Sie an eine beliebige Stelle auf der Benutzeroberfläche.
- durch Auswahl eines Werts aus der Dropdown-Liste der Auswahlliste für diese Eigenschaft. Klicken Sie in der Auswahlliste auf den Abwärtspfeil, um die Liste der Stilwerte zu sehen. In der Abbildung unten werden die Optionen für die Eigenschaft `(background-)repeat` angezeigt. Wählen Sie den gewünschten Wert aus der Dropdown-Liste aus.
- durch Verwendung der Schaltfläche rechts von der Wertespalte für diese Stileigenschaft. Es stehen zwei Arten von Schaltflächen zur Verfügung und zwar nur für Eigenschaften, für die diese relevant sind: (i) eine Farbpalette zur Auswahl von Farben (in der Abbildung unten siehe die Eigenschaft `(background-)color`), und (ii) ein Dialogfeld zur Dateisuche (in der Abbildung unten siehe die Eigenschaft `(background-)image`).

© 2019-2025 Altova GmbH



- Werte für Stile können auch mit Hilfe eines XPath-Ausdrucks[353] zugewiesen werden.

Ändern oder Löschen eines Stilwerts

Wenn ein falscher oder ungültiger Stilwert eingegeben wird, wird er rot angezeigt. Um den Wert zu ändern, verwenden Sie eine der im vorherigen Abschnitt Eingabe von Eigenschaftswerten[352] beschriebenen Methoden.

Um einen Stilwert zu löschen (oder anders ausgedrückt um einen Stilwert zurückzusetzen), klicken Sie in der Symbolleiste der Seitenleiste "Stil" auf die Schaltfläche "Zurücksetzen". Alternativ dazu können Sie in die Wertespalte der Eigenschaft doppelklicken, den Wert mit der **Entf-** und/oder der **Rücktaste** löschen und anschließend die **Eingabetaste** drücken.

8.4.5 Stileigenschaften über XPath

Stile können Design-Komponenten mittels XPath-Ausdrücken zugewiesen werden. Auf diese Art können Stileigenschaftswerte aus XML-Daten oder dem XPath-Ausdruck selbst übernommen werden. Auch mit Hilfe der `doc()` Funktion von XPath 2.0/3.0 können Nodes in jedem XML-Dokument, auf das Zugriff besteht, adressiert werden. Es können nicht nur Stildefinitionen aus XML-Daten abgerufen werden; dank dieser Funktion können auch Stile auf Basis von Bedingungen gewählt werden, die von der Struktur oder dem Inhalt der XML-Daten abhängen. So können z.B. mit Hilfe der `if...else` Bedingung von XPath 2.0/3.0 je nach Position eines Elements in einer Sequenz zwei unterschiedliche Hintergrundfarben ausgewählt werden. Auf diese Art können bei Darstellung dieser Elemente als Zeilen einer Tabelle die ungerade nummerierten Zeilen in einer Hintergrundfarbe dargestellt werden, während die gerade nummerierten Zeilen in einer anderen Farbe dargestellt werden (*siehe Beispiel unten*). Die Darstellung kann auch je nach Inhalt eines Node variiert werden.

Stileigenschaften, für die XPath-Ausdrücke verwendet werden können

XPath-Ausdrücke können für die folgenden Stileigenschaften eingegeben werden:

- Alle Eigenschaften in der Seitenleiste "Stil"
- die Eigenschaftsgruppen *Allgemein*, *Ereignis* und *HTML* in der Seitenleiste "Eigenschaften"

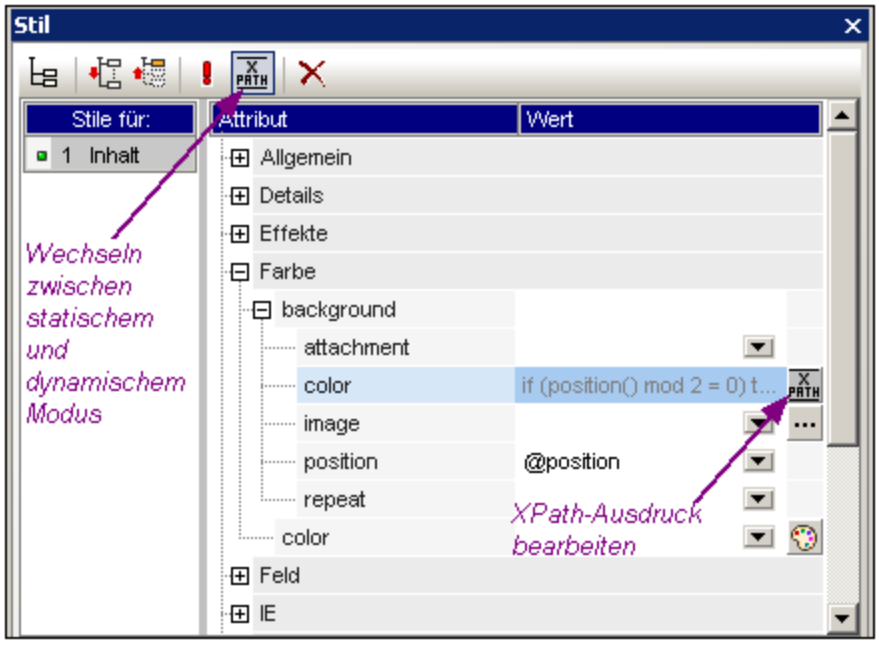
Statischer Modus und dynamischer (XPath)-Modus bei Eigenschaftswerten

Bei den Eigenschaften, bei denen XPath-Ausdrücke aktiviert sind[353], stehen zwei Modi zur Verfügung:

- Statischer Modus, wobei der Eigenschaftswert direkt in die Wertespalte der Seitenleiste eingegeben wird. So kann z.B. bei der Hintergrundfarbe einer Design-Komponente der Wert `red` direkt in die Seitenleiste eingegeben werden.
- Dynamischer oder XPath-Modus, wobei ein XPath-Ausdruck eingegeben wird. Der Ausdruck wird zur Laufzeit ausgewertet und das Ergebnis wird als Wert der Eigenschaft eingegeben. So kann z.B. für die Hintergrundfarbe einer Design-Komponente der folgende XPath-Ausdruck eingegeben werden: `/root/colors/color1`. Der Inhalt des Node `/root/colors/color1` wird zur Laufzeit abgerufen und als Wert der Eigenschaft "background-color" eingegeben.

Wechseln zwischen dem statischen und dem dynamischen (XPath)-Modus

Bei jeder Eigenschaft, für die XPath-Ausdrücke aktiviert sind, ist standardmäßig der statische Modus ausgewählt. Um bei einer Eigenschaft in den dynamischen (XPath) Modus zu wechseln, wählen Sie die Eigenschaft aus und klicken Sie in der Symbolleiste der Seitenleiste auf das XPath-Symbol (*Abbildung unten*).



Wenn für diese Eigenschaft ein statischer Wert vorhanden war, wird dieser nun gelöscht und der Modus wird auf dynamisch umgestellt. Das Dialogfeld "XPath-Ausdruck bearbeiten"[440] wird angezeigt. In diesem Dialogfeld können Sie den XPath-Ausdruck für die Eigenschaft eingeben. Klicken Sie anschließend auf **OK**.

© 2019-2025 Altova GmbH

Nach Eingabe eines XPath-Ausdrucks für die Eigenschaft wird in der Wertespalte für diese Eigenschaft eine "XPath-Ausdruck bearbeiten" Schaltfläche angezeigt. Klicken Sie auf diese Schaltfläche um den XPath-Ausdruck anschließend zu bearbeiten. Wenn Sie wieder in den statischen Modus zurückwechseln möchten, klicken Sie auf das XPath-Symbol in der Symbolleiste. Daraufhin wird der XPath-Ausdruck gelöscht und die Eigenschaft wechselt in den statischen Modus zurück.

Anmerkung: Beachten Sie bitte zwei wichtige Punkte. 1.) Es ist immer jeweils nur ein Modus (statisch oder dynamisch) und der Wert/Ausdruck für diesem Modus aktiv. Alle Werte/Ausdrücke, die vorher im anderen Modus vorhanden waren, werden gelöscht; wenn Sie daher in den anderen Modus wechseln, wird in diesem Modus ein leeres Eingabefeld angezeigt. (Um zu einem vorher verwendeten Wert/Ausdruck zurückzugehen, verwenden Sie den Befehl Rückgängig 493.) 2.) Wenn Sie eine Eigenschaft nach dem Bearbeiten des SPS erneut auswählen, wird diese Eigenschaft in dem Modus geöffnet, in dem sie sich zuvor befunden hat.

Erstellen und Bearbeiten der XPath-Definition

Die XPath-Definition wird im Dialogfeld "XPath-Ausdruck bearbeiten" 440 erstellt und bearbeitet. Dieses Dialogfeld kann auf zwei Arten aufgerufen werden:

- Jedes Mal, wenn Sie vom statischen Modus in den dynamischen Modus einer Eigenschaft wechseln (durch Klicken Auf das XPath-Symbol in der Symbolleiste der Seitenleiste), wird das Dialogfeld "XPath-Ausdruck bearbeiten" 440 angezeigt. Sie können den XPath-Ausdruck nun erstellen. (Beachten Sie: Wenn Sie auf die Symbolleisten-Schaltfläche klicken, wenn Sie sich bereits im dynamischen Modus befinden, wechseln Sie wieder in den statischen Modus; das Dialogfeld "XPath-Ausdruck bearbeiten" wird in diesem Fall nicht aufgerufen.)
- Das Dialogfeld "XPath-Ausdruck bearbeiten" 440 wird auch angezeigt, wenn Sie im Wertefeld einer Eigenschaft, für die bereits ein XPath-Ausdruck definiert wurde, auf die Schaltfläche "XPath-Ausdruck bearbeiten" klicken. Das Dialogfeld enthält den bereits für diese Eigenschaft definierten XPath-Ausdruck, den Sie nun bearbeiten können.

Nachdem Sie den XPath-Ausdruck in das Eingabefeld eingegeben haben, klicken Sie zum Fertigstellen auf **OK.**

Von einem XPath-Ausdruck zurückgegebene Werte

Die wichtigsten Vorteile der Verwendung von XPath-Ausdrücken zum Definieren eines Eigenschaftswerts sind, dass: (i) der Eigenschaftswert aus einer XML-Datei übernommen werden kann (anstatt direkt eingegeben werden zu müssen); und/oder (ii) ein XPath-Ausdruck eine Bedingung zum Inhalt oder der Struktur des gerade verarbeiteten XML-Dokuments überprüfen kann und dann entsprechend dem Resultat einen Wert auswählen kann. XPath-Ausdrücke geben Werte in den folgenden beiden Kategorien zurück:

- *XML-Node-Inhalt*
  Der XPath-Ausdruck kann für folgende Nodes verwendet werden: (i) Nodes in dem vom SPS verarbeiteten XML-Dokument oder (ii) Nodes in jedem XML-Dokument, auf das Zugriff besteht. So würde z.B. mit dem Ausdruck `Format/@color` das `color`-Attribut des Child-Elements `Format` des Kontext-Node aufgerufen. Der Wert des `color`-Attributs wird als der Wert der Eigenschaft definiert, für die der XPath-Ausdruck definiert wurde. Ein Node in einem anderen XML-Dokument kann mit Hilfe der `doc()` Funktion von XPath 2.0 aufgerufen werden. So würde z.B. mit dem Ausdruck `doc('Styles.xml')//colors/color-3` der Wert des Elements `color-3` im XML-Dokument `Styles.xml` abgerufen und als Wert der Eigenschaft definiert, für die der XPath-Ausdruck definiert wurde.

- *XPath-Ausdruck*

Der Wert der Eigenschaft kann aus dem XPath-Ausdruck selbst stammen und nicht aus dem XML-Dokument. So können Sie z.B. für ein Element, das als Zeile in einer Tabelle ausgeben wird, je nachdem, ob es sich um eine gerade oder ungerade nummerierte Zeile handelt, unterschiedliche Hintergrundfarben definieren. Dazu könnten Sie den folgenden XPath 2.0/3.0-Ausdruck verwenden: `if (position() mod 2 = 0) then 'red' else 'green'`. Beachten Sie, dass der Rückgabewert dieses Ausdrucks entweder der String `red` oder der String `green` ist. Dieser Wert wird auch als der Wert der Eigenschaft definiert, für die der XPath-Ausdruck definiert wurde. Im gerade erwähnten Beispiel wurden die Eigenschaftswerte als String Literale eingegeben. Alternativ dazu könnten Sie aus einem XML-Dokument kommen, z.B.: `if (position() mod 2 = 0) then doc('Styles.xml')//colors/color-1 else doc('Styles.xml')//colors/color-2`. Umgekehrt könnte es sich beim XPath-Ausdruck auch um einen ganz einfachen String wie z.B. `'green'`handeln. Dies ist dasselbe, wie wenn der statische Wert `green` für die Eigenschaft eingegeben wird.

8.4.6 Kombistile

Bei einem Kombistil handelt es sich um eine Gruppe von CSS-Textstileigenschaften für Textstile, die mit einem Attribut eines XML-Instanzdokument-Nodes verknüpft sind. Zusätzlich dazu wird auch jede im Stylesheet gespeicherte Gruppe von CSS-Textstileigenschaften als Kombistil bezeichnet. Kombistile können für die folgenden Designkomponenten definiert werden:

- automatische Berechnungen 257
- den Platzhalter (Inhalt) 111
- Absatz (Block)-Design-Elemente 113
- Tabellenzellen 126

Vorteile von Kombistilen

Kombistile bieten die folgenden Vorteile:

- Die Stileigenschaften befinden sich in den XML-Daten und können daher vom Benutzer geändert werden.
- Bei den Stileigenschaften der oben aufgelisteten Designkomponenten kann es sich um eine Kombination aus in den XML-Daten gespeicherten Eigenschaften und im SPS zugewiesenen Eigenschaften handeln.
- Sie können beim Erstellen des SPS-Designs schnell zwischen den zahlreichen mit einem Element verknüpften Kombistilen wechseln.

Eingabe des Kombistils in das XML-Attribut

Ein Kombistil (der aus mehreren Stileigenschaften besteht) wird als Attributwert eines Elements in das XML-Quelldokument eingegeben. So enthält z.B. das Attribut `desc-style` in unten gezeigten XML-Quelldokument einen Standard-Kombistil:

```
<Desc desc-style="font-family:Verdana; font-size:12pt; color:blue">
```

Sie können auch mehr als einen Kombistil für ein Element definieren. In diesem Fall muss jeder Kombistil in ein separates Attribut eingegeben werden:

```
<Desc styleBlue="font-family:Verdana; font-size:12pt; color:blue"
      styleRed ="font-family:Verdana; font-size:12pt; color:red">
```


© 2019-2025 Altova GmbH

Wenn für ein Element mehrere Kombistile zur Verfügung stehen, können Sie beim Definieren eines Werts für die Eigenschaft *Kombistil* einer Design-Komponente zwischen Kombistilen wechseln (*siehe unten*).

Anmerkung: Die Attribute, mit denen die Kombistile aufgerufen werden, müssen im Quellschema definiert sein, damit das XML-Dokument gültig ist.

Unterstützte CSS-Textstileigenschaften

Die folgenden CSS-Stile können in Kombistilen verwendet werden:

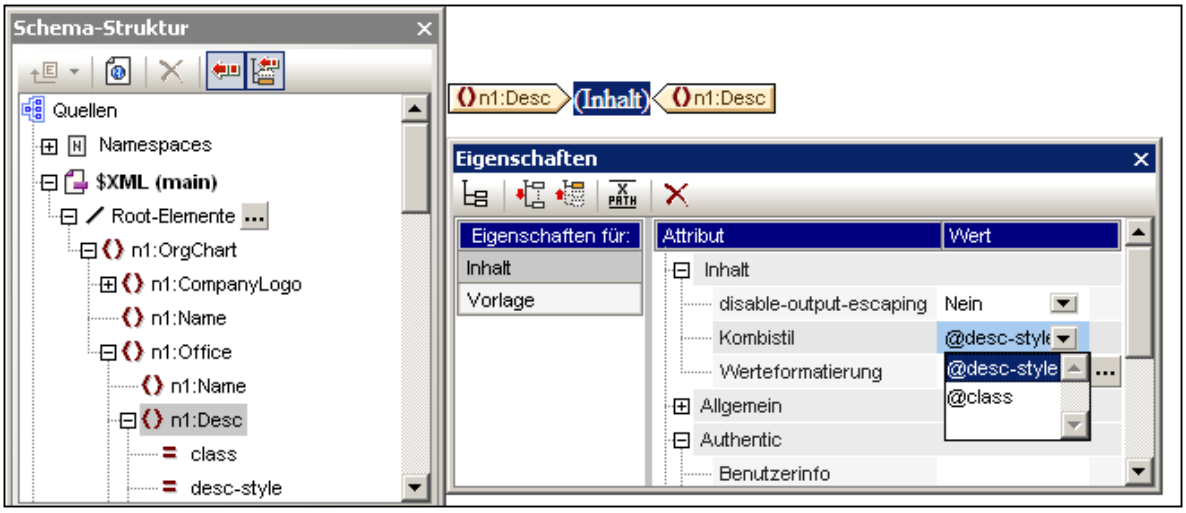
`font-family`	`font-size`	`font-weight`	`font-style`
`color`	`background-color`	`text-align`	`text-decoration`

Definieren eines Attributs als Wert für einen Kombistil

Wenn Sie den Kombistil einer Design-Komponente als Attribut definieren, kann der Benutzer der Authentic-Ansicht diesen Kombistil bearbeiten.

Um ein Attribut als Kombistil einer Design-Komponente zu definieren, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Design-Ansicht die Design-Komponente aus, der Sie ein Attribut als Kombistil zuweisen möchten. In der Abbildung unten wurde der Platzhalter `(Inhalt)` des Elements `Desc` ausgewählt.



2. In der Auswahlliste der Eigenschaft `Kombistile` der Komponente *Inhalt* (*siehe Seitenleiste "Eigenschaften" rechts unten in der obigen Abbildung*) werden die Attribute des Inhalt-Elements angezeigt. Wählen Sie das Attribut aus, das als Kombistil der Design-Komponente definiert werden soll.

Definieren eines XPath-Ausdrucks als Kombistilwert

Sie können als Wert der Eigenschaft *Kombistil* auch einen XPath-Ausdruck eingeben. In diesem Fall kann der Benutzer den Kombistil jedoch in der Authentic-Ansicht nicht bearbeiten, da der Kombistil im SPS (und nicht im XML-Quelldokument) gespeichert ist.

Um einen XPath-Ausdruck als Wert der Eigenschaft *Kombistil* zu definieren, klicken Sie in der Symbolleiste der Seitenleiste "Eigenschaften" auf die Schaltfläche **XPath** und geben Sie den XPath-Ausdruck anschließend in das Dialogfeld "XPath" ein, das daraufhin angezeigt wird. Der XPath-Ausdruck wird als Attributwertvorlage ausgewertet; der zurückgegebene Wert ist der Wert eines `style` Attributs (und seines Äquivalents in Nicht-HTML-Ausgabeformaten).

Nehmen wir dazu als Beispiel den folgenden am Platzhalter `(Inhalt)` des Elements `n1:Person` erstellten XPath-Ausdruck.

```
if (number(n1:Shares) gt 1000) then 'color:red' else 'color:green'
```

Das Ergebnis dieses Ausdrucks ist das folgende: Wenn das Element `n1:Person` ein Subelement `n1:Shares` mit einer Zahl hat, die größer als 1000 ist, so wird der Inhalt des Elements `n1:Person` rot in der Ausgabe dargestellt. Andernfalls werden alle `n1:Person` Elemente in der Ausgabe grün dargestellt. Der vom XPath-Ausdruck zurückgegebene Wert wird als Wert des HTML `style` Attributs (und seines Äquivalents in Nicht-HTML-Ausgabeformaten) an das Ausgabedokument übergeben.

In dem anhand des SPS generierten XSLT Stylesheet wird dieser XPath-Ausdruck etwa folgendermaßen als Attributwertvorlage ausgewertet:

```
<span style="{if (number(n1:Shares) gt 1000) then &apos;color:red&apos; else
&apos;color:green&apos;}">
```

In der HTML-Ausgae würde je nachdem, welche Bedingung ausgewertet wurde, eine der folgenden Zeilen stehen:

```
<span style="color:red">
```

oder

```
<span style="color:green">
```

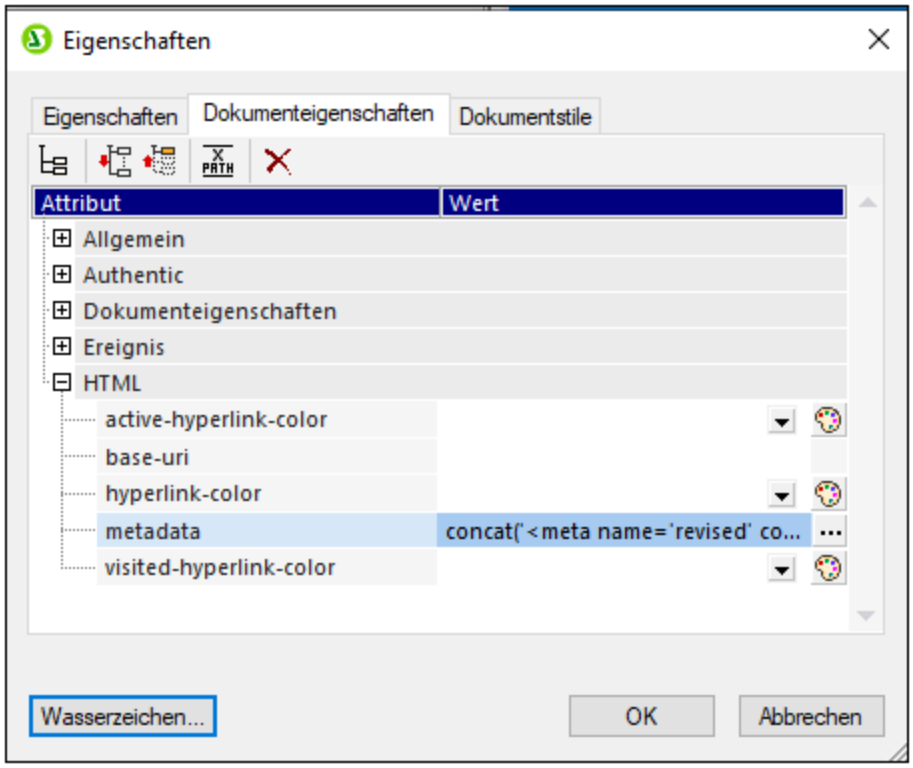
Anmerkung: Attributwertvorlagen sind XSLT-Konstrukte, mit Hilfe derer der Wert eines Attributs als XPath-Ausdruck gelesen werden kann. Sie stehen innerhalb von geschwungenen Klammern und ermöglichen eine dynamische Zuweisung des Attributwerts.

© 2019-2025 Altova GmbH

8.5 HTML-Dokumenteigenschaften

Die Eigenschaften des HTML-Ausgabedokuments können entweder über das Register *Dokumenteigenschaften* des Dialogfelds "Eigenschaften" des Dokumentanfangsabschnitts oder, wenn im Design-Struktur-Fenster die Hauptvorlage[41] ausgewählt ist, im Bereich "Eigenschaften" bearbeitet werden.

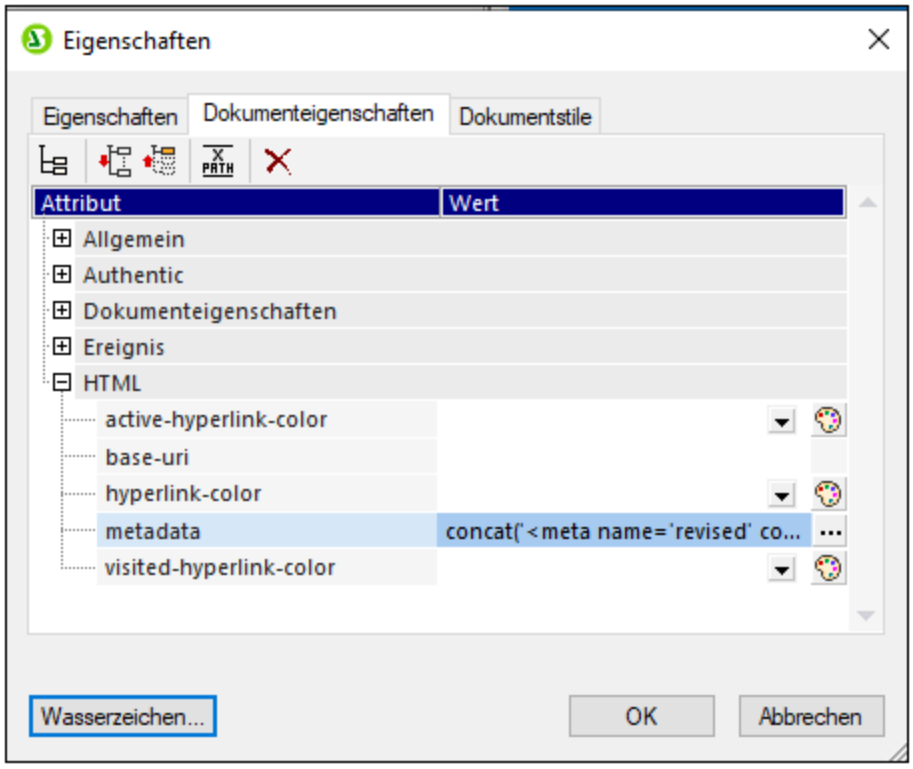
Über die Eigenschaften des Dokumentanfangsabschnitts
Klicken Sie in der Titelleiste des Dokumentanfangsabschnitts auf den Hyperlink *Eigenschaften bearbeiten* und anschließend auf das Register *Dokumenteigenschaften*.



Hier können Sie verschiedene Eigenschaften des zu generierenden HTML-Ausgabedokuments definieren. Beachten Sie die folgenden Punkte:

- Es handelt sich hierbei um HTML-Eigenschaften, die auf Dokumentebene angewendet werden, z.B. die Eigenschaften **class** und **id** des Eigenschaftsabschnitts *Allgemein*.
- Die Abschnitte *Dokumenteigenschaften* und *HTML* enthalte generische Eigenschaften im Zusammenhang mit dem HTML-Dokument als Ganzes, wie z.B. Link-Farben und Informationen, die in die **meta**-Tags des HTML-Dokuments gesetzt werden.
- Über die Eigenschaft **metadata** des Abschnitts "HTML" können Sie beliebigen Text für das **HEAD**-Element des HTML-Dokuments definieren (*siehe Abbildungen oben und unten, aus der Enterprise Edition von StyleVision, die für diese Edition spezifische Funktionalitäten enthalten*). Als Text könnten Sie z.B. ein oder mehrere **script**- oder **meta**-Elemente eingeben. Sie können diese HTML-Elemente entweder direkt als Text (ohne Anführungszeichen) oder, wie in den Abbildungen, als XPath-Ausdrücke eingeben. Im Beispiel in der Abbildung wird im XPath-Ausdruck ein Meta-Tag für das Überarbeitungsdatum eines Dokuments definiert.

Über die Eigenschaften der Hauptvorlage
Wählen Sie in der Seitenleiste Design-Struktur[41] die Hauptvorlage aus. Sie können nun in der Seitenleiste "Eigenschaften" (*Abbildung unten*) die Eigenschaften des HTML-Ausgabedokuments definieren.



Die Eigenschaften in diesem Dialogfeld sind genau dieselben, die auch auf dem oben beschriebenen Register *Dokumenteigenschaften* definiert werden können.

© 2019-2025 Altova GmbH

9 Zusätzliche Funktionen

Zusätzlich zu den Funktionen zur Bearbeitung[110], Gliederung[184] und Darstellung[327] von Inhalt sowie den komplexen SPS-Funktionen[256] bietet StyleVision eine Reihe von weiteren nützlichen Funktionen. Im Folgenden sehen Sie eine Liste dieser Funktionen, die in den Unterabschnitten dieses Abschnitts genauer beschrieben sind.

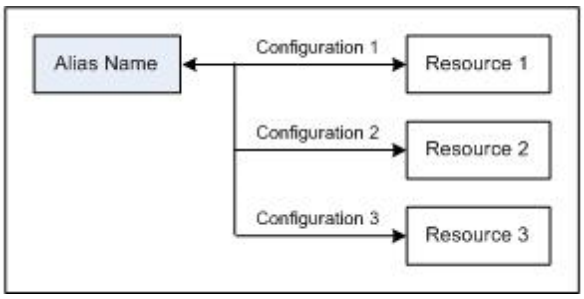
- Globale Ressourcen[362]. Globale Ressourcen bieten Flexibilität beim Auswählen der Ressourcen. So können z.B. einem Alias mehrere Ressourcen (wie z.B. Dateien) zugewiesen werden. Wenn ein Alias als Quelle (XML, XSD, usw.) eines SPS verwendet wird, können Sie zwischen den verschiedenen Ressourcen, die dem Alias zugewiesen wurden, wechseln.
- Arbeiten mit Datumsangaben[399].Datumsangaben können bedarfsgemäß bearbeitet und formatiert werden.
- URIs von ungeparsten Entities[377]. URIs können in der DTD, auf der ein XML-Dokument basiert, in ungeparsten Entities gespeichert werden. Dank der Funktion zur Angabe von URIs für ungeparste Entities können diese URIs bei Bildern und Hyperlinks als Ziel-URIs verwendet werden.
- Verwendung von Skripts[403]. StyleVision enthält einen JavaScript Editor, in dem JavaScript-Funktionen definiert werden können. Diese Funktionen stehen anschließend an beliebigen Stellen im SPS als Event Handler zur Verfügung und führen Ihre Funktion im HTML-Ausgabedokument aus.
- HTML-Import[408]. Ein HTML-Dokument kann in StyleVision importiert werden und anhand dieser Datei kann eine XML-, eine XSD-i und eine SPS-Datei erstellt werden.
- Neu von XSLT-Datei[379]: Eine SPS-Datei kann anhand einer XSLT-für-HTML- oder einer XSLT-für-FO-Datei erstellt werden. Die Vorlagenstruktur und die Stile der XSLT-Datei werden im SPS erstellt. Sie können die SPS-Komponenten anschließend ändern und Inhalt sowie Formatierungen zur SPS-Datei hinzufügen.

Siehe auch

- Seitenleiste "Eigenschaften"[49]

9.1 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. So können z.B. bei Zuweisung eines XSLT Stylesheet zur Transformationen eines XML-Dokuments über eine globale Ressource (einen Alias) mehrere Konfigurationen für die globale Ressource definiert werden, von denen jede auf eine andere XSLT-Datei verweist. Nachdem Sie die globale Ressource auf diese Art definiert haben, wird beim Wechseln der Konfiguration die für die Transformationen verwendete XSLT-Datei gewechselt.



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

- Definieren globaler Ressourcen 362: 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 372: In StyleVision können Dateien über eine globale Ressource anstatt über einen Dateipfad zugewiesen werden. Der Vorteil ist, dass die verwendete Ressource durch Wechseln der Konfiguration in StyleVision 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, MobileTogether Designer und DatabaseSpy..

9.1.1 Definieren globaler Ressourcen

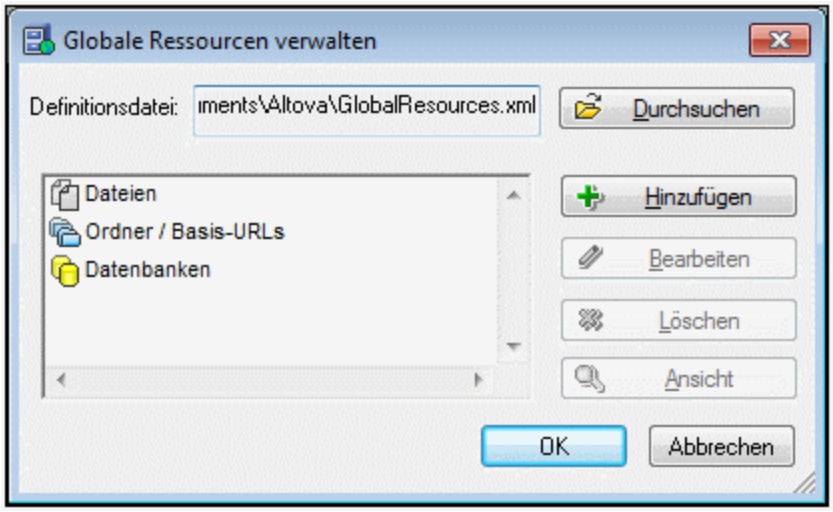
Globale Altova-Ressourcen werden im Dialogfeld **Globale Ressourcen verwalten** definiert, welches auf zwei Arten aufgerufen werden kann:

- 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*)

© 2019-2025 Altova GmbH

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:\Benutzer\<Benutzername>\Eigene Dokumente\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[365], Ordnern[370] und Datenbanken[362] 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.

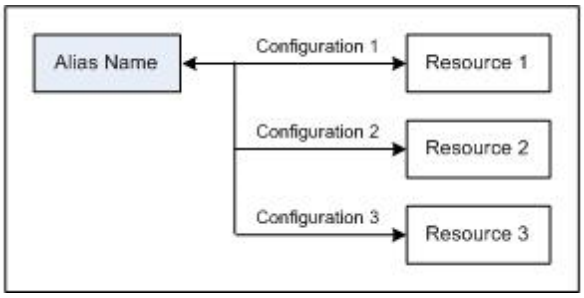
Um eine ***globale Ressource zu bearbeiten***, wählen Sie sie aus und klicken Sie auf **Bearbeiten**. Daraufhin wird das entsprechende Dialogfeld **Globale Ressource** angezeigt, in dem Sie die nötigen Änderungen vornehmen können (*siehe dazu die Beschreibungen in den Unterabschnitten Dateien[365], Ordner[370] 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*).



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

 © 2019-2025 Altova GmbH

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.

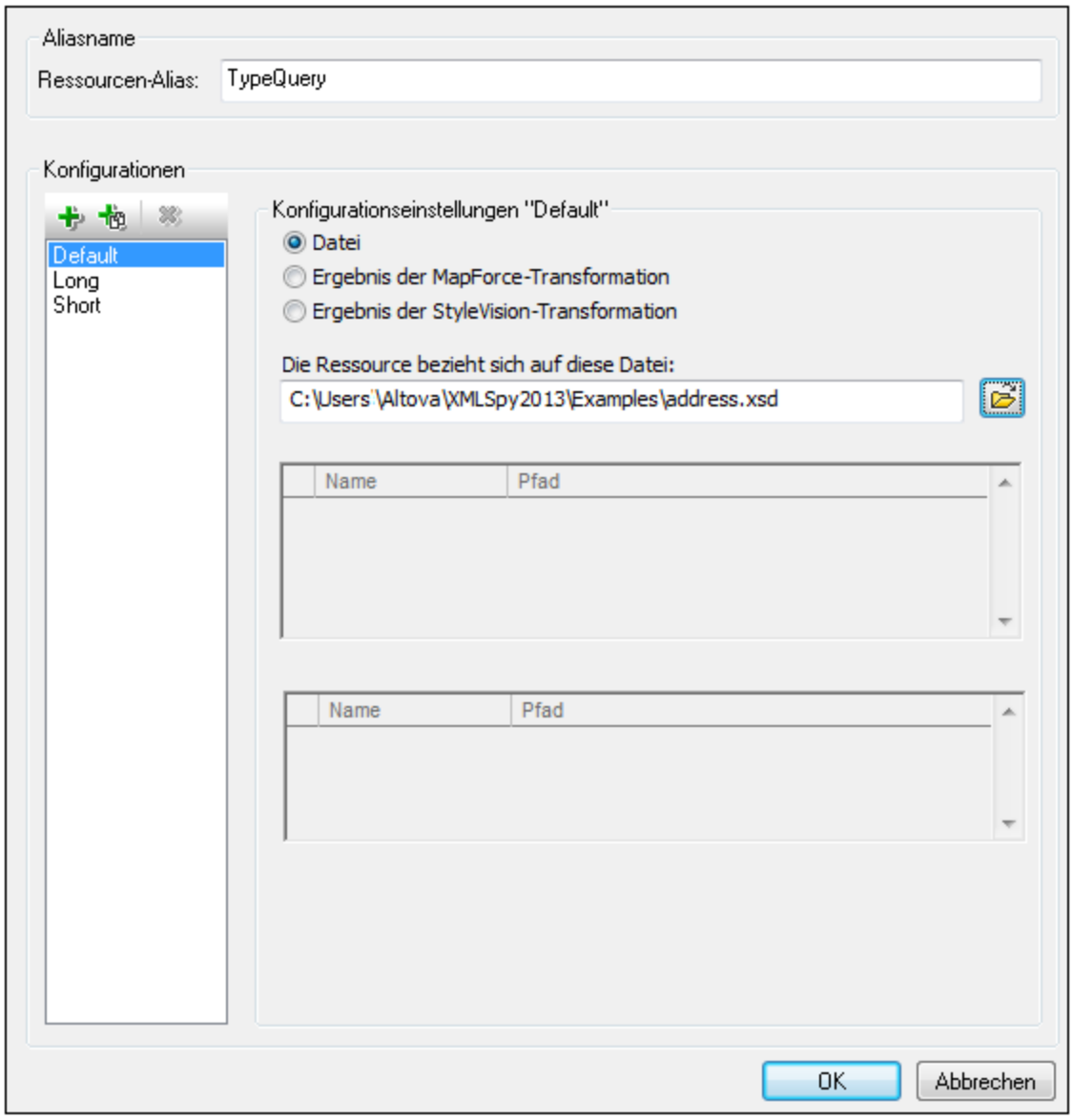
9.1.1.1 Dateien

Das Dialogfeld **Globale Ressource** für Dateien (*Abbildung unten*) wird über den Befehl **Hinzufügen | Datei** im Dialogfeld "Globale Ressourcen verwalten" 362 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") 362 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.

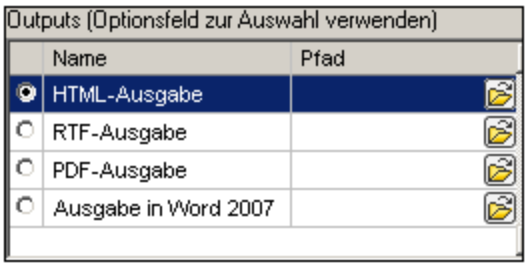
Öffnen: Damit können Sie zur Datei navigieren, die als globale Ressource erstellt werden soll.

© 2019-2025 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.

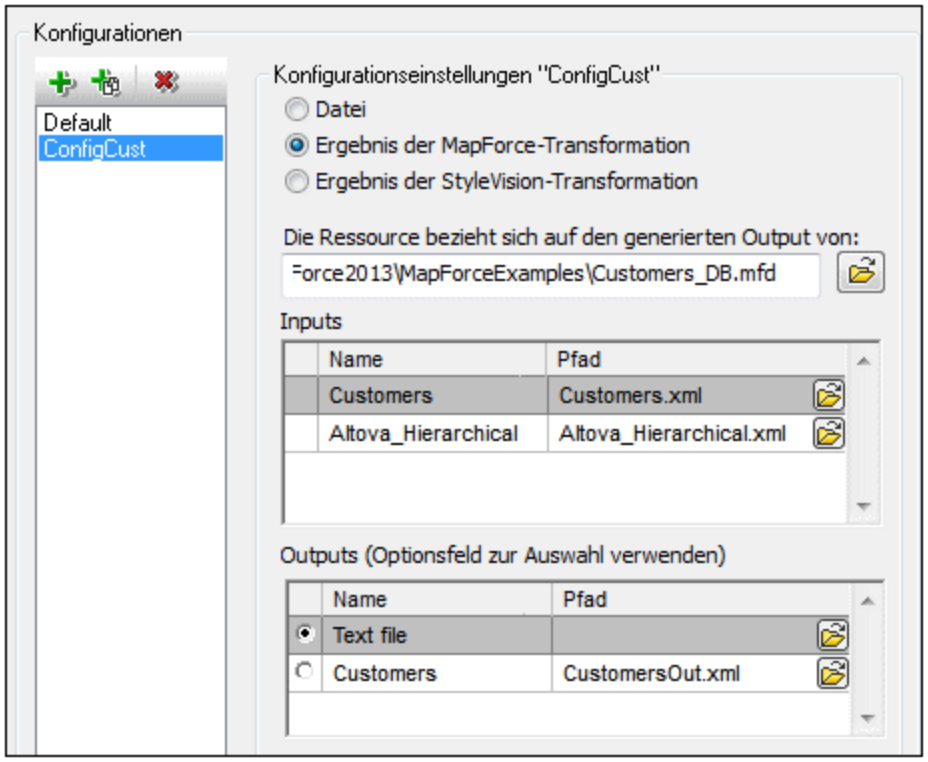
5. *Definieren Sie, falls nötig, mehrere Konfigurationen:* Sie können mehrere Konfigurationen hinzufügen und für jede davon eine Ressource definieren. Wiederholen Sie dazu für jede Konfiguration die Schritte 3 und 4. Sie können jederzeit eine neue Konfiguration zur Aliasdefinition hinzufügen.

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" 362 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-

© 2019-2025 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.

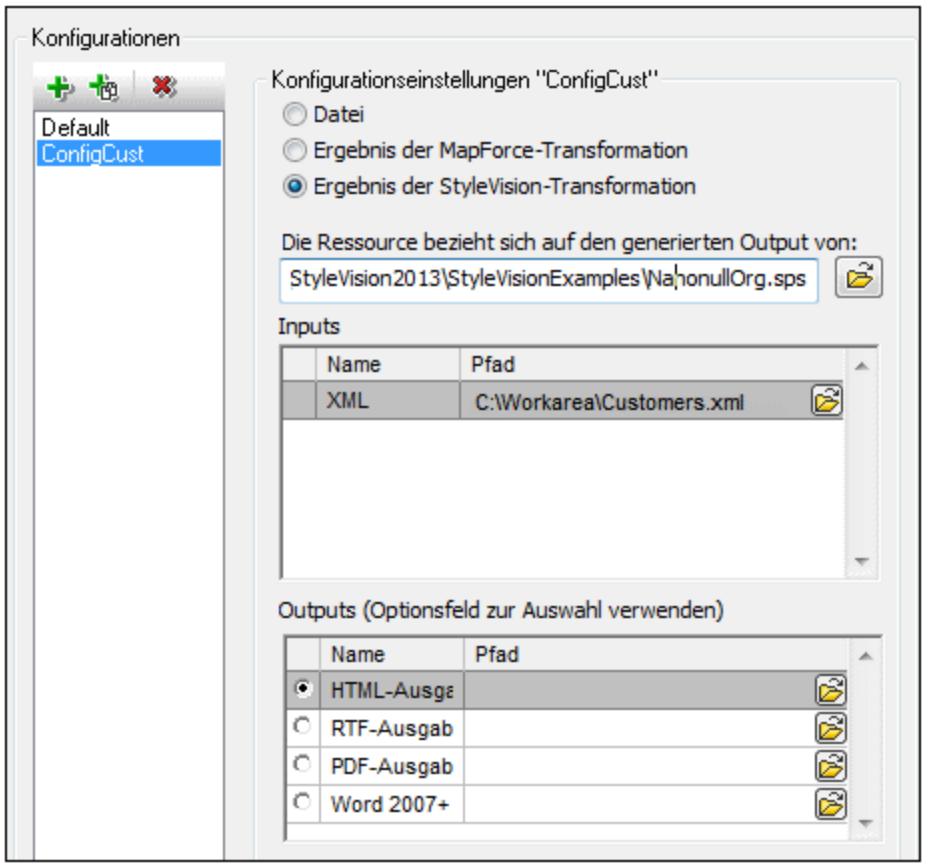
Anmerkung: Der Vorteil dieser Option (Ergebnis der MapForce-Transformation) ist, dass die Transformation zum Zeitpunkt des Aufrufs der globalen Ressource durchgeführt wird. Dadurch enthält die globale Ressource die aktuellsten Daten (aus der/den Input-Datei(en).

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.

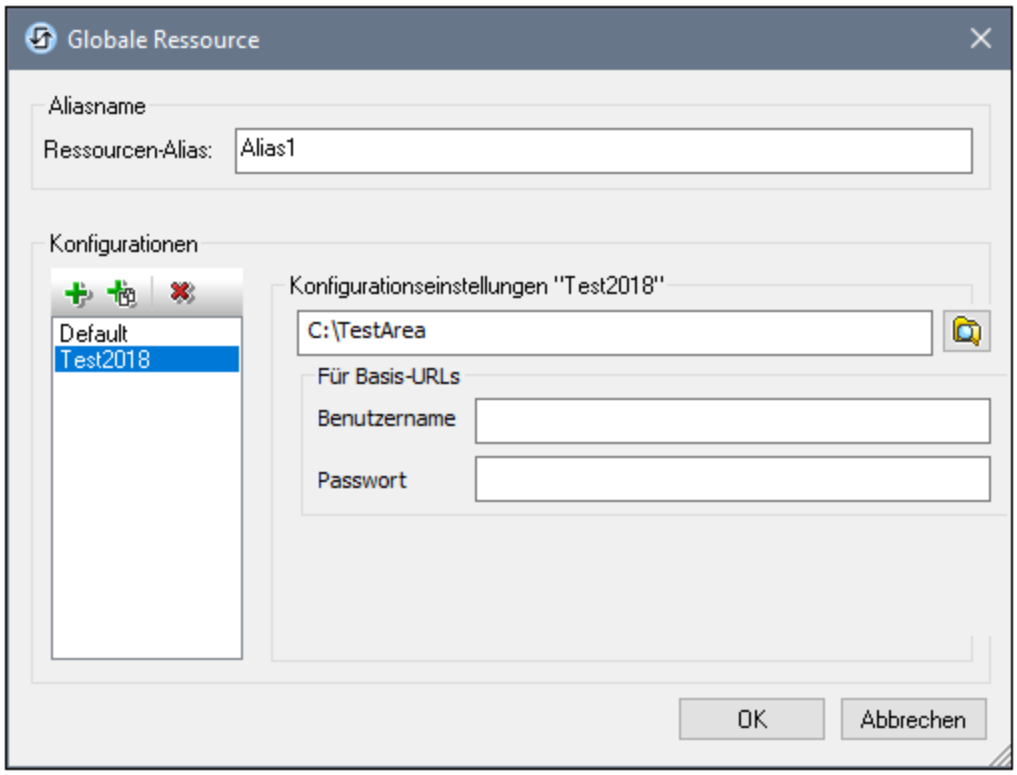
Anmerkung: Der Vorteil dieser Option (Ergebnis der StyleVision-Transformation) ist, dass die Transformation zum Zeitpunkt des Aufrufs der globalen Ressource durchgeführt wird. Dadurch enthält die globale Ressource die aktuellsten Daten (aus der/den Input-Datei(en).

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

9.1.1.2 Ordner

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

© 2019-2025 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" 362 unter "Ordner" aufgelistet.

9.1.2 Verwendung globaler Ressourcen

Es gibt mehrere Arten von globalen Ressourcen (Datei, Ordner). Einige Szenarien, in denen Sie globale Ressourcen in StyleVision verwenden können, sind hier aufgelistet: Dateien und Ordner 372.

Auswahloptionen, die festlegen, welche Ressource verwendet wird

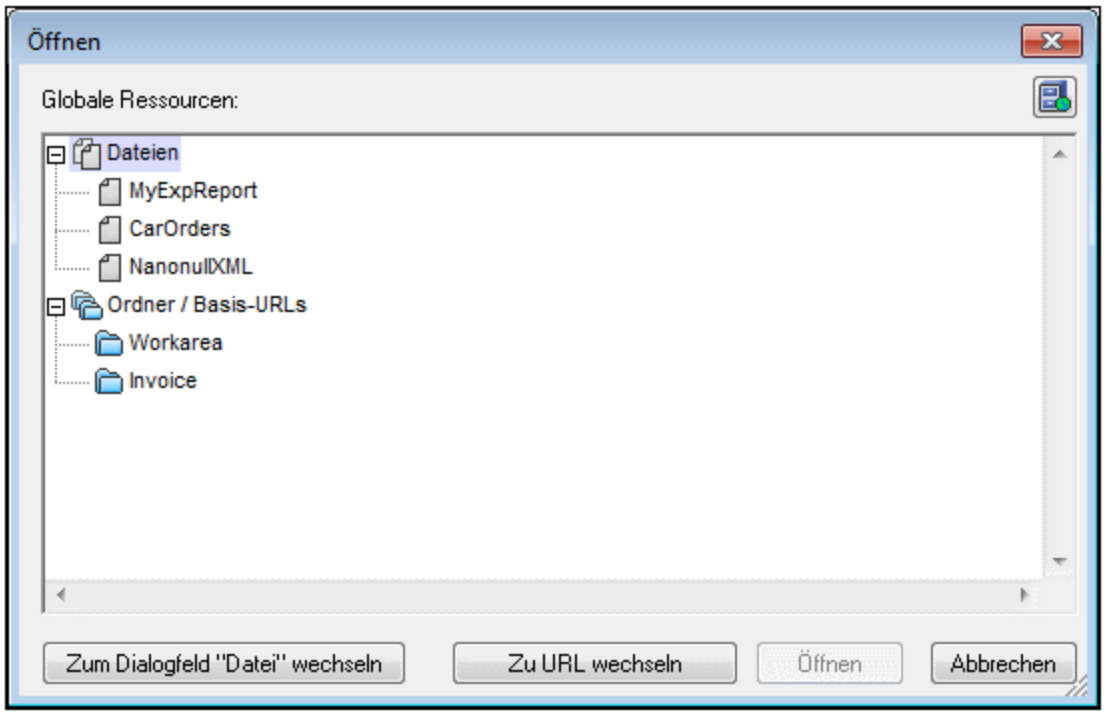
Es gibt zwei applikationsweite Auswahlmöglichkeiten, um festzulegen, welche globalen Ressourcen verwendet werden können und welche globalen Ressourcen zu einem bestimmten Zeitpunkt tatsächlich verwendet werden:

- *Die aktive XML-Datei für globale Ressourcen* wird im Dialogfeld "Globale Ressource" 362 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** 545 oder über die Symbolleiste "Globale Ressourcen" 465 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.

9.1.2.1 Zuweisen von Dateien und Ordnern

Die Zuweisung für globale Ressourcen vom Typ Datei und Ordner erfolgt auf unterschiedliche Weise. In jedem der unten beschriebenen Szenarien 372 erscheint bei Auswahl des Befehls **Globale Ressourcen** das Dialogfeld "Globale Ressource öffnen" (*Abbildung unten*).

© 2019-2025 Altova GmbH



Globale Ressourcen verwalten: Ruft das Dialogfeld Globale Ressourcen verwalten[362] 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[362].

Verwendungsszenarien

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

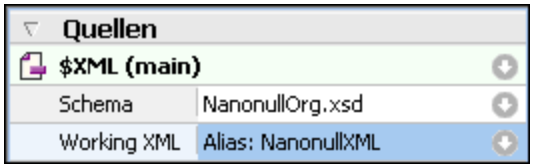
- Hinzufügen und Ändern von Schemaquellen und XML-Arbeitsdateien[372]
- Speichern als globale Ressource[374]
- Hinzufügen von Modulen und CSS-Dateien[374]
- Hinzufügen von globalen Ressourcen zu einem Projekt[375]

Schema, XML-Arbeitsdatei

In der Seitenleiste "Design-Übersicht" (*Abbildung unten*) enthält das Kontextmenü für das Schema und die XML-Arbeitsdatei einen Befehl, mit dem das Dialogfeld "Öffnen" aufgerufen wird, in dem Sie das Schema[468]

oder die XML-Arbeitsdatei[36] über eine globale Ressource zuweisen können. Wenn Sie auf den Befehl **Globale Ressourcen** klicken, wird ein Dialogfeld mit einer Liste aller globalen Ressourcen vom Typ Datei angezeigt, die in der in StyleVision gerade aktiven XML-Datei für globale Ressourcen definiert sind. (Eine Beschreibung, wie man die gerade aktive XML-Datei für globale Ressourcen festlegt, finden Sie im Abschnitt Definieren globaler Ressourcen[362].)

Wenn eine globale Ressource als Dateiquelle ausgewählt wurde, wird sie im entsprechenden Eintrag in der Seitenleiste "Design-Übersicht" angezeigt (*Abbildung unten*).



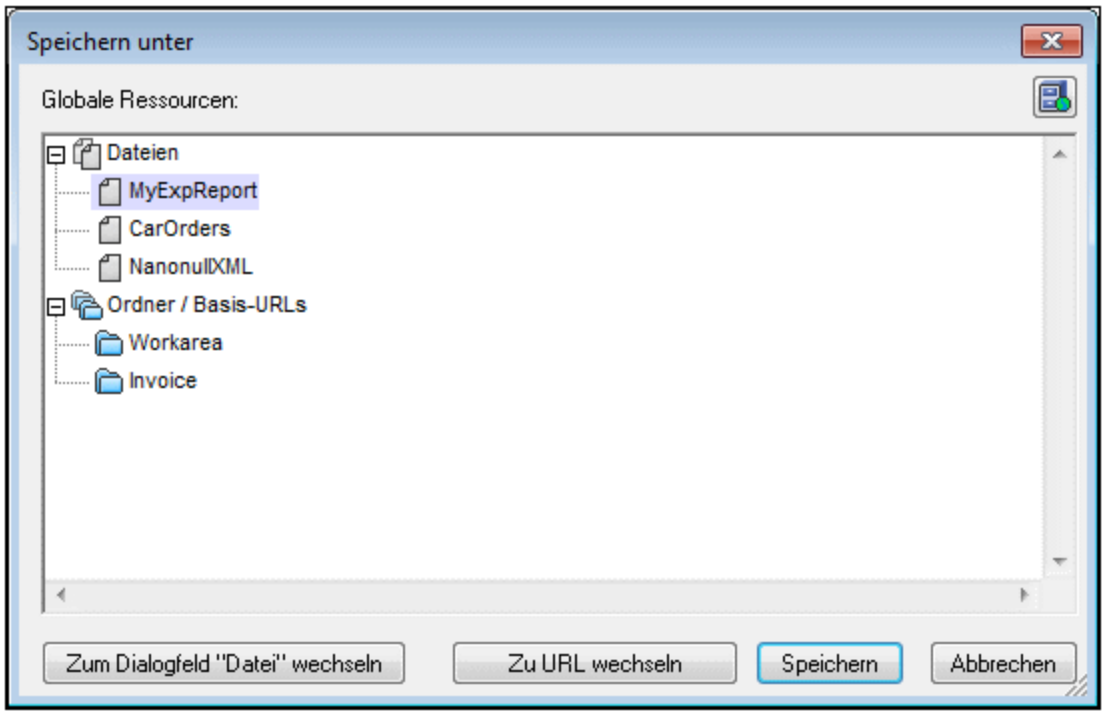
Hinzufügen von Modulen und CSS-Dateien aus einer globalen Ressource

Mit dem Befehlen **Neues Modul hinzufügen** und **Neue CSS-Datei hinzufügen** aus der Seitenleiste "Design-Übersicht"[36] wird das Dialogfeld "Öffnen" aufgerufen, in dem Sie auf "Globale Ressourcen" klicken können, um die gewünschte globale Ressource auswählen zu können. Die Module und CSS-Dateien können später durch Wechseln der Konfigurationen ausgetauscht 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 Ressourcen,** um die verfügbaren globalen Ressourcen (*Abbildung unten*) aufzurufen. Dies sind die Aliasnamen, die in der aktuellen XML-Datei für globale Ressourcen gespeichert sind.

© 2019-2025 Altova GmbH



Wählen Sie einen Alias aus und klicken Sie auf "Speichern". Wenn es sich beim Alias um einen Dateialias 365 handelt, wird die Datei direkt gespeichert. Wenn es sich um einen Ordneralias 370 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 375 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 StyleVision zu Bearbeitungs- und Validierungsfehlern kommen. Der Grund dafür ist, dass StyleVision die Datei unter der Annahme öffnet, dass die Datei den in der Konfigurationsdefinition angegebenen Dateityp aufweist.

Globale Ressourcen in Projekten

Globale Ressourcen können zum aktuell aktiven Projekt auch über den Befehl **Projekt | Globale Ressource zu Projekt hinzufügen** hinzugefügt werden. Daraufhin erscheint ein Dialogfeld, in dem die globalen Ressourcen vom Typ Datei in der gerade aktiven XML-Datei für globale Ressourcen 362 aufgelistet werden. Wählen Sie eine globale Ressource aus und klicken Sie auf **OK,** um sie zum Projekt hinzuzufügen. Die globale Ressource wird in der Projekt-Seitenleiste angezeigt und kann wie jede andere Datei verwendet werden.

9.1.2.2 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.

© 2019-2025 Altova GmbH

9.2 URIs von ungeparsten Entities

Wenn Sie eine DTD verwenden und darin eine ungeparste Entity deklariert haben, können Sie die mit dieser Entity verknüpfte URI für Bild- und Hyperlinkziele im SPS verwenden. Dies ist eine nützliche Methode, wenn Sie dieselbe URI mehrmals im SPS verwenden wollen. Diese Funktion bedient sich der XSLT-Funktion `unparsed-entity-uri`, um die URI der ungeparsten Entity von der DTD an die Ausgabe zu übergebenzur Verfügung und nicht in der Authentic-Ansicht. Beachten Sie, dass ungeparste Entity URIs für die Textausgabe nicht verarbeitet werden..

Bei Verwendung dieser Funktion müssen die DTD, das XML-Dokument und das SPS folgendermaßen geändert werden:

1. In der DTD, (i) müssen die ungeparsten Entities deklariert werden[377], wobei (i) URI und (ii) die Notation (anhand der StyleVision den Ressourcentyp der Entity ermittelt) angegeben werden müssen.
2. Im XML-Dokument muss die ungeparste Entity referenziert[377] werden, indem die Namen der erforderlichen ungeparsten Entities angegeben werden.
3. Im SPS können mit Hilfe von Entities Bilder[152] und Hyperlinks[322] referenziert werden. Dazu werden die entsprechenden dynamischen Node-Werte als ungeparste Entities korrekt aufgerufen[377].

Deklarieren und Referenzieren von ungeparsten Entities im XML-Dokumentinhalt

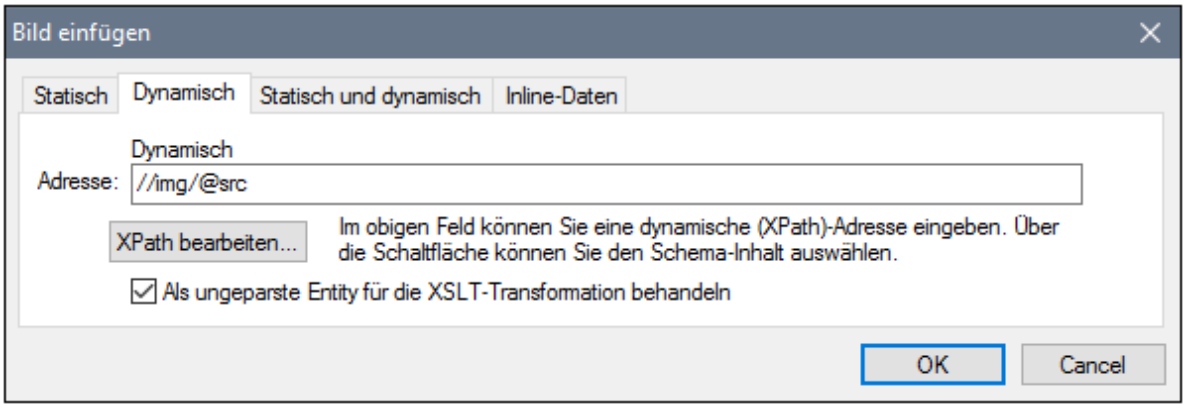
Unten sehen Sie einen Ausschnitt aus einer Liste eines XML-Dokuments. Es enthält eine interne DTD-Untermenge, in der zwei ungeparste Entities, eine mit einer `GIF` Notation (Verweis auf ein GIF-Bild) und die andere mit einer `LNK` Notation (Verweis auf einen Hyperlink), deklariert sind. Die Nodes `img/@src` und `link/@href` im XML-Code referenzieren die ungeparsten Entities durch Angabe ihrer Namen.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE document SYSTEM "UEURIDoc.dtd" [
<!ENTITY Picture SYSTEM "nanonull.gif" NDATA GIF>
<!ENTITY AltovaURI SYSTEM "http://www.altova.com" NDATA LNK>
]>
<document>
   <header>Example of How to Use Unparsed Entity URIs</header>
   <para>...</para>
   <img src="Picture"/>
   <link href="AltovaURI">Link to the Altova Website.</link>
</document>
```

SPS-Bilder und Hyperlinks, die ungeparste Entities verwenden

Bilder und Hyperlinks im SPS, die ungeparste Entity URIs referenzieren, werden folgendermaßen verwendet:

1. Fügen Sie das Bild oder den Hyperlink über das Menü **Einfügen** ein.
2. Wählen Sie im "Bearbeiten"-Dialogfeld des Objekts die Eigenschaften auf dem Register "Dynamisch" aus (*Abbildung unten)* und geben Sie einen XPath-Ausdruck ein, der den Node, der den Namen der ungeparsten Entities enthält, auswählt. In dem Beispiel des XML-Dokuments oben wäre dies der Node `//img/@src` bzw. `//link/@href`.



3. Aktivieren Sie anschließend am unteren Rand des Dialogfelds das Kontrollkästchen "Als ungeparste Entity für die XSLT-Transformation behandeln", damit der Inhalt des ausgewählten Node als ungeparste Entity gelesen wird; wenn eine ungeparste Entity dieses Namens deklariert ist, wird die mit der ungeparsten Entity verknüpfte URI zum Auffinden der Ressource (Bild oder Hyperlink) verwendet.

Während der Verarbeitung des Stylesheet wird der Name der Entity durch die mit dem Entity-Namen verknüpfte URI ersetzt.

Anmerkung: Wenn die URI eine relative URI ist, erweitert der XSLT-Prozessor diese zu einer absoluten URI, indem er die Basis-URI der DTD darauf anwendet. Wenn also die ungeparste Entity mit der relativen URI "`nanonull.gif`" verknüpft ist, dann wird diese URI zum Pfad `file:///c:/someFolder/nanonull.gif` erweitert, wobei sich die DTD im Ordner `someFolder` befindet.

© 2019-2025 Altova GmbH

9.3 Neu von XSLT-, XSL-FO- oder FO-Datei

Einem SPS-Design können vorhandene XSLT-Dateien, die für die HTML-Ausgabe oder XSLT-Dateien mit XSL-FO-Befehlen, die für die Ausgabe in PDF oder FO erstellt wurden, zugrunde gelegt werden, d.h. die SPS-Datei muss nicht von Grund auf neu erstellt werden, sondern kann anhand einer vorhandenen XSLT-Datei weiterentwickelt werden.

Schritte zur Erstellung eines SPS anhand einer XSLT-Datei

Die Schritte zur Erstellung einer SPS-Datei anhand einer XSLT-, XSLT-for-FO- oder FO-Datei sind die folgenden.

1. Wählen Sie den Befehl **Datei | Neu | Neu von XSLT-, XSLT-FO- oder FO-Datei.**
2. Wählen Sie im daraufhin angezeigten Dialogfeld "Öffnen" die gewünschte Datei aus.
3. Im nächsten Dialogfeld werden Sie aufgefordert, ein Schema auszuwählen, auf dem die SPS-Datei basieren soll. Wählen Sie das gewünschte Schema aus.
4. Daraufhin wird ein SPS, das auf der Struktur und Formatierung in der XSLT- oder FO-Datei basiert, erstellt und in der Design-Ansicht 29 angezeigt.
5. Sie können die SPS-Datei nun wie gewohnt bearbeiten. So könnten Sie z.B. Nodes aus der Schema-Struktur 38 in das Design ziehen, die Stile und Darstellung ändern oder zusätzliche Stile hinzufügen sowie StyleVision-Funktionen wie z.B. automatische Berechnungen 257 und Bedingungsvorlagen 262 verwenden.
6. Sie können das SPS speichern und anhand einer XML-Arbeitsdatei 36 eine Vorschau der verschiedenen Ausgabeformate 31 anzeigen. Anschließend können Sie mit Hilfe des Befehls **Generierte Dateien speichern** 486 Stylesheets und Ausgabedateien generieren 486.

Beispiel

Das unten beschriebene Beispiel befindet sich im Ordner (Eigene) Dateien 25, `C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2025\StyleVisionExamples/Tutorial/NewFromXSLT`. Dieser Ordner enthält die Dateien: `SimpleExample.xslt`, `SimpleExample.xsd` und `SimpleEample.xml`.

Unten sehen Sie die XML-Datei.

Im Diagrammbeispiel YearlySales.xml verwendete XML-Daei

```
<?xml version="1.0" encoding="UTF-8"?>
<Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="YearlySales.xsd">
       <ChartType>Pie Chart 2D</ChartType>
       <Region id="Americas">
              <Year id="2005">30000</Year>
              <Year id="2006">90000</Year>
              <Year id="2007">120000</Year>
              <Year id="2008">180000</Year>
              <Year id="2009">140000</Year>
              <Year id="2010">100000</Year>
       </Region>
       <Region id="Europe">
              <Year id="2005">50000</Year>
```

```
            <Year id="2006">60000</Year>
            <Year id="2007">80000</Year>
            <Year id="2008">100000</Year>
            <Year id="2009">95000</Year>
            <Year id="2010">80000</Year>
      </Region>
      <Region id="Asia">
            <Year id="2005">10000</Year>
            <Year id="2006">25000</Year>
            <Year id="2007">70000</Year>
            <Year id="2008">110000</Year>
            <Year id="2009">125000</Year>
            <Year id="2010">150000</Year>
      </Region>
</Data>
```

Die XSLT-Datei sieht folgendermaßen aus:

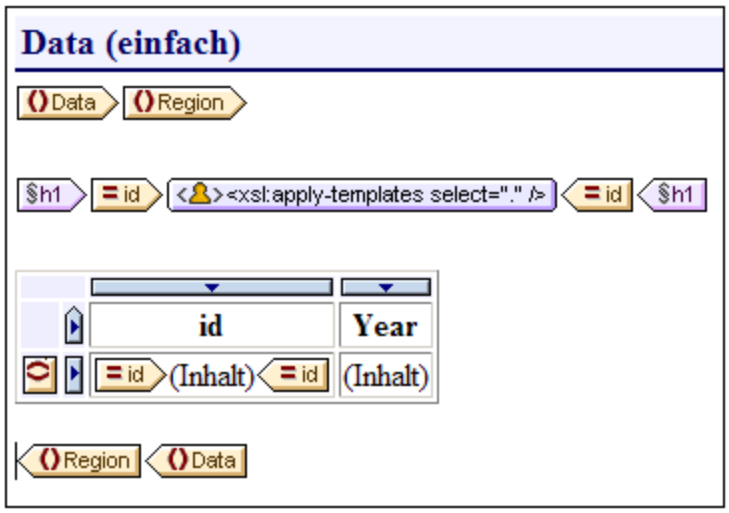
```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:fn="
http://www.w3.org/2005/xpath-functions">
   <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
   <xsl:template match="/">
      <html>
         <head>
            <title>Simple Example for New From XSLT</title>
         </head>
         <body>
            <xsl:apply-templates/>
         </body>
      </html>
   </xsl:template>
   <xsl:template match="Data">
      <xsl:for-each select="Region">
         <h1 style="color:red">
            <xsl:apply-templates select="@id"/>
         </h1>
      </xsl:for-each>
   </xsl:template>
</xsl:stylesheet>
```

Führen Sie die oben aufgelisteten Schritte 1 bis 4 durch, um das SPS in der Design-Ansicht zu erhalten. Das SPS sollte ungefähr folgendermaßen aussehen:

© 2019-2025 Altova GmbH

Beachten Sie, dass die beiden Vorlagen im XSLT im SPS erstellt wurden. Wechseln Sie nun in die HTML-Vorschau (*Abbildung unten*). Hier sehen Sie, dass der Stil des Elements `h1` (`color:red`) ebenfalls an das SPS übergeben wurde.

Wählen Sie das `h1`-Element in der Design-Ansicht aus und ändern Sie (in der Seitenleiste "Stile" in der Eigenschaftsgruppe *Farbe*) seine Farbe in Schwarz. Ziehen Sie anschließend das Element `Year` aus der Schema-Struktur ins Design und erstellen Sie es an der in der Abbildung unten gezeigten Stelle als Tabelle. Kehren Sie den Inhalt der beiden Spalten um, sodass die Jahres-ID sich in der ersten Spalte befindet.



Sie können weitere Änderungen am Inhalt, der Struktur und den Darstellungseigenschaften des Dokuments machen, dann eine Vorschau der Ausgabe anzeigen und die Dateien mit dem Befehl **Generierte Dateien speichern** 486 speichern.

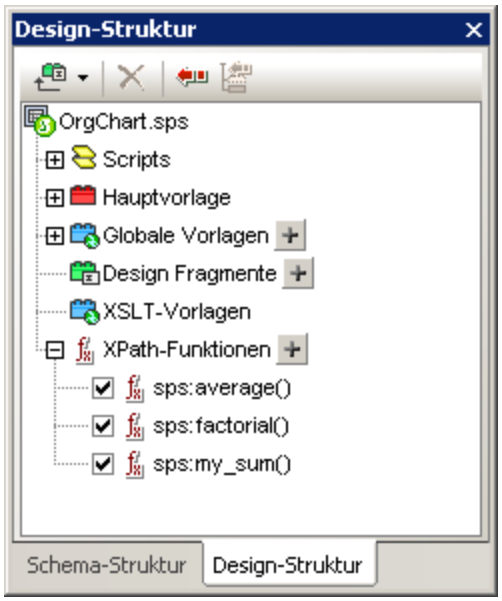
© 2019-2025 Altova GmbH

9.4 Benutzerdefinierte XPath-Funktionen

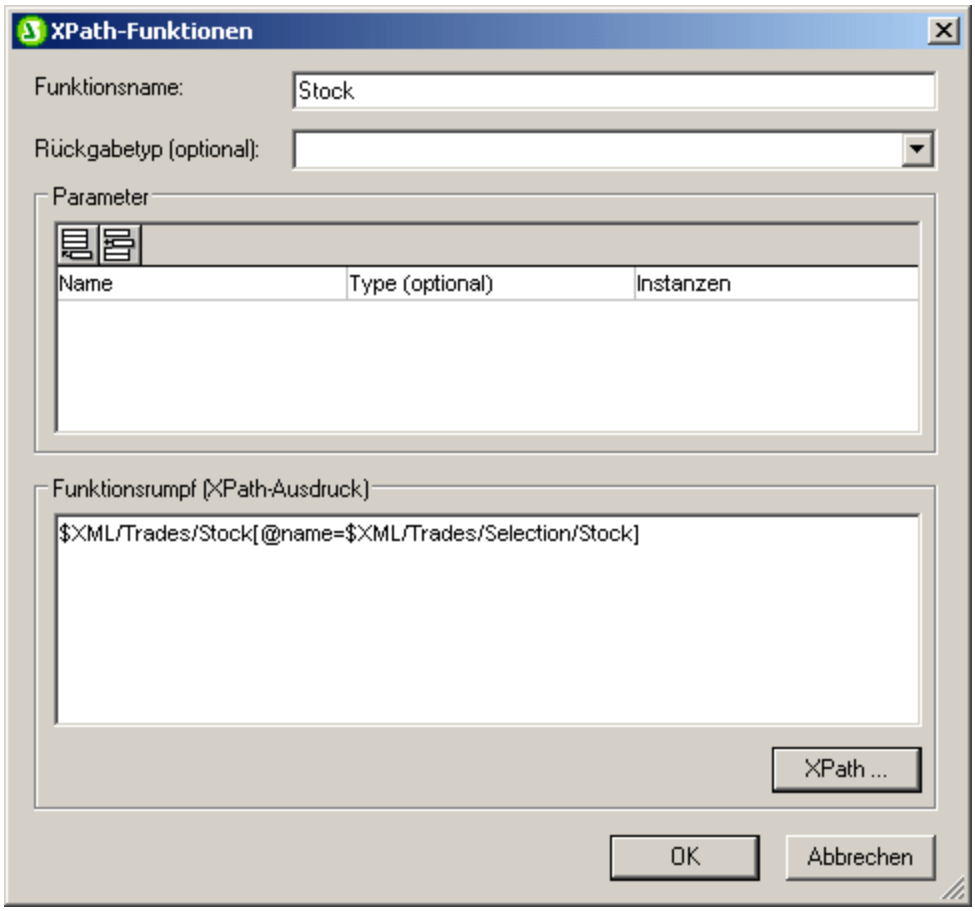
Der SPS-Designer kann angepasste **XPath 2.0/3.0**-Funktionen definieren. Eine benutzerdefinierte XPath-Funktion kann in jeder Design-Komponente , die XPath-Ausdrücke akzeptiert, wiederverwendet werden, zum Beispiel in automatischen Berechnungen, Bedingungen und Auswahllisten.

Definieren und Bearbeiten von benutzerdefinierten XPath-Funktionen

Benutzerdefinierte XPath-Funktionen werden entweder in der Seitenleiste Schema-Struktur oder in der Seitenleiste Design-Struktur (*siehe Abbildung unten*) erstellt (und in weiterer Folge zum Bearbeiten aufgerufen). Alle benutzerdefinierten XPath-Funktionen in einem SPS sind sowohl in der Seitenleiste Schema-Struktur als auch in der Seitenleiste Design-Struktur unter dem Element "XPath-Funktionen) aufgelistet und können über beide Seitenleisten aufgerufen werden.



Um eine benutzerdefinierte XPath-Funktion zu erstellen, klicken Sie auf das Symbol des Elements "XPath-Funktionen". Dadurch erscheint das Dialogfeld "XPath-Funktionen" (*Abbildung unten*). Wenn Sie eine Funktion, die bereits erstellt wurde, bearbeiten möchten, doppelklicken Sie den entsprechenden Eintrag in der Liste der XPath-Funktionen. Das Dialogfeld "XPath-Funktionen" (*Abbildung unten*) erscheint und die Funktionsdefinition kann bearbeitet werden.



Nachdem eine benutzerdefinierte XPath-Funktion erstellt worden ist, kann sie an einer beliebigen Stelle des Designs verwendet werden.

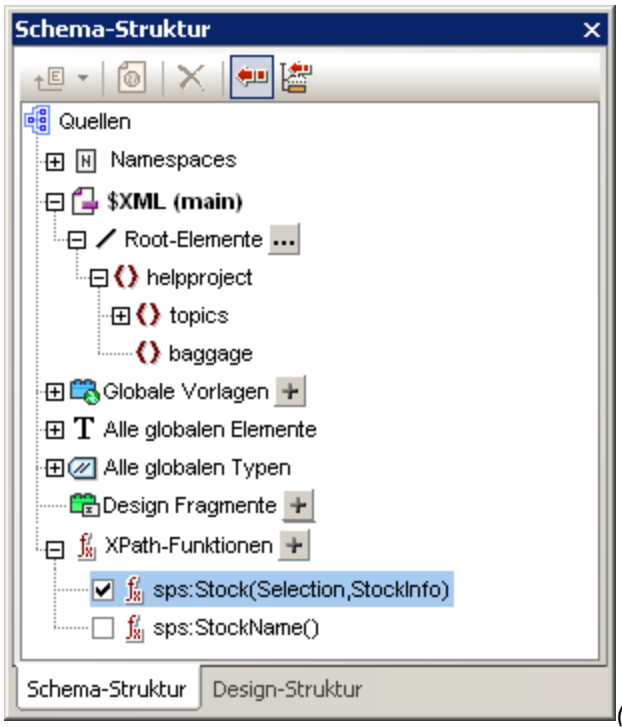
Namespace von benutzerdefinierten XPath-Funktionen

Benutzerdefinierte XPath-Funktionen werden im Namespace `http://www.altova.com/StyleVision/user-xpath-functions` erstellt. Dieser Namespace ist an den Präfix `sps:` gebunden, daher müssen benutzerdefinierte SPath-Funktionen müssen unter Verwendung dieses Namespace-Präfix aufgerufen werden. Beispiel: `sps:MyFunction()`.

Aktivieren und Deaktivieren von benutzerdefinierten XPath-Funktionen

Jede benutzerdefinierte XPath-Funktion kann aktiviert oder deaktiviert werden, indem das Kontrollkästchen links neben dem Eintrag der Funktion in der Liste der Benutzerdefinierten XPath-Funktionen aktiviert bzw. deaktiviert wird (*siehe Abbildung unten*).

© 2019-2025 Altova GmbH



Diese Funktion ist nützlich, wenn zwei Funktionen den gleichen Namen haben. Eine solche Situation könnte zum Beispiel entstehen, wenn ein importiertes SPS-Modul eine Funktion mit dem gleichen Namen enthält.

Aufrufen einer benutzerdefinierten XPath-Funktion

Eine benutzerdefinierte XPath-Funktion kann in einem XPath-Ausdruck an jeder beliebigen Stelle im Design aufgerufen werden. Zum Beispiel kann die oben definierte benutzerdefinierte XPath-Funktion `sps:MyFunction` mit dem folgenden XPath-Ausdruck in einer automatischen Berechnung aufgerufen werden:

```
sps:MyFunction()/@name.
```

Dieser XPath-Ausdruck würde wie folgt ausgewertet werden:

1. Die Funktion `sps:MyFunction()` wird ausgewertet. Angenommen, die Funktion ist wie folgt definiert: `$XML/Trades/Stock[@name=$XML/Trades/Selection/Stock]`. Wenn die Funktion ausgewertet ist, gibt sie das Element `/Trades/Stock` zurück, das ein `name`-Attribut mit einem Wert hat, der dem Inhalt des Elements `/Trades/Selection/Stock` entspricht.
2. Das Ergebnis von Schritt 1 wird dem XPath-Ausdruck im Funktionsaufruf zurückgegeben. Nun wird das `name`-Attribut dieses `/Trades/Stock`-Elements als Wert der automatischen Berechnung zurückgegeben.

Löschen einer Funktion

Um eine Funktion zu löschen, markieren Sie sie in der Liste der XPath-Funktionen in der Seitenleiste Schema-Struktur oder in der Seitenleiste Design-Struktur und klicken Sie dann in der Symbolleiste der Seitenleiste auf das Symbol **Objekt entfernen**. Alternativ dazu können Sie mit der rechten Maustaste auf die XPath-Funktion klicken und **Entfernen** aus dem Kontextmenü auswählen.

9.4.1 Definieren einer XPath-Funktion

Eine benutzerdefinierte XPath-Funktion benötigt (i) einen Namen (einen Text-String) und (ii) eine Definition (einen XPath-Ausdruck).

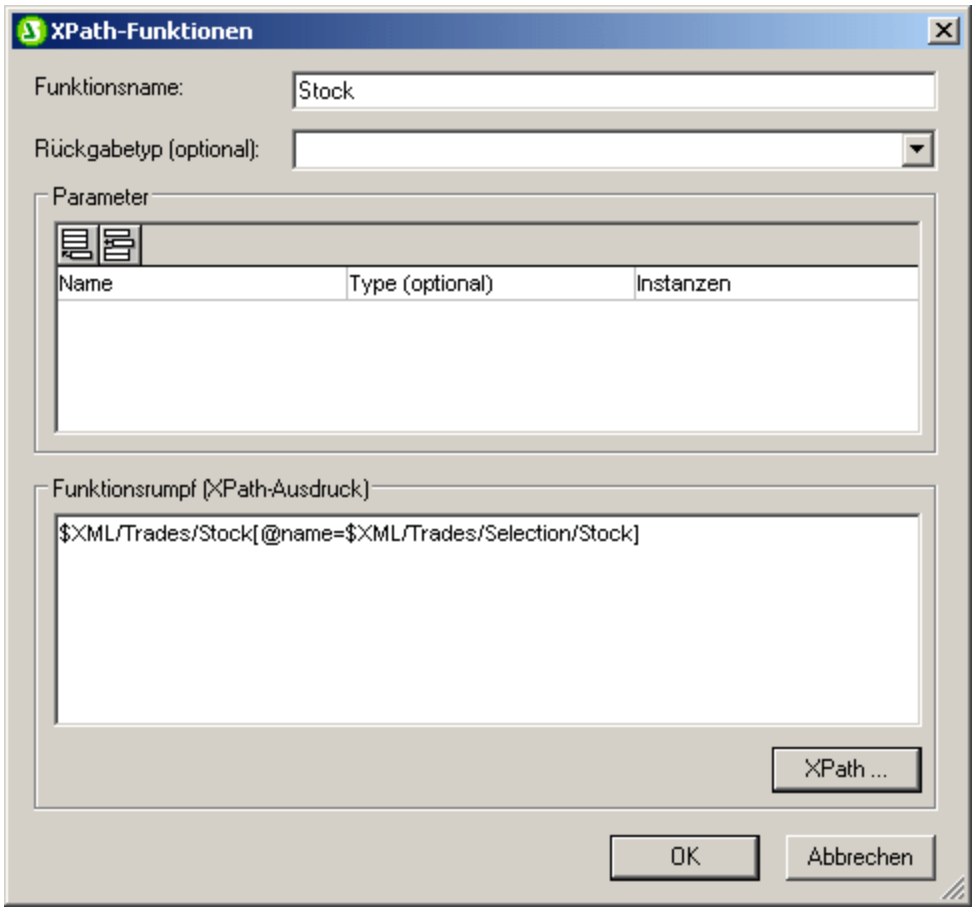
Darüber hinaus können Sie für die Funktion einen oder mehrere **Parameter** definieren. Eine benutzerdefinierte XPath-Funktion kann auch einen optionalen **Rückgabetyp** haben (wird durch Auswählen eines Typs aus der Dropdown-Liste des Listenfelds *Rückgabetyp* angegeben). Ein Rückgabetyp kann sinnvoll sein, wenn Sie sicherstellen möchten, dass der Datentyp des Rückgabewerts mit dem ausgewählten Datentyp übereinstimmt. Beachten Sie, dass der Rückgabewert nicht in den ausgewählten Datentyp konvertiert wird. Wenn die Typen nicht übereinstimmen, wird eine Fehlermeldung ausgegeben. Wenn kein Rückgabewert angegeben wird, wird der Datentyp nicht überprüft.

Nachdem eine benutzerdefinierte XPath-Funktion erstellt worden ist, kann sie an einer beliebigen Stelle des Designs verwendet werden. Im XSLT-Stylesheet wird sie als `xsl:function`-Element erstellt, das ein Child des Elements `xsl:stylesheet` ist, wie im unten stehenden Listing gezeigt wird.

```
<xsl:stylesheet>
   ...
   <xsl:function name="sps:Stock">
       <xsl:sequence select="$XML/Trades/Stock[@name=$XML/Trades/Selection/Stock]"/>
   </xsl:function>
   <xsl:function name="sps:Average" as="xs:decimal">
       <xsl:param name="a" as="xs:integer"/>
       <xsl:param name="b" as="xs:integer"/>
       <xsl:param name="c" as="xs:integer"/>
       <xsl:sequence select="avg( ($a, $b, $c) )"/>
  </xsl:function>
</xsl:stylesheet>
```

Die in der unten stehenden Abbildung und im obigen Listing gezeigte `sps:Stock`-Funktion gibt das Element `/Trades/Stock` zurück, das ein `name`-Attribut mit einem Wert hat, der dem Inhalt des Elements `/Trades/Selection/Stock` entspricht. Die oben aufgelistete Funktion `sps:Average` gibt den Mittelwert der drei Eingabeparameterwerte zurück. Die Funktionsdefinition verwendet die Funktion `avg()` von XPath 2.0/3.0. Der Rückgabedatentyp ist auf den Typ `xs:decimal` festgelegt, das ist der Datentyp, der von der Funktion `avg()`zurückgegeben wird, wenn Eingabewerte des Datentyps `xs:integer` überprüft werden. Wenn der Rückgabetyp angegeben wird, wird der Datentyp des Rückgabewerts dahingehend überprüft, ob er mit dem angegebenen Typ übereinstimmt. Falls dem nicht so ist, wird eine Fehlermeldung ausgegeben.

© 2019-2025 Altova GmbH



Definieren der Funktion

Um eine Definition zu definieren, klicken Sie in der Schema-Struktur oder in der Design-Struktur auf das Symbol des Elements "XPath-Funktionen". Dadurch erscheint das Dialogfeld "XPath-Funktionen" (*Abbildung oben*). Wenn Sie eine Funktion bearbeiten möchten, die bereits erstellt worden ist, doppelklicken Sie ihren Eintrag in der Liste der XPath-Funktionen. Geben Sie dann einen Namen für die Funktion und im Bereich Funktionsrumpf eine Definition ein. Parameterdefinitionen können, falls nötig, eingegeben werden (Details dazu finden Sie in den nächsten beiden Abschnitten Parameter und Sequenzen 391 und Parameter und Nodes 396). Ein Rückgabetyp für die Funktion kann ebenfalls angegeben werden (*siehe oben*).

Der wichtigste Punkt, den Sie beim Schreiben des XPath-Ausdrucks, der die XPath-Funktion definiert, beachten müssen, ist, dass es **keinen Kontext-Node** für den XPath-Ausdruck gibt. Wenn der XPath-Ausdruck einen Node lokalisieren muss, dann kann der Kontext-Node auf eine der folgenden Arten festgelegt werden:

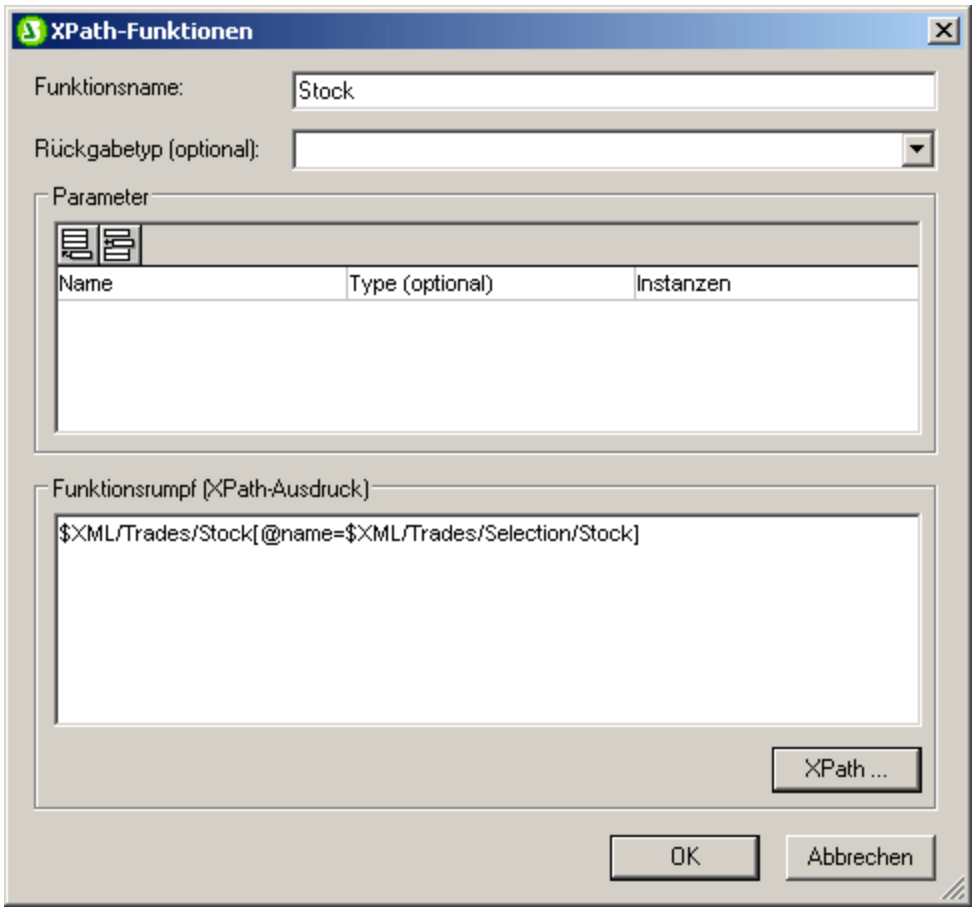
1. Der XPath-Ausdruck beginnt mir der Root des Dokuments. Die Dokument-Root wird im ersten Lokalisierungsschritt des XPath-Ausdrucks als `$XML` angegeben. Der XPath-Ausdruck `$XML/Trades/Stock[1]` findet zum Beispiel das erste `Stock` Child-Element des Elements `/Trades`. Die Variable `$XML` (die die Dokument-Root des Hauptschemas lokalisiert) wird von StyleVision global in allen SPS-Designs definiert.
2. Der Kontext-Node kann als Parameter ausgegeben werden. Eine Erklärung dazu finden Sie weiter unten im Abschnitt Parameter und Nodes 396.

In den folgenden Fällen werden Fehlermeldungen zurückgegeben:

- Wenn ein Parameter definiert ist, aber im Rumpf der Definition nicht verwendet wird.
- Wenn der Datentyp eines Werts, der von der Funktion zurückgegeben wird, nicht dem Rückgabetyp entspricht, der für die Funktion definiert wurde.
- Wenn eine beliebige Funktion im SPS einen Fehler enthält, wird ein XSLT-Fehler für das gesamte Design generiert, selbst wenn die Funktion, die den Fehler enthält, nicht aufgerufen wird. Beachten Sie jedoch, dass eine Funktion deaktiviert werden kann, indem ihr Kontrollkästchen in der Liste der benutzerdefinierten XPath-Funktionen deaktiviert wird. Wenn die Funktion im Design deaktiviert ist, wird sie nicht in das XSLT-Dokument aufgenommen, das aus dem Design generiert wird. Auf diese Weise kann ein XPath-Ausdruck, der einen Fehler enthält vom XSLT ausgeschlossen werden und es wird kein XSLT-Fehler generiert.

9.4.2 Funktionen zum Auffinden von Nodes wiederverwenden

Im vorhergehenden Abschnitt haben wir erfahren, wie ein XPath-Ausdruck zum Auffinden eines Nodes erstellt werden kann. Die Funktion `sps:Stock`, die wie in unten stehenden Abbildung definiert ist, gibt das Element `/Trades/Stock` zurück, das ein `name`-Attribut mit einem Wert hat, der dem Inhalt des Elements `/Trades/Selection/Stock` entspricht.



 © 2019-2025 Altova GmbH

Wir könnten die Lokalisierungsschritt des XPath-Ausdrucks `$XML/Trades/Stock[@name=$XML/Trades/Selection/Stock]` in einzelne Module von XPath-Ausdrücken zerlegen; beispielsweise wie folgt:

- Die Funktion `sps:Stocks()`, mit der Definition: `$XML/Trades/Stock`
- Die Funktion `sps:SelectedStock()`, mit der Definition: `$XML/Trades/Selection/Stock`

Der gesamte XPath-Ausdruck kann dann in einem anderen XPath-Ausdruck folgendermaßen geschrieben werden:

```
sps:Stocks()[@name=sps:SelectedStock()]
```

Wenn XPath-Funktionen auf diese Weise erstellt werden, um einen Node oder ein Nodeset zu lokalisieren, können diese Funktionen innerhalb des SPS-Designs wiederverwendet werden und so das Schreiben von komplexen XPath-Ausdrücken erheblich vereinfachen.

9.4.3 Parameter in XPath-Funktionen

Einer benutzerdefinierte XPath-Funktion kann eine beliebige Anzahl von Parametern zugewiesen werden. Die Parameter der Funktion werden im Bereich "Parameter" des Dialogfelds "XPath-Funktionen" (*siehe Abbildung unten*) definiert. Diese Parameter können dann bei der Definition der benutzerdefinierten XPath-Funktion (im Bereich *Funktionsrumpf*) verwendet werden.

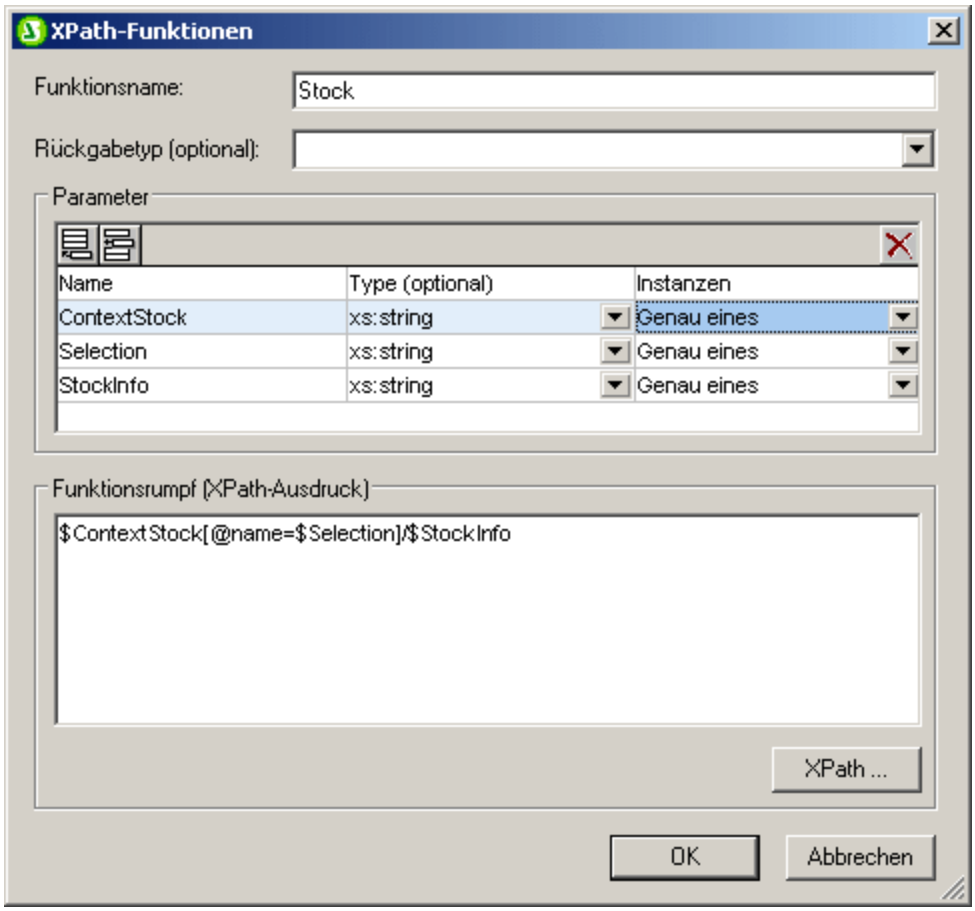
Funktionsweise einer benutzerdefinierten XPath-Funktion

Nachfolgend wird beschrieben, wie sich eine XPath-Funktion verhält.

1. Die Anzahl der Argumente in einem Funktionsaufruf (z.B. in einer automatischen Berechnung) muss mit der Anzahl der für die benutzerdefinierte Funktion definierten Parameter (die im Bereich *Parameter* der benutzerdefinierten Funktion definiert sind; *siehe Abbildung unten*) übereinstimmen. Zusätzlich dazu muss die Anzahl der von jedem Argument (im Funktionsaufruf) übergebenen Objekte mit der *Instanzen*-Definition des entsprechenden Parameters übereinstimmen. Wenn für einen Parameter (im Bereich P*arameter* in der Spalte "Typ") eine Datentypeinschränkung definiert wurde, so muss/müssen der/die vom Argument gelieferten Wert(e) mit diesem Datentyp übereinstimmen.
2. Die an die Parameter der Funktion übergebenen Argumente werden anschließend in der (im Bereich *Funktionsrumpf; siehe Abbildung unten*) definierten XPath-Funktion verwendet. Das durch die Auswertung des XPath-Ausdrucks ermittelte Ergebnis wird anschließend anhand der optionalen Definition *Rückgabetyp* überprüft (*siehe Abbildung unten*). Wenn der Datentyp dem erwarteten entspricht, wird das Ergebnis in dem XPath-Ausdruck, über den die Funktion aufgerufen wurde, verwendet.

Reihenfolge von Parametern

Die Reihenfolge der Parameter der benutzerdefinierten Funktion ist wichtig, weil die Argumente beim Funktionsaufruf den Parametern in der Reihenfolge, wie sie im Bereich "*Parameter*" (*siehe Abbildung unten*) definiert sind, zugewiesen werden.



Wenn also die benutzerdefinierte XPath-Funktion `sps:Stock` XPath, so wie in der Abbildung oben gezeigt, definiert ist und wenn Sie mit dem XPath-Ausdruck

```
sps:Stock($XML, Node1, Node2)
```

aufgerufen wird, dann werden diese drei Argumente – `$XML, Node1, Node2` – den Parametern – `$ContextStock`, `$Selection` und `$StockInfo` – in dieser Reihenfolge zugewiesen.

Beachten Sie, dass jedes Argument im Funktionsaufruf vom nächsten durch ein Komma getrennt wird, sodass jedes Argument an den entsprechenden Parameter übergeben wird (Reihenfolge siehe Bereich *Parameter in der Abbildung oben)*.

Die Reihenfolge der Parameter im Bereich *Parameter* kann mit den dort verfügbaren Symbolen **Anhängen**, **Einfügen** und **Löschen** geändert werden.

Datentyp von Parametern

Optional kann der Datentyp der Parameter der benutzerdefinierten Funktion definiert werden. Wenn ein Datentyp angegeben ist, wird der Datentyp des Eingabearguments anhand des Datentyps des Parameters überprüft und eine Fehlermeldung ausgegeben, wenn die Typen nicht übereinstimmen. Mit Hilfe dieser Funktion können die Eingabedaten (aus den Argumenten des Funktionsaufrufs) überprüft werden.

© 2019-2025 Altova GmbH

Instanzen

Man kann jeden Parameter der benutzerdefinierten XPath-Funktion als Sequenz betrachten. Die Eigenschaft *Instanzen* eines Parameters gibt an, wie viele Objekte vom entsprechenden Argument des Funktionsaufrufens für diesen Parameter geliefert werden müssen.

Sowohl in Funktionsdefinitionen als auch in Funktionsaufrufen werden die Parameter bzw. Argumente, aber auch die einzelnen Objekte innerhalb einer Sequenz durch Kommas voneinander getrennt. Daher muss unbedingt der Kontext, in dem die Kommas verwendet werden, beachtet werden: ob das Komma zum Trennen von Parametern/Argumenten oder von Sequenzobjekten dient.

- In Parametern/Argumenten werden Sequenzen - in der Funktionsdefinition (Parameter) oder im Funktionsaufruf (Argumente), falls erforderlich, durch Klammern getrennt.
- In Sequenzen werden Klammern ignoriert.

In diesem Zusammenhang sollten die folgenden Beispiele und Punkte beachtet werden:

- **Klammern in Parametern/Argumenten:** Einige XPath-Funktionen verwenden eine einzige Sequenz als Argument, z.B. die Funktionen `avg()` und `count()`. Wenn diese Sequenz mittels Kommatrennzeichen oder Bereichsoperatoren enumeriert wird, muss sie in Klammern eingeschlossen werden, damit eindeutig klar ist, dass es sich um eine einzige Sequenz - und nicht um mehrere, durch Kommas getrennte Sequenzen - handelt. So erhält z.B. in der Funktion `avg((count($a), $b, $c))` die XPath 2.0 `avg()` Funktion die Einzelsequenz `count($a),$b,$c` als Argument. Da die einzelnen Elemente der Sequenz enumeriert sind und eine Sequenz von drei Elementen bilden, muss die Sequenz in Klammern eingeschlossen werden und als einziges Argument an die `avg()` Funktion: `avg((count($a),$b,$c))` übergeben werden. Ohne das innere Klammerpaar hätte die Definition der `avg()` Funktion drei Parameter. Dies wäre ein Fehler, da die `avg()` Funktion ein Argument bestehend aus einer einzigen Sequenz erwartet.
- **Keine Klammern in Parametern/Argumenten**: Ähnlich wie oben, verwendet auch die `count()` Funktion eine einzige Sequenz als ihr Ein-Parameter-Argument. Da aber die einzige Sequenz in unserem Beispiel `count($a)` keine kommagetrennte enumerierte Liste ist, sondern stattdessen von der Variablen/dem Parameter `$a`, abgerufen wird, muss das Argument nicht in eine innere Klammer eingeschlossen werden. Daher ist der Ausdruck `count($a)` korrekt.
- **Klammern und Kommas in Funktionsaufrufen**: Im Funktionsaufruf müssen die Klammern korrekt gesetzt werden, damit jedes Argument einem Parameter (wie im Dialogfeld "XPath-Funktionen im Bereich *Parameter* definiert) entspricht. Wenn, zum Beispiel, die benutzerdefinierte XPath-Funktion mit dem Namen `MyAverage()` mit dem XPath 2.0-Ausdruck `avg((count($a),$b,$c))` definiert wird, dann wäre der folgende Funktionsaufruf gültig: `MyAverage((1,2,3),4,5)`. Die Werte, die den drei Parametern `$a`, `$b`, und `$c` entsprechen, wären die Sequenz `(1,2,3)`, die Einzelsequenz `4` und die Einzelsequenz `5`. Einzelsequenzen können optional in Klammern eingeschlossen werden. Der von `MyAverage()` in diesem Fall zurückgegebene Wert wäre `4`.

9.4.3.1 Parameter und Sequenzen

Es ist wichtig die Beziehung zwischen Parametern und Sequenzen zu beachten und zu wissen, wie Parameter und Sequenzen in XPath-Ausdrücken verwendet werden. Die folgenden Definitionen sollten diese Beziehungen verdeutlichen:

- Eine ***Sequenz*** besteht aus Objekten, bei denen es sich entweder um atomare Werte oder Nodes handelt. Die Objekte der Sequez werden mittels Kommas getrennt.
- Eine XPath-Funktion kann ***Parameter*** erhalten**.** So ist z.B. im XPath 2.0-Ausdruck `count($a)` der Teil innerhalb der Klammern der Funktion der Parameter der Funktion und muss eine Sequenz von Objekten sein.
- Ein ***Argument*** besteht aus einem oder mehreren Objekten in einem Funktionsaufruf. So hat z.B. die Funktion `count(//Person)` ein Argument: `//Person`. Dieses Argument ist gültig, weil es eine Node-Sequenz zurückgibt, die der Signatur der `count()` Funktion entspricht. (Die Signatur einer Funktion definiert die Anzahl der Parameter und den erwarteten Datentyp der einzelnen Parameter. Außerdem definiert sie, was die Funktion zurückgibt sowie den Datentyp des zurückgegebenen Objekts).
- Die Funktion `substring('StyleVisionExamples', 6, 6)` - die den String `Vision` zurückgibt - hat drei Argumente und ist laut der Signatur der `substring()` Funktion gültig und wird von dieser Funktion definiert. Wenn ein Funktionsaufruf mehrere Argumente hat, werden diese durch Kommas voneinander getrennt.

Klammern als Sequenz-Trennzeichen

Beachten Sie beim Erstellen von XPath-Ausdrücken unbedingt Folgendes: *Klammern werden zum Trennen von Sequenzen verwendet, in denen zum Enumerieren von Sequenzen das Komma-Trennzeichen oder ein Bereichsoperator verwendet wird. Folglich wird jede durch Klammern getrennte Sequenz (in Funktionsdefinitionen) als ein einziger Parameter bzw. (in Funktionsaufrufen) als ein einziges Argument gelesen.*

Ein Path- (oder Locator)-Ausdruck muss nicht unbedingt in Klammern gesetzt werden, da ein XPath-Ausdruck (Beispiel für einen XPath-Ausdruck `//Person/@salary`) in einem Argument eindeutig als ein einziger Parameter bzw. ein einziges Argument gelesen werden kann. Es handelt sich um einen Parameter/ein Argument, der/das aus einer einzigen Sequenz besteht.

Hier finden Sie ein paar Beispiele zur Veranschaulichung der obigen Punkte:

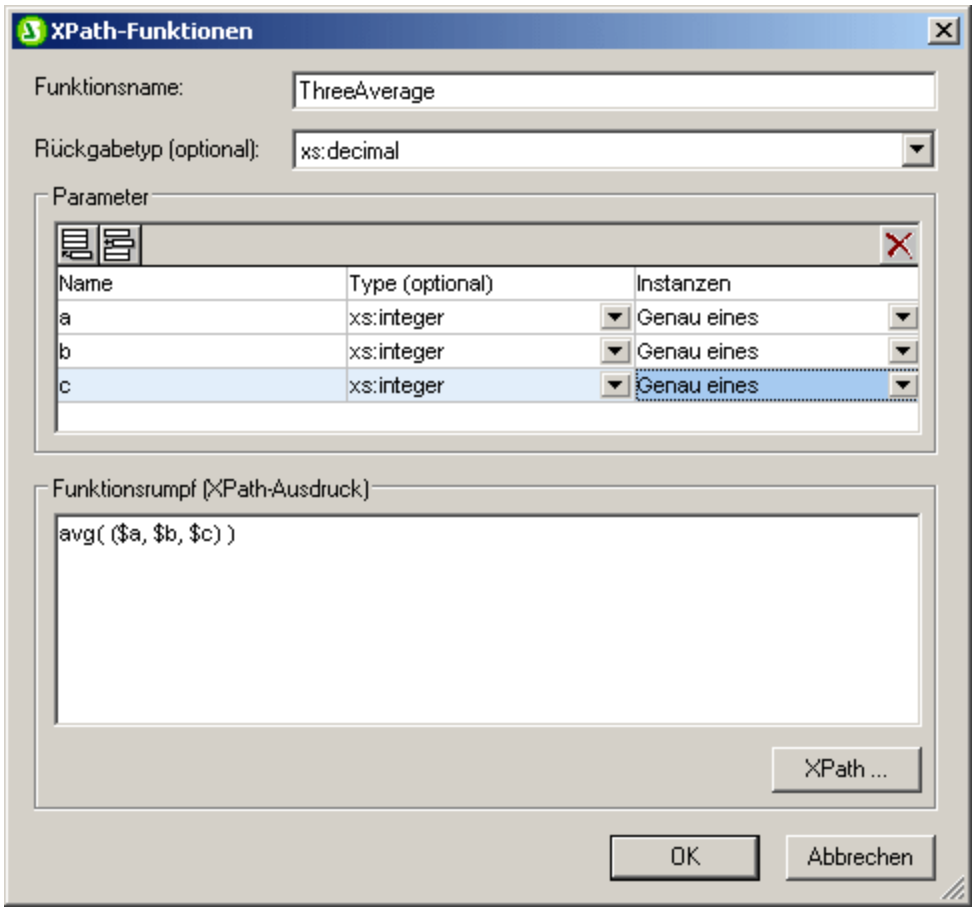
- `avg((10, 20, 30))` Die XPath 2.0-Funktion `avg` verwendet eine Sequenz von Objekten als ihr einziges Argument. Da es sich bei dieser Sequenz um eine durch Kommas getrennte Enumeration handelt, wird das innere Klammerpaar zum Trennen der obligatorischen einzigen Sequenz benötigt. Ohne die inneren Klammern hätte das Argument drei Argumente und wäre daher ungültig. (Das äußere Klammernpaar sind die Klammern der Funktion.)
- `avg(//Person/@salary)` Mit diesem XPath-Ausdruck werden die `salary`-Attribut-Nodes aller `Person`-Elemente ausgewählt. Ihre Attributwerte werden als die auszuwertende Sequenz zurückgegeben (d. h. es soll der Mittelwert errechnet werden). Es sind keine Klammern notwendig, weil die Sequenz vor der Auswertung nicht enumeriert ist. Das Argument ist der einzige XPath (oder Locator)-Ausdruck. Der XPath-Ausdruck wird ausgewertet und die Rückgabewerte werden als die Elemente einer Sequenz an die Funktion übergeben.
- `count((10 to 34))` Dies ist eine Enumeration über den Bereichsoperator. Der Bereichsoperator `'to'` generiert eine Sequenz von kommagetrennten Objekten (die Ganzzahlen von 10 bis 34), bevor das Argument gelesen wird. Als Ergebnis hat die `count()` Funktion in Ihrem Argument eine kommagetrennte Sequenz von 25 Objekten. Damit diese als ein aus einer einzigen Sequenz bestehendes Argument gelesen wird, werden Klammern als Trennzeichen benötigt. Ohne diese Klammern hätte der Funktionsaufruf 25 Argumente anstelle von einem, wodurch der Funktionsaufruf ungültig würde, da die `count()` Funktion gemäß ihrer Signatur nur ein Argument akzeptiert.
- `count((10 to 34, 37))` Die inneren Klammern geben an, dass es sich bei allen Objekten innerhalb davon um das einzige Argument des Funktionsaufrufs handelt - nämlich eine Sequenz bestehend aus 26 Elementen.

© 2019-2025 Altova GmbH

- `count(//Person)` Rund um dieses einzige Argument werden keine Klammern zum Trennen der Sequenz benötigt. Beim Argument handelt es sich um einen XPath-Ausdruck, der die `//Person` Nodes im XML-Dokument sammelt und diese Nodes als Objekte der zu zählenden Sequenz zurückgibt.

Verwendung von XPath-Parametern in XPath-Funktionen

Wenn Parameter in der Definition einer benutzerdefinierten XPath-Funktion verwendet werden, stellen Sie sicher, dass (i) die Anzahl der Argumente in einem Funktionsaufruf dieser XPath-Funktion korrekt ist und (ii) das Ergebnis der Auswertung der Argumente dem erwarteten Typ und den erwarteten Instanzen entspricht.



In der Abbildung oben sind (im Bereich *Parameter*) drei Parameter definiert, die anschließend (im Bereich *Funktionsrumpf*) zum Definieren einer XPath-Funktion verwendet werden.

Jeder unter *Parameter* definierte Parameter kann als Einzelsequenz betrachtet werden. Die Anzahl von Objekten, die innerhalb dieser Sequenz zulässig sind, wird mit der Eigenschaft *Instanzen* angegeben. In der obigen Definition ist zum Beispiel jeder Parameter (in seiner Eigenschaft *Instanzen*) als eine Einzelsequenz (das heißt, als Sequenz mit genau einem Objekt) definiert. Jedes Argument muss daher als eine Sequenz bestehend aus einem Objekt betrachtet werden. Die Eigenschaft *Type* definiert den Datentyp der Objekte der Sequenz.

In der Definition unserer XPath-Beispielfunktion (im Bereich *Funktionsrumpf*) liefert jeder Parameter ein Element der Sequenz, deren Mittelwert zu errechnen ist. Da die XPath-Parameter zusammen eine Sequenz bilden, muss die Sequenz in Klammern gesetzt werden, um sicherzustellen, dass die gesamte Sequenz als ein

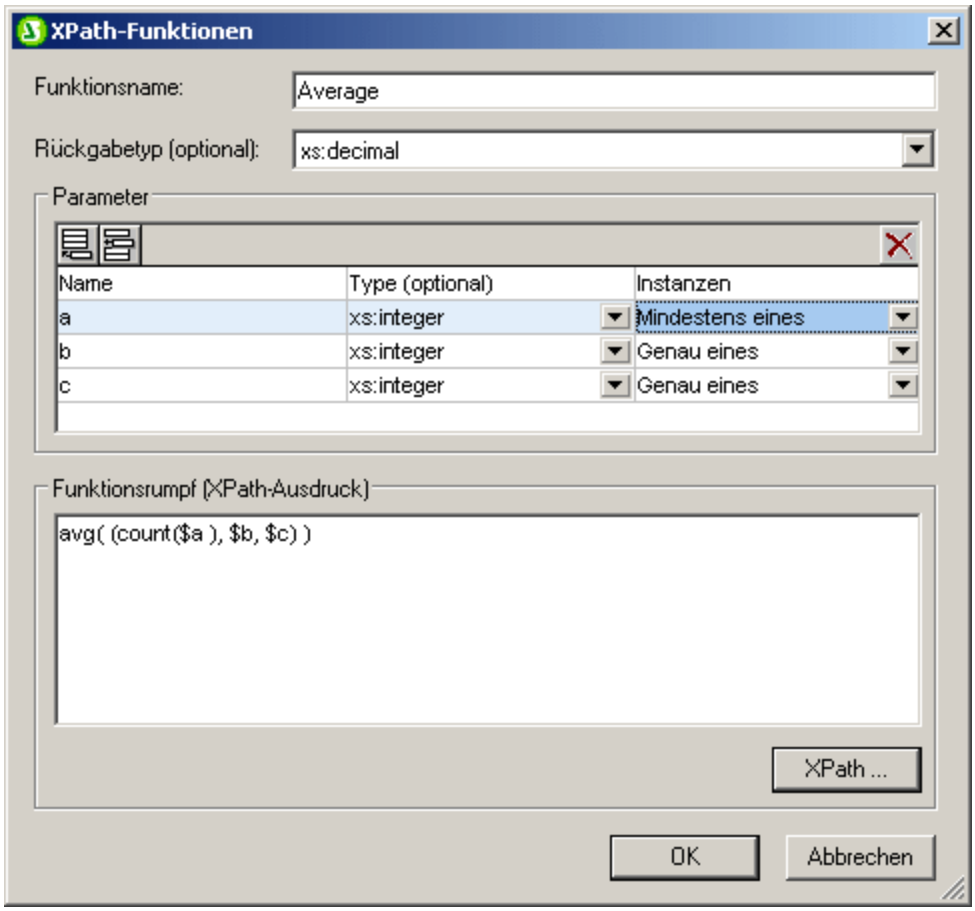
einziger Parameter der `avg()` Funktion gelesen wird. Wenn eines der Argumente im Funktionsaufruf (die den drei Parametern entsprechen) zur Laufzeit keine Einzelsequenz ist, wird ein Fehler zurückgegeben.

Im Folgenden finden Sie Beispiele für die Verwendung von XPath-Parametern in **Aufrufen** der XPath-Funktion `ThreeAverage()`, die in der obigen Abbildung zu sehen ist. Sie können in der Design-Ansicht eine automatische Berechnung einfügen und ihr die unten aufgelisteten XPath-Ausdrücke zuweisen, um die Ergebnisse zu sehen. Die Funktion wurde als bestehend aus einer Sequenz von drei Ganzzahlen definiert, deren Mittelwert berechnet werden soll.

- `sps:ThreeAverage(10,20,30)` gibt `20` zurück. Es gibt drei gültige Argumente im Funktionsaufruf, die den drei XPath-Parametern entsprechen.
- `sps:ThreeAverage( (10),(20),(30) )` Es gibt drei gültige Eingabeargumente, die den drei XPath-Parametern entsprechen. Jedes Eingabeargument steht innerhalb von Klammern (die redundant, aber kein Fehler sind, da jede Sequenz eine Einzelsequenz ist).
- `sps:ThreeAverage( (10),20,30 )` gibt `20` zurück. Es gibt drei gültige Eingabeargumente, die den drei XPath-Parametern entsprechen. Der erste Argument steht innerhalb von Klammern (dies ist nicht notwendig, aber kein Fehler).
- `sps:ThreeAverage( (10,20),(30),(40) )` gibt einen Fehler zurück, weil das erste Argument ungültig ist. Es handelt sich nicht um eine Einzelsequenz, wie in der Definition der Eigenschaft des ersten `$a`-Parameters (`'Genau eines'`)' gefordert.
- `sps:ThreeAverage( (10,20,30) )` gibt einen Fehler zurück, weil innerhalb der Klammern nur ein Argument übergeben wird. Außerdem ist das Argument ungültig, weil es sich bei der Sequenz nicht um eine Einzelsequenz handelt.

Wenn die Eigenschaft *Instanzen* eines Parameters auf `Mindestens eines` gesetzt wird (*wie in der Definition in der Abbildung unten gezeigt*), dann wird dieser Parameter als Sequenz bestehend aus einem oder mehreren Objekten definiert.

© 2019-2025 Altova GmbH



In der obigen Definition wurde der erste Parameter als Sequenz von einem oder mehreren Objekten und die nächsten beiden als Einzelsequenzen definiert. Die Funktion wurde so definiert, dass die Anzahl der Objekte, die vom ersten Parameter übergeben werden, ermittelt wird, das Resultat zu den beiden Integer-Werten, die von den beiden Parametern übergeben werden, hinzugefügt und dann das Resultat durch drei dividiert wird, um den Durchschnitt zu erhalten. Beachten Sie Folgendes:

- Die Sequenz, die der Parameter der `avg()`-Funktion ist, steht innerhalb von Klammern. Damit wird angegeben, dass die `avg()`-Funktion eine Einzelsequenz enthält, die aus drei Objekten besteht, die die Parameter bilden. Die Einzelsequenz besteht aus drei Integer-Werten: dem ersten, der von der `count()`-Funktion übergeben wird; und dem zweiten und dem dritten Wert, also den Parametern `b` und `c`.
- Das Argument der `count()` Funktion steht nicht innerhalb von Klammern zum Trennen der Sequenz, da es sich beim Argument eindeutig um eine Einzelsequenz handelt.

Hier finden Sie Beispiele zur Verwendung von Parametern in Aufrufen der oben gezeigten XPath-Funktion `Average()`.

- `sps:Average((1,2),3,4)` gibt `3` zurück. Es gibt drei gültige Eingabeargumente, die den drei Parametern entsprechen. Das erste Argument steht zur Abgrenzung innerhalb von Klammern. Wenn die `count()`-Funktion damit operiert, gibt die Funktion den Wert `2` zurück. Dieser Wert ist das erste Objekt der Sequenz, die an die `avg()`-Funktion übergeben wird.

- `sps:Average( 4,4,4 )` gibt `3` zurück. Es gibt drei gültige Eingabeargumente. Das erste Argument darf eine Sequenz bestehend aus einem Objekt sein (siehe die Eigenschaft *Instanzen* des entsprechenden Parameters). Es werden keine Klammern benötigt um separate Argumente anzugeben.

Weitere interessante Punkte

Die folgenden zusätzlichen Punkte sollten beachtet werden:

- Wenn ein Parameter mit `Mindestens einer`-Instanz definiert ist, könnte eine Funktion wie `MyAverage()` mit einem XPath-Ausdruck wie `avg(($a))` definiert werden. Diese Funktion würde ein Argument, das eine Einzelsequenz bestehend aus einem oder mehreren Objekten ist, akzeptieren. Die Funktion könnte wie folgt aufgerufen werden: `sps:MyAverage((2,3,4))` und würde den Wert `3` zurückgeben. Das Eingabeargument muss in Klammern eingeschlossen werden, um sicherzustellen, dass die Eingabe als Einzelsequenz und nicht als drei einzelne Sequenzen gelesen wird (was der Fall sein würde, wenn die Objekte nicht in Klammern stünden).
- Wenn ein XPath-Parameter `$a` mit einer `Keines oder eines`-Instanz definiert ist, könnte eine Funktion wie `MyAverage()` mit einem XPath-Ausdruck wie `avg(($a, $b, $c))` definiert werden. Diese Funktion würde drei Sequenzen als ihr Argument akzeptieren, wobei die Möglichkeit besteht, dass die erste Sequenz leer ist. Wenn die erste Sequenz leer sein soll, dann muss die leere Sequenz explizit als erstes Eingabeargument übergeben werden. Andernfalls wird ein Fehler ausgegeben. Wenn die Funktion wie folgt aufgerufen würde: `sps:MyAverage(30,20,10)`, würde sie den Wert `20` zurückgeben. Die Funktion könnte auch mit `sps:MyAverage((),20,10)` aufgerufen werden und würde den Wert `15` zurückgeben (beachten Sie, dass die leere Sequenz als Eingabewert "leer" gezählt wird). Folgendes würde jedoch einen Fehler generieren: `sps:MyAverage(20,10)`, weil keine leere erste Sequenz geliefert wird und das dritte Argument daher als fehlend betrachtet wird.

Komplexe Beispiele

Neben dem Vorteil der Wiederverwendung eines XPath-Ausdrucks ermöglichen benutzerdefinierte XPath-Funktionen auch die Konstruktion von komplexen, angepassten XPath-Funktionen, die nicht im Funktionsumfang von XPath 2.0 enthalten sind. Zum Beispiel könnte eine faktorielle Funktion einfach mit einem XPath-Ausdruck erstellt werden, die eine Einzelsequenz als ihren einzigen Parameter hat. Wenn der Parameter `$num` die Zahl ist, von der der Faktor gebildet werden soll, dann würde der XPath-Ausdruck zum Erstellen der Funktion so aussehen:

```
if ($num < 2) then 1 else $num * sps:factorial($num - 1)
```

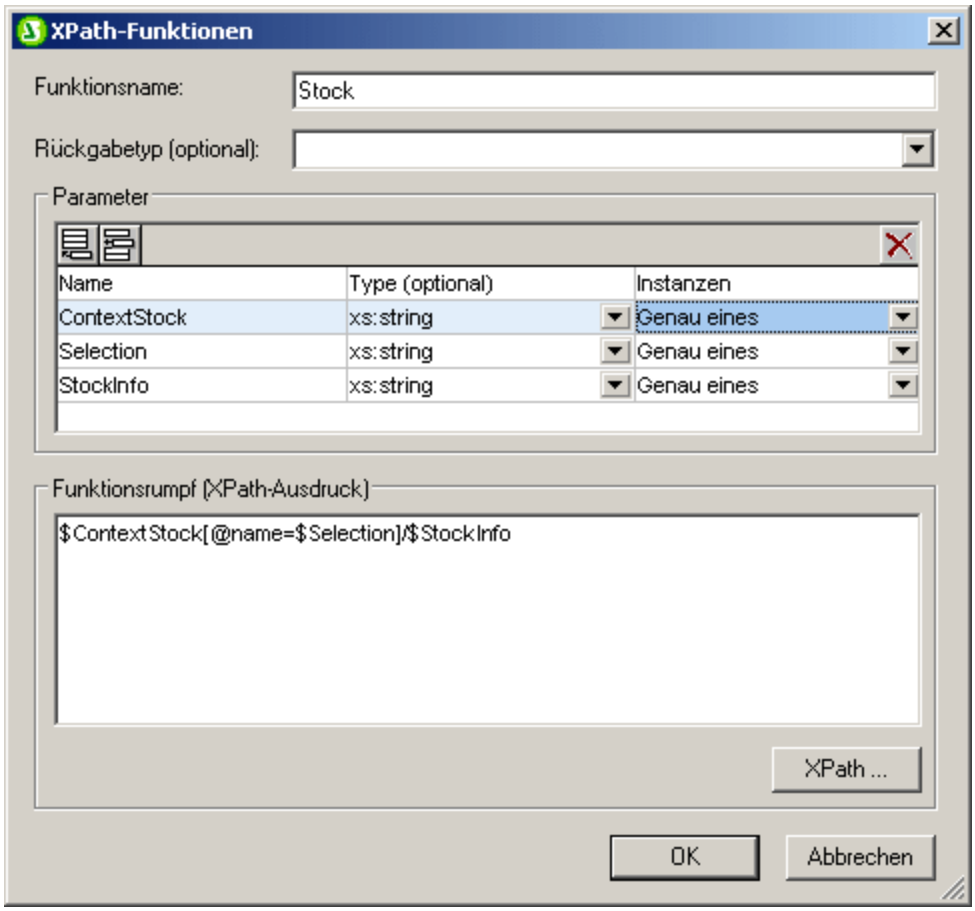
Wenn diese Funktion den Namen `Factorial()` hätte, dann könnte der Faktor von beispielsweise `6` berechnet werden, indem die Funktion mit `sps:Factorial(6)` aufgerufen würde.

9.4.3.2 Parameter und Nodes

Wenn Parameter in XPath-Funktionen, die Nodes auffinden, verwendet werden, ist es wichtig zu beachten, dass die Funktion keinen Kontext-Node hat, ganz gleich von wo im Design sie aufgerufen wird. Der Kontext-Node kann entweder durch den XPath-Ausdruck, der zur Definition der Funktion verwendet wird, (das heißt im Bereich "Funktionsrumpf") oder durch den XPath-Ausdruck, der zum Aufrufen der XPath-Funktion verwendet wird, übergeben werden. Im letzteren Fall kann der Kontext über Argumente im Funktionsaufruf übergeben werden.

© 2019-2025 Altova GmbH

Betrachten Sie die benutzerdefinierte XPath-Funktion `Stock()`, die, wie in der Abbildung unten zu sehen ist, mit drei Parametern definiert ist.



Die Definition im Funktionsrumpf ist `$ContextStock[@name=$Selection]/$StockInfo`. Diese verwendet die drei Parameter, enthält jedoch keine Information zum Kontext-Node. Die Information zum Kontext-Node kann vom XPath-Ausdruck bezogen werden, der die Funktion aufruft, beispielsweise auf diese Art:

```
sps:Stock( $XML/Trades/Stock, $XML/Trades/Selection/Stock, @name )
```

Der Funktionsaufruf hat drei Argumente, von denen jedes entweder Informationen zum Kontext-Node oder Information zur Node-Lokalisierung bietet. Alternativ dazu kann der folgende XPath-Ausdruck als Funktionsaufruf verwendet werden. Er liefert dieselben Ergebnisse:

```
sps:Stock( /Trades/Stock, /Trades/Selection/Stock, @name )
 sps:Stock( /Trades/Stock, //Selection/Stock, @name )
```

Die Variable `$XML`, die die Dokument-Root zurückgibt, kann in Funktionsaufrufen von Design-Komponenten ausgelassen werden, weil im XPath-Ausdruck von Design-Komponenten der Kontext-Node bekannt ist.

Beachten Sie, dass es im obigen Funktionsaufruf drei Eingabeargumente gibt, die jeweils den drei für die benutzerdefinierte XPath-Funktion definierten Parametern entsprechen:

- `$ContextStock = $XML/Trades/Stock` (das Element `/Trades/Stock`)
- `$Selection = $XML/Trades/Selection/Stock` (das Element `/Trades/Selection/Stock`)

- `$StockInfo = @name`

Der XPath-Ausdruck in der Funktionsdefinition lautet:

```
$ContextStock[@name=$Selection]/$StockInfo
```

Wenn die Eingabeargumente ersetzt werden, wird aus dem XPath-Ausdruck in der Funktionsdefinition:

```
$XML/Trades/Stock[@name=$XML/Trades/Selection/Stock]/@name
```

Es wichtig zu beachten, dass die Nodesets an die Funktion übergeben werden und nicht die Text-Strings.

Der Kontext-Node und die Schritte zur Auffindung des Node werden auf diese Weise über Parameter an die Funktion übergeben. Anschließend kann die Funktion ausgewertet werden, um die erforderlichen Nodes zu finden und zurückzugeben.

© 2019-2025 Altova GmbH

9.5 Arbeiten mit Datumswerten

Wenn das Quelldokument Nodes für Datumswerte enthält, stehen Ihnen über die Datentypen `xs:date` oder `xs:dateTime` aus dem zugrundeliegenden XML-Schema die leistungsstarken XPath 2.0/3.0-Funktionen zur Bearbeitung von Datum und Uhrzeit zur Verfügung (*siehe Beispiele unten*399). StyleVision bietet zur Unterstützung der Datentypen `xs:date` oder `xs:dateTime` über die Funktion Werteformatierung333 eine breite Palette an Funktionen zur Datenformatierung399.

Anmerkung: Datums- und Uhrzeitwerte können mit XPath 1.0 nicht bearbeitet werden. Sie können jedoch auch mit XPath 1.0 die Datumswerte mit Hilfe der Werteformatierung formatieren399.

Datumsberechnungen mit XPath 2.0

Datumswerte können in automatischen Berechnungen257 mit Hilfe von XPath 2.0-Ausdrücken bearbeitet werden. Im Folgenden finden Sie einige Beispiele dafür, welche Ergebnisse Sie mit XPath 2.0 Ausdrücken erhalten können.

- Mit Hilfe der XPath 2.0-Funktionen `current-date()` und `current-dateTime()` können Sie das aktuelle Datum bzw. das aktuelle Datum und die aktuelle Uhrzeit erhalten.
- Datumswerte können voneinander subtrahiert werden. Z.B.: `current-date() - DueDate` hätte als Ergebnis den Wert `xdt:dayTimeDuration`; z,B. den Wert `P24D`, d.h. eine positive Differenz von 24 Tagen.
- Mit Hilfe von XPath 2.0-Funktionen können Zeiteinheiten aus einer Zeitdauer extrahiert werden. Das Ergebnis von: `days-from-duration(xdt:dayTimeDuration('P24D'))` wäre die Ganzzahl `24`.

Hier sehen Sie einen XPath 2.0 Ausdruck in einer automatischen Berechnung. Berechnet wird ein Zinssatz von 4 % für einen ausstehenden Betrag pro Tag. Das Ergebnis ist die Summe des ursprünglichen Betrags plus die angefallenen Zinsen:

```
if      (current-date() gt DueDate)
then  (round-half-to-even(InvoiceAmount +
        (InvoiceAmount*0.04 div 360 *
          days-from-duration((current-date() - DueDate))), 2))
else  InvoiceAmount
```

Eine solche Berechnung wäre mit XPath 2.0 nur dann möglich, wenn das Element `DueDate` z.B. als Datentyp `xs:date` definiert wäre und wenn der Inhalt des Elements in lexikalisch korrekter Form eingegeben würde, also `JJJJ-MM-TT[±HH:MM]`, wobei die Zeitzonenkomponente (durch das Präfix ±) optional ist.

9.5.1 Formatieren von Datumswerten

Ein Datum wird in einem XML-Dokument in dem Format gespeichert, das für den Datentyp dieses Node spezifisch ist. So hat z.B. der Wert eines Node `xs:date` das Format `JJJJ-MM-TT[±HH:MM]`, während der Wert eines Node `xs:dateTime` das Format `JJJJ-MM-TTTHH:MM:SS[±HH:MM]` hat. Diese Formate sind die lexikalische Darstellung dieser Daten. Standardmäßig ist es die lexikalische Darstellung der in der Authentic-Ansicht und der Ausgabe angezeigten Daten. Im SPS kann jedoch die Werteformatierungsfunktion verwendet

werden, um Daten in der Authentic-Ansicht und in einigen Fällen optional in der Ausgabe in Alternativformaten anzuzeigen.

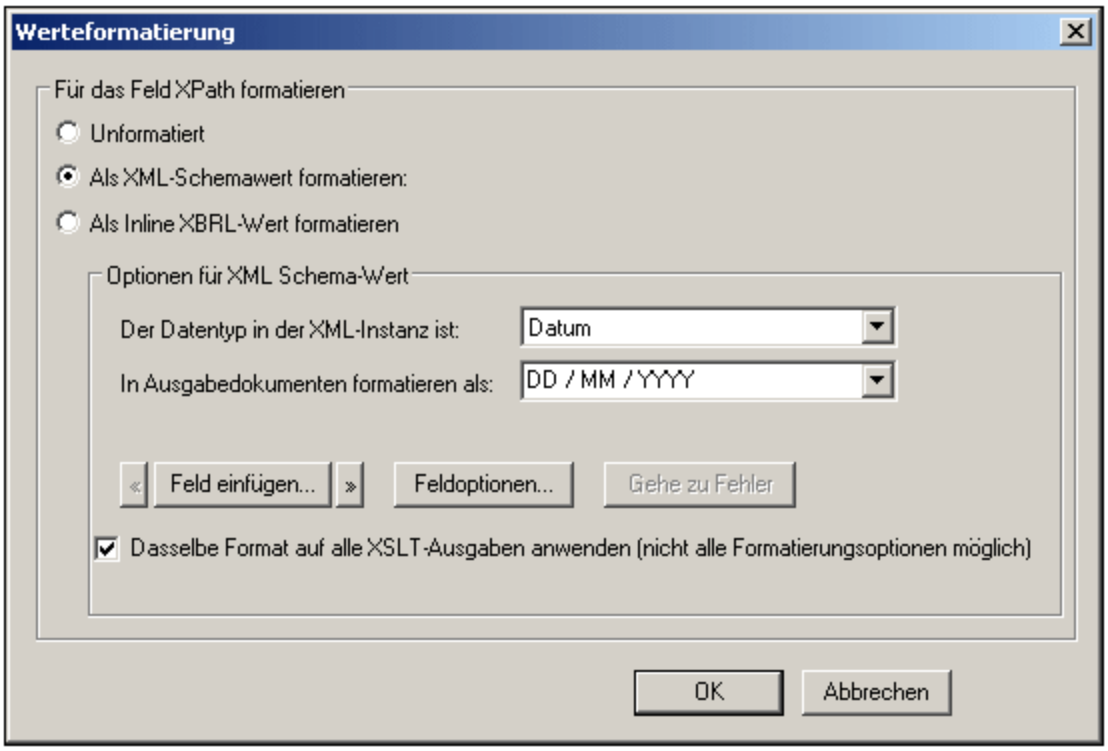
Mit Hilfe der Werteformatierung können Sie für die folgenden Datentypen benutzerdefinierte Formate für Nodes und automatische Berechnungen definieren:

- `xs:date`
- `xs:dateTime`
- `xs:duration`
- `xs:gYear`
- `xs:gYearMonth`
- `xs:gMonth`
- `xs:gMonthDay`
- `xs:gDay`

Verwendung der Werteformatierung zum Formatieren von Datums-Nodes

Um Datumswerten ein anderes als das lexikalische Format des Datums-Node zu geben, gehen Sie folgendermaßen vor:

1. Wählen Sie auf den Platzhalter `(Inhalt)` oder das Eingabefeld des Node aus. Beachten Sie, dass die Werteformatierung nur auf Nodes angewendet werden kann, die **als Inhalt und nicht als Eingabefeld erstellt wurden.**
2. Wählen Sie in der Seitenleiste "Eigenschaften" den Eintrag `XPath` und anschließend die Eigenschaftsgruppe *XPath* aus. Klicken Sie nun auf die Bearbeiten Schaltfläche [...] der Eigenschaft `Werteformatierung`. Daraufhin erscheint das Dialogfeld "Werteformatierung" (*Abbildung unten*).



 © 2019-2025 Altova GmbH

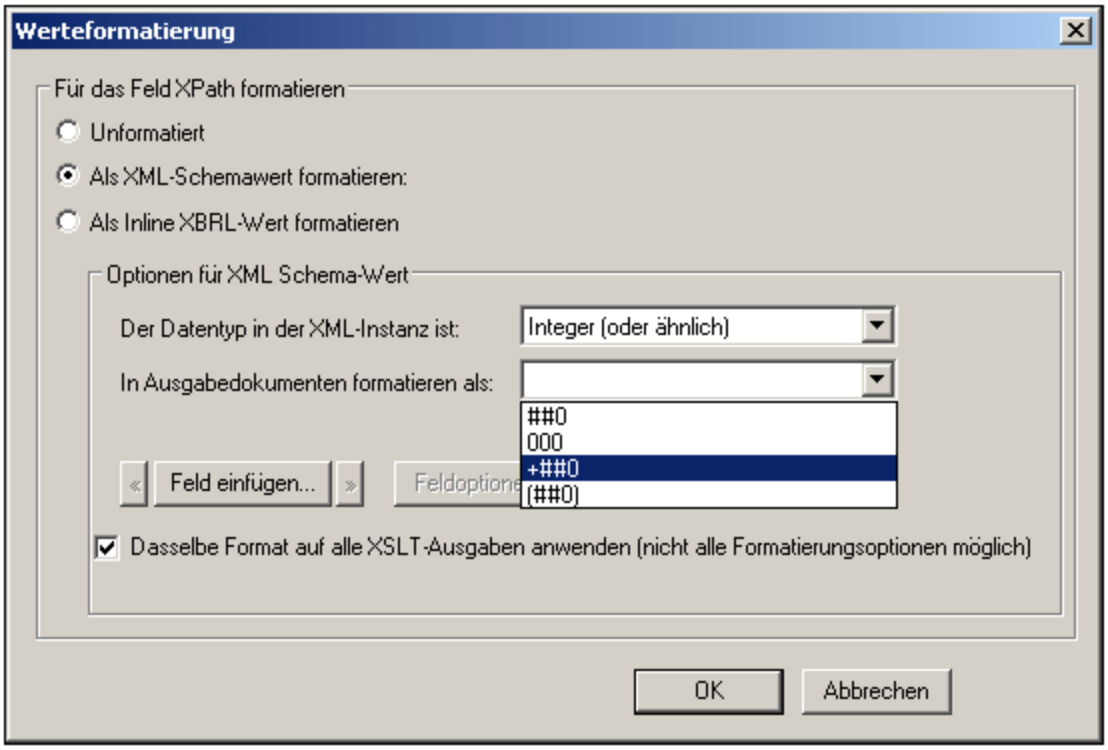
Standardmäßig ist das Optionsfeld "Unformatiert" (das lexikalische Standardformat für den Datentyp des Node) ausgewählt.

3. Um ein Alternativformat zu definieren, aktivieren Sie die Schaltfläche "Formatiert".
4. Nun können Sie aus der Dropdown-Liste (*Abbildung unten*) ein vordefiniertes Datumsformat auswählen, oder im Eingabefeld der Auswahlliste Ihr eigenes Format definieren. Nähere Informationen zur Syntax, die beim Definieren Ihres eigenen Formats zu verwenden ist, finden Sie unter Syntax der Werteformatierung 336.

Verwendung der Werteformatierung zum Formatieren von automatischen Berechnungen

Wenn das Ergebnis einer automatischen Berechnung ein Wert ist, bei dem es sich um ein lexikalisches Datumsformat handelt, können Sie das Ergebnis mit Hilfe der Werteformatierung anzeigen. Gehen Sie dazu folgendermaßen vor:

1. Wählen Sie die automatische Berechnung im Design aus.
2. Wählen Sie in der Seitenleiste "Eigenschaften" den Eintrag `Inhalt` aus, klicken Sie anschließend auf die Eigenschaftsgruppe *XPath*. Klicken Sie nun auf die Bearbeiten-Schaltfläche ... der Eigenschaft `Werteformatierung`. (*Abbildung unten*).



Standardmäßig ist das Optionsfeld "Unformatiert" aktiviert.

3. Um ein anderes Format zu definieren, wählen Sie das Optionsfeld "..formatieren" aus.
4. Wählen Sie im Bereich "Optionen für XML-Schema-Wert" in der Datentyp-Auswahlliste den Datentyp für das Datum aus, als das das Ergebnis der automatischen Berechnung angezeigt werden soll. Nun können Sie in der Auswahlliste "...formatiert" ein vordefiniertes Format auswählen (welche Optionen zur Verfügung stehen, hängt vom ausgewählten Datentyp ab) oder im Eingabefeld der Auswahlliste Ihr eigenes Format definieren. Nähere Informationen zur Syntax für die Definition Ihres eigenen Formats finden Sie unter Syntax der Werteformatierung 336.

Anwenden der Werteformatierung auf die Ausgabe

Die von Ihnen definierte Werteformatierung wird auf die Authentic-Ansicht angewendet. Zusätzlich dazu können einige Werteformatierungsdefinitionen - aber nicht alle - auch auf die HTML-Ausgabe angewendet werden. Aktivieren Sie dazu das Kontrollkästchen "Dasselbe Format auf alle XSLT-Ausgaben anwenden". Wenn diese Option deaktiviert oder nicht verfügbar ist, wird die Werteformatierung nur in der Authentic-Ansicht angezeigt; in der Ausgabe wird der Wert im lexikalischen Format (für Nodes) angezeigt oder im Fall von automatischen Berechnungen im Format für das Ergebnis der automatischen Berechnung.

© 2019-2025 Altova GmbH

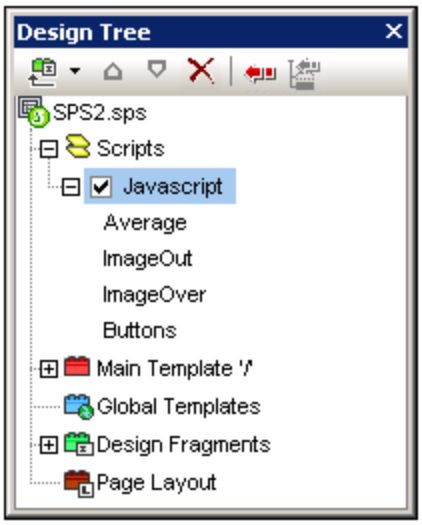
9.6 Verwenden von Skripts

Sie können in StyleVision in einem JavaScript Editor (der auf einem Register in der Design-Ansicht zur Verfügung steht) für jedes SPS JavaScript-Funktionen definieren. Die auf diese Art erstellten Funktionsdefinitionen sind im "Header" des HTML-Dokuments gespeichert und können über den "Body" des HTML-Dokuments aufgerufen werden. Diese Funktionen eignen sich in folgenden Fällen:

- Um mit Hilfe mehrerer Skript-Anweisungen ein komplexes Ergebnis zu erhalten. In diesem Fall sollten alle erforderlichen Skripts an einer Stelle (im Header) als separate Funktionen erstellt und anschließend im Design-Dokument referenziert werden.
- Wenn Sie ein bestimmtes Skript an mehreren Stellen im Design-Dokument verwenden möchten.

Die Definition von Funktionen im JavaScript Editor wird im Unterabschnitt Definition von JavaScript-Funktionen 404 beschrieben.

Auf der Benutzeroberfläche werden alle JavaScript-Funktionen, die im JavaScript Editor für ein bestimmtes SPS definiert wurden, im Fenster "Design-Struktur" unter dem Eintrag "Skripts" aufgelistet (*Abbildung unten*). In der Abbildung unten sehen Sie, dass derzeit im aktiven SPS vier JavaScript-Funktionen definiert sind: `Average`, `ImageOut`, `ImageOver` und `Buttons`.



Die im JavaScript Editor definierten Funktionen stehen auf der Benutzeroberfläche als Event Handler zur Verfügung. Wenn eine Komponente in der Design-Ansicht ausgewählt wird, kann jede der definierten Funktionen einer Event Handler-Eigenschaft in der Seitenleiste "Eigenschaften" in der Eigenschaftsgruppe "Ereignis" zugewiesen werden. Informationen zum Zuweisen einer JavaScript-Funktion zu einem Event Handler finden Sie im Abschnitt Zuweisen von Funktionen zu Event Handlern 405.

Skripts in modularen SPSs

Wenn ein SPS Modul zu einem anderen SPS Modul hinzugefügt wird 216, stehen die Skripts im hinzugefügten Modul im referenzierenden SPS zur Verfügung und können über die Seitenleiste "Eigenschaften" für Komponenten im referenzierenden SPS als Event Handler verwendet werden. Nähere Informationen zur Verwendung von modularen SPSs finden Sie im Abschnitt Modulare SPSs 216.

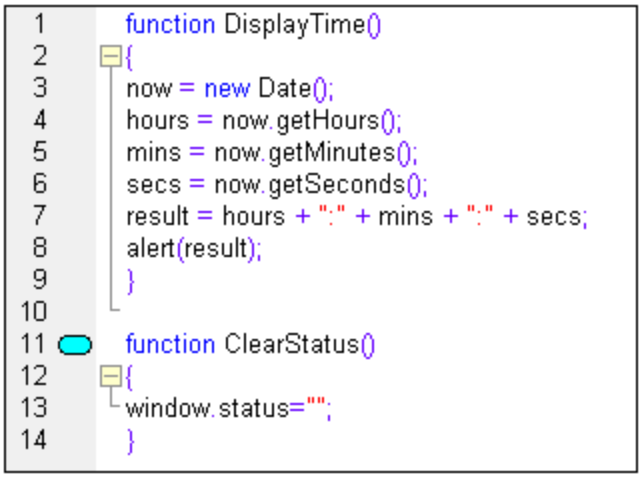
9.6.1 Definieren von JavaScript-Funktionen

Gehen Sie bei der Definition von JavaScript-Funktionen folgendermaßen vor:

1. Wechseln Sie in der Design-Ansicht zum JavaScript Editor. Klicken Sie dazu auf das Register "Design" und wählen Sie "JavaScript" (*Abbildung unten*).

1.

2. Geben Sie im JavaScript Editor die Funktionsdefinitionen ein (*siehe Abbildung unten*).



```
1   function DisplayTime()
2   {
3   now = new Date();
4   hours = now.getHours();
5   mins = now.getMinutes();
6   secs = now.getSeconds();
7   result = hours + ":" + mins + ":" + secs;
8   alert(result);
9   }
10
11  function ClearStatus()
12  {
13  window.status="";
14  }
```

In der Abbildung oben sehen Sie die Definitionen von zwei JavaScript-Funktionen: `DisplayTime` und `ClearStatus`. Diese wurden für das aktive SPS beschrieben. Sie werden folgendermaßen in den Header-Abschnitt der HTML-Datei eingegeben:

```
<script language="javascript">

<!-- function DisplayTime()
{
   now = new Date();
   hours = now.getHours();
   mins = now.getMinutes();
   secs = now.getSeconds();
   result = hours + "." + mins + "." + secs;
   alert(result)
}

function ClearStatus()
{
   window.status="";
}
-->
```

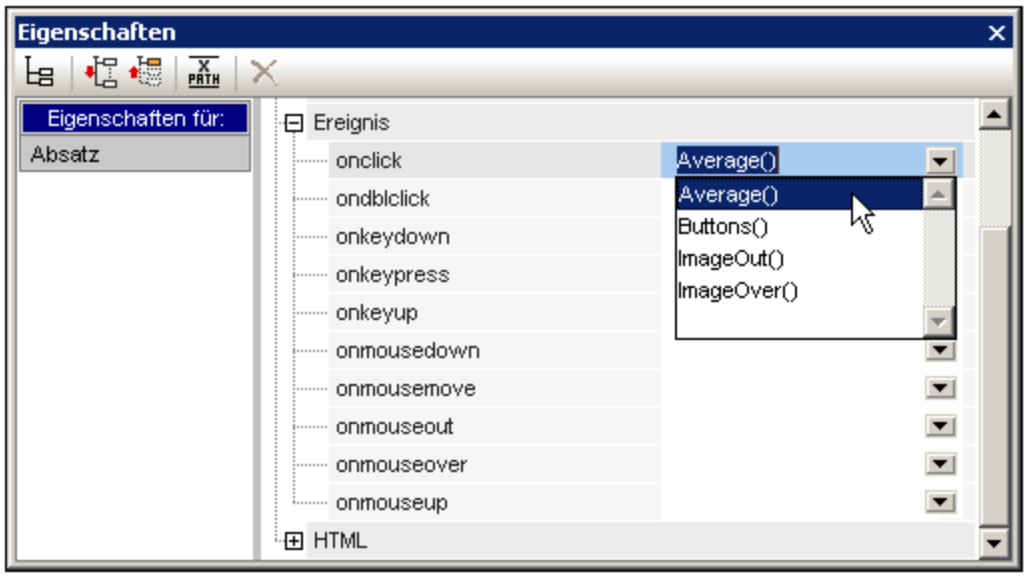

© 2019-2025 Altova GmbH

```
</script>
```

Diese Funktionen können nun von jeder Stelle im HTML-Dokument aus aufgerufen werden. In StyleVision stehen alle definierten Funktionen als Optionen zur Verfügung, die einer Event Handler-Eigenschaft in der Seitenleiste "Eigenschaften" in der Ereignisgruppe "*Ereignis*" zugewiesen werden können. Nähere Informationen siehe Zuweisen von Funktionen zu Event Handlern 405.

9.6.2 Zuweisen von Funktionen als Event Handler

Auf der Benutzeroberfläche von StyleVision können Sie Ereignissen, die bei der HTML-Wiedergabe von SPS-Komponenten vorkommen, JavaScript-Funktionen als Ereignis-Handler zuweisen. Diese Ereignis-Handler werden in der HTML-Ausgabedatei verwendet. Der Ereignis-Handler für ein verfügbares Ereignis - z.B. `onclick` - wird durch Zuweisung einer globalen Funktion als Ereignis-Handler definiert. In der Seitenleiste "Eigenschaften" stehen die im JavaScript Editor definierten globalen Funktionen als Ereignis-Handler in den Dropdown-Feldern der einzelnen Ereignisse in der Eigenschaftsgruppe *Ereignisse* für die gewählte Komponente zur Verfügung (*Abbildung unten*).



Um einem Ereignis-Handler eine Funktion zuzuweisen, gehen Sie folgendermaßen vor:

1. Wählen Sie die Komponente im SPS, für die der Ereignis-Handler definiert werden soll. Dabei kann es sich um einen Node oder Inhalt jeder Art - dynamischen oder statischen - handeln.
2. Wählen Sie in der Seitenleiste "Eigenschaften" die Gruppe "*Ereignis*". Daraufhin werden die verfügbaren Ereignisse in der Spalte "Attribute" angezeigt (*Abbildung oben*).
3. Klicken Sie in der Spalte "Wert" des gewünschten Ereignisses auf den Abwärtspfeil der Auswahlliste. Daraufhin erscheint eine Liste aller im JavaScript Editor definierten Funktionen.
4. Wählen Sie in der Dropdown-Liste die gewünschte Funktion als Ereignis-Handler für dieses Ereignis aus.

In der HTML-Ausgabe wird bei Auslösung dieses Ereignisses bei der Komponente, für die der Ereignis-Handler definiert wurde, die JavaScript-Funktion ausgeführt.

9.6.3 Externe JavaScript-Dateien

Ein SPS kann externe JavaScript-Dateien auf zwei Arten aufrufen:

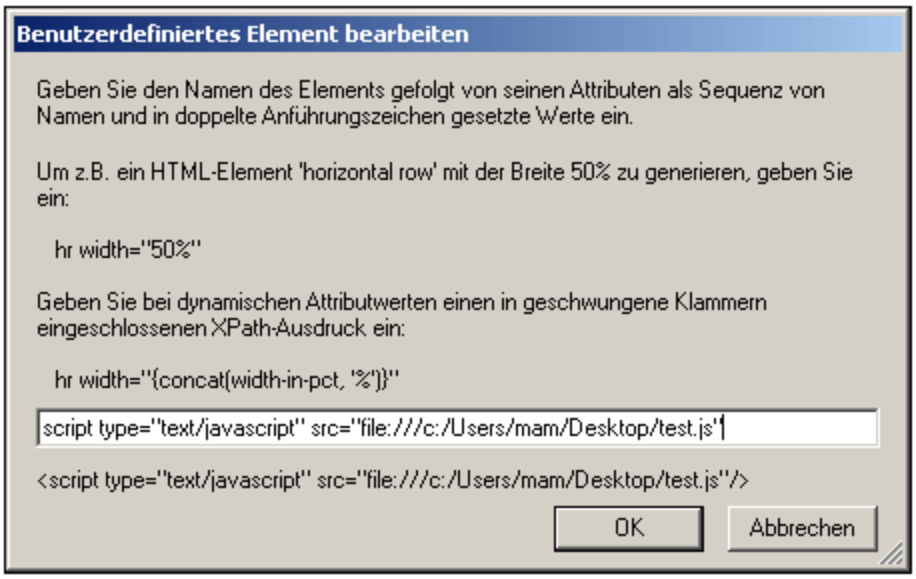
1. Durch Erstellung eines benutzerdefinierten Elements oder XML-Blocks[406]. Diese Design-Objekte können ein `SCRIPT` Element enthalten, das die externe JavaScript-Datei aufruft. Beachten Sie, dass sich das benutzerdefinierte Element bzw. der benutzerdefinierte XML-Block innerhalb des `BODY` Elements des Designs (und daher innerhalb des `BODY`-Elements der HTML-Ausgabe und nicht innerhalb des `HEAD`-Elements) befindet.
2. Durch Hinzufügen eines Skripts im Javascript Editor[407], das die extern Datei aufruft. Ein Skript, das auf diese Art hinzugefügt wird, befindet sich im `HEAD` Element der HTML-Ausgabe.

Benutzerdefinierte Elemente und benutzerdefinierte XML-Blöcke

Externe JavaScript-Dateien können über benutzerdefinierte Elemente[123] und benutzerdefinierte XML-Blöcke[124] aufgerufen werden. Mit Hilfe dieser Mechanismen kann ein `SCRIPT` Element, das die externe JavaScript-Datei aufruft, an jeder beliebigen Stelle im `BODY` Element des HTML-Ausgabedokuments eingefügt werden.

Ein benutzerdefiniertes Element[123] könnte folgendermaßen eingefügt werden:

1. Platzieren Sie den Cursor an die Stelle im Design, an der das `SCRIPT` Element, das die JavaScript-Datei aufruft, eingefügt werden soll.
2. Wählen Sie im Menü **Einfügen** oder im Kontextmenü den Befehl zum Einfügen eines benutzerdefinierten Elements[123].



3. Geben Sie im Dialogfeld, das daraufhin angezeigt wird (*siehe Abbildung oben*), das `SCRIPT` Element wie oben gezeigt ein und geben Sie die URL der JavaScript-Datei als Wert des `src` Attributs des `SCRIPT` Elements an. Beispiel: `script type="text/javascript" src="file:///c:/Users/mam/Desktop/test.js"`
4. Klicken Sie zum Fertigstellen auf **OK**.

© 2019-2025 Altova GmbH

Sie können dasselbe Ergebnis auch mit einem benutzerdefinierten XML-Block[124] erzielen. Verwenden Sie dazu dasselbe oben beschriebene Verfahren wie bei benutzerdefinierten Elementen. Der einzige Unterschied ist, (i) dass anstelle eines benutzerdefinierten Elements[123] ein benutzerdefinierter XML-Block[124] eingefügt wird und dass (ii) das `SCRIPT` Element als kompletter XML-Block, d.h. mit einem Start- und einem End-Tag eingefügt wird.

JavaScript Editor

Mit Hilfe des JavaScript Editors[404] können Sie ein externes Skript in das `HEAD` Element der HTML-Ausgabe einfügen. Geben Sie dazu das folgende Skript-Fragment außerhalb aller anderen von Ihnen erstellten Funktionsdefinitionen in den JavaScript Editor ein.

```
var script = document.createElement('script');
script.type = 'text/javascript';
    script.src = 'file:///c:/Users/Desktop/test.js';
var head = document.getElementsByTagName('head')[0];
head.appendChild(script)
```

Die externe JavaScript-Datei, deren Pfad in der URL in `script.src` definiert ist, wird von einer Stelle innerhalb des `HEAD` Elements des HTML-Ausgabedokuments aufgerufen.

9.7 HTML-Import

In StyleVision können Sie ein HTML-Dokument importieren und anhand dieses Dokuments die folgenden Dokumente generieren:

- ein SPS Dokument basierend auf Design und Struktur der importierten HTML-Datei.
- ein XML-Schema, in dem HTML-Dokumentkomponenten als Schemaelemente oder Attribute erstellt werden. Optional dazu können im benutzerdefinierten Schema zusätzliche Elemente und Attribute, die nicht mit dem HTML-Dokument in Zusammenhang stehen, erstellt werden.
- ein XML-Dokument mit: (i) einer Struktur, die auf dem von Ihnen erstellten XML-Schema basiert und (ii) Inhalt aus der HTML-Datei.
- XSLT Stylesheets basierend auf dem Design in der Design-Ansicht.

Schritte bei der Konvertierung von HTML in XML

Der HTML-in-XML-Konvertierungsprozess besteht aus den folgenden Schritten:

1. Erstellen eines neuen SPS mittels HTML-Import 408 .Wenn eine HTML-Datei in StyleVision importiert wird, wird ein neues SPS-Dokument erstellt. Die HTML-Datei wird in der Design-Dokumentansicht mit Markup-Tags angezeigt. Gleichzeitig wird im Schemafenster eine Schema-Struktur mit einem einzigen Element - einem Dokumentelement mit dem Namen `BenutzerRoot` - erstellt. Dies ist das Schema, auf dem das SPS basiert. Der Inhalt des HTML-Dokuments und die Markup-Tags, die in der Design-Ansicht angezeigt werden, werden als statischer Inhalt in das SPS eingefügt.
2. Erstellen de Schemas und des SPS-Designs 410 . Erstellen Sie das Schema durch (i) Ziehen von Nodes aus dem HTML-Dokument an die gewünschte Stelle in der Schema-Struktur (im Fenster "Schema-Struktur"); und optionales (ii) Hinzufügen Ihrer eigenen Nodes zur Schema-Struktur. Im Design-Fenster wurde HTML-Inhalt verwendet, um in der Schema-Struktur Nodes zu erstellen. Dieser Inhalt wird nun innerhalb von Node-Tags angezeigt. HTML-Inhalt, der keinem Schema-Node entspricht, wird weiterhin ohne Schema-Node-Tags dargestellt.
3. Im Design-Dokument erfolgt nun die Zuweisung von Formatierung zu Nodes, genauere Definition von Verarbeitungsregeln oder das Hinzufügen von statischem Inhalt, falls erforderlich. Diese Änderungen wirken sich nur auf das SPS und die generierte XSLT-Datei aus, nicht aber auf das generierte Schema oder die XML-Datei.
4. Nachdem Sie die Schema-Struktur und das Design des SPS fertig gestellt haben, können Sie die folgenden Dateien generieren und speichern 414 :

   - ein XML-Schema, das der von Ihnen erstellten Schema-Struktur entspricht;
   - eine XML-Datei mit einer Struktur, die auf dem Schema und dem Inhalt für Schema-Nodes basiert, die im SPS -Design mit dem Platzhalter `(Inhalt)` erstellt werden;
   - ein SPS (`.sps` Datei) und/oder ein XSLT Stylesheet basierend auf Ihrem Design.

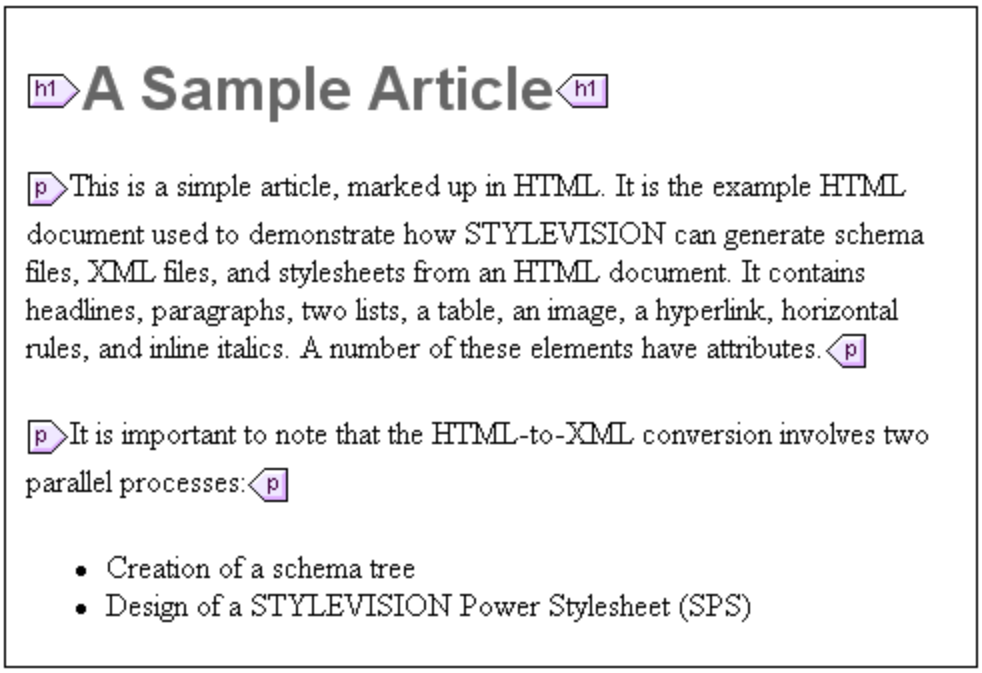
9.7.1 Erstellen eines neuen SPS anhand einer HTML-Datei

Um anhand einer HTML-Datei ein neues SPS zu erstellen, gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Datei | Neu | Neu von HTML-Datei**.
2. Suchen Sie im Dialogfeld "Öffnen" die gewünschte HTML-Datei. Wählen Sie sie aus und klicken Sie auf **Öffnen**.
3. Sie werden gefragt, ob relative Pfade in absolute Pfade konvertiert werden sollen. Wählen Sie die gewünschte Option, indem Sie entweder auf **Ja** oder **Nein** klicken.

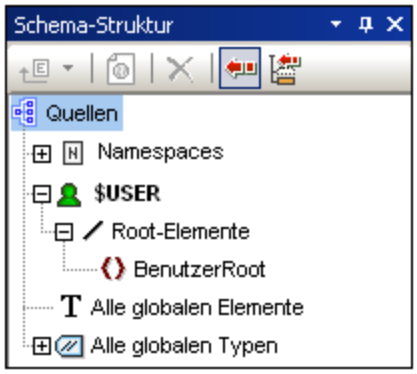
© 2019-2025 Altova GmbH

Daraufhin wird ein neues SPS-Dokument erstellt. Das Dokument wird in der Design-Ansicht angezeigt und enthält die Markup-Tags mit den in StyleVision verfügbaren vordefinierten HTML-Formatierungen (*siehe Abbildung unten*).



Beachten Sie, dass das HTML-Dokument in der Hauptvorlage angezeigt wird. Es gibt keine globale Vorlage.

In der Seitenleiste "Schema-Struktur" wird ein benutzerdefiniertes Schema mit einem Root-Element (Dokumentelement) mit dem Namen `BenutzerRoot` erstellt (*siehe Abbildung unten).*



Beachten Sie, dass es in der Liste "Alle globalen Elemente" kein globales Element gibt.

Struktur und Design des SPS

Das SPS enthält eine einzige Vorlage - die Hauptvorlage - die auf den Dokument-Node einer temporären internen XML-Datei angewendet wird. Dieses XML-Dokument hat die Struktur des benutzerdefinierten Schemas, das im Fenster "Schema-Struktur" erstellt wurde. In der Design-Ansicht werden die Komponenten des HTML-Dokuments in der Hauptvorlage **zu diesem Zeitpunkt** als statische Komponenten in das SPS

eingefügt. Diese HTML-Komponenten werden in der Authentic-Ansicht als nicht editierbarer nicht-XML-Inhalt dargestellt. Das XSLT Stylesheet enthält diese HTML-Komponente als literale Ergebniselemente. Das Schema hat zu diesem Zeitpunkt nur das Dokumentelement `Root`; daher enthält das temporäre interne XML-Dokument nur das Dokumentelement `Root` und keine Child-Elemente.

Um bestimmte Abschnitte aus dem HTML-Dokument im benutzerdefinierten Schema als Elemente und Attribute zu erstellen, gibt es zwei Möglichkeiten:

1. Durch **Konvertierung** der Auswahl in ein Element oder Attribut. Die Node-Tags werden im Design innerhalb von Tags mit einem Platzhalter `(Inhalt)` eingefügt. Im Schema wird ein Element oder Attribut erstellt. Im XML-Dokument wird die Auswahl in den Textinhalt des Schema-Node konvertiert, welcher im XML-Dokument erstellt wird. Der Inhalt des im XML-Dokument erstellten Node wird dynamisch in das Ausgabedokument, das mit dem SPS generiert wird, eingefügt.
2. Durch **Einschließen** der Auswahl in ein Element oder Attribut. Im Design wird die Auswahl innerhalb der Node-Tags eingeschlossen; es wird kein Platzhalter `(Inhalt)` eingefügt. D.h. die Auswahl ist im SPS Design als statischer Inhalt vorhanden. Im Schema wird ein Element oder Attribut erstellt. Im XML-Dokument wird ein Node erstellt, der jedoch leer ist. Der statische Text innerhalb der Schema Node-Tags im Design wird ausgegeben; es wird für diesen Node kein dynamischer Inhalt ausgegeben, es sei denn, es wird für diesen Node explizit ein Platzhalter `(Inhalt)` in das Design eingefügt.

Der Platzhalter `(Inhalt)` gibt an, wo im Design Daten aus dem XML-Dokument (in der Ausgabe) angezeigt werden und (in der Authentic-Ansicht) bearbeitet werden können.

9.7.2 Erstellen des Schemas und SPS Designs

Das Schema wird durch Ziehen ausgewählter Nodes aus der Design-Ansicht in das benutzerdefinierte Schema erstellt. Dies geschieht Node für Node. Die Auswahl wird auf einen Node in der Schemastruktur gezogen (relativ zu welchem der neue Node entweder als Child-Element oder gleichrangiges Element erstellt werden soll). Sie wählen den zu erstellenden Node-Typ aus (Element oder Attribut) und ob die Auswahl in den neuen Node konvertiert oder in ihn eingeschlossen werden soll.

Die Auswahl

Bei der Auswahl in der Design-Ansicht kann es sich um eine der folgenden handeln:

- einen Node im HTML-Dokument.
- einen Textstring in einem Node.
- benachbarte Textstrings über Nodes hinweg.
- ein Bild.
- einen Link.
- eine Tabelle.
- eine Liste.
- eine beliebige Kombination aus den oben genannten Möglichkeiten.

In diesem Abschnitt wird der allgemeine Vorgang bei einer Auswahl beschrieben. Die Sonderfälle Tabelle und Listen werden im Abschnitt Erstellen von Tabellen und Listen als Elemente/Attribute 412 näher behandelt.

Um eine Auswahl zu treffen, klicken Sie auf eine HTML-Dokumentkomponente oder markieren Sie den gewünschten Textstring. Bei Auswahl mehrerer Komponenten klicken Sie an die gewünschte Stelle und ziehen Sie die Maus über die gewünschten Komponenten um den Abschnitt zu markieren. Beachten Sie, dass die

 © 2019-2025 Altova GmbH

Auswahl in StyleVision am Beginn und Ende der Auswahl auf Elemente auf höheren Ebenen erweitert wird, bis das erste und letzte ausgewählte Element zum selben Parent gehören.

Position in der Schemastruktur

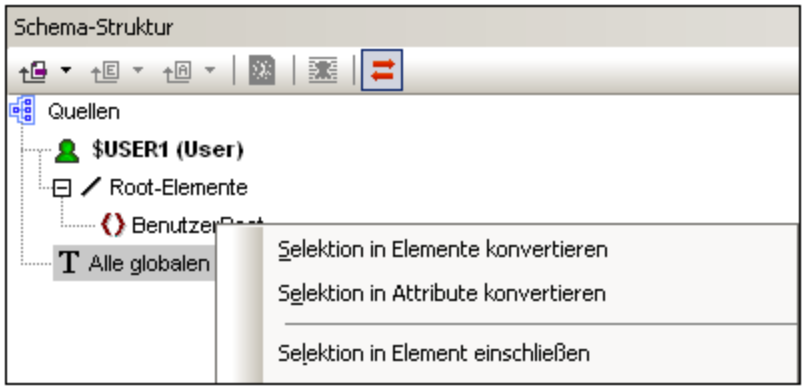
Wenn Sie die Auswahl über den gewünschten Node der Schemastruktur ziehen, verwandelt sich der Mauszeiger in eines der folgenden Symbole:

- Wenn Sie die Maustaste loslassen, wenn das Symbol "Als gleichrangiges Element erstellen" erscheint, wird die Auswahl als gleichrangiger Node zum Node, in den die Auswahl gezogen wird, erstellt.
- Wenn Sie die Maustaste loslassen, wenn das Symbol "Als Child erstellen" erscheint, wird die Auswahl als Child Node des Nodes erstellt, in den die Auswahl gezogen wird.

Wählen Sie den Node, in den die Auswahl gezogen wird, nach dem Kriterium aus, ob die Auswahl als gleichrangiger Node oder als Child dieses Node erstellt werden soll.

Auswählen, wie der Node erstellt wird

Beim Loslassen der Maustaste (*siehe vorheriger Abschnitt)* erscheint ein Kontextmenü (*siehe Abbildung unten*), in dem Sie zwischen zwei Optionen wählen können: (i) ob der Node als Element oder als Attribut erstellt werden soll; (ii) ob die Auswahl in den Node konvertiert werden soll oder ob die Auswahl einfach in den Node eingeschlossen werden soll.



Beachten Sie bitte die folgenden Punkte:

- Wenn eine Auswahl in einen Node (ein Element oder ein Attribut) konvertiert wird, ersetzen die Node Tags zusammen mit dem darin enthaltenen Platzhalter `(Inhalt)` die Auswahl im Design. Der Textinhalt der Auswahl wird im Design und in der Ausgabe aus dem statischen Inhalt entfernt. In der Ausgabe wird der Text der Auswahl als dynamischer Inhalt des Node im XML-Dokument angezeigt.
- Wenn ein HTML Node in einen XML Node konvertiert wird, werden die Tags des XML Node innerhalb der HTML Node Tags eingefügt.
- Wenn eine Auswahl (einschließlich der Auswahl der HTML Nodes) in einen XML Node eingeschlossen wird, werden die XML Node Tags vor und hinter die Auswahl platziert. Im Design und in der Ausgabe bleibt der Textinhalt der Auswahl als statischer Text erhalten.
- Die eingefügten Node Tags werden mit dem erforderlichen Pfad eingefügt (d.h. mit den Tags der übergeordneten Nodes, die einen Pfad relativ zum enthaltenden Node bilden). Je nach dem Kontext des Node im Design ist der Pfad absolut oder relativ.

- Eine Beschreibung zum Erstellen von Nodes anhand einer Tabellen- oder Listenauswahl finden Sie im Abschnitt Erstellen von Tabellen und Listen als Elemente/Attribute 412.

Hinzufügen und Löschen von Nodes im Schema

Sie können zusätzliche Nodes (welche nicht auf einer HTML-Auswahl basieren) zum benutzerdefinierten Schema hinzufügen. Rechtsklicken Sie dazu auf einen Node und wählen Sie den gewünschten Befehl aus dem Kontextmenü aus. Alternativ dazu können Sie auch die Symbolleistenschaltflächen der Seitenleiste "Schema-Struktur" verwenden.

Um einen Node zu löschen, wählen Sie ihn aus und verwenden Sie anschließend entweder das Kontextmenü oder die Symbolleistenschaltfläche. Beachten Sie jedoch: Wenn ein Node gelöscht wird, könnten einige Pfade im Design ungültig werden.

Ändern des Designs

Sie können die Struktur des Designs durch Ziehen der Komponenten und Einfügen von statischen und dynamischen Komponenten ändern. Stile können auch durch Verwendung der verschiedenen Stilfunktionen von StyleVision geändert werden.

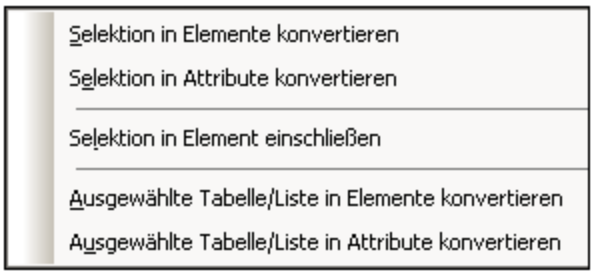
9.7.3 Erstellen von Tabellen und Listen als Elemente/Attribute

Tabellen und Listen im HTML-Dokument können in Element- oder Attribut-Nodes im XML-Schema konvertiert werden, sodass die Tabellen- oder Listenstruktur im Schema erhalten bleibt.

Konvertieren einer Tabelle in Elemente/Attribute

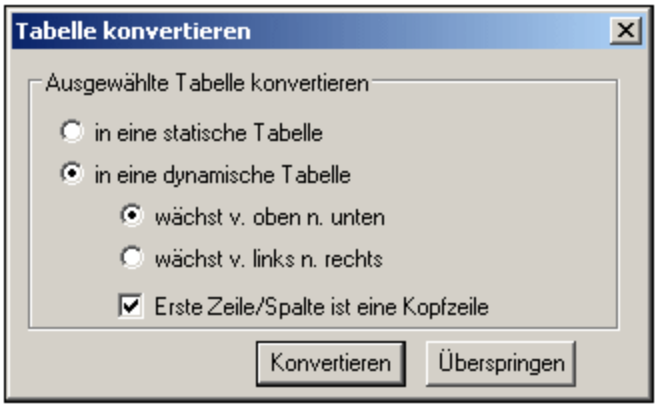
Um eine Tabelle in Schema-Nodes zu konvertieren, gehen Sie folgendermaßen vor:

1. Wählen Sie die HTML-Tabelle aus, indem Sie Text darin markieren.
2. Ziehen Sie sie in den Node in der Schemastruktur, als dessen gleichrangiges oder untergeordnetes Element Sie ihn erstellen möchten.
3. Lassen Sie die Maustaste los, wenn das Symbol "Als gleichrangiges Element erstellen" oder "Als Child erstellen" erscheint.
4. Wählen Sie nun im Kontextmenü (*Abbildung unten*), je nachdem, ob der Inhalt von Tabellenzellen als Elemente oder Attribute erstellt werden soll, entweder den Befehl **Ausgewählte Tabelle/Liste in Elemente konvertieren** oder **Ausgewählte Tabelle/Liste in Attribute konvertieren** aus.



5. Wählen Sie im Dialogfeld "Tabelle konvertieren" (*Abbildung unten*) aus, ob die im SPS erstellte Tabelle als statische oder dynamische Tabelle erstellt werden soll.

 © 2019-2025 Altova GmbH



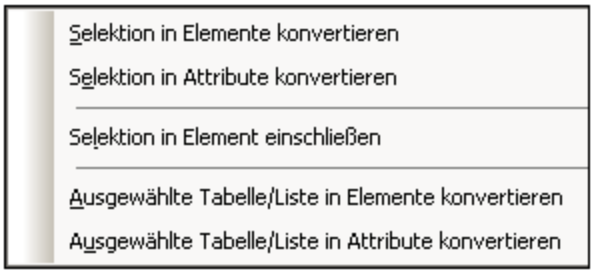
Wenn Sie die Option **statische Tabelle** auswählen, wird für jede Zelle in der Tabelle ein Schema-Node erstellt. Im Design wird jeder Node mit dem Platzhalter `(Inhalt)` erstellt. Die Daten in den Tabellenzellen werden in das temporäre XML-Dokument (und in das generierte XML-Dokument) kopiert. Die Option **dynamische Tabelle** ist verfügbar, wenn die Struktur aller Zellen in der Tabelle identisch ist. Wenn die Tabelle im SPS erstellt wird, werden die Zeilen der dynamischen Tabelle durch eine einzige Zeile im Design dargestellt (da jede Zeile dieselbe Struktur hat). Die Tabellendaten werden in die XML-Datei kopiert. Die dynamische Tabelle kann von oben nach unten wachsen (die Zeilen werden vertikal übereinander angeordnet) oder von links nach rechts (die Zeilen werden zu Spalten und werden von links nach rechts angeordnet). Wenn Sie angeben, dass die erste Zeile/Spalte die Überschrift sein soll, (i) so wird eine Kopfzeile in das Design eingefügt, die die Spaltenüberschriften als statischen Text enthält; und (ii) die Schemaelement/Attribut-Nodes erhalten den Text der Kopfzeile als Namen. Wenn die erste Zeile/Spalte nicht als Kopfzeile definiert wird, so wird keine Kopfzeile in das Design eingefügt.

6. Klicken Sie nach Auswahl der gewünschten Option(en) zum Fertigstellen auf **Konvertieren**.

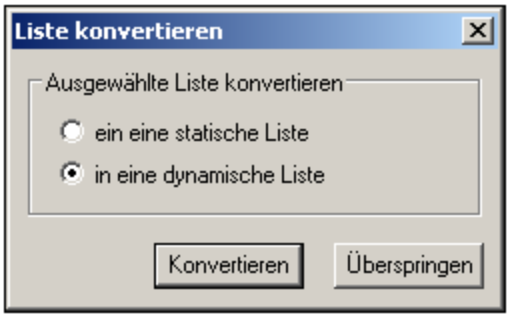
Konvertieren einer Liste in Elemente/Attribute

Um eine Liste in Schema-Nodes zu konvertieren, gehen Sie folgendermaßen vor:

1. Wählen Sie die HTML-Liste durch Markieren von Text darin aus.
2. Ziehen Sie sie in den Node in der Schemastruktur, als dessen gleichrangiges oder untergeordnetes Element Sie ihn erstellen möchten.
7. Lassen Sie die Maustaste los, wenn das Symbol "Als gleichrangiges Element erstellen" oder "Als Child erstellen" erscheint.
8. Wählen Sie nun im Kontextmenü (*Abbildung unten*), je nachdem, ob der Inhalt von Tabellenzellen als Elemente oder Attribute erstellt werden soll, entweder den Befehl **Ausgewählte Tabelle/Liste in Elemente konvertieren** oder **Ausgewählte Tabelle/Liste in Attribute konvertieren** aus.



3. Wählen Sie im Dialogfeld "Liste konvertieren" (*Abbildung unten*) aus, ob die im SPS erstellte Tabelle als statische oder dynamische Tabelle erstellt werden soll.



Wenn Sie die Option **statische Liste** auswählen, wird für jeden Listeneintrag in der Tabelle ein Schema-Node erstellt. Im Design wird jeder Node mit dem Text des HTML-Listeneintrags, der als statischer Inhalt des Listeneintrags eingefügt wird, erstellt. Bei Auswahl der Option **dynamische Liste** wird jeder Listeneintrag im Design durch einen einzelnen Listen-Node dargestellt. Im Design wird das Listeneintragselement mit dem Platzhalter `(Inhalt)` eingefügt.

4. Klicken Sie nach Auswahl der gewünschten Option(en) zum Fertigstellen auf **Konvertieren.**

9.7.4 Generieren der Ausgabedateien

Nach Fertigstellung des SPS können Sie mit Hilfe des Befehls **Datei | Generierte Dateien speichern** die folgenden Ausgabedateien erzeugen:

- ein generiertes benutzerdefiniertes Schema. Dies ist das Schema, das Sie in der Seitenleiste "Schema-Struktur" erstellt haben.
- generierte benutzerdefinierte XML-Daten. Dies ist das XML-Dokument basierend auf dem von Ihnen erstellten Schema, das die aus der HTML-Datei importierten Daten enthält.
- XSLT Stylesheets für die HTML-Ausgabe.
- HTML-Ausgabedatei.

© 2019-2025 Altova GmbH

9.8 ASPX-Schnittstelle für Web-Applikationen

Wenn in StyleVision mit einem SPS ein HTML-Bericht zu Datenbank- oder XML-Daten für das Internet erstellt werden soll, dann erfolgt dies gewöhnlich in den folgenden Schritten:

1. Wenn es sich bei den Quelldaten um eine Datenbank handelt, generieren Sie, während das SPS in StyleVision aktiv ist, eine XML-Datei anhand der Datenbank. (Wenn es sich bei den Quelldaten um eine XML-Datei handelt, fällt dieser Schritt weg)
2. Generieren Sie mit diesem SPS auch die XSLT-für-HTML-Datei.
3. Transformieren Sie die XML-Datei mit Hilfe der XSLT-für-HTML-Datei.
4. Speichern Sie die erzeugte HTML-Datei auf dem Server.

Bei einer Web-Applikation kann es vorkommen, dass die HTML-Datei bald nicht mehr auf aktuellem Stand ist, wenn die Quelldaten (Datenbank oder XML-Daten) geändert werden. Um die HTML-Datei auf dem Webserver zu aktualisieren, damit die neuen Daten darin enthalten sind, müsste Folgendes geschehen: (i) Die XML-Datei, die die geänderten DB-Daten enthält, muss neu generiert werden (ii) die neue XML-Datei muss mit Hilfe der XSLT-für-HTML-Datei transformiert werden und (iii) die erzeugte HTML-Datei muss auf dem Server gespeichert werden.

Zu diesem Zweck bietet StyleVision eine Lösung, um HTML-Webseiten schnell aktualisieren zu können, nämlich eine Funktion zur automatischen Generierung einer ASPX-Applikation. Alle erforderlichen ASPX-Applikationsdateien (die `.aspx` Datei, die XSLT-Datei und die Codedateien) werden von StyleVision generiert und können anschließend zusammen mit der Datenbank- oder XML-Datei und der XSLT-für-HTML-Datei auf dem Server gespeichert werden. Bei jeder Aktualisierung der `.aspx`-Datei, also der Schnittstellendatei zum Internet, geschieht Folgendes: (i) Bei Datenbankdaten wird anhand der Datenbank eine neue XML-Datei generiert; dieser Schritt fällt bei XML-Dateien weg (ii) die XML-Datei wird mit Hilfe der XSLT-für-HTML-Datei auf dem Server transformiert und (iii) die Transformationsausgabe wird auf der Seite, die als Webschnittstelle dient, angezeigt. Auf diese Art werden immer die neuesten und aktuellsten Datenbank- bzw. XML-Daten auf dieser Seite angezeigt.

Generierung der Dateien für eine ASPX-Lösung

Nachdem Sie das Datenbank- oder XML-basierte SPS erstellt haben, gehen Sie folgendermaßen vor, um eine ASPX-Lösung zu erstellen:

1. Generieren Sie, während das SPS in StyleVision aktiv ist, durch Auswahl des Befehls **Datei | Web Design | ASPX-Web-Applikation generieren** die ASPX-Dateien. Die ASPX-Datei wird im angegebenen Ordner erstellt. Der Ordner, in dem die ASPX-Applikation generiert wird, wird unter anderem auch die folgenden Dateien enthalten:

   - `Readme.doc`
   - `SPSFilename.aspx`
   - `SPSFilename.xslt`
   - `SPSFilename.cs`

2. Speichern Sie die Datenbank bzw. die XML-Datei auf dem Server im selben Ordner wie die ASPX-Applikationen. Über die `.aspx`-Datei wird die Applikation aufgerufen. Wenn Sie diese Datei aktualisieren, werden die darin angezeigten Datenbank- bzw. XML-Daten aktualisiert.

Hinweis: Damit die XSLT-Transformation korrekt ausgeführt werden kann, muss die Altova-Applikation RaptorXML installiert sein. Bei Problemen mit der Transformation, finden Sie in der Datei `ReadMe.doc` Informationen zum Einrichten von RaptorXML.

Funktionsweise

Der Ordner, in dem Sie die ASPX-Applikation generieren, wird unter anderem die folgenden Dateien enthalten:

- `Readme.doc`
- `SPSFilename.aspx`
- `SPSFilename.xslt`
- `SPSFilename.cs`

`SPSFilename.aspx` ist die URL des Ausgabedokuments. `SPSFilename.aspx` führt den in der Datei `SPSFilename.cs` gespeicherten C#-Code aus. Dieser C#-Code liest den XML-Inhalt (je nach Erfordernis aus Dateien oder einer Datenbank) und übergibt ihn zusammen mit der Datei `SPSFilename.xslt` an RaptorXML. (RaptorXML enthält den XSLT-Transformationsprozessor von Altova. RaptorXML kann von der Altova Website heruntergeladen werden.) RaptorXML transformiert den XML-Inhalt anhand der zur Verfügung gestellten XSLT-Datei. Das Ergebnis ist ein HTML-Dokument, das anschließend von der Web-Applikation im Browser angezeigt wird. Wenn sich der XML-Inhalt ändert, z.B. weil an der Datenbank Änderungen vorgenommen wurden, werden bei Aufruf von `SPSFilename.aspx` (bzw. bei Aktualisierung der Seite im Browser) automatisch die neuesten Daten aus der Datenbank oder XML-Datei abgerufen, so dass ein aktualisiertes Dokument angezeigt wird.

9.8.1 Beispiel: Lokaler Host auf Windows 7

Mit dem unten beschriebenen Verfahren konfigurieren Sie ihren lokalen Rechner als Host für eine ASPX-Applikation. Nähere Informationen dazu finden Sie im Applikationsordner in der Datei Readme.doc. Dieser Ordner und diese Datei werden generiert, wenn Sie den Befehl **Datei | Web Design | ASPX-Web-Applikation generieren** aufrufen, während eine SPS-Datei aktiv ist.

Installieren Sie RaptorXML

Stellen Sie sicher, dass die neueste Version von RaptorXML installiert ist. RaptorXML enthält den Transformationsprozessor von Altova, der zum Transformieren der anhand der Datenbank generierten XML-Datei verwendet wird.

Aktivieren Sie Internet Information Services (Webserver von Microsoft)

Wenn die Internet Information Services (IIS) nicht aktiviert sind, führen Sie die folgenden Schritte aus, um ihn zu aktivieren. In Schritt 5 wird beschrieben, wie Sie testen können, ob IIS aktiviert wurde.

1. Gehen Sie zu *Systemsteuerung* | *Programme und Funktionen* | *Windows-Funktionen aktivieren oder deaktivieren*.
2. Aktivieren Sie das Kontrollkästchen *Internetinformationsdienste*. Das Kontrollkästchen zeigt nun an, dass einige der Statusarten aktiviert sind.
3. Aktivieren Sie zusätzlich dazu das Kontrollkästchen *Internetinformationsdienste* | *WWW-Dienste* | *Anwendungsentwicklungsfeatures* | *ASP.NET*.
4. Klicken Sie auf **OK**. Wenn der Vorgang abgeschlossen ist, haben Sie einen Ordner namens `C:/inetpub/wwwroot`. Dies ist der Root-Ordner für den Webserver.

© 2019-2025 Altova GmbH

5. Gehen Sie zum Testen in einem Browser zu `localhost`. Daraufhin wird der IIS-Willkommensbildschirm angezeigt.

Generieren Sie die ASP.NET-Applikation in StyleVision

Generieren Sie die ASP.NET-Applikation folgendermaßen:

1. Es wird empfohlen, die Datenbank und die SPS-Datei im selben Ordner zu haben.
2. Rufen Sie nach Fertigstellung der SPS-Datei den Befehl *Dateien* | *Web Design* | *ASPX-Web-Applikation generieren*.
3. Erstellen Sie im Dialogfeld, das daraufhin angezeigt wird, einen Ordner unter `C:/inetpub/wwwroot` und wählen Sie den Ordner aus, z.B.: `C:/inetpub/wwwroot/Test1`.
4. Nachdem Sie die Ordnerauswahl bestätigt haben, generiert StyleVision darin die folgenden Dateien: `<Dateiname>.aspx`, `<Dateiname>_AltovaDataBaseExtractor.cs`, `Web.config` und einen Ordner namens `App_Code`, der die verschiedenen Dateien enthält.

Anmerkung: Um Dateien unter to `C:/inetpub/wwwroot` speichern zu können, müssen Sie StyleVision als Administrator ausführen. Starten Sie dazu StyleVision neu. Klicken Sie mit der rechten Maustaste auf dieStyleVision Exe-Datei oder auf einen Shortcut zu dieser Datei und wählen Sie den Befehl **Als Administrator ausführen**.

Melden Sie die generierte Applikation bei ASPX an

Mit den folgenden Schritten melden Sie die Applikation, die Sie mit StyleVision generiert haben, bei ASPX an:

1. Gehen Sie zu *Systemsteuerung* | *Verwaltung* | *Internetinformationsdienste (IIS)-Manager*.
2. Erweitern Sie die Struktur im Fenster *Verbindungen,* um den Ordner (z.B. `Test1`) zu sehen. Als Ordnersymbol wird zu diesem Zeitpunkt das gelbe Ordnersymbol als Standardsymbol angezeigt.
3. Rufen Sie im Kontextmenü des Ordners den Befehl *In Applikation konvertieren* auf und klicken Sie anschließend im Dialogfeld auf **OK**. Daraufhin wird als Ordnersymbol ein Globus angezeigt.
4. Erweitern Sie die Struktur im Fenster *Verbindungen* um den Eintrag *Anwendungspools* zu sehen und wählen Sie diesen aus.
5. Wählen Sie im Kontextmenü von *DefaultAppPool* (im Hauptfenster, wo Sie im Bereich "Verbindungen" den Eintrag *Anwendungspools* auswählen), den Befehl **Erweiterte Einstellungen** aus.
6. Wählen Sie für die Eigenschaft *Identität* den Eintrag *Benutzerdefiniertes Konto* aus und geben Sie ihren Windows-Benutzernamen und Ihr Passwort ein.
7. Geben Sie bei der Eigenschaft *32-Bit-Anwendungen aktivieren* `True` ein (damit die Datenbanktreiber Zugriff haben). Dieser Schritt gilt nur für 64-Bit-Versionen von Windows 7.

Führen Sie die Applikation aus

Gehen Sie im Browser zu `localhost/Test1/<Dateiname>.aspx` (unter der Voraussetzung, dass `Test1` der Name des Ordners ist, in dem die ASPX-Applikation gespeichert wurde). Die transformierte HTML-Datei wird im Browser angezeigt. Wenn Sie die ASPX-Seite aktualisieren, werden die neuesten Daten aus der Datenbank bzw. HTML-Datei angezeigt.

Anmerkung: Wenn der Browser nun hängt, stellen Sie sicher, dass RaptorXML korrekt lizenziert ist.

9.9 PXF-Datei: Behälter für SPS und dazugehörige Dateien

Ein SPS-Design, in dem XSLT 2.0 oder 3.0 verwendet wird, kann als Portable XML Form (PXF)-Datei gespeichert werden. Das PXF-Dateiformat wurde von Altova eigens dafür entwickelt, um SPS-Designs mit dazugehörigen Dateien (wie z.B. der Schema-Datei, der XML-Quelldatei, den im Design verwendeten Bilddateien und der für die Transformation der XML-Quelldatei verwendeten XSLT-Datei) in einer Datei zu verpacken. Der Vorteil des PXF-Formats ist, dass alle für die Bearbeitung in der Authentic-Ansicht und die Generierung einer Ausgabe anhand der Authentic-Ansicht benötigten Dateien in einer einzigen Datei weitergeleitet werden können.

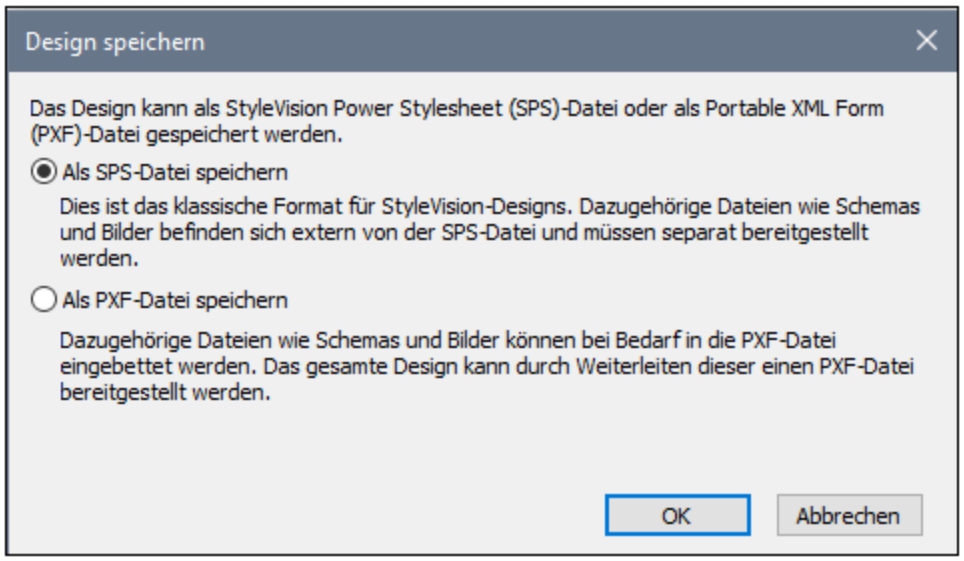
In diesem Abschnitt wird die Verwendung von PXF-Dateien beschrieben. Der Abschnitt ist in zwei Teile gegliedert:

- Erstellen einer PXF-Datei 418
- Bearbeiten einer PXF-Datei 421
- Bereitstellen einer PXF-Datei 422

Anmerkung: PXF-Dateien können nur anhand von SPS-Dokumenten, die mit XSLT 2.0 oder 3.0 erstellt wurden, erstellt werden.

9.9.1 Erstellen einer PXF-Datei

Um eine PXF-Datei zu erstellen, die ein SPS-Design sowie die dazugehörigen Dateien enthält, öffnen Sie das SPS-Design in StyleVision und wählen Sie den Befehl **Datei | Speichern unter**. Daraufhin wird das Dialogfeld "Design speichern" (*Abbildung unten)* geöffnet.

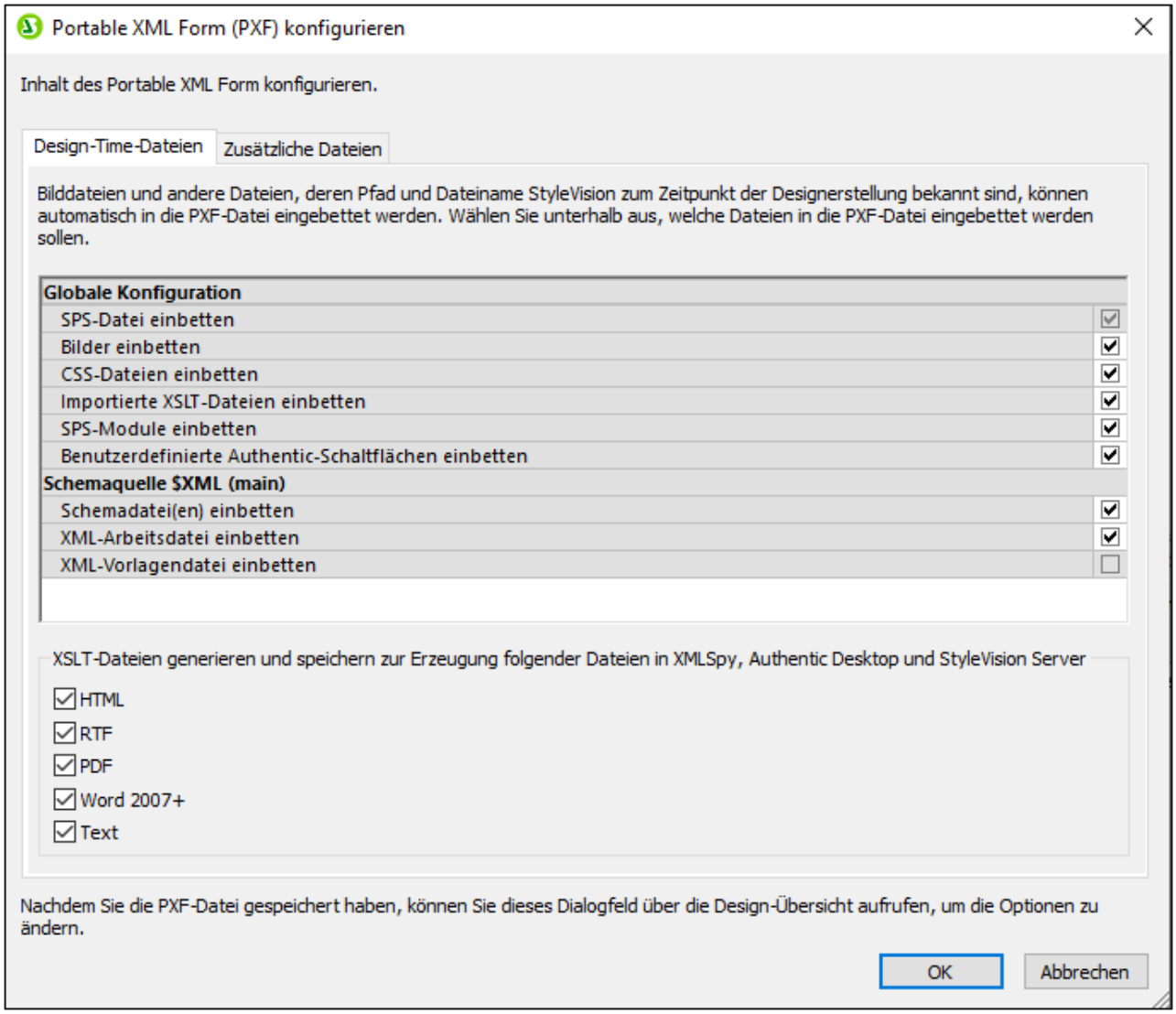


Das **SPS-Format** ist das Altova Standardformat für StyleVision Designs. In diesem Abschnitt wird das PXF-Format beschrieben, daher wird das SPS-Format hier nicht näher behandelt. Das Speichern einer Datei als SPS wird im Abschnitt Benutzerreferenz 485 im Detail beschrieben.

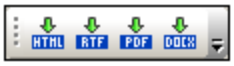
© 2019-2025 Altova GmbH

Als PXF-Datei speichern

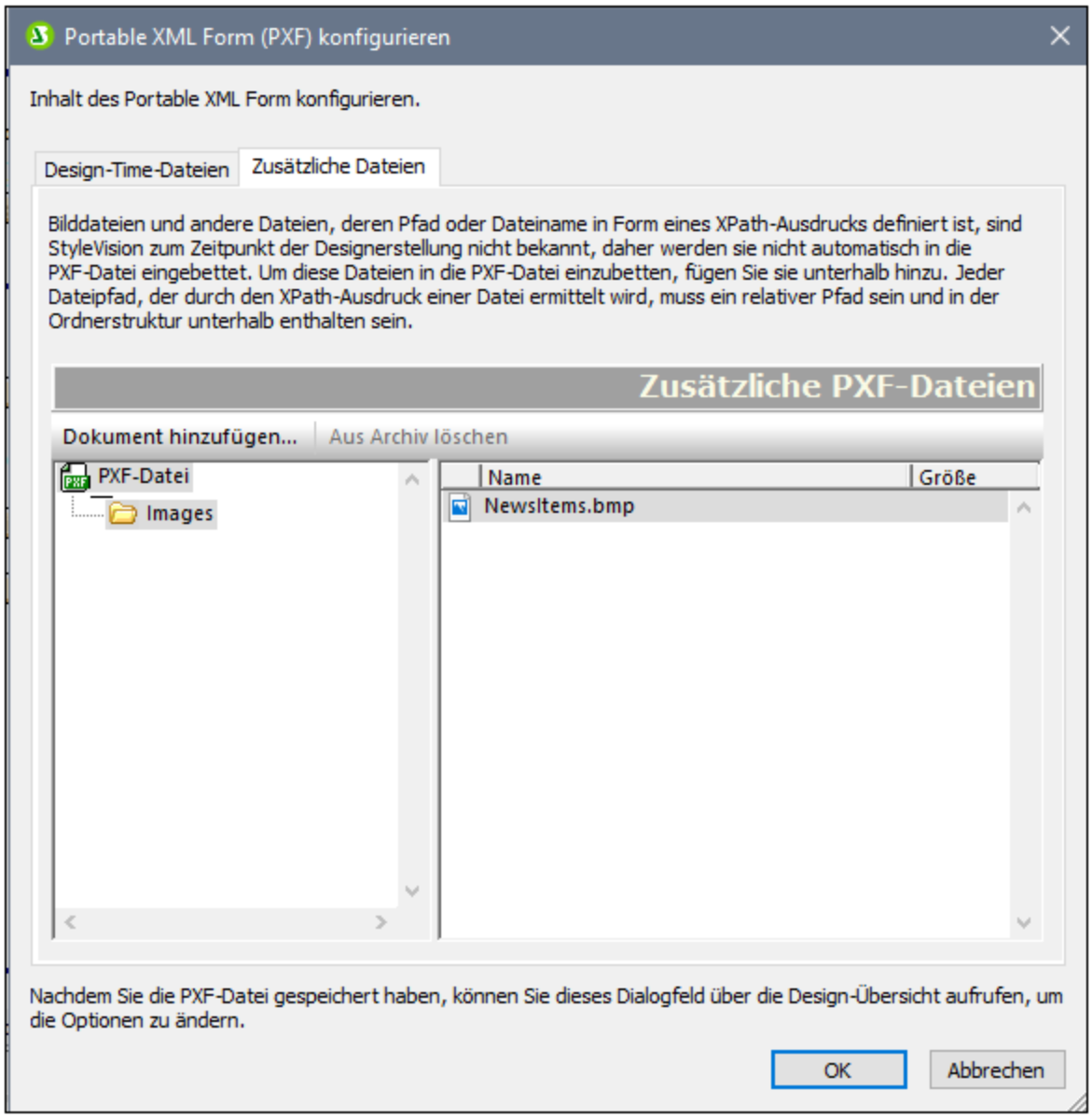
Bei Auswahl der PXF-Option wird das bekannte Windows-Dialogfeld "Speichern unter" angezeigt. Dieser Speicherbefehl funktioniert genau so, wie der **Befehl "Speichern unter"** [480], mit dem zusätzlichen Schritt, dass die Dateien, die in die PFX-Datei inkludiert werden sollen, ausgewählt werden müssen. Nachdem Sie den PXF-Dateinamen definiert haben, wird das Dialogfeld "Portable XML Form (PXF) konfigurieren" (*Abbildung unten*) aufgerufen, in dem Sie die einzubettenden Dateien auswählen/deselektieren können.



Auf dem Register *Design-Time-Dateien* können Sie im Bereich "Globale Konfiguration" auswählen/deselektieren, welche mit dem Design in Zusammenhang stehenden Quelldateien eingebettet bzw. weggelassen werden sollen. Zusätzlich dazu können Sie festlegen, ob anhand des Designs generierte XSLT-Dateien ebenfalls eingebettet werden sollen. Wählen Sie im Bereich "XSLT-Dateien" die Ausgabeformate aus, für die XSLT-Dateien generiert und eingebettet werden sollen. Wenn eine XSLT-Datei in die PXF-Datei inkludiert wird und die PXF-Datei in der Authentic-Ansicht eines Altova-Produkts geöffnet wird, so ist die Symbolleisten-Schaltfläche zum Generieren und Anzeigen des Ausgabeformats in der Authentic-Ansicht aktiv (*Abbildung unten)*



Auf dem Register *Zusätzliche Dateien* (*Abbildung unten*) können Sie alle zusätzlichen Dateien angeben, bei denen es sich nicht um Design Time-Dateien handelt und die ebenfalls in die PXF-Datei inkludiert werden sollen. Dabei könnte es sich z.B. um Bilddateien handeln, die im Design durch eine mit einem XPath-Ausdruck generierte URL referenziert werden. In der Abbildung unten wurde die Bilddatei `NewsItems.bmp` aus dem Ordner `Images` ausgewählt, um in die PXF-Datei inkludiert zu werden.



Um eine zusätzliche Datei in die PXF-Datei zu inkludieren, klicken Sie auf die Schaltfläche **Dokument hinzufügen** und navigieren Sie dann zur gewünschten Datei. Im Dialogfeld "Öffnen" (in dem Sie zur gewünschten Datei navigieren) wird der Ordner geöffnet, in dem sich die SPS-Datei befindet. Es können Dateien aus diesem Ordner oder jedem beliebigen diesem untergeordneten Ordner ausgewählt werden.

© 2019-2025 Altova GmbH

Nachdem eine weitere Datei zur PXF-Datei hinzugefügt wurde, werden diese Datei und die übergeordnete Ordnerstruktur angezeigt. In der Abbildung oben sehen Sie, dass sich die zusätzliche Datei `NewsItems.bmp` in einem Ordner namens `Images` befindet, der sich wiederum selbst in dem Ordner befindet, der die SPS-Datei enthält.

Wenn eine Datei aus einem Ordner ausgewählt wird, der sich hierarchisch über dem Ordner mit der SPS-Datei befindet, so wird ein Fehler ausgegeben.

Alle Verweise auf zusätzliche Dateien in einem SPS-Design müssen mit einem relativen Pfad angegeben werden und der im Bereich *Zusätzliche Dateien* gezeigten Ordnerstruktur entsprechen. So muss z.B. die Datei `NewsItems.bmp` in der Abbildung oben durch den relativen Pfad: `Images/NewsItems.bmp` referenziert werden.

Anmerkung: Um PXF-Dateien speichern zu können, muss die Option *Bilder für RTF und Word 2007+ einbetten* (**Datei | Eigenschaften | Bilder**) aktiviert sein.

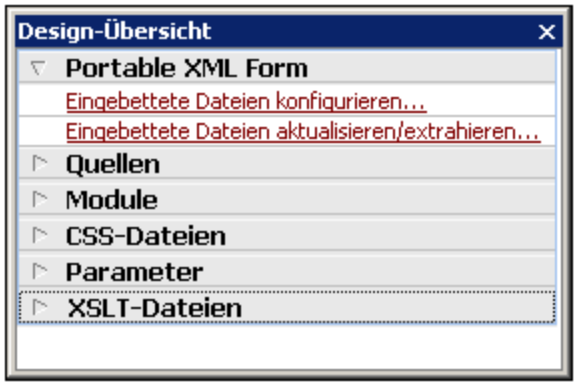
9.9.2 Bearbeiten einer PXF-Datei

Eine PXF-Datei kann in StyleVision mit dem Befehl **Datei | Öffnen** 475 geöffnet und bearbeitet werden. Die Datei kann auf zwei Arten bearbeitet werden:

- Die Konfiguration der PXF-Datei kann bearbeitet werden
- Der Inhalt der einzelnen Komponentendateien wie z.B. das SPS und die Authentic XML-Datei können in StyleVision bearbeitet werden, während andere Komponentendateien (wie z.B. Bilder und CSS-Dateien) in externen Applikationen bearbeitet werden können. Alle Komponentendateien müssen allerdings explizit in StyleVision aktualisiert werden.

Ausgangspunkt für die PXF-Bearbeitung

Der Ausgangspunkt für die Bearbeitung der PXF-Konfiguration und die Aktualisierung der PXF-Datei ist der Eintrag PXF in der Seitenleiste "Design-Übersicht" (*Abbildung unten*).

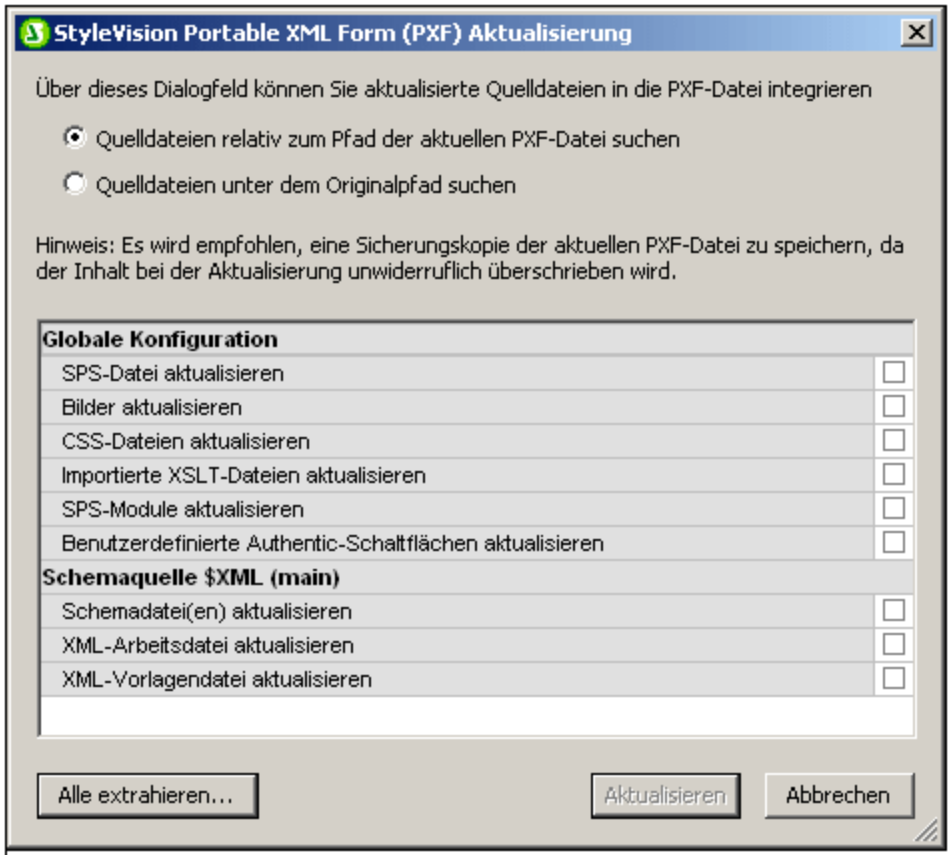


Konfigurieren der eingebetteten Dateien

Wenn Sie in der Design-Übersicht auf den Link *Eingebettete Dateien konfigurieren* klicken (*siehe Abbildung oben*), wird das Dialogfeld "Portable XML Form (PXF) konfigurieren" geöffnet. Die Konfigurationsoptionen sind genau dieselben wie die im Abschnitt Erstellen einer PXF-Datei 418 beschriebenen.

Aktualisieren eingebetteter Dateien

Wenn Sie in der Seitenleiste "Design-Übersicht" auf den Link *Eingebettete Dateien aktualisieren* klicken (*siehe Abbildung oben*) wird das Dialogfeld "StyleVision Portable XMLForm (PXF) Aktualisierung" geöffnet (*Abbildung unten*).



Wählen Sie zuerst aus, ob die Quelldateien relativ zu ihren aktuellen PXF-Ordnern oder vom Originalpfad aus aufgerufen werden sollen. Klicken Sie anschließend auf die Dateien, die aktualisiert werden sollen und klicken Sie auf **Aktualisieren** . Daraufhin wird eine neue PXF-Datei erstellt, die die bestehende PXF-Datei überschreibt. Es empfiehlt sich daher vor der Aktualisierung der PXF-Datei eine Sicherungskopie davon anzulegen.

9.9.3 Bereitstellen einer PXF-Datei

Nachdem eine PXF-Datei erstellt wurde, kann sie wie jede andere Datendatei weitergeleitet, heruntergeladen, kopiert und gespeichert werden. Da die PXF-Datei alle Dateien enthalten kann, die zum Bearbeiten einer XML-Datei in der Authentic-Ansicht und zum Generieren von Ausgabeberichten benötigt werden, ist sie die einzige Datei, die der Benutzer der Authentic-Ansicht für seine Arbeit benötigt und mit der er auch Ausgabedokumente erzeugen kann.

© 2019-2025 Altova GmbH

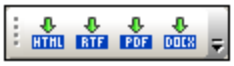
Eine PXF-Datei kann in der Authentic [19] von Altova-Produkten geöffnet werden. Um Ihnen einen Überblick über die Verwendungsmöglichkeiten einer PXF-Datei zu verschaffen, sind hier einige Verwendungsszenarios in XMLSpy aufgelistet:

- Die PXF-Datei wird mit dem Befehl **Datei | Öffnen** geöffnet. Die eingebettete XML-Datei wird unter Verwendung des eingebetteten SPS in der Authentic-Ansicht angezeigt und kann dort bearbeitet werden. Mit dem Befehl **Datei | Speichern** werden die Änderungen in der PXF-Datei gespeichert (die eingebettete XML-Datei wurde geändert).

- Die PXF-Datei enthält keine eingebettete XML-Datei und wird mit dem Befehl **Datei | Öffnen** geöffnet. Wenn keine XML-Datei inkludiert ist, wird eine XML-Vorlagendatei basierend auf dem SPS-Design in der Authentic-Ansicht geöffnet. Mit dem Befehl **Datei | Speichern** wird diese XML-Datei in der PXF-Datei als eingebettete Datei gespeichert.

- Im Altova-Produkt XMLSpy kann eine XML-Datei mit einer PXF-Datei verknüpft werden, sodass das in die PXF-Datei eingebettete SPS zur Bearbeitung in der Authentic-Ansicht verwendet wird. Die Verknüpfung erfolgt über den Menübefehl **Authentic | StyleVision Stylesheet zuweisen**. Wenn Änderungen gespeichert werden, so werden diese in der XML-Datei gespeichert; die PXF-Datei ändert sich nicht.

- Wenn ein XSLT-Stylesheet für eines der Ausgabeformate in die PXF-Datei eingebettet wurde, so kann der Benutzer der Authentic-Ansicht eine Ausgabedatei in diesem Format generieren. Dies geschieht über die Symbolleisten-Schaltfläche für das entsprechende Ausgabeformat (*Abbildung unten*). In der Authentic-Ansicht sind die einzelnen Symbolleisten-Schaltflächen zur Generierung von Ausgabedokumenten nur aktiv, wenn die PXF-Datei so konfiguriert wurde, dass sie das XSLT Stylesheet für diese Ausgabe enthält. Wenn die PXF-Datei z.B. so konfiguriert wurde, dass sie XSLT Stylesheets für HTML und PDF enthält, so sind nur die Symbolleisten-Schaltflächen für die HTML- und PDF-Ausabe aktiv, während diejenigen für die RTF- und DocX (Word 2007+)-Ausgabe deaktiviert sind.



Anmerkung: Wenn sich eine PXF-Datei auf einem Webserver befindet und mit dem Authentic Browser Plug-in verwendet werden soll, stellen Sie bitte sicher, dass der Server die Datei nicht blockiert. Fügen Sie dazu (z.B. über den IIS-Verwaltungsbereich) den folgenden MIME Type für PXF (`.pxf`) Dateierweiterungen hinzu: `application/x-zip-compressed`.

10 Automatisierte Verarbeitung

Die Funktionalitäten von StyleVision bieten zusammen mit den verschiedenen von StyleVision generierten XSLT- und Ausgabedateien zahlreiche Möglichkeiten zur Automation. Diese Funktionalitäten werden in diesem Abschnitt beschrieben.

Die Dateigenerierungsfunktionalität von StyleVision

Nachdem Sie mit StyleVision ein SPS erstellt haben, können Sie je nachdem, welche Version von StyleVision Sie verwenden (Enterprise, Professional oder Basic) über die Benutzeroberfläche mehrere Arten von XSLT und Ausgabedateien generieren. Die folgenden Dateien können mit dem Befehl **Datei | Generierte Dateien speichern**[486] generiert werden:

- XSLT-Dateien für die HTML-Ausgabe.
- HTML-Ausgabe.

Wie Sie aus der obigen Liste ersehen, gibt es zwei Arten von Dateien, die mit StyleVision gespeichert werden können:

1. die vom SPS Design generierten XSLT-Dateien und
2. die endgültigen Ausgabedateien (wie z.B.HTML).

Die Generierung der endgültigen HTML-Ausgabedateien erfolgt in einem Schritt, in der das XML-Dokument anhand eines XSLT Stylesheet in das Ausgabeformat transformiert wird.

StyleVision Server und RaptorXML: Generieren von Dateien von außerhalb der Benutzeroberfläche

Neben der Generierung von XSLT-Stylesheets und der erforderlichen Ausgabedokumente über die Benutzeroberfläche von StyleVision (Befehl **Datei | Generierte Dateien speichern)**[486], stehen zwei weitere Methoden zur Verfügung, um Ausgabedateien zu generieren:

1. mit StyleVision Server, der die Dateigenerierungsfunktionen von StyleVision aufruft, ohne die Benutzeroberfläche zu öffnen, können Sie verschiedene Arten von Ausgabedateien erzeugen.
2. mit RaptorXML[428], einer Standalone-Applikation von Altova, die den Altova XML- (+XBRL)-Validator und den Altova XSLT- und XQuery-Prozessor enthält. Die XSLT-Prozessoren in RaptorXML können für die Transformation von XML in ein Ausgabeformat (durch Verarbeitung der XML-Dokumente mittels eines XSLT-Dokuments) verwendet werden. Zuerst muss die XSLT-Datei erstellt werden, damit sie RaptorXML verwendet werden kann. (RaptorXML akzeptiert kein %SPS%> als Input-Parameter.) Die Vorteile der Verwendung von RaptorXML sind: (i) schnellere Transformationen großer Dateien. (ii) RaptorXML bietet zusätzlich zu einer Befehlszeilenschnittstelle eine COM-, Java- und .NET-Schnittstelle und kann daher jederzeit von diesen Umgebungen aus aufgerufen werden. Eine Beschreibung zur Verwendung von RaptorXML für Transformationen finden Sie im Unterabschnitt RaptorXML[428].
3. In einem Altova FlowForce Server Workflow können mit Hilfe von Altova StyleVision Server anhand von vordefinierten Triggern (wie z.B. einer Tageszeit) mehrere Transformationen ausgeführt werden. Eine Beschreibung dazu finden Sie im Abschnitt Automatisierung mit FlowForce Server[431].

© 2019-2025 Altova GmbH

10.1 Befehlszeilenschnittstelle

Die StyleVision-Funktionalitäten können auf zwei Arten über die Befehlszeile aufgerufen werden:

- Durch Aufruf der ausführbaren StyleVision-Datei[425]. Auf diese Art erhalten Sie Zugriff auf die StyleVision-Funktionen zur XSL-Dateigenerierung. Die XSLT-Dateien werden anhand der SPS-Datei generiert.
- Durch Verwendung von StyleVision Server[427] zur Generierung von Ausgabedateien (HTML, usw.). Die Ausgabedateien werden anhand einer PXF-Datei generiert, welche ein Paket bestehend aus einer SPS-Datei und den dazugehörigen Dateien (XML, XSD, Bilddateien, usw.) ist. Die PXF-Datei wird über StyleVision generiert.

Verwendung der Befehlszeile

Es gibt zwei Arten, wie die Befehlszeile verwendet werden kann:

- Befehle können einzeln über die Befehlszeile eingegeben und sofort ausgeführt werden. So können Sie z.B. im Befehlszeileneingabefenster einen Befehl für StyleVision oder StyleVision Server[427] eingeben und zur Ausführung des Befehls die **Eingabetaste** drücken.
- Es kann in einer Batch-Datei eine Reihe von Befehlen für die Stapelverarbeitung eingegeben werden. Z.B.:

```
@ECHO OFF
CLS
StyleVision TestEN.sps -outxslt=HTML-EN.xslt
StyleVision TestDE.sps -outxslt=HTML-DE.xslt
StyleVision TestES.sps -outxslt=HTML-ES.xslt
```

Bei der Verarbeitung der Batch-Datei werden die Befehle ausgeführt und die Dateien werden generiert.

StyleVision Funktionalitäten in geplanten Aufgaben

Mit Hilfe der geplanten Aufgaben in Windows können Sie die Ausführung von StyleVision Befehlen nach einem vordefinierten Zeitplan definieren. Als auszuführende Aufgabe kann entweder ein einziger Befehl oder eine Batch-Datei definiert werden. Eine Anleitung zur Erstellung solcher Befehle als geplante Aufgaben finden Sie unter Anleitung zum Automatisieren der Verarbeitung[433]

10.1.1 StyleVision

Die Syntax für Benutzer der Befehlszeile lautet:

```
StyleVision [<SPS File>] [<options>]
```

wobei

`StyleVision`	StyleVision aufruft, das sich im StyleVision Applikationsordner befindet
`<SPS File>`	die SPS-Datei angibt
`<options>`	Eine oder mehrere der unten aufgelisteten Optionen auflistet

Wenn ein Befehl ausgeführt wird, wird StyleVision im Hintergrund (d.h. ohne dass die Benutzeroberfläche geöffnet wird) ausgeführt, die erforderlichen Ausgabedateien werden generiert und das Programm wird geschlossen. Wenn ein Fehler oder eine Warnung auftritt, wird die Benutzeroberfläche geöffnet und die entsprechende Meldung wird in einem Meldungsfeld angezeigt.

Anmerkung: Damit das SPS in StyleVision korrekt geladen wird, müssen sich die vom SPS verwendete die XSD- und XML-Arbeitsdatei in den im SPS definierten Ordnern befinden.

Optionen

Die Optionen können in jeder beliebigen Reihenfolge eingegeben werden. In der Liste unten sind sie zur besseren Übersichtlichkeit in Gruppen gegliedert. Beachten Sie, dass die Optionen für die FO-, RTF-, PDF- und Word 2007+-Ausgabe nur in der Enterprise Edition zur Verfügung stehen; diese Optionen sind in der Liste unten mit dem Eintrag *Enterprise Edition* oder *Enterprise Edition und Professional editions*gekennzeichnet..

- ***XSLT-Dateiausgabe***

`-OutXSLT=<file>`	Schreibt die XSLT-für-HTML-Ausgabe in die angegebene Datei
`-OutXSLRTF=<file>`	Schreibt die XSLT-für-RTF-Ausgabe in die angegebene Datei*(nur Enterprise und Professional Edition)*
`-OutXSLText=<file>`	Schreibt die XSLT-für-Text-Ausgabe in die angegebene Datei *(Enterprise und Professional editions)*
`-OutXSLFO=<file>`	Schreibt die XSLT-für-FO-Ausgabe in die angegebene Datei *(nur Enterprise Edition)*
`-OutXSLWord2007=<file>`	Schreibt die XSLT-für-Word 2007+-Ausgabe in die angegebene Datei *(nur Enterprise Edition)*

Beispiele

```
StyleVision "QuickStart.sps" -outxslt="QuickStartHTML.xslt"
StyleVision "C:\Test\QuickStart.sps" -outxslt="C:\Test\QuickStartHTML.xslt"
```

Bitte beachten Sie die folgenden Punkte

Im Folgenden finden Sie einige ergänzende Erklärungen zur StyleVisionBatch Syntax und der Verwendung der Befehlszeile.

- Pfade können absolut oder relativ sein und sollten umgekehrte Schrägstriche verwenden.
- Wenn der Dateiname oder der Pfad zur Datei ein Leerzeichen enthält, sollte der gesamte Pfad innerhalb von Anführungszeichen gesetzt werden. Z.B.: `"c:\My Files\MyXML.xml"` oder `"c:\MyFiles\My XML.xml"`.
- Die Groß- und Kleinschreibung wird bei Befehlen, Pfaden und Ordner- oder Dateinamen nicht beachtet.

© 2019-2025 Altova GmbH

10.1.2 StyleVision Server

StyleVision Server kann über seine Befehlszeilenschnittstelle auf Windows-, Linux- und Mac OS-Systemen verwendet werden, um XML-Dateien in HTML-, PDF-, RTF- und DOCX-Ausgabedokumente zu transformieren. Der `generate` Befehl der StyleVision Server Befehlszeilenschnittstelle erhält als seine beiden Argumente eine XML-Datei und eine PXF-Datei[418] und als seine Parameter die gewünschten Ausgabeformate. Die XSLT-Stylesheets für die Transformation stammen aus der als Input angegebenen PXF-Datei[418].

Ein Vorteil der Verwendung des CLI von StyleVision Server anstelle des CLI von RaptorXML Server ist, dass StyleVision Server als Input-Datei eine PXF-Datei akzeptiert (RaptorXML akzeptiert als Input eine XSLT-Datei). Bei Verwendung im Altova FlowForce Workflow empfiehlt sich die Verwendung von StyleVision Server. Bei einem FlowForce Workflow können Transformationsaufträge entsprechend vordefinierter Trigger gestartet werden: Wenn ein FlowForce Auftrag durch einen Trigger gestartet wird, können mehrere Dateien innerhalb eines Netzwerks automatisch transformiert werden. Nähere Informationen dazu finden Sie im Abschnitt Automatisierung mit FlowForce Server[431].

Nähere Informationen zum StyleVision Server CLI finden Sie in der Dokumentation zu StyleVision Server.

Ausgabedateien

Mit Hilfe von StyleVision Server können Sie anhand der definierten PXF-Datei eine oder mehr der folgenden Dateien generieren:

- HTML (`.html`)-Datei unter Verwendung der in der PXF-Datei definierten XML- und XSLT-für-HTML-Datei oder unter Verwendung einer anderen XML-Datei

10.2 Verwendung von RaptorXML

Altova RaptorXML ist der ultraschnelle XML- und XBRL-Prozessor der dritten Generation und wurde speziell für die neuesten Standards und parallele Rechnerumgebungen optimiert. RaptorXML lässt sich plattformübergreifend einsetzen und verarbeitet XML- und XBRL-Daten dank Nutzung moderner Multi-Core Computer ultraschnell.

RaptorXML ist in zwei Editionen erhältlich:

- der RaptorXML Server Edition, die über ein Netzwerk aufgerufen werden kann und die mehrere Dateien gleichzeitig transformieren kann.
- der RaptorXML+XBRL Server Edition, die über ein Netzwerk aufgerufen werden kann und die mehrere Dateien gleichzeitig transformieren kann und die XBRL-Validierung unterstützt.

Nähere Informationen zu RaptorXML finden Sie auf der Altova Website.

Typische Einsatzmöglichkeiten

Am wichtigsten für Benutzer von StyleVision sind die XSLT-Transformationsfunktionen von RaptorXML. Normalerweise werden die Funktionen folgendermaßen verwendet:

1. Anhand eines SPS wird mit Hilfe des Befehls **Datei | Generierte Dateien speichern** 486 ein XSLT-Stylesheet generiert. Beachten Sie, dass RaptorXML nicht dazu verwendet werden kann, XSLT Stylesheets anhand eines SPS zu generieren.
2. Mit Hilfe des generierten XSLT Stylesheet werden XML-Dokumente mit RaptorXML generiert.Mit RaptorXML können Sie HTML-Ausgabedokumente generieren.

Vorteile von RaptorXML

Die Vorteile der Verwendung von RaptorXML sind die folgenden:

- RaptorXML bietet eine sehr schnelle Validierung und XSLT-Transformation und eignet sich daher zum Verarbeitung großer Dateien.
- Einfache Verwendung mit der Befehlszeilenschnittstelle, der COM-, der Java- und .NET-Schnittstelle.
- Automatisierung und Ansetzen von geplanten Aufgaben 433 mittels Batch-Dateien und über die "geplanten Aufgaben" von Windows.

Eine Beschreibung, wie RaptorXML zum Automatisieren der Erzeugung von Ausgabedokumenten (wie z.B. HTML) anhand von XML-Quelldokumenten verwendet werden kann, finden Sie unter Anleitung zur Automatisierung der Verarbeitung 433.

Weitere ausführlichere Informationen zur Verwendung von RaptorXML, u.a. die COM-, Java- und .NET-Schnittstelle finden Sie in der Benutzerdokumentation zu RaptorXML.

10.2.1 PDF-Ausgabe

Die Generierung eines PDF-Ausgabedokuments anhand eines XML-Dokuments erfolgt in zwei Schritten:

© 2019-2025 Altova GmbH

1. Das XML-Dokument wird mittels eines XSLT-Stylesheet transformiert. Für die Transformation wird ein XSLT-Transformationsprozessor (wie z.B. der von RaptorXML) verwendet. Das Ergebnis ist ein FO-Dokument.
2. Das FO-Dokument wird von einem FO-Prozessor (wie z.B. dem Apache FOP) verarbeitet und eine PDF-Ausgabedatei wird generiert. StyleVision kann so konfiguriert werden, dass das FO-Ergebnis einer XSLT-Transformation an einen FO-Prozessor übergeben wird. In StyleVision wird das Ergebnis der PDF-Generierung im Fenster "PDF-Vorschau" angezeigt. Das Ergebnis kann auch mit dem Befehl **Datei | Generierte Dateien speichern** 486 als Datei gespeichert werden.

RaptorXML und PDF

Da RaptorXML keine Parameter bereitstellt, um die FO-Ausgabe einer XSLT-Transformation an einen FO-Prozessor weiterzuleiten, erhalten Sie als Ergebnis der XSLT-Transformation ein FO-Dokument (der erste von zwei Schritten zur PDF-Generierung).

Das FO-Dokument muss nun an einen FO-Prozessor übergeben werden, der den zweiten Schritt der Transformation von FO in PDF durchführt. Die Anweisungen zur Durchführung dieses Schritts variieren je nachdem, welcher Prozessor verwendet wird. Bei Verwendung des Apache FOP-Prozessors wird z.B. zum Definieren des Input-FO-Dokuments und zur Angabe des Namens und Pfads des PDF-Ausgabedokuments der folgende einfache Befehl verwendet:

```
fop -fo input.fo -pdf output.pdf
```

FOP bietet andere Parameter, die in der FOP-Benutzerreferenz aufgelistet sind.

FOP und XSLT

Es gibt eine FOP-Option, um eine XML-, XSLT-Input-Datei und eine PDF-Ausgabedatei zu definieren.

```
fop -xml input.xml -xslt input.xslt -pdf output.pdf
```

In dieser Situation verwendet FOP den integrierten XSLT-Prozessor um den ersten Schritt der XML-in-FO-Transformation auszuführen. Das erzeugte FO-Dokument wird anschließend für den zweiten Schritt der FO-zu-PDF-Verarbeitung an den FOP Prozessor übergeben.

Beachten Sie bitte, dass der integrierte FOP-Prozessor nicht unbedingt alle von StyleVision und RaptorXML unterstützten XSLT-Funktionen unterstützt. Daher kann es zu Fehlern kommen, wenn für die XML-Transformation mit dem integrierten FOP XSLT-Prozessor ein mit StyleVision generiertes XSLT Stylesheet definiert wurde. Verwenden Sie in diesem Fall den Altova XSLT-Prozessor in RaptorXML+XBRL Server, um das XML-Dokument in FO zu transformieren. Übergeben Sie die FO-Datei dann für die Erzeugung eines PDF-Dokuments an FOP.

Batch-Verarbeitung zu PDF

Eine schnelle und einfache Methode zur Generierung eines PDF-Dokuments mittels RaptorXML für die XSTL-Transformation und FOP für die FO-Verarbeitung wäre, eine Batch-Datei zu schreiben, die die beiden Befehle kombiniert. z.B.:

```
raptorxmlserver xslt --input=Test.xml --output=Test.fo Test.xslt
fop -fo test.fo -pdf test.pdf
```

Der erste Befehl ruft RaptorXML auf und erzeugt als Ausgabedatei `test.fo`. Der zweite Befehl übergibt `test.fo` an den FOP-Prozessor, der die PDF-Datei `test.pdf`.generiert. Nähere Informationen zur Batch-Verarbeitung und eine Beschreibung zur Verwendung von Batch-Dateien für automatisierte Vorgänge finden Sie im folgenden Abschnitt: Anleitung zum Automatisieren der Verarbeitung 433.

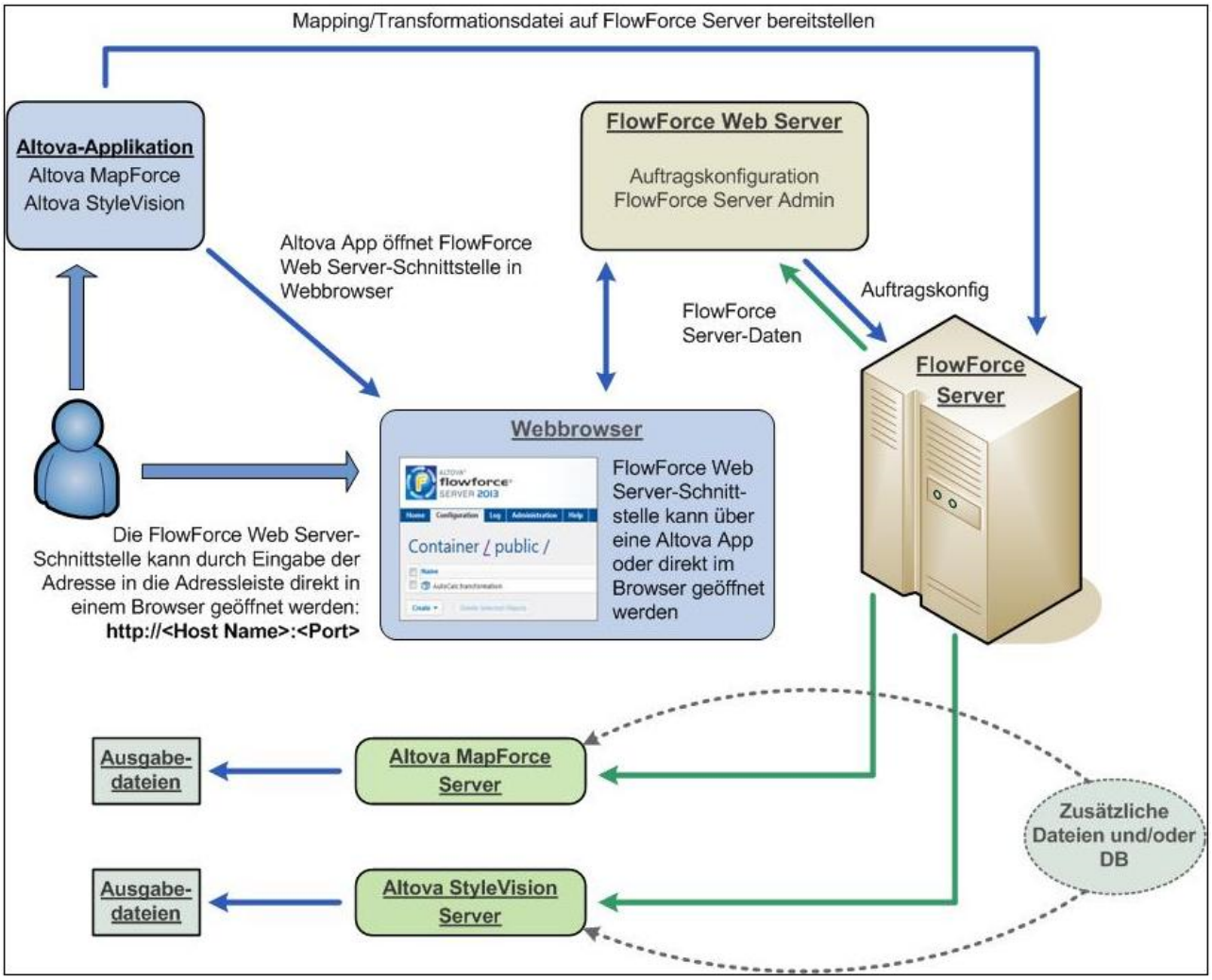
 © 2019-2025 Altova GmbH

10.3 Automatisierung mit FlowForce Server

Mit Hilfe von FlowForcer Server von Altova, welcher für Windows, Linux und Mac OS-Systeme zur Verfügung steht, können Transformationen über ein Netzwerk automatisiert werden. Dabei wird folgendermaßen verfahren:

1. Von StyleVision aus wird (mit dem Befehl **Datei | In FlowForce bereitstellen)** 487 eine PXF-Datei 418 als `.transformation`-Datei auf FlowForce Server bereitgestellt 487. Die `.transformation` -Datei enthält alle im SPS für die Transformation definierten Dateien und Daten. (Die Bereitstellung der Datei ist im Diagramm unten durch die Verbindungslinie entlang des oberen Randes dargestellt.)
2. Nachdem die `.transformation`-Datei auf FlowForce Server bereitgestellt wurde, können in FlowForce Aufträge erstellt werden, in denen die `.transformation`-Datei verwendet wird, um entsprechend der in der Auftragsdefinition definierten Trigger Transformationen zu generieren. (Als Trigger könnte z.B. eine bestimmt Uhrzeit jeden Tag verwendet werden.) FlowForce-Aufträge werden über die FlowForce Webserver-Schnittstelle (siehe Mitte des Diagramms unten) erstellt. Diese Webserver-Schnittstelle kann von StyleVision aus oder über eine HTTP-Adresse aus aufgerufen werden. Nähere Informationen zum Erstellen von FlowForce-Aufträgen finden Sie in der Dokumentation zu FlowForce.
3. Zum Zeitpunkt der Auftragsausführung übergibt FlowForce Server die Transformationsanweisungen und die relevanten Dateien an StyleVision Server, der dann die Transformation ausführt (*siehe Diagramm unten*).

Im unten gezeigten Diagramm wird die Rolle von StyleVision Server im FlowForce Workflow dargestellt. (Da auch FlowForce-Aufträge, die Altova MapForce Mappings zur Ausführung an den Altova MapForce Server senden, erstellt werden können, ist auch die Rolle von MapForce Server im Workflow dargestellt.)



Beachten Sie, dass StyleVision Server nicht nur durch einen FlowForce-Auftrag, sondern auch über seine Befehlszeilenschnittstelle aufgerufen werden kann. Eine Beschreibung dazu finden Sie in der Dokumentation zu StyleVision Server.

 © 2019-2025 Altova GmbH

10.4 Anleitung zum Automatisieren der Verarbeitung

Eine Batch-Datei (eine mit der Dateierweiterung `.bat` gespeicherte Textdatei) enthält eine Befehlsfolge, die über die Befehlszeile ausgeführt wird. Bei Ausführung der Batch-Datei werden die einzelnen Befehle in der Batch-Datei der Reihe nach hintereinander ausgeführt. Eine Batch-Datei ist daher in den folgenden Situationen nützlich:

- Automatische Ausführung einer Reihe von Befehlen (*siehe unten*).
- Erstellung einer Kette von Verarbeitungsbefehlen, wobei für einen Befehl ein durch den vorhergehenden Befehl erzeugter Input benötigt wird. (z.B. eine XML-Datei, die als Ausgabe einer Transformation erzeugt wurde, wird als Input für eine nachfolgende Transformation verwendet.) *Siehe unten*.
- Ansetzen einer Reihe von Aufgaben, die zu einem bestimmten Zeitpunkt ausgeführt werden.

Batch-Datei mit einer Folge von Befehlen

Eine Folge auszuführender Befehle wird wie folgt eingegeben:

```
@ECHO OFF
CLS
StyleVision TestEN.sps -outxslt=HTML-EN.xslt
StyleVision TestDE.sps -outxslt=HTML-DE.xslt
StyleVision TestES.sps -outxslt=HTML-ES.xslt
```

Bei Verarbeitung der Batch-Datei werden die Befehle ausgeführt und die Dateien generiert. Die obige Batch-Datei generiert mittels StyleVision drei XSLT-Dateien anhand einer SPS-Datei

11 Menübefehle und Referenz

Dieser Abschnitt enthält eine vollständige Beschreibung der Symbolleisten, der Design-Ansicht-Symbole und der Menübefehle von StyleVision. Der Abschnitt ist in die folgenden Unterabschnitte gegliedert:

- eine Beschreibung aller in der Design-Ansicht verwendeten Symbole 435
- Beschreibungen des Dialogfelds "XPath-Ausdruck bearbeiten 440
- eine Beschreibung aller Symbolleisten mit ihren Schaltflächen 457 sowie eine Anleitung zum Anpassen der Symbolleisten
- alle Menübefehle.

Während die Benutzerreferenz eine Beschreibung der einzelnen Befehle enthält, werden die Mechanismen, die verschiedenen StyleVision Funktionen zugrunde liegen, in den entsprechenden Abschnitten näher beschrieben. Die Mechanismen wurden in die folgenden Gruppen gegliedert:

- Inhalt von SPS-Dateien 110
- Struktur der SPS-Datei 184
- SPS-Datei - Komplexe Funktionen 256
- SPS-Datei - Darstellung 327
- Zusätzliche Funktionen für die SPS-Datei 361

Siehe auch

- Benutzeroberfläche 27
- Schnellstart-Tutorial 55

© 2019-2025 Altova GmbH

11.1 Symbole in der Design-Ansicht

Ein SPS Design enthält normalerweise mehrere Arten von Komponenten. Jede Komponente wird durch ein bestimmtes Symbol im Design dargestellt. Diese Symbole sind nachfolgend aufgelistet und in die folgenden Gruppen unterteilt:

- Nodes im XML-Dokument 435
- XML-Dokument-Inhalt
- Dateneingabelelemente
- Vordefinierte Formate
- XPath-Objekte
- URI-Objekte

Jeder dieser Komponententypen kann:

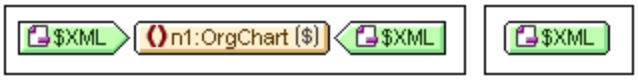
- mit Drag und Drop verschoben werden;
- (i) mit Hilfe der Befehle im Menü "Ansicht" 501 (ii) oder der Windows-Standardtastaturkürzel für diese Befehle ausgeschnitten, kopiert, eingefügt und gelöscht werden;
- mit einer Formatierung versehen werden;
- Bei Rechtsklick darauf wird ein Kontextmenü aufgerufen.

Nodes im XML-Dokument

Element- und Attribut-Nodes im XML-Dokument werden im SPS Design-Dokument durch Tags dargestellt. Jeder Node hat einen Start-Tag und einen End-Tag. Durch Doppelklick auf den Start- oder End-Tag wird der Node reduziert. Dabei wird der gesamte Node-Inhalt ausgeblendet. Wenn Sie auf einen reduzierten Node doppelklicken, wird er erweitert und der Inhalt wird angezeigt.

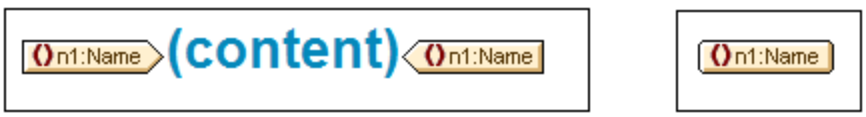
Die folgenden Arten von Nodes können dargestellt werden:

- ***Dokument-Node***



Der **Dokument-Node** (dargestellt durch `$XML`) steht für das XML-Dokument in seiner Gesamtheit. Er wird durch einen grünen `$XML` Tag gekennzeichnet, wenn die Schemaquelle mit einem XML-Dokument verknüpft ist und mit `$DB`, wenn die Schemaquelle mit einer Datenbank verknüpft ist. Der Dokument-Node in der Abbildung oben links ist erweitert und enthält das Element `OrgChart` in seiner reduzierten Form. Der Dokument-Node in der Abbildung auf der rechten Seite ist reduziert; sein Inhalt wurde ausgeblendet.

- ***Element-Node***



Ein **Element-Node** wird mit seinen übergeordneten Elementen eingefügt, wenn die übergeordneten Elemente am Einfügepunkt nicht vorhanden sind. In der Abbildung oben sehen Sie den Element-Node `Name` in erweiterter Form *(links)* und in reduzierter Form *(rechts)*.

- ***Attribut-Node***

Ein **Attribut-Node** wird mit seinen übergeordneten Elementen eingefügt, wenn die übergeordneten Elemente am Einfügepunkt nicht vorhanden sind. Attribut-Namen enthalten das Präfix `=`. In der Abbildung oben sehen Sie den Attribut-Node `href` in erweiterter Form *(links)* und in reduzierter Form *(rechts)*.

Nodes werden in das Design als Node-Vorlagen eingefügt. Informationen über die verschiedenen Arten von Vorlagen, die in das Design inkludiert werden können, finden Sie im Abschnitt Vorlagen und Design-Fragmente 230.

XML-Dokument-Inhalt

Der XML-Dokument-Inhalt wird durch zwei Platzhalter dargestellt:

- `(Inhalt)`
- `(Restlicher Inhalt)`

Der Platzhalter **`Inhalt`** steht für den Inhalt eines einzelnen Nodes. Der gesamte Textinhalt des Node wird ausgegeben. Wenn der Node ein Attribut-Node oder ein reiner Text-Element-Node ist, wird der Wert des Node ausgegeben. Wenn der Node ein Element-Node, der gemischten Inhalt enthält, oder ein reiner Element-Node ist, wird der Textinhalt aller untergeordneten Elemente ausgegeben. In XSLT bedeutet dies, dass der Platzhalter `Inhalt` dem Element `xsl:apply-templates` entspricht, wobei sein `select` Attribut für diesen Node festgelegt ist.

Anmerkung: Bei Anwendung auf einen Element-Node gibt der Platzhalter `Inhalt` die Attributwerte dieses Elements nicht aus. Zur Ausgabe von Attribut-Nodes müssen Sie das Attribut explizit in die Vorlage (Hauptvorlage oder globale Vorlage) inkludieren.

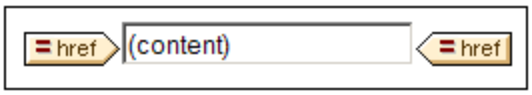
Mit dem Platzhalter **`restlicher Inhalt`** werden Vorlagen auf die restlichen Child-Element des aktuellen Node angewendet. Bei der Vorlage, die für die einzelnen Child-Elemente angewendet wird, handelt es sich in diesem Fall entweder um eine globale Vorlage (wenn für dieses Element eine definiert ist) oder die Standardvorlage für Elemente (welche nur den Text von reinen Textelementen ausgibt und Vorlagen auf Child-Elemente anwendet). Angenommen Sie haben ein Element `book`, welches die Child-Elemente: `title`, `author`, `isbn` und `pubdate` enthält. Wenn in der Definition von `book` festgelegt ist, dass nur das Child-Element `title` ausgegeben werden soll, wird bei der Verarbeitung der Definition keines der anderen Child-Elemente (`author`, `isbn`, and `pubdate`) ausgegeben. Wenn die Definition von `book` dagegen den Platzhalter `restlicher Inhalt` nach der Definition für das Element `title` enthält, wird für jedes der anderen Child-Elemente (`author`, `isbn` und `pubdate`), eine globale Vorlage (wenn eine für dieses Element vorhanden ist) oder die Standardvorlage für Elemente angewendet.

© 2019-2025 Altova GmbH

Dateneingabeelemente

Um dem Benutzer der Authentic-Ansicht die korrekte Eingabe gültiger Daten zu erleichtern, können im Design Dateneingabeelemente verwendet werden. Sie können einem Node jedes der folgenden Dateneingabeelemente zuweisen:

- ***Eingabefelder (ein- oder mehrzeilig)***



- ***Auswahllisten***

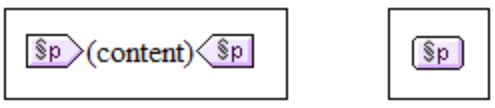
- ***Kontrollkästchen***

- ***Optionsfelder***

Diese Tags können durch Doppelklick auf den erweiterten bzw. reduzierten Tag reduziert bzw. erweitert werden. Eine ausführliche Beschreibung zur Verwendung der einzelnen Dateneingabeelemente finden Sie unter Dateneingabeelemente[158].

Vordefinierte Formate

Verdefinierte Formate werden mit violetten Tags angezeigt, die durch Doppelklick erweitert und reduziert werden können.



In der Abbildung oben sehen Sie Tags für das vordefinierte Format `p (para)`, auf der linken Seite in erweiterter und auf der rechten Seite in reduzierter Form. Um ein vordefiniertes Format anzuwenden, markieren Sie das Objekt, das in das vordefinierte Format eingeschlossen werden soll (durch Klicken auf die Komponente und/oder Markieren von Text) und fügen Sie das vordefinierte Format ein[328].

XPath-Objekte

In StyleVision werden XPath-Ausdrücke auf zwei Arten verwendet:

- ***Bedingungsvorlagen***

Bedingungstags sind blau. Der Start-Tag enthält Zellen. Die Zelle ganz links enthält ein Fragezeichen. Andere Zellen enthalten entweder (i) eine fortlaufende Zahl für jede `when` Bedingung - beginnend mit eins; und/oder (ii) ein Sternchen für die Bedingung `andernfalls`. Eine Bedingungsverzweigung kann durch Anklicken ausgewählt werden. Die Nummer der ausgewählten Bedingungsverzweigung erscheint im Start-Tag markiert und die Vorlage für diese Verzweigung wird (innerhalb des Start- und End-Tags der Bedingung) angezeigt. Der XPath-Ausdruck für die ausgewählte Bedingungsverzweigung wird auch in der Design-Struktur markiert. Beachten Sie, dass Tags für Bedingungen nicht erweitert/reduziert werden können.

- ***Automatische Berechnungen***

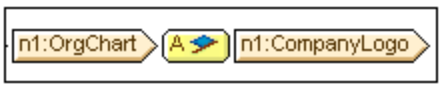
Automatische Berechnungen werden in der Design-Ansicht durch das Objekt `=(XPath)` dargestellt (*siehe Abbildung oben*). Der XPath-Ausdruck für die ausgewählte automatische Berechnung wird in der Design-Struktur hervorgehoben. Das Dialogfeld zum Bearbeiten von automatischen Berechnungen wird über die Seitenleiste "Eigenschaften" aufgerufen 257.

URI-Objekte

Es gibt drei URI-basierte Objekte, die in ein Design eingefügt werden können:

- ***Bilder***
  Wenn ein Bild in das SPS Design eingefügt wird und von StyleVision aufgerufen werden kann, wird es in der Design-Ansicht angezeigt. Falls es nicht aufgerufen werden kann, wird die Stelle, an der es im SPS eingefügt wurde, durch einen Bild-Platzhalter markiert.

- ***Lesezeichen (Anchors)***



Lesezeichen-Tags sind gelb und werden durch den Buchstaben `A` markiert (*Abbildung oben*). Ein Lesezeichen wird mit Hilfe des Befehls **Einfügen | Lesezeichen einfügen** erstellt und kann entweder

© 2019-2025 Altova GmbH

leer sein oder Inhalt enthalten. Der Inhalt muss immer erst nach Erstellung des Lesezeichens eingefügt werden. Lesezeichen-Tags können erweitert (*Abbildung oben links*) oder reduziert (*Abbildung oben rechts*) werden.

- ***Links***

Link-Tags sind gelb und sind durch den Buchstaben A gekennzeichnet (*Abbildung oben*). Ein Link wird mit dem Befehl **Einfügen | Hyperlink** erstellt. Das Link-Objekt kann vor oder nach der Erstellung des Link erstellt werden. Um ein Objekt als Link zu erstellen, wählen Sie es aus und schließen Sie es in einen Link ein. Link-Tags können erweitert (*Abbildung oben links*) oder reduziert (*Abbildung oben rechts*) werden.

Siehe auch

- Symbolleisten 457
- Design-Seitenleisten 33
- Verfahren zur Bearbeitung von Inhalt 110

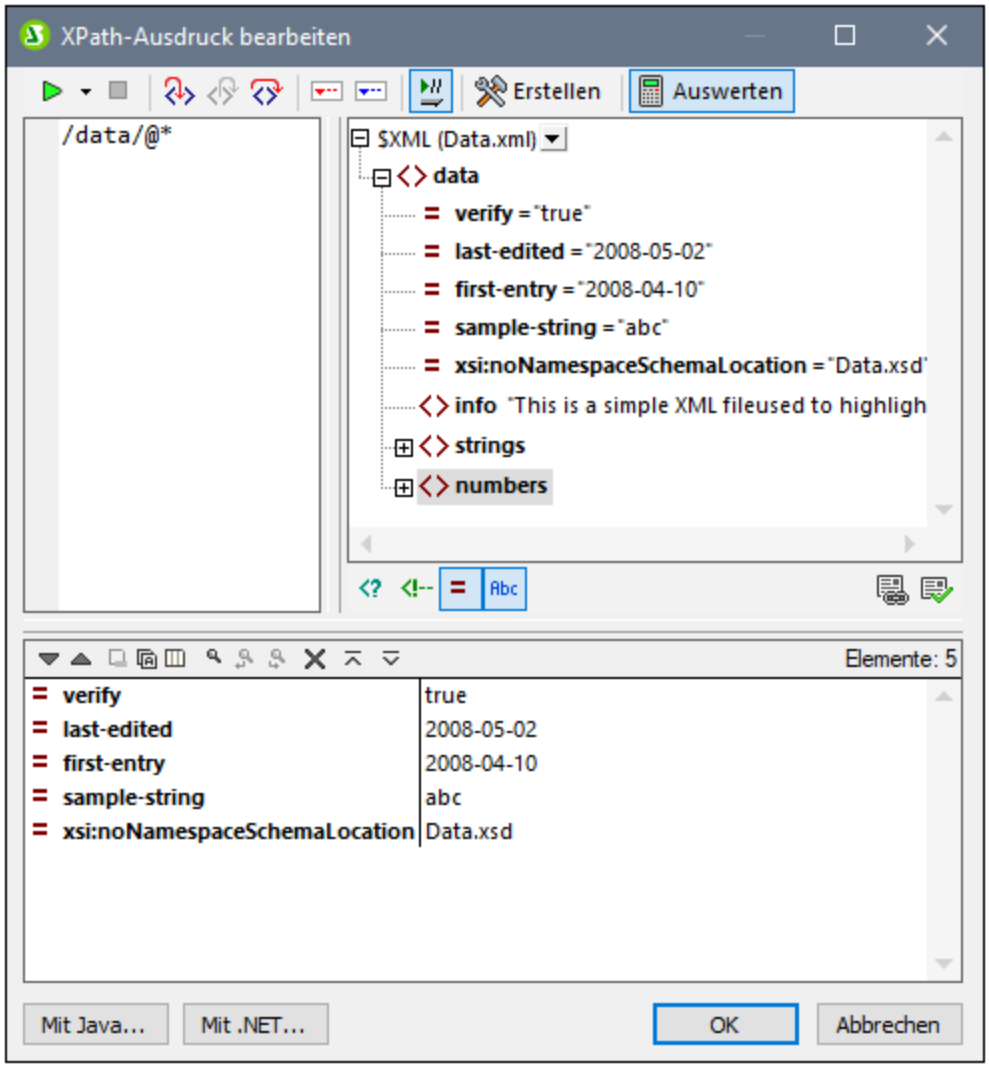
11.2 Dialogfeld "'XPath-Ausdruck bearbeiten"

Über das Dialogfeld **XPath-Ausdruck bearbeiten** (*Abbildung unten)* können XPath-Ausdrücke erstellt, getestet und bearbeitet werden. Es kann von allen Stellen in der Design-Ansicht, an denen ein XPath-Ausdruck eingegeben werden kann, wie z.B. bei der Eingabe von Ausdrücken für eine bedingte Verarbeitung[262] oder der Werte von Stilen[47] und Eigenschaften[49], aufgerufen werden.

Das Dialogfeld unterstützt automatisch die **XPath-Version**, die der XSLT-Version[100] des SPS (XPath 1.0 für XSLT 1.0, XPath 2.0 für XSLT 2.0 und XPath 3.1 für XSLT 3.0) entspricht. Um die XPath-Version zu wechseln, wechseln Sie die XSLT-Version des SPS[100].

Dialogfeld-Layout

Das Dialogfeld "XPath-Ausdruck bearbeiten" enthält die folgenden Bereiche *(siehe Abbildung unten)*: (i) einen Ausdrucksbereich *(links oben)*; (ii) einen Bereich "Quellen" *(rechts oben)*; (iii) einen Ergebnisbereich *(unten)*. Im Erstellungsmodus sehen Sie statt eines Ergebnisbereichs Eingabehilfebereiche.



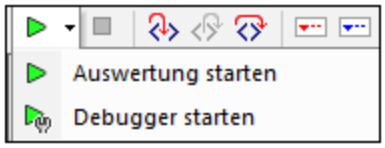
 © 2019-2025 Altova GmbH

Auswertungsmodus und Debug-Modus

Das Dialogfeld "XPath-Ausdruck bearbeiten" kann in zwei Modi verwendet werden:

- Auswertungsmodus[441]: In diesem Modus wird ein XPath-Ausdruck anhand der zugewiesenen XML-Arbeitsdatei(en) ausgewertet. Der Ausdruck wird in den **Ausdrucksbereich** eingegeben, woraufhin das Ergebnis im daneben liegenden **Ergebnisbereich** angezeigt wird. Wenn Sie auf Nodes im Ergebnis klicken, gelangen Sie zu diesem Node im Bereich "Quellen" des Dialogfelds.
- Debug-Modus[445]: In diesem Modus können Sie einen XPath-Ausdruck während der Anwendung an der/den zugewiesenen XML-Arbeitsdatei(en) debuggen. Sie können Breakpoints und Tracepoints setzen und sich Schritt für Schritt durch die Auswertung bewegen. Bei jedem Schritt sehen Sie den Inhalt der Variablen. Außerdem können Sie benutzerdefinierte Überwachungsausdrücke festlegen, um zusätzliche Aspekte der Auswertung zu überprüfen.

Um zwischen den beiden Modi zu wechseln, wählen Sie den entsprechenden Befehl im Dropdown-Menü **Auswertung/Debugging starten** in der linken Ecke der Symbolleiste des Fensters (*siehe Abbildung unten*) .



Eine Anleitung zur Verwendung der beiden Modi finden Sie in den Unterabschnitten dieses Abschnitts.

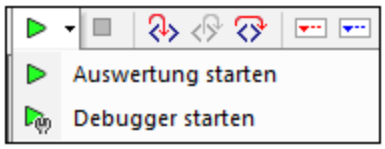
XPath/XQuery-Ausdruckserstellung

In beiden Modi können Sie mit Hilfe der Ausdruckserstellung[452] syntaktisch korrekte Ausdrücke erstellen.

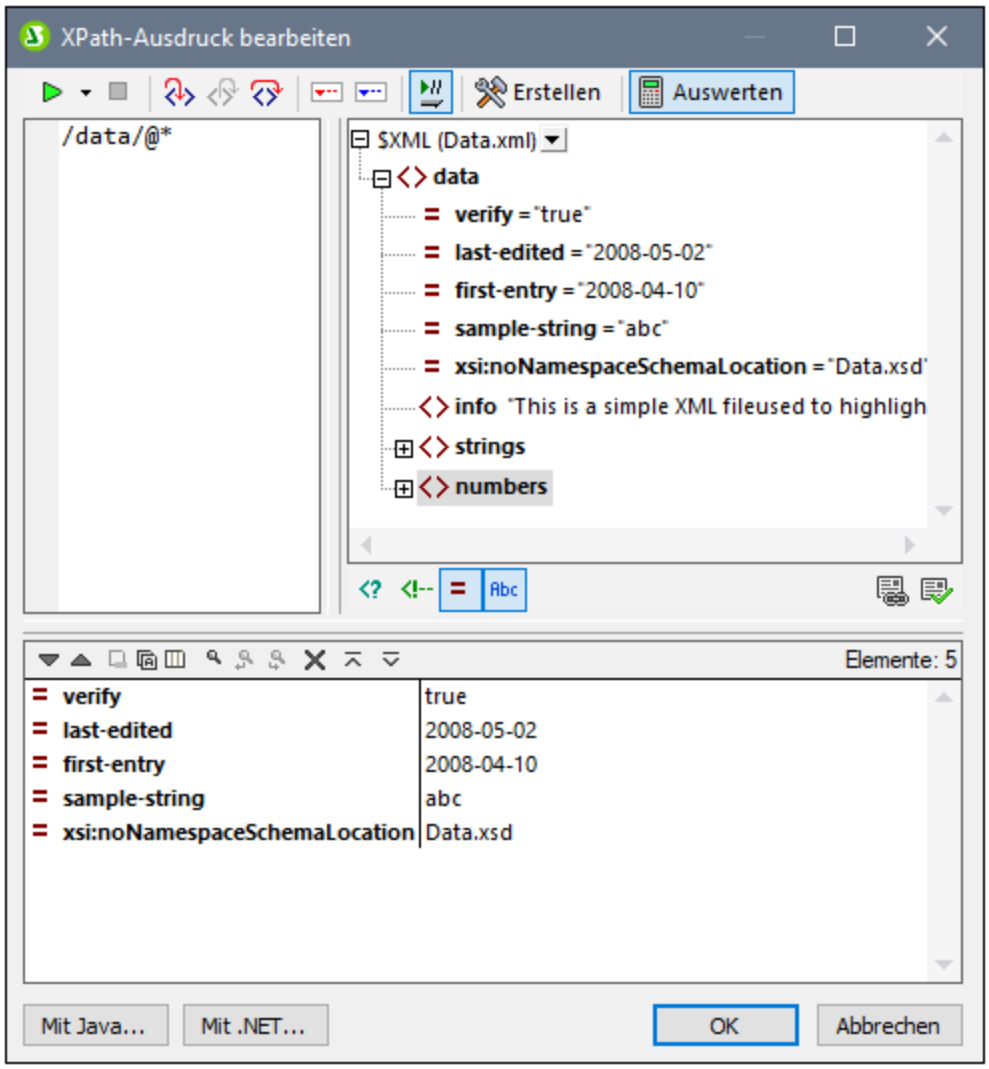
Über die **Erstellungsmodus**-Schaltfläche in der Hauptsymbolleiste können Sie die Ausdruckserstellung[452] ein und ausschalten.

11.2.1 Auswertung

Wählen Sie im Dropdown-Menü des Befehls **Auswertung/Debugging starten (F5)** *(siehe Abbildung unten)* den Befehl **Auswertung starten** aus.



Klicken Sie im Auswerungsmodus auf die Schaltfläche **Auswertung** (*siehe Abbildung unten*). Der Auswertungsmodus hat die folgenden Bereiche *(siehe Abbildung unten)*: (i) einen Ausdrucksbereich *(links oben)*; (ii) einen Bereich "Quellen" *(rechts oben)*; (iii) einen Ergebnisbereich *(unten)*.



Der XPath-Ausdruck und seine Auswertung

Der XPath-Ausdruck wird in den Ausdrucksbereich eingegeben. Die Ergebnisse der Auswertung werden im Ergebnisbereich (*siehe Abbildung oben*) angezeigt.

Beachten Sie die folgenden Punkte:

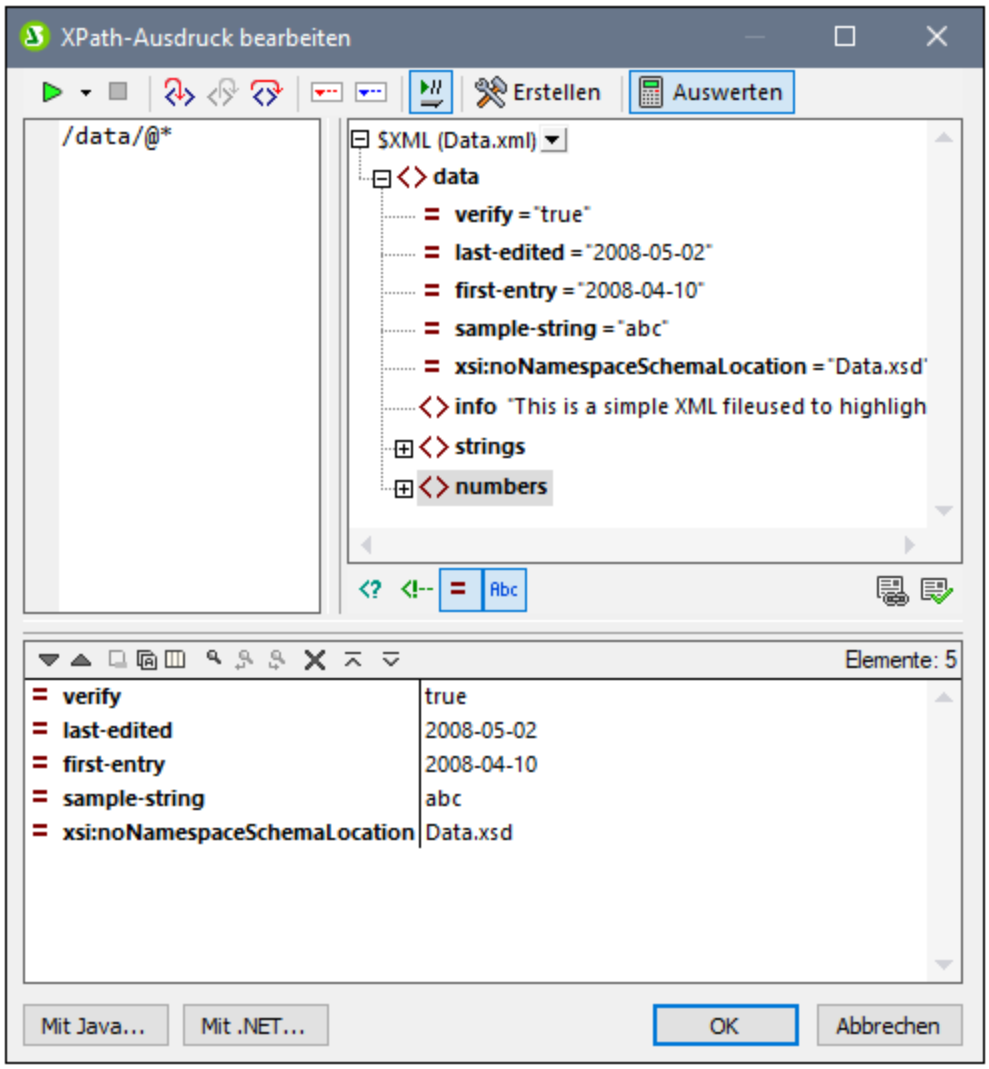
- Um einen Ausdruck anhand einer XML-Datei auszuwerten, muss die entsprechende Datei als XML-Arbeitsdatei einer der Quellen des SPS zugewiesen sein 98.
- Ergebnisse können auch schon während der Eingabe des Ausdrucks angezeigt werden (aktivieren Sie dazu in der Symbolleiste die Schaltfläche **Bei der Eingabe auswerten.** Oder diese können angezeigt werden, wenn Sie auf die Symbolleisten-Schaltfläche **Auswertung/Debugging starten (F5)** klicken (links oben in der Symbolleiste).
- Um den XPath-Locator-Ausdruck eines Node in einer Quellstruktur einzugeben, doppelklicken Sie im Bereich "Quellen" auf den entsprechenden Node.

© 2019-2025 Altova GmbH

- Im Bereich "Quellen" können Sie über die Schaltflächen unterhalb des Bereichs "Quellen" die Anzeige von (i) Processing Instructions, (ii) Kommentaren, (iii) Attributen und (iv) Elementen ein- und ausschalten.
- Der Kontext-Node ist der Node der Design-Komponente, in dem der Ausdruck erstellt wird. Um einen anderen Node als den Kontext-Node des Ausdrucks zu definieren, (i) wählen Sie den Node im Bereich "Quellen" aus und (ii) klicken Sie (unterhalb des Bereichs "Quellen") auf **Auswertungskontext definieren**. Um diesen Kontext-Node für den Ausdruck zu speichern, klicken Sie (unterhalb des Bereichs "Quellen") auf den Ein/Aus-Befehl Auswertungskontext merken. Beachten Sie jedoch, dass als eigentlicher Kontext-Node für den Ausdruck der Kontext-Node der aktuellen Design-Komponente verwendet wird und dasss dies der Kontext-Node ist, der zur Laufzeit verwendet wird.
- Sie können in XPath-Ausdrücken die Funktionen der Programmiersprachen Java und .NET verwenden. Bei Klick auf die Schaltflächen **Mit Java** und **Mit .NET** am unteren Rand des Dialogfelds werden Infokästen mit Erklärungen, wie man Java- und .NET-Erweiterungsfunktionen in XPath-Ausdrücken verwendet, angezeigt. Nähere Informationen zu diesem Thema finden Sie in dieser Dokumentation im Abschnitt Erweiterungsfunktionen 676.
- Um den Ausdruck aus Gründen der Lesbarkeit in mehreren Zeilen zu erstellen, drücken Sie die **Eingabetaste**.
- Um den Text im Ausdrucksfeld zu vergrößern/zu verkleinern, klicken Sie in das Ausdrucksfeld und drehen Sie bei gedrückter **Strg**-Taste das Mausrad. **Beachten Sie, dass dies auch für das Ergebnisfenster gilt.**
- Anstatt den Locator-Pfad-Ausdruck eines Node manuell einzugeben, können Sie folgendermaßen vorgehen: (i) Platzieren Sie den Cursor an die Stelle im XPath-Ausdruck, an der der Locator-Pfad eingegeben werden soll; (ii) Setzen Sie den Cursor im Bereich "Quellen" in den Node, der als Ziel-Node verwendet werden soll; Daraufhin wird der Locator-Pfad des ausgewählten Node in den Ausdruck eingegeben. Der Locator-Pfad wird als absoluter Pfad, der am Root-Node des Dokuments beginnt, eingegeben.

Ergebnisfenster

Das Ergebnisfenster ist der untere Bereich in der Abbildung unten. Beachten Sie, dass dieser eine eigene Symbolleiste hat.



Das Ergebnisfenster verfügt über die folgenden Funktionalitäten:

- Die Ergebnisliste besteht aus zwei Spalten: (i) einem Node-Namen oder Datentyp; (ii) dem Inhalt des Node.
- Wenn der XPath-Ausdruck Nodes zurückgibt - wie z.B. Elemente oder Attribute - können Sie auswählen, ob der gesamte Inhalt der Nodes als Wert des Node angezeigt werden soll. Klicken Sie dazu auf die Schaltfläche *Gesamtes Ergebnis anzeigen.*
- Wenn das Ergebnis - im Gegensatz zu mit einem Ausdruck generierten Literalen - einen Node enthält (auch einen Text-Node), wird der entsprechende Node in der Quellstruktur im XML-Dokument markiert, wenn Sie im Ergebnisbereich darauf klicken.
- Sie können beide Spalten einer Ergebnis-Subzeile oder nur die Wertespalte kopieren. Um alle Spalten zu kopieren, klicken Sie mit der rechten Maustaste auf eine Subzeile und aktivieren Sie den Befehl **Beim Kopieren werden alle Spalten inkludiert**. (Alternativ dazu können Sie diesen Befehl über die Symbolleisten-Schaltfläche im Ergebnisbereich ein- und ausschalten.) Klicken Sie anschließend mit der rechten Maustaste auf die gewünschte Subzeile und wählen Sie entweder **Subzeile kopieren** (für diese Subzeile) oder **Alles kopieren** (für alle Subzeilen).

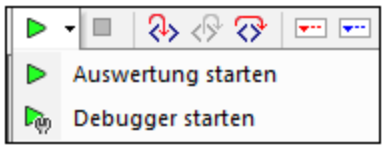
 © 2019-2025 Altova GmbH

Symbolleiste des Ergebnisfensters
Die Symbolleiste des Ergebnisbereichs enthält Symbole für Navigations-, Such- und Kopierfunktionen. In der unten stehenden Tabelle finden Sie eine Beschreibung der Symbole von links nach rechts. Die dazugehörigen Befehle stehen auch über das Kontextmenü von Einträgen der Ergebnisliste zur Verfügung.

Symbol	Funktion
Weiter, Zurück	Wählt jeweils den nächsten bzw. vorhergehenden Eintrag in der Ergebnisliste aus
Ausgewählte Textzeile in die Zwischenablage kopieren	Kopiert die Wertespalte des ausgewählten Ergebniseintrags in die Zwischenablage. Um alle Spalten zu kopieren, aktivieren Sie den Befehl *Beim Kopieren werden alle Spalten inkludiert (siehe unten)*
Alle Nachrichten in die Zwischenablage kopieren	Kopiert die Wertespalte aller Ergebniseinträge, darunter auch leere Werte, in die Zwischenablage. Jeder Eintrag wird als separate Zeile kopiert.
Beim Kopieren werden alle Spalten inkludiert	Wechselt zwischen der Funktion zum Kopieren (i) aller Spalten und (ii) nur der Wertespalte. Das Spaltentrennzeichen ist ein einziges Leerzeichen.
Suchen	Öffnet ein *Suchdialogfeld* zum Suchen nach einem beliebigen String - darunter auch von Sonderzeichen - in der Ergebnisliste.
Vorheriges suchen	Sucht nach der vorhergehenden Instanz des zuletzt im *Suchdialogfeld* gesuchten Eintrags.
Weitersuchen	Sucht nach der nächsten Instanz des zuletzt im *Suchdialogfeld* gesuchten Eintrags.
Mit Child Nodes erweitern	Erweitert den ausgewählten Eintrag und alle seine Nachfahren
Mit Child Nodes reduzieren	Reduziert den ausgewählten Eintrag und alle seine Nachfahren
Zurücksetzen	Löscht die Ergebnisliste.

11.2.2 Debugger

Mit Hilfe des Debuggers können Sie einen XPath-Ausdruck im Kontext einer XML-Arbeitsdatei debuggen 98. Um den Debugger aufzurufen, klicken Sie im Dropdown-Menü des Befehls **Auswertung/Debugging starten** (*siehe Abbildung unten*) auf **Debugger starten**. Dadurch wird der Debug-Modus ausgewählt. Sie können anschließend zwischen der Ausdruckserstellung (Hilfe bei der Erstellung des Ausdrucks) und der Auswertung (zum Debuggen des Ausdrucks) wechseln. Um mit dem Debuggen zu beginnen, klicken Sie auf **Auswertung/Debugging starten (F5).**



Nachdem Sie einen Ausdruck eingegeben haben, können Sie durch Klick auf **Auswertung/Debugging starten (F5)** mit dem Debuggen beginnen (Stellen Sie sicher, dass Sie sich im Debug-Modus befinden).

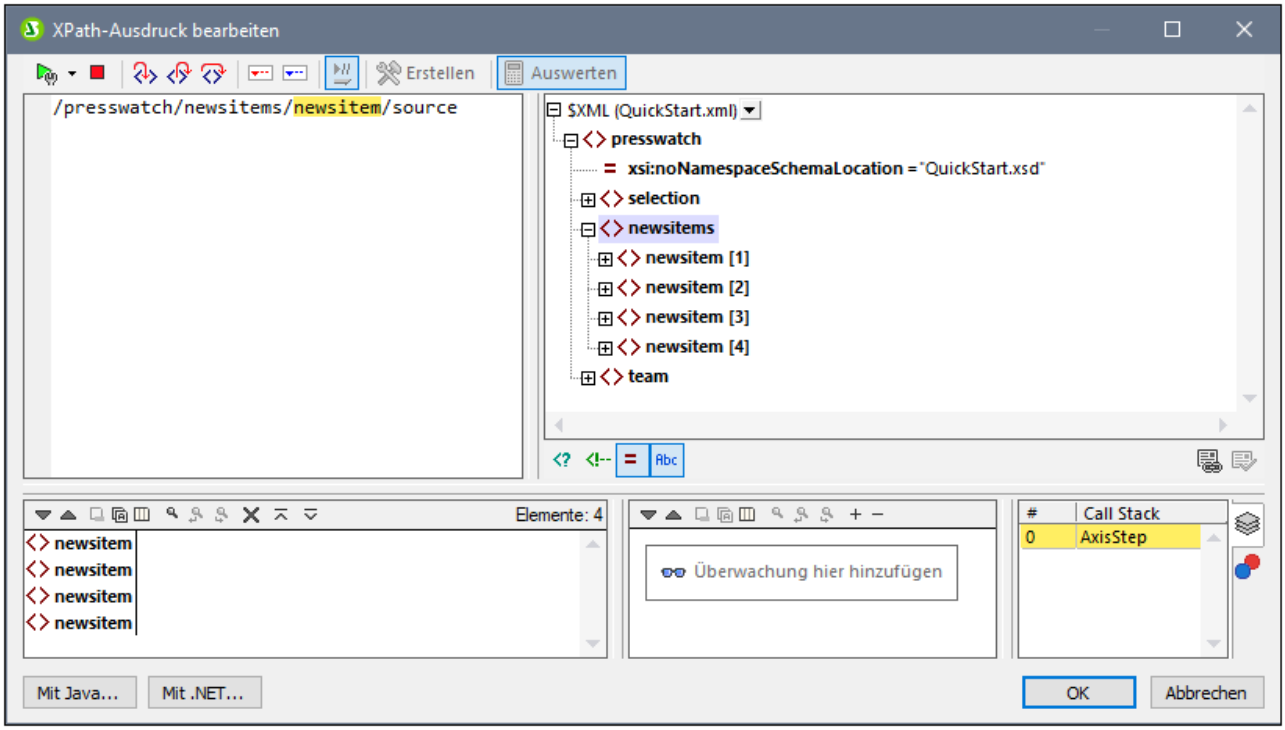
Schaltflächen für die Konfiguration des Debug-Modus

	Auswertung/Debugging starten (F5)	Startet den Debugger.
	Zum Erstellungsmodus wechseln	Wechselt in den Ausdruckserstellungsmodus, in dem Ihnen bei der Erstellung von Ausdrücken kontextsensitive Eingabehilfen zur Verfügung stehen.
	Bei der Eingabe auswerten	Aktiviert die Auswertung des Ausdrucks, während dieser eingegeben wird.

Layout des Debug-Modus

Im Debug-Modus werden zum Ergebnisbereich zwei zusätzliche Bereiche hinzugefügt (*siehe Abbildung unten*):

- der Call Stack- und der Debug-Punkte-Bereich. Jeder davon hat ein eigenes Register im Fenster.
- das Bereich für Variablen und Überwachungsausdrücke; sowohl Überwachungsausdrücke als auch Variablen werden im selben Bereich angezeigt



Im Debugger-Modus stehen die folgenden Funktionalitäten zur Verfügung:

- Hier können Sie in den XPath-Auswertungsvorgang einsteigen und Schritt für Schritt sehen, wie der XPath-Ausdruck ausgewertet wird. Verwenden Sie dazu die Symbolleisten-Schaltfläche **Einsteigen (F11)**. Bei jedem Auswertungsschritt wird der Teil des Ausdrucks, der soeben ausgewertet wird, gelb markiert (*siehe Abbildung oben*), während das Ergebnis der Auswertung dieses Schritts im

 © 2019-2025 Altova GmbH

Ergebnisbereich angezeigt wird. So wurden z.B. in der Abbildung oben alle `section`-Nachfahren des Elements `book` ausgewählt.

- Sie können Breakpoints setzen, wo die Auswertung angehalten werden soll, damit Sie Ergebnisse an diesen Stellen überprüfen können. Sie können die Auswertung durchlaufen lassen, indem Sie nur an den Breakpoints anhalten. Verwenden Sie dafür die Symbolleisten-Schaltfläche **Debugger starten (F5).** Dies geht schneller, als wenn Sie bei jedem Schritt mit **Einsteigen (F11)** anhalten.
- Sie können Tracepoints setzen, um einen Bericht über die Ergebnisse an den als Tracepoints markierten Stellen zu sehen. Die Auswertung wird dabei nicht angehalten (außer bei Breakpoints), sondern die Tracepoint-Ergebnisse werden in einer Liste im Ergebnisbereich angezeigt.
- Mit Hilfe von Überwachungsausdrücken können bestimmte Informationen (wie Dokumentdaten oder Auswertungsaspekte) überprüft werden. Dies ist vor allem an Breakpoints nützlich.
- Variablen, die sich im Geltungsbereich befinden, werden einschließlich ihrer Werte im Bereich für Variablen und Überwachungsausdrücke angezeigt.
- Prozessoraufrufe eines Auswertungsschritts werden auf dem Register "Call Stack" des Call Stack- und Debug-Punkte-Bereichs angezeigt.
- Wenn Breakpoints und Tracepoints gesetzt wurden, werden diese auf dem Register "Debug-Punkte" des Call Stack- und Debug-Punkte-Bereichs angezeigt.

Nähere Informationen zu diesen Funktionalitäten finden Sie in der jeweiligen Beschreibung dazu weiter unten.

Ausführen des Debuggers

Das Debuggen eines XPath-Ausdrucks erfolgt in den folgenden Schritten:

1. Geben Sie den XPath-Ausdruck in den Ausdrucksbereich ein.
2. Setzen Sie die gewünschten Breakpoints oder Tracepoints. Ein Breakpoint ist ein Punkt, an dem die Auswertung unterbrochen wird. Ein Tracepoint ist ein Punkt in der Auswertung, der aufgezeichnet wird, sodass Sie einen zurückverfolgbaren Pfad der Auswertungsergebnisse zur Verfügung haben.
3. Wenn Sie auf **Debugger starten** klicken, wird die Auswertung in einem einzigen Schritt bis zum Ende ausgeführt, es sei denn, es wurde ein Breakpoint im Ausdruck markiert. Klicken Sie wiederholt auf **Debugger starten,** um nach jedem Breakpoint fortzufahren und bis zum Ende der Auswertung zu gelangen.
4. Mit Hilfe der Funktionen Einsteigen/Aussteigen/Überspringen können Sie sich Schritt für Schritt durch die Auswertung bewegen.

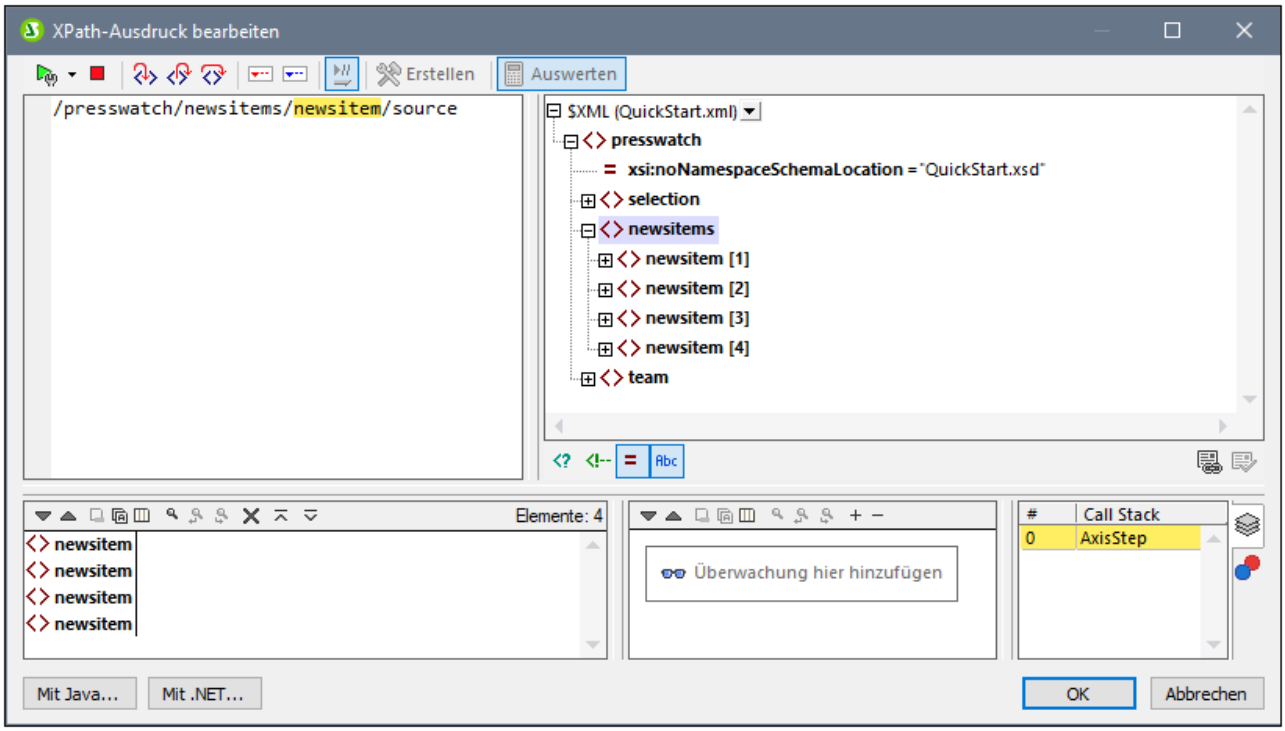
Schaltflächen für das Debuggen

	Debugger starten (F5)	Startet den Debugger. Die Auswertung wird direkt bis zum Ende durchgeführt und wird nur bei Breakpoints angehalten.
	Debugging beenden (Umschalt+F5)	Beendet die Auswertung und stoppt den Debugger.
	Einsteigen (F11)	Steigt Schritt für Schritt durch die Auswertung.
	Aussteigen (Umschalt+F11)	Steigt aus dem aktuellen Auswertungsschritt aus und geht zum übergeordneten Schritt.
	Überspringen (Strg+F11)	Überspringt untergeordnete Schritte.
	Breakpoint einfügen/löschen (F9)	Fügt an dem Ausführungsschritt, in den Sie den Cursor platzieren, einen Breakpoint ein, bzw. entfernt diesen.

	Tracepoint einfügen/löschen (Umschalt+F9)	Fügt an dem Ausführungsschritt, in den Sie den Cursor platzieren, einen Tracepoint ein, bzw. entfernt diesen.

Einsteigen, Aussteigen und Überspringen von Auswertungsschritten
Mit der Funktion "*Einsteigen*" können Sie die Auswertung Schritt für Schritt durchgehen. Jeder Klick leitet Sie durch den entsprechenden Schritt der Auswertung. Der aktuelle Schritt wird im Ausdruck markiert (*siehe Abbildung unten*). Mit der Funktion *Aussteigen* gelangen Sie zur nächsthöheren Ebene und mit *Überspringen* werden Schritte auf niedrigeren Ebenen übersprungen und Sie gelangen zum nächsten Schritt auf derselben Ebene. Um die Schrittfunktionen zu probieren, verwenden Sie den unten stehenden Ausdruck und klicken Sie auf die drei *Schritt*-Schaltflächen, um zu sehen, wie diese funktionieren.

In der Abbildung unten sehen Sie die Auswertung an der Stelle, an der die Verarbeitung bei Erreichen des Locator-Schritts Additionsschritts **`newsitem`** angehalten wird. In diesem Schritt wird der Node **`newsitem`** angezeigt.



Breakpoints

Breakpoints sind Punkte, an denen der Debugger angehalten werden soll, nachdem er mit **Debugger starten** gestartet wurde. Sie eignen sich dazu, einen bestimmten Teil eines Ausdrucks zu analysieren. Wenn der Debugger an einem Breakpoint anhält, können Sie das Ergebnis überprüfen. Mit Hilfe der Funktion **Einsteigen** können Sie anschließend die Ergebnisse des nächsten Schritts der Auswertung anzeigen. Um einen Breakpoint zu setzen, platzieren Sie den Cursor an der Stelle in den Ausdruck, an der Sie den Breakpoint setzen möchten und klicken Sie auf die Symbolleisten-Schaltfläche **Breakpoint einfügen/löschen (F9)**. Der Breakpoint wird daraufhin mit einem rot strichlierten Überstrich markiert (*siehe Abbildung unten*). Um einen Breakpoint zu löschen, wählen Sie ihn aus und klicken Sie auf **Breakpoint einfügen/löschen (F9).**

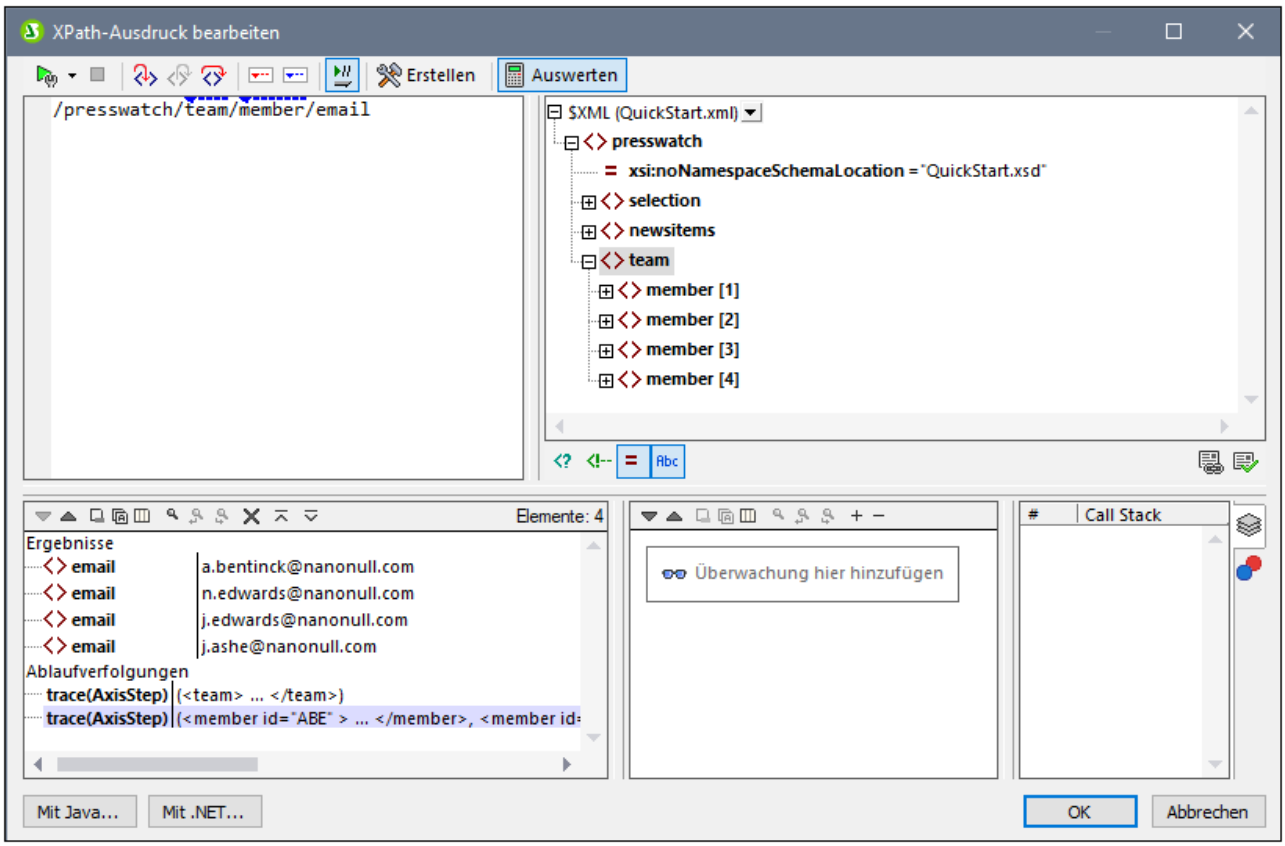
© 2019-2025 Altova GmbH

Siehe auch Debug-Punkte[451] weiter unten.

Tracepoints

Tracepoints sind Breakpoints, an denen die Ergebnisse aufgezeichnet und in der *Ablaufverfolgungsstruktur* des Registers *Ergebnis* (*siehe Abbildung unten*) angezeigt werden. Dadurch sehen Sie alle Auswertungsergebnisse bestimmter Teile des Ausdrucks. So wurden z.B. in der Abbildung unten Tracepoints in den Nodes **`team`** und **`member`** gesetzt. Die Ergebnisse dieser Tracepoints werden in der Ablaufverfolgungsstruktur angezeigt.

Um einen Tracepoint zu setzen, platzieren Sie den Cursor an der Stelle in den Ausdruck, an der Sie den Tracepoint setzen möchten und klicken Sie auf die Symbolleisten-Schaltfläche **Tracepoint einfügen/löschen (Umschalt+F9)**. Der Tracepoint wird daraufhin mit einem blau strichlierten Überstrich markiert (*siehe Abbildung unten*). Um einen Tracepoint zu löschen, wählen Sie ihn aus und klicken Sie auf **Tracepoint einfügen/löschen (Umschalt+F9).**

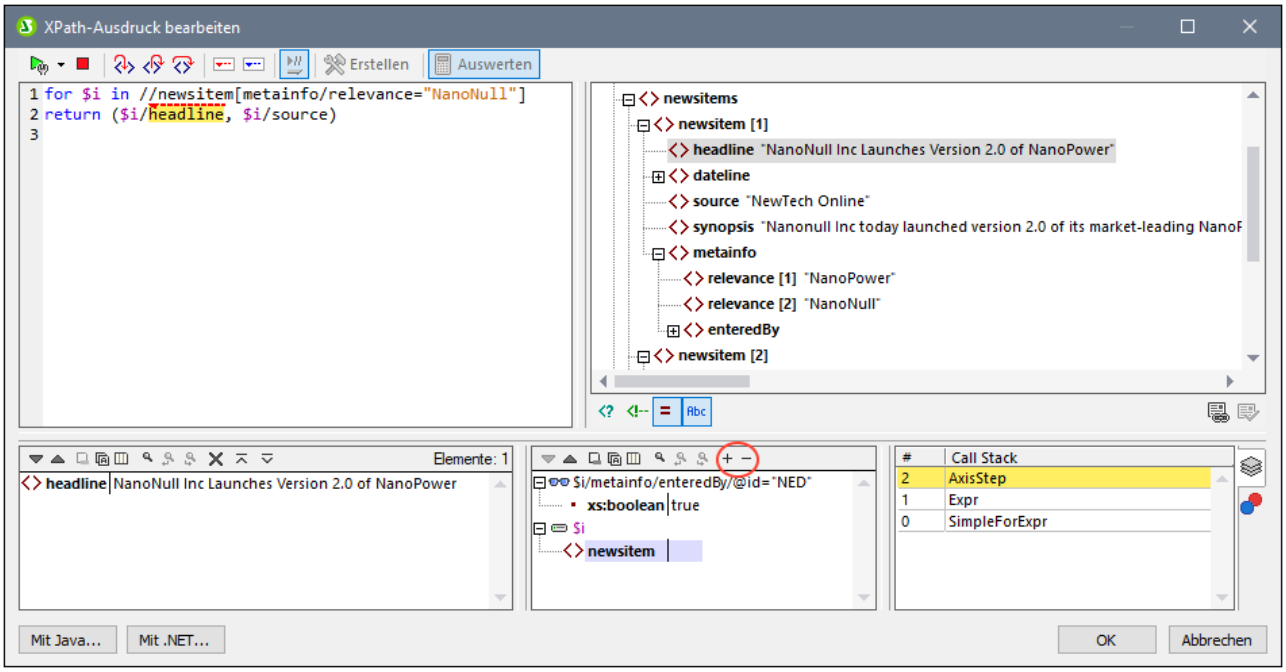


Anmerkung: Wenn an einer Stelle eines Ausdrucks sowohl ein Breakpoint als auch ein Tracepoint gesetzt wurde, so besteht der Überstrich aus abwechselnd gesetzten roten und blauen Strichen.

Siehe auch Debug-Punkte[451] weiter unten.

Variablen, Überwachungsausdrücke und Call Stack

Variablen und Überwachungsausdrücke werden im Variablen- und Überwachungsausdrucksbereich (*mittlerer unterer Bereich in der Abbildung unten*) angezeigt.



Variablen

Variablen, die im Ausdruck deklariert wurden und sich im aktuellen Auswertungsschritt im Geltungsbereich befinden, werden zusammen mit den jeweiligen aktuellen Werten angezeigt. So wurde z.B. in der Abbildung oben die Verarbeitung am Breakpoint in `headline` unterbrochen. Die Variable `$i` befindet sich in diesem Auswertungsschritt im Geltungsbereich, daher wird `$i` mit ihrem aktuellen Wert angezeigt. In der Abbildung oben ist dies der erste `newsitem` Node.

Überwachungsausdrücke

Überwachungsausdrücke sind Ausdrücke, die Sie entweder vor Beginn einer Auswertung oder während einer Unterbrechung der Auswertung eingeben können. Sie können für folgende Zwecke verwendet werden:

- um bestimmte Bedingungen zu überprüfen. So wird z.B. in der Abbildung oben mit dem Überwachungsausdruck `$i/metainfo/enteredBy/@id="NED"` überprüft, ob dieses Nachrichtenelement (newsitem) vom Team-Mitglied mit der ID `NED` eingegeben wurde. Im Fall des ersten `newsitem`-Node besagt das Ergebnis `true`, dass diese Bedingung erfüllt wird.
- um Daten in einem bestimmten Kontext zu suchen. So könnten wir z.B. im Kontext eines Elements `Company` den Überwachungsausdruck `@id` eingeben, um den Kundencode des Unternehmens im XML-Zieldokument nachzuschlagen.
- um zusätzliche Daten zu generieren. So könnte etwa ein String generiert werden, um die Gesamtanzahl der `newsitem`-Nodes anzugeben.

Um einen Überwachungsausdruck einzugeben, klicken Sie in der Symbolleiste des Fensters auf **Überwachung hinzufügen** (*in der Abbildung oben rot umrandet*), geben Sie anschließend den Ausdruck ein und drücken Sie anschließend die **Eingabetaste**. Um einen Überwachungsausdruck zu entfernen, wählen Sie ihn aus und klicken Sie in der Symbolleiste auf **Ausgewählten Überwachungsausdruck entfernen**. Wenn der Ausdruck während des Debuggens aus irgendeinem Grund (z.B. wenn sich eine seiner Variablen nicht im Geltungsbereich befindet) nicht korrekt ausgewertet werden kann, wird der Überwachungsausdruck rot angezeigt.

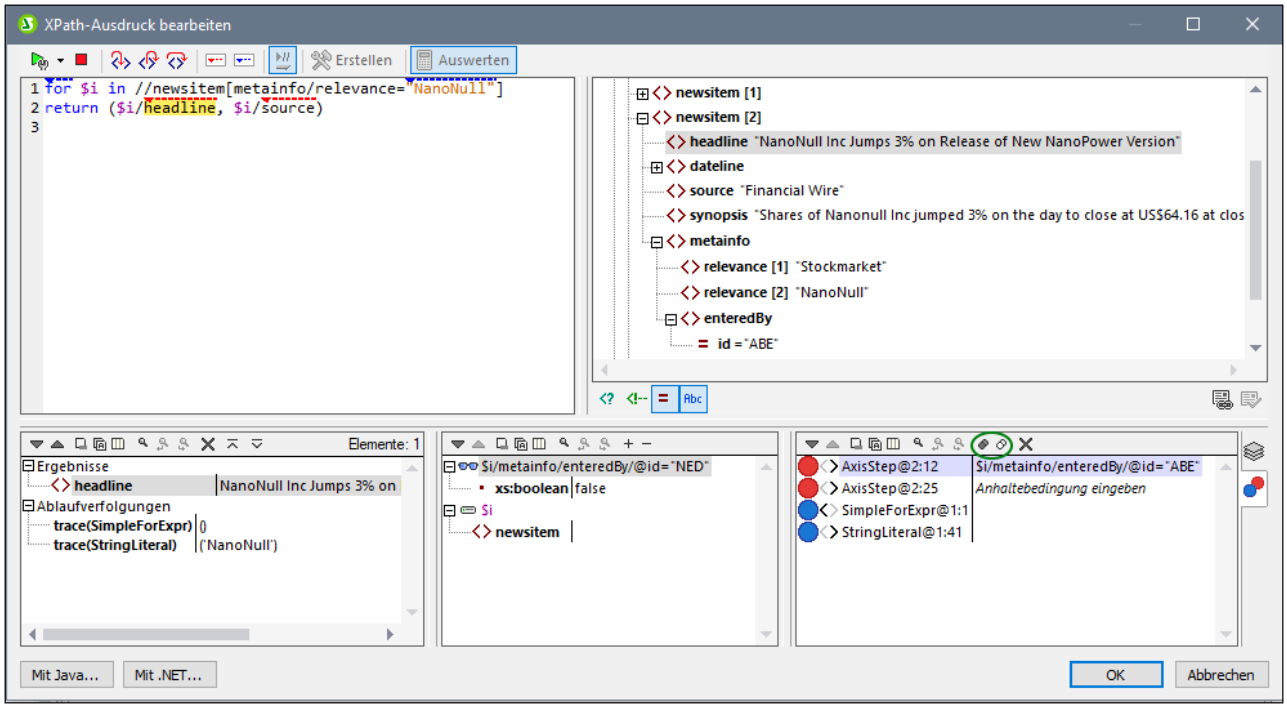
 © 2019-2025 Altova GmbH

Call Stack

Auf dem Register *Call Stack* des Call Stack- und Debug-Punkte-Bereichs (*rechter unterer Bereich in der Abbildung oben*) werden die Prozessoraufrufe bis zu diesem Punkt im Debugging angezeigt. Der aktuelle Prozessoraufruf erscheint gelb markiert. Beachten Sie, dass nur die Aufrufe, die direkt zum aktuellen Auswertungsschritt geführt haben, angezeigt werden.

Debug-Punkte

Auf dem Register *Debug-Punkte* des Call Stack- und Debug-Punkte-Bereichs (*rechter unterer Bereich in der Abbildung oben*) werden die im Ausdruck gesetzten Breakpoints (rote Kreise) und Tracepoints (blaue Kreise) angezeigt. Jeder Debug-Punkt wird mit der jeweiligen Zeilen- und Zeichennummer angezeigt. So bedeutet etwa `AxisStep@2:12`, dass sich in Zeile 2, bei Zeichen 12 des Ausdrucks im Ausdrucksbereich ein Debug-Punkt befindet.



Beachten Sie die folgenden Funktionalitäten:

- Durch Doppelklick auf *Anhaltebedingung eingeben* im Bereich Debug-Punkte, Eingabe des Ausdrucks für die Bedingung und Drücken der **Eingabetaste** können Sie eine **Break-Bedingung** für Breakpoints eingeben. Dieser Breakpoint ist nur dann aktiv, wenn das Resultat der Auswertung der Bedingung `true` ist. So aktiviert die Break-Bedingung `$i/metainfo/enteredBy/@id="ABE"` die Breakpoints in der Überschrift jedes Nachrichtenelements `newsitem`, das vom Team-Mitglied mit der ID `ABE` eingegeben wurde. In der Abbildung sehen Sie, dass die Auswertung an diesem Breakpoint angehalten wurde. (Beachten Sie außerdem, dass der Überwachungsausdruck an diesem Breakpoint `false` zurückgibt).
- Durch Klick auf die entsprechende Symbolleisten-Schaltfläche **Alle Debug-Punkte aktivieren** und **Alle Debug-Punkte deaktivieren** (*grün umrandete Schaltflächen in der Abbildung oben*) können Sie alle Debug-Punkte aktivieren bzw. deaktivieren. Wenn ein Debug-Punkt deaktiviert ist, bleibt er bis zu seiner erneuten Aktivierung für alle Auswertungen deaktiviert.
- Einzelne Breakpoints können im dazugehörigen Kontextmenü aktiviert bzw. deaktiviert werden.

Symbolleisten-Schaltflächen in den Fenstern

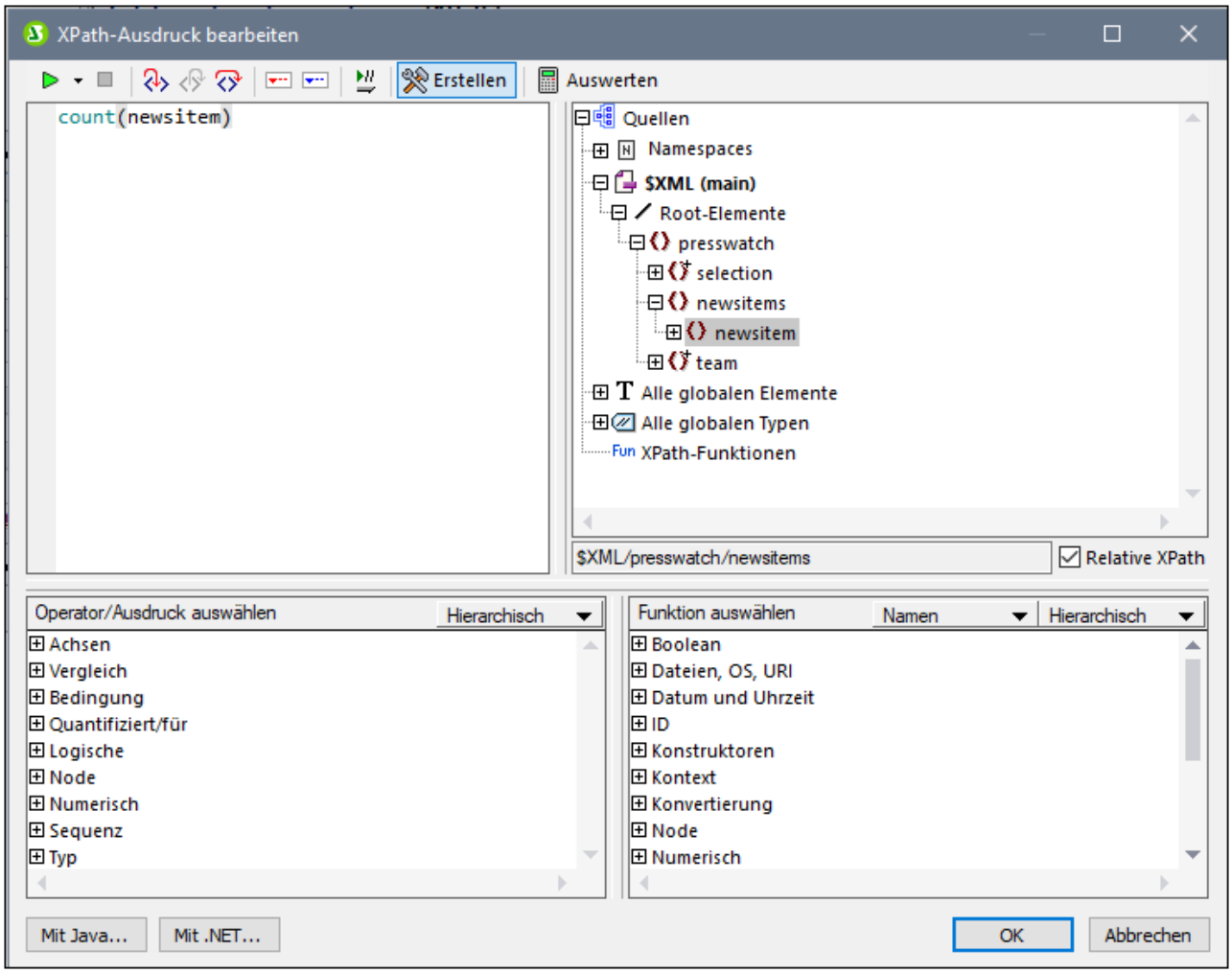
Die Bereiche des Dialogfelds "XPath-Ausdruck bearbeiten" im Debug-Modus (*siehe Abbildung oben*) enthalten Schaltflächen für die Navigation, Suche und Kopierfunktionen. Diese Schaltflächen werden in der Tabelle unten von links nach rechts beschrieben. Die dazugehörigen Befehle finden Sie auch im Kontextmenü der aufgelisteten Einträge.

Symbol	Funktion
Weiter, Zurück	Wählt jeweils den nächsten bzw. vorhergehenden Eintrag in der Ergebnisliste aus
Ausgewählte Textzeile in die Zwischenablage kopieren	Kopiert die Wertespalte des ausgewählten Ergebniseintrags in die Zwischenablage. Um alle Spalten zu kopieren, aktivieren Sie den Befehl *Beim Kopieren werden alle Spalten inkludiert (siehe unten)*
Alle Nachrichten in die Zwischenablage kopieren	Kopiert die Wertespalte aller Ergebniseinträge, darunter auch leere Werte, in die Zwischenablage. Jeder Eintrag wird als separate Zeile kopiert.
Beim Kopieren werden alle Spalten inkludiert	Wechselt zwischen der Funktion zum Kopieren (i) aller Spalten und (ii) nur der Wertespalte. Das Spaltentrennzeichen ist ein einziges Leerzeichen.
Suchen	Öffnet ein *Suchdialogfeld* zum Suchen nach einem beliebigen String - darunter auch von Sonderzeichen - in der Ergebnisliste.
Vorheriges suchen	Sucht nach der vorhergehenden Instanz des zuletzt im *Suchdialogfeld* gesuchten Eintrags.
Weitersuchen	Sucht nach der nächsten Instanz des zuletzt im *Suchdialogfeld* gesuchten Eintrags.
Mit Child Nodes erweitern	Erweitert den ausgewählten Eintrag und alle seine Nachfahren
Mit Child Nodes reduzieren	Reduziert den ausgewählten Eintrag und alle seine Nachfahren
Zurücksetzen	Löscht die Ergebnisliste.

11.2.3 Ausdruckserstellung

Wenn Sie im Dialogfeld "XPath-Ausdruck bearbeiten" auf die Schaltflächen **Erstellen** klicken (*siehe Abbildung unten*), werden Eingabehilfen zum Erstellen eines XPath-Ausdrucks angezeigt. Wenn Sie auf einen der Einträge darin doppelklicken, wird dieser an der aktuellen Cursorposition in den XPath-Ausdruck eingefügt.

© 2019-2025 Altova GmbH



Es gibt drei Eingabehilfebereiche:

- Eine Schemastruktur zur Eingabe der Element- und Attribut-Nodes in den XPath-Ausdruck. Wenn das Kontrollkästchen *Relativer XPath* aktiviert ist, so wird der Pfad zum ausgewählten Node relativ zum Kontext-Node (dem Node im Design, in dem der XPath-Ausdruck erstellt wird) eingegeben. Der Kontext-Node wird unterhalb des Schemastrukturbereichs angezeigt. Ein absoluter XPath-Ausdruck beginnt am Dokument-Root-Element und wird für den ausgewählten Node verwendet, wenn das Kontrollkästchen *Relativer XPath* deaktiviert ist.
- Eine Eingabehilfenbereich für Operatoren und Ausdrücke: Dazu gehören (i) Achsen (`ancestor::`, `parent::`, usw.) , (ii) Operatoren (z.B. `eq` und `div`) und (iii) Ausdrücke (`for # in # return #`, usw.). Die Elementen in diesem Bereich können entweder in alphabetischer Reihenfolge oder nach Funktionskategorie gruppiert aufgelistet werden. Wählen Sie die gewünschte Option, indem Sie aus dem Dropdown-Menü in der Titelleiste des Bereichs entweder *Hierarchisch* oder *Flache Hierarchie* auswählen.
- Eine Eingabehilfe mit den Funktionen der aktiven XPath-Version, die alphabetisch oder nach Funktionskategorie aufgelistet werden. Wählen Sie die gewünschte Option, indem Sie aus dem Dropdown-Menü in der Titelleiste des Bereichs entweder *Hierarchisch* oder *Flache Hierarchie* auswählen. Über die Option *Namen/Typen* können Sie auswählen, ob die Argumente von Funktionen als Namen oder Datentypen angezeigt werden sollen.

Funktionen des Bereichs "Erstellung":

- Um eine Textbeschreibung eines Eintrags in einem der Bereiche zu sehen, platzieren Sie den Mauszeiger darüber.
- Jede Funktion wird mit ihrer Signatur angezeigt (d.h. mit ihren Argumenten, den Datentypen der Argumente und dem Datentyp der Funktionsausgabe).
- Die Signaturen werden entweder nach dem Namen oder dem Datentyp der Funktionsargumente und der Funktionsausgabe aufgelistet. Wählen Sie aus dem Dropdown-Menü in der Titelleiste des Bereichs entweder *Namen* oder *Typen* aus.
- Wenn Sie in einem der Bereiche auf einen Eintrag doppelklicken (Operator, Ausdruck oder Funktion), so wird dieser an der Cursorposition in den Ausdruck eingefügt. Funktionen werden mit ihren Argumenten in Form von Platzhaltern (**#**-Symbol) eingefügt.
- Wenn (i) im Bearbeitungsfeld des XPath-Ausdrucks Text ausgewählt ist und (ii) Sie auf einen Ausdruck oder eine Funktion mit einem Platzhalter doppelklicken, um sie einzufügen, so wird anstelle des Platzhalters der ausgewählte Text eingefügt.

Nachdem Sie eine Funktion in den Ausdruck eingefügt haben, werden bei Platzieren der Maus über den Funktionsnamen die Signatur der Funktion und eine Textbeschreibung der Funktion angezeigt. Wenn für eine Funktion unterschiedliche Signaturen mit demselben Namen vorhanden sind, so werden diese am unteren Rand der Anzeige mit einem Überladungsfaktor angezeigt. Wenn Sie den Cursor innerhalb der Klammern der Funktion platzieren und **Strg+Umschalt+Leertaste** drücken, sehen Sie die Signaturen der verschiedenen Überladungen dieses Funktionsnamens.

Erstellen von XPath-Ausdrücken

Über das Dialogfeld "XPath-Ausdruck bearbeiten" können Sie auf folgende Arten XPath-Ausdrücke erstellen.

- *Kontext-Node und Schemastruktur*
  Im Auswahl-Textfeld im Bereich "Quellen" sehen Sie sofort den Kontext-Node. Der Ausdruck wird an einer Stelle innerhalb dieses Kontext-Node eingefügt und wird im Kontext dieses Node ausgewertet.

- *Einfügen eines Node aus der Schemastruktur*
  Im Bereich *Quellen* wird das gesamte Schema angezeigt. Durch Doppelklick auf den entsprechenden Node können Sie diesen in den XPath-Ausdruck einfügen. Wenn das Kontrollkästchen *Relativer XPath* aktiviert ist, wird der ausgewählte Node mit einem Pfadausdruck, die relativ zum Kontext-Node ist, eingefügt.

- *Namespace-Informationen*
  Die Schemastruktur im Bereich *Quellen* enthält einen Namespace-Eintrag. Wenn Sie diesen Eintrag erweitern, werden alle im Stylesheet deklarierten Namespaces angezeigt. Diese Informationen sind oft hilfreich um die Präfixe eines Namespace zu überprüfen, den Sie in einem XPath-Ausdruck verwenden möchten.

- *Einfügen von XPath-Achsen, Operatoren und -Ausdrücken*
  Im Bereich *Operator/Ausdruck auswählen* werden je nach XPath-Version XPath-Achsen (`ancestor::`, `parent::`, usw.), Operatoren (z.B. `eq` und `div`) und Ausdrücke (`for # in # return #`, usw.) aufgelistet. Sie können zwischen einer alphabetischen und einer hierarchischen Liste (in der die Elemente nach Funktion gruppiert werden) wechseln. Um eine Achse oder einen Operator in den XPath-Ausdruck einzufügen, doppelklicken Sie auf den gewünschten Eintrag.

- *Einfügen von XPath-Funktionen*

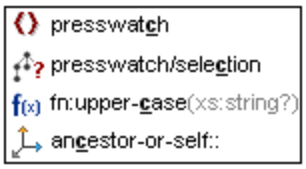
 © 2019-2025 Altova GmbH

Im Bereich *Funktion auswählen* werden die XPath-Funktionen alphabetisch oder nach Funktionalität gruppiert aufgelistet (Klicken Sie auf das entsprechende Symbol oben im Bereich, um zwischen den beiden Anzeigearten zu wechseln). Jede Funktion wird mit ihrer Signatur aufgelistet. Wenn eine Funktion mehrere Signaturen hat, wird sie so oft aufgelistet, wie sie Signaturen hat. Argumente in einer Signatur sind durch Kommas getrennt und jedes Argument kann einen Instanzindikator haben (`?` bedeutet eine Sequenz von null oder einem Objekt des angegebenen Typs; `*` bedeutet eine Sequenz von null oder mehr Objekten des angegebenen Typs). Die Argumente können als Namen oder Datentypen angezeigt werden; wählen Sie in der Titelleiste des Bereichs *Namen* oder *Datentypen* aus. Für jede Funktion ist auch der Rückgabetyp dieser Funktion definiert. So zeigt z.B. `=> date ?` an, dass es sich beim erwarteten Rückgabedatentyp um eine Sequenz von keinem oder einem `date` Element handelt. Wenn Sie den Mauszeiger über eine Funktion positionieren, wird eine kurze Beschreibung der Funktion angezeigt. Um eine Funktion in den XPath-Ausdruck einzufügen, doppelklicken Sie auf die gewünschte Funktion.

- In XPath-Ausdrücken können Java- und .NET-Erweiterungsfunktionen verwendet werden, wodurch Sie Zugriff auf die Funktionen dieser Programmiersprachen erhalten. Über die Schaltflächen **Java** und **.NET** am unteren Rand des Dialogfelds können Sie Info-Felder mit Erklärungen zur Verwendung von Java- und .NET-Erweiterungsfunktionen in XPath-Ausdrücken aufrufen. Nähere Informationen dazu finden Sie in dieser Dokumentation im Abschnitt Erweiterungsfunktionen 657.

Intelligente Bearbeitung während der Texteingabe

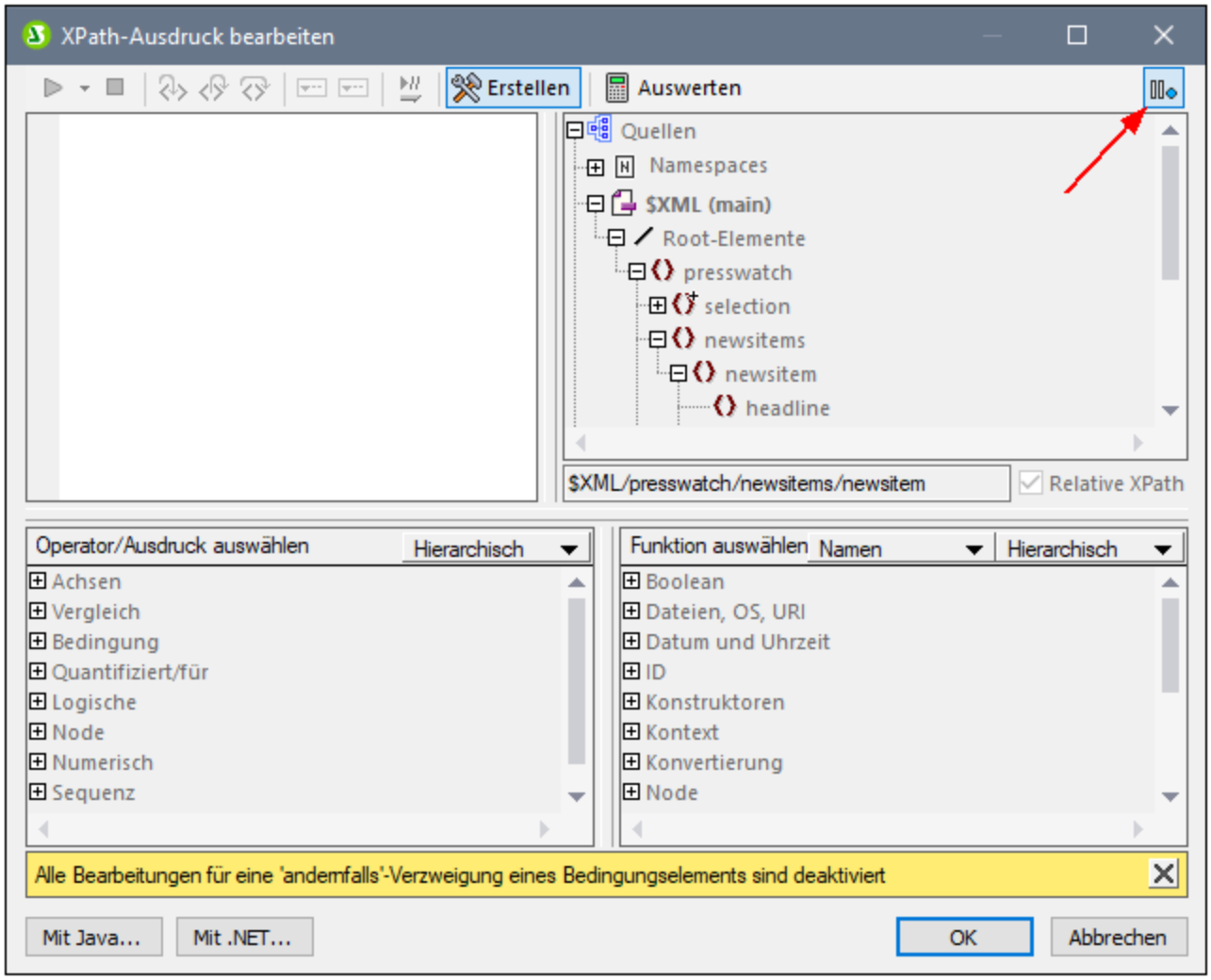
Wenn Sie einen Ausdruck in das Textfeld *Ausdruck* eingeben, werden die verfügbaren Optionen in einem Popup-Fenster angezeigt (*Abbildung unten*).



Zur Auswahl stehen Elemente, XPath-Funktionen und XPath-Achsen. Mit Hilfe der **Nach oben** und **Nach unten**-Tasten können Sie durch die Liste der Optionen navigieren. Drücken Sie die **Eingabetaste**, um eine Option auszuwählen und in den Ausdruck einzufügen.

Das Kontrollkästchen "Andernfalls"

Über die Ein/Aus-Schaltfläche *Andernfalls* (*siehe roter Pfeil in der Abbildung unten*) wird als letzte Verzweigung eine Andernfalls-Verzweigung zu einer Bedingungsvorlage hinzugefügt. In einer Bedingungsvorlage darf nur eine einzige Andernfalls-Verzweigung vorhanden sein. Wenn eine Bedingungsvorlage ausgewertet wird, wird die erste Verzweigung, deren Ergebnis `true` ergibt, ausgeführt. Wenn keine der Verzweigungen zu `true` ausgewertet wird, so wird die Andernfalls-Verzweigung ausgeführt, falls vorhanden. Andernfalls wird die Bedingungsvorlage geschlossen, ohne dass irgendeine ihrer Verzweigungen ausgeführt wird. Da die Andernfalls-Bedingung nur dann ausgeführt wird, wenn keine der vorhergehenden Bedingungen zu `true` ausgewertet wurde, muss keine Bedingung dafür definiert werden. Wenn Sie daher das Kontrollkästchen "Andernfalls" aktivieren, ist das Eingabefeld des XPath-Ausdrucks deaktiviert.



Nähere Informationen zur Verwendung der Andernfalls-Bedingung finden Sie unter Bedingungsvorlagen[262].

© 2019-2025 Altova GmbH

11.3 Symbolleisten

Eine Reihe von StyleVision Befehlen können über die Symbolleisten aufgerufen werden. Es stehen die folgenden Symbolleisten zur Verfügung:

- Formatierung 459
- Tabelle 460
- Einfügen von Design-Elementen 462
- Designfilter 464
- Globale Ressourcen 465
- Standard 466

Die Symbole in den einzelnen Symbolleisten werden in den Unterabschnitten dieses Abschnitts aufgelistet. Zu jedem Symbol erhalten Sie eine kurze Beschreibung des entsprechenden Befehls.

Positionieren der Symbolleisten

Symbolleisten können freischwebend oder entlang des Rands der Benutzeroberfläche platziert werden. Meist werden die Symbolleisten entlang des oberen Rands der Benutzeroberfläche, direkt unterhalb der Menüleiste positioniert. Sie können jedoch auch am seitlichen oder unteren Rand der Benutzeroberfläche positioniert werden.

Um eine Symbolleiste an eine bestimmte Stelle zu verschieben, gehen Sie folgendermaßen vor:

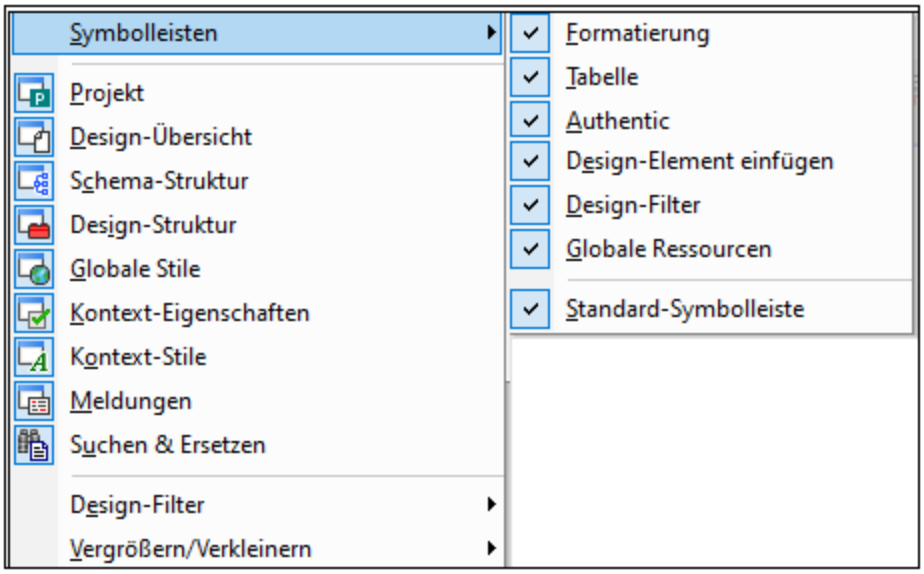
1. Ziehen Sie die Symbolleiste an ihrem Ziehpunkt (wenn sich die Symbolleiste bereits im Symbolleistenbereich befindet) oder an der Titelleiste (bei einer freischwebenden Symbolleiste).
2. Ziehen Sie die Symbolleiste in den gewünschten Symbolleistenbereich und lassen Sie die Maustaste an der gewünschten Stelle los. Wenn an dem Rand, an dem Sie die Symbolleiste platzieren wollen, kein Symbolleistenbereich vorhanden ist, wird beim Loslassen der Maustaste automatisch ein solcher Bereich an dieser Stelle erstellt.

Um die Symbolleiste freischwebend anzuzeigen, ziehen Sie sie am Ziehpunkt aus dem Symbolleistenbereich mitten in den Bildschirmbereich hinein (nicht in einen Symbolleistenbereich am Rand).

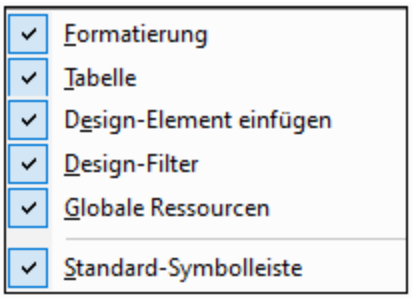
Ein- und Ausblenden der Symbolleisten

Einzelne Symbolleisten können mit Hilfe der folgenden drei Methoden ein- oder ausgeschaltet werden:

- Aktivieren bzw. deaktivieren Sie die gewünschte Symbolleiste im Menü **Ansicht | Symbolleisten** (*Abbildung unten*) um sie ein- bzw. auszuschalten.



- Rechtsklicken Sie auf eine Symbolleiste, um ein Kontextmenü anzuzeigen (*Abbildung unten*), über das Sie die einzelnen Symbolleisten ein- und ausschalten können.



- Schalten Sie die Anzeige der einzelnen Symbolleisten im Dialogfeld "Anpassen" 547 (**Extras | Anpassen** 547) auf dem Register "Symbolleisten" ein oder aus, indem Sie das Kontrollkästchen für die jeweilige Symbolleiste aktivieren bzw. deaktivieren. Klicken Sie anschließend auf **Schließen**, um das Dialogfeld zu schließen.

Hinzufügen und Entfernen von Symbolleisten-Schaltflächen

Sie können einzelne Symbolleisten-Schaltflächen zu einer Symbolleiste hinzufügen oder daraus entfernen, d.h. ein- oder ausblenden. Gehen Sie dazu folgendermaßen vor:

1. Klicken Sie in der Symbolleiste, in der die Schaltfläche hinzugefügt oder entfernt werden soll, auf die Schaltfläche **Mehr Schaltflächen** (wenn sich die Symbolleiste in einem Symbolleistenbereich befindet) oder auf die Schaltfläche **Symbolleisten-Optionen** (wenn die Symbolleiste freischwebend angezeigt wird). Die Schaltfläche **Mehr Schaltflächen** ist ein Pfeil, der (in einem horizontalen Symbolleistenbereich) rechts von der Symbolleiste oder (in einem vertikalen Symbolleistenbereich) am unteren Rand der Symbolleiste angezeigt wird. Die Schaltfläche "Symbolleisten-Optionen" ist ein Pfeil, der am rechten Rand der freischwebenden Symbolleiste angezeigt wird.
2. Platzieren Sie den Cursor im Menü **Schaltflächen hinzufügen oder löschen,** das daraufhin angezeigt wird, über den Menüeintrag **Schaltflächen hinzufügen oder löschen** (*Abbildung unten*).

 © 2019-2025 Altova GmbH

Daraufhin erscheint ein Menü mit den Namen der Symbolleisten in diesem Symbolleistenbereich sowie dem Menübefehl **Anpassen** (*Abbildung unten*).

3. Platzieren Sie den Cursor über die Symbolleiste, die die Schaltfläche enthält, die Sie hinzufügen oder löschen möchten (*Abbildung oben*).
4. Klicken Sie im Menü, das daraufhin angezeigt wird (*Abbildung oben*), auf den Namen der gewünschten Symbolleisten-Schaltfläche.
5. Wenn Sie auf den Befehl "Anpassen" klicken, wird das Dialogfeld "Anpassen" 547 angezeigt.

Über die Schaltfläche **Symbolleiste zurücksetzen,** die unterhalb der Schaltflächen im jeweiligen Symbolleisten-Menü angezeigt wird kann die Symbolleiste in den Originalzustand zurückgesetzt werden. In diesem Zustand werden alle Schaltflächen für diese Symbolleiste angezeigt.

Anmerkung: Die Schaltflächen, die eine Symbolleiste enthält, sind vordefiniert und können nicht mit einer anderen Symbolleiste verknüpft werden. Im oben beschriebenen Verfahren wird die Schaltfläche in der Symbolleiste, die auf der Benutzeroberfläche angezeigt wird, ein- oder ausgeblendet.

Siehe auch

- Benutzeroberfläche 27

11.3.1 Format

Die Symbolleiste **Format** (*Abbildung unten*) ist in der Design-Ansicht aktiv und enthält Befehle, die dem/den im SPS-Design ausgewählten Objekt(en) gebräuchliche Eigenschaften zur Formatierung von Blöcken und Texten innerhalb der Zeile zuweist.

Vordefinierte HTML-Formate
Das in der Dropdown-Liste ausgewählte HTML-Format wird in der Design-Ansicht auf die Auswahl angewendet. So wird z.B. bei Auswahl von `div` das HTML-Element `Block (div)` rund um die aktuelle Auswahl in der Design-Ansicht gesetzt.

Texteigenschaften
Die zeileninternen Texteigenschaften "fett", "kursiv" und "unterstrichen" können durch Klicken auf die entsprechende Schaltfläche direkt auf die aktuelle Auswahl in der Design-Ansicht angewendet werden. Auch der Schriftstil, die Schriftgröße sowie die Vorder- und Hintergrundfarbe können über Symbolleisten-Schaltflächen angewendet werden.

Ausrichtung
Ausrichtungseigenschaften (linksbündig, zentriert, rechtsbündig und Blocksatz) können direkt auf die Auswahl in der Design-Ansicht angewendet werden.

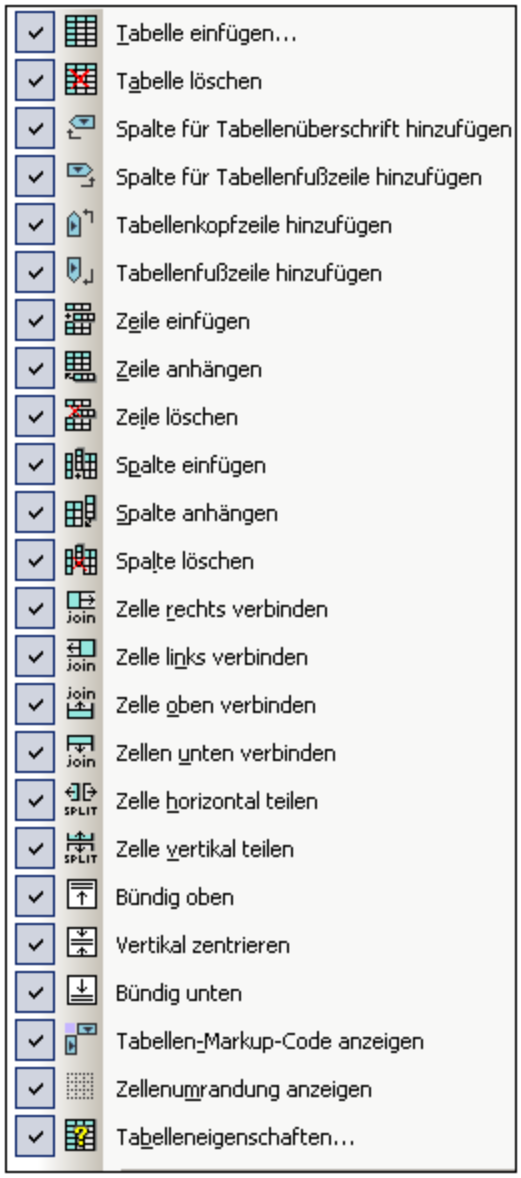
Listen
Am Cursoreinfügepunkt können Listen eingefügt werden oder die Auswahl im SPS kann in eine Liste umgewandelt werden.

Hyperlinks
Fügt an der Cursorposition einen Hyperlink ein. Eine Beschreibung zur Verwendung des Befehls finden Sie unter Hyperlink 517.

11.3.2 Tabelle

Die **Tabellen-Symbolleiste** enthält Befehle zum Strukturieren und Formatieren von statischen und dynamischen Tabellen in der Design-Ansicht. Diese Befehle sind in der Abbildung unten zu sehen (hier sehen Sie das Menü zum Anpassen der Symbolleiste "Tabelle", das Sie über die Schaltfläche "Anpassen" rechts von der Symbolleiste aufrufen können).

© 2019-2025 Altova GmbH



Zeilen- und Spaltenoperationen
Zeilen und Spalten in einer SPS Tabelle (statisch oder dynamisch) können eingefügt, angehängt oder gelöscht werden. Zeilen und Spalten werden vor der aktuellen Cursorposition eingefügt oder nach allen Zeilen/Spalten angehängt. Die Zeile/Spalte, in der sich der Cursor befindet, kann auch gelöscht werden. Diese Operationen werden mit Hilfe der Schaltflächen **Zeile/Spalte einfügen, Zeile/Spalte anhängen** oder **Zeile/Spalte löschen** durchgeführt. Sie können Tabellenkopf- und -fußzeilen auch als Spalten oder Zeilen hinzufügen.

Operationen an Zellen
Die Zelle einer SPS Tabelle, in der der Cursor sich befindet, kann mit jeder der vier benachbarten Zellen verbunden werden. Diese Operation ist der HTML-Operation zum Zusammenfügen von Tabellenzellen ähnlich. Die Schaltflächen für diese Operationen sind **Zelle rechts verbinden/Zelle links verbinden/Zelle oben verbinden/Zelle unten verbinden**. Außerdem kann die Zelle einer SPS Tabelle, in der sich der Cursor befindet, entweder in horizontaler oder vertikaler Richtung mit Hilfe der Schaltflächen **Horizontal teilen** bzw. **Vertikal teilen** geteilt werden. Der Inhalt der SPS Tabellenzelle kann oben, in der Mitte und am unteren Rand

der Zelle ausgerichtet werden. Die Umrandungen der Zelle können mit Hilfe der Schaltfläche **Zellumrandung anzeigen** ein- und ausgeblendet werden.

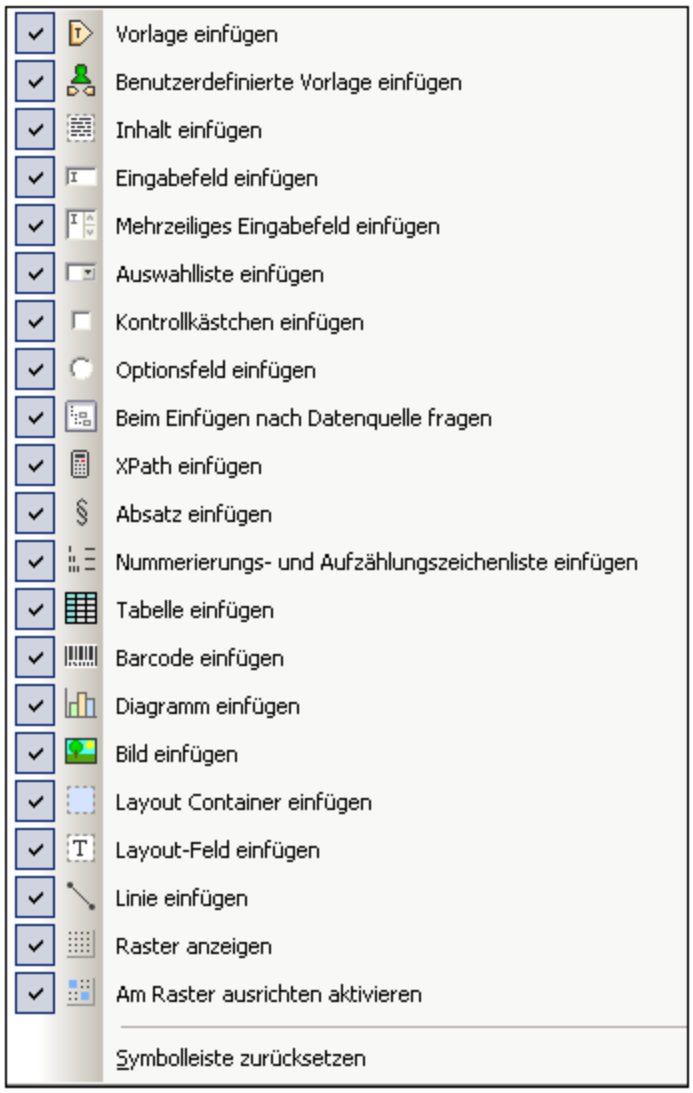
Tabellenoperationen, Eigenschaften, Anzeige
Wenn Sie den Cursor in eine statische oder dynamische Tabelle platzieren und auf **Tabelle löschen** 532 klicken, wird die Tabelle gelöscht. Tabellen-Markup-Tags können mit dem Befehl "Tabellen-Markup-Code anzeigen" ein- und ausgeblendet werden. Mit dem Befehl "Tabelleneigenschaften" wird das Dialogfeld "Tabelleneigenschaften" aufgerufen, in dem die Eigenschaften der Tabelle definiert werden können.

11.3.3 Design-Elemente einfügen

Die **Symbolleiste "Design-Elemente einfügen"** enthält Schaltflächen für Befehle zum Einfügen von Design-Elementen in das SPS Design und damit im Zusammenhang stehende Befehle. In der Abbildung unten sehen Sie die verschiedenen Design-Elemente, die über diese Symbolleisten-Schaltflächen eingefügt werden können. Die Symbolleiste enthält drei Arten von Elementen:

1. Design-Elemente 463, die Kontext-Node-sensitiv sind (die Mehrzahl der Elemente in der Symbolleiste).
2. Layout-Elemente 464, die unabhängig vom Node-Kontext sind,
3. Ein/Aus-Befehle zur Rasteranzeige 464, die als Design-Hilfe dienen.

© 2019-2025 Altova GmbH



Design-Elemente

Die Design-Elemente sind die Kontext-Node-sensitiven Elemente, die im Menü "**Einfügen**" zur Verfügung stehen. Um ein Design-Element über die entsprechende Symbolleisten-Schaltfläche einzufügen, gehen Sie folgendermaßen vor:

1. Wählen Sie die Symbolleisten-Schaltfläche für das gewünschte Element aus.
2. Klicken Sie an die Stelle im Design, an der Sie das Element einfügen möchten. Daraufhin wird ein Dialogfeld "Design-Element einfügen" für das ausgewählte Element angezeigt. Darin sehen Sie die Schema-Struktur, wobei der Kontext-Node markiert ist. Der Kontext-Node ist der Node, in den der Cursor zum Einfügen des Design-Elements platziert wurde.

3. Wenn Sie das Design-Element in den aktuell ausgewählten Node einfügen möchten, klicken Sie auf **OK.** Wenn Sie einen anderen Kontext-Node auswählen möchten, tun Sie dies in der Schema-Struktur und klicken Sie auf **OK**.
4. Bei einigen Design-Elementen wie z.B. bei automatischen Berechnungen ist ein weiterer Schritt erforderlich, wie z.B. die Definition der automatischen Berechnung. In anderen Fällen wie z.B. beim Einfügen einer benutzerdefinierten Vorlage wird das Dialogfeld "Design-Element einfügen"

übersprungen. In diesen Fällen wird ein weiteres Dialogfeld wie z.B. das Dialogfeld XPath-Ausdruck bearbeiten[440] angezeigt. Führen Sie die gewünschten Schritte aus und klicken Sie auf **OK**.

Das Design-Element wird, je nachdem, um welches Design-Element es sich handelt, am Ende von Schritt 3 oder 4 eingefügt.

Layout-Elemente

Es gibt in der Symbolleiste "Design-Elemente einfügen" drei Layout-Element-Befehle: zum Einfügen (i) eines Layout Containers; (ii) eines Layout-Felds; und (iii) einer Linie. Beachten Sie, dass Layout-Felder und Linien nur in einen Layout Container eingefügt werden können.

Um einen Layout Container einzufügen, klicken Sie auf die Schaltfläche **Layout Container einfügen** und klicken Sie dann an die Stelle, an der der Layout Container eingefügt werden soll. Sie werden gebeten, die Größe des Layout Containers zu definieren. Anschließend wird er in das Design eingefügt. Um ein Layout-Feld einzufügen, klicken Sie auf die Schaltfläche **Layout-Feld einfügen** und platzieren Sie den Cursor dann an die Stelle im Layout Container, an die das Layout-Feld eingefügt werden soll und klicken Sie. Daraufhin wird das Layout-Feld eingefügt. Klicken Sie in das Layout-Feld, um mit der Eingabe zu beginnen. Um eine Linie einzufügen, klicken Sie auf die Schaltfläche **Linie einfügen** und platzieren Sie dann den Cursor an die Stelle im Layout Container, an der Sie mit dem Zeichnen der Linie beginnen möchten. Klicken Sie, um den Anfangspunkt der Linie zu definieren und ziehen Sie die Linie dann mit der Maus an den gewünschten Endpunkt und lassen Sie dann die Maustaste los. Die Linie wird zwischen dem Anfangs- und dem Endpunkt eingefügt.

Um die Größe von Layout Containern und Layout-Feldern anzupassen, platzieren Sie den Cursor über den rechten oder unteren Rand des Layout Containers bzw. des Layout-Felds und ziehen Sie den Rand, bis Sie die gewünschte Größe erhalten haben. Um ein Layout-Feld zu verschieben, platzieren Sie den Cursor über den linken oder oberen Rand des Layout-Felds. Sobald der Cursor als Kreuz angezeigt wird, ziehen Sie das Layout-Feld an die gewünschte Stelle.

Ein/Aus-Befehle zur Rasteranzeige

Mit dem Befehl **Raster anzeigen** wird das Zeichenraster ein- und ausgeblendet. Wenn der Befehl **An Raster ausrichten** aktiviert ist, werden in einem Layout Container erstellte Elemente wie z.B. Layout-Felder und Linien an Rasterlinien und -punkten ausgerichtet. Die Eigenschaften des Rasters können im Dialogfeld "Optionen" auf dem Register "Design" (**Extras | Optionen**) definiert werden.

11.3.4 Design-Filter

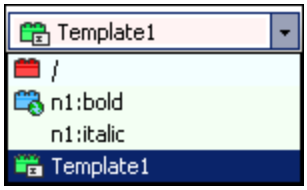
Die Symbolleiste **Design-Filter** (*Abbildung unten*) enthält Befehle, mit Hilfe derer Sie filtern können, welche Vorlagen im Design angezeigt werden. Die einzelnen Schaltflächen der Symbolleiste werden unten erläutert.

Schaltfläche	Befehl	Beschreibung
	Immer nur eine Vorlage anzeigen	Zeigt nur die ausgewählte Vorlage an. Platzieren Sie den Cursor in eine Vorlage und klicken Sie, um nur diese Vorlage anzuzeigen.

© 2019-2025 Altova GmbH

Schaltfläche	Befehl	Beschreibung
	Alle Vorlagentypen anzeigen	Zeigt alle Vorlagen im SPS an (Hauptvorlage, globale Vorlagen, Design-Fragmente und Layout-Vorlagen) .
	Importierte Vorlagen anzeigen	Blendet importierte Vorlagen ein und aus.
	Hauptvorlage ein-/ausblenden	Blendet die Hauptvorlage ein und aus.
	Globale Vorlagen ein-/ausblenden	Blendet die globalen Vorlagen ein und aus.
	Design-Fragmente ein-/ausblenden	Blendet die Design-Fragmente ein und aus.

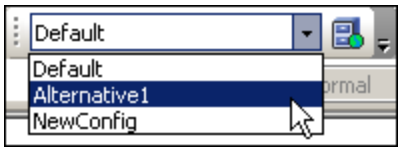
Die Design-Filter-Auswahlliste (Abbildung unten) zeigt eine Liste aller Vorlagen im SPS an.



Bei Auswahl einer Vorlage in der Auswahlliste, wird die Vorlage im Design ausgewählt. Mit Hilfe der Auswahlliste können Sie daher schnell zur gewünschten Vorlage im Design navigieren, was sich in Designs mit mehreren Vorlagen, von denen möglicherweise einige ausgeblendet sind, als hilfreich erweist.

11.3.5 Globale Ressourcen

Über die Symbolleiste **Globale Ressourcen** (*Abbildung unten*) können Sie: (i) die aktive Konfiguration für die Applikation auswählen und (ii) das Dialogfeld "Globale Altova-Ressourcen" 362 aufrufen.



Wählen Sie die aktive Konfiguration aus den Optionen in der Dropdown-Liste der Auswahlliste aus. Klicken Sie auf die Schaltfläche "Globale Ressourcen verwalten", um das Dialogfeld "Globale Altova-Ressourcen" aufzurufen.

11.3.6 Standard

Die **Standard-Symbolleiste** enthält Schaltflächen für Befehle, die wichtige Funktionen zur Bearbeitung und für die Arbeit mit Dateien zur Verfügung stellen. Unten sehen Sie eine Liste dieser Schaltflächen mit einer kurzen Beschreibung. Um eine ausführlichere Beschreibung des Befehls zu erhalten, klicken Sie darauf und gehen Sie zur Beschreibung im Abschnitt "Referenz".

Schfl.	Befehl	Tastenkürzel	Beschreibung
	Neues Design von XML-Schema / DTD 468	**Strg+N**	Erstellt ein neues SPS Dokument auf Basis eines Schemas. Wenn Sie auf den Nach-unten-Pfeil klicken, können Sie das SPS anhand einer Datenbank, eines HTML-Dokuments oder ein leeres SPS erstellen.
	Öffnen 475	**Strg+O**	Öffnet ein vorhandenes SPS Dokument.
	Design speichern 480	**Strg+S**	Speichert das aktive SPS Dokument.
	Alles speichern 480	**Strg+Umschalt+S**	Speichert alle geöffneten SPS Dokumente.
	Drucken 491	**Strg+P**	Druckt die Authentic-Ansicht der XML-Arbeitsdatei.
	Druckvorschau 491		Zeigt eine Druckvorschau der Authentic-Ansicht der XML-Arbeitsdatei an.
	Ausschneiden 493	**Umschalt+Entf**	Schneidet die Auswahl aus und platziert sie in die Zwischenablage.
	Kopieren 493	**Strg+C**	Kopiert die Auswahl in die Zwischenablage.
	Einfügen 493	**Strg+P**	Fügt den Inhalt der Zwischenablage in der Cursorposition ein.
	Löschen 493	**Entf**	Löscht die Auswahl.
	Rückgängig 493	**Alt+Rücktaste**	Macht eine Bearbeitung rückgängig. Aktionen können unbegrenzt rückgängig gemacht werden.
	Wiederherstellen 493	**Strg+Y**	Stellt eine rückgängig gemachte Bearbeitung wieder her.
	Suchen 493	**Strg+F**	Sucht Text in der Authentic -Ansicht und den Ausgabeansichten.
	Weitersuchen 493	**F3**	Sucht die nächste Instanz des gesuchten Texts.
	XSLT 1.0 100		Legt als Stylesheet Sprache XSLT 1.0 fest.
	XSLT 2.0 100		Legt als Stylesheet Sprache XSLT 2.0 fest.

© 2019-2025 Altova GmbH

Schfl.	Befehl	Tastenkürzel	Beschreibung
	XSLT 3.0 100		Legt als Stylesheet Sprache XSLT 3.0 fest.
	Rechtschreibprüfung 541		Führt am SPS Dokument eine Rechtschreibprüfung durch.

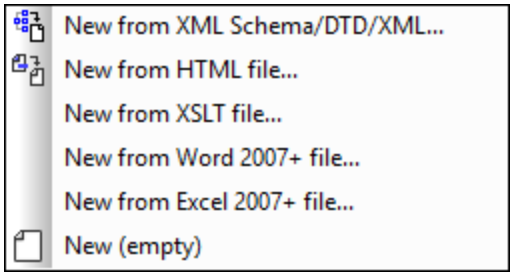
11.4 Menü "Datei"

Das Menü **Datei** enthält Befehle für die Arbeit mit SPSs und verwandten Dateien. Es stehen die folgenden Befehle zur Verfügung:

- Neu 468, zur Erstellung eines neuen SPS anhand von verschiedensten Quellen.
- Öffnen, Neu laden, Schließen, Alle schließen 475 zum Öffnen und Schließen und zum Neuladen der aktiven Datei.
- Design speichern, Design speichern unter, Alles speichern 480: Befehle zum Speichern des aktiven SPS und aller geöffneten SPS Dateien.
- Als MobileTogether Design-Datei exportieren 486: Generiert anhand der aktiven SPS-Datei ein MobileTogether Design.
- Generierte Dateien speichern 486: Dient zum Speichern von Ausgabedateien, die mit Hilfe des SPS generiert werden können.
- Web Design 489: Generiert alle zur Ausführung einer ASPX-Applikation erforderlichen Dateien im angegebenen Ordner.
- Eigenschaften 489: Dient zum Einstellen des Encodings (Kodierung) der Ausgabedokumente und des CSS-Kompatibilitätsmodus des Browsers, der Art der Auflösung von relativen Bildpfaden in der Authentic-Ansicht und ob Bilder in der RTF- (*Enterprise und Professional Edition*) und Word 2007+-Ausgabe (l*Enterprise Edition*) eingebettet oder durch Verknüpfung eingefügt werden sollen.
- Druckvorschau, Drucken 491: Ist in der Ausgabeansicht aktiv. Mit Hilfe dieser Befehle wird die Ansicht in den Vorschauen gedruckt.
- Zuletzt verwendete Dateien, Beenden 492: Dient zum Öffnen einer zuletzt verwendeten Datei bzw. zum Beenden des Programms.

11.4.1 Neu

Wenn Sie den Cursor über den Befehl **Neu** platzieren, erscheint ein Untermenü (*Abbildung unten*), über das Sie ein neues SPS Dokument einer der folgenden Arten erstellen können:



- Ein neues SPS basierend auf einem XML-Schema, einer DTD oder einem anhand einer XML-Datei generierten XML-Schema (**Neu von XML-Schema / DTD / XML**). Das ausgewählte Schema wird zur Seitenleiste "Design-Übersicht" 36 hinzugefügt, zur Struktur der Schemaquellen wird (in der Seitenleiste "Schema-Struktur" 38) eine grafische Darstellung der Struktur hinzugefügt und in der Design-Ansicht 29 wird ein SPS mit einer leeren Hauptvorlage erstellt. Ein neues SPS kann auch

© 2019-2025 Altova GmbH

anhand einer Datei (Schema oder XML) über eine URL oder globale Ressource erstellt werden (*siehe unten*).

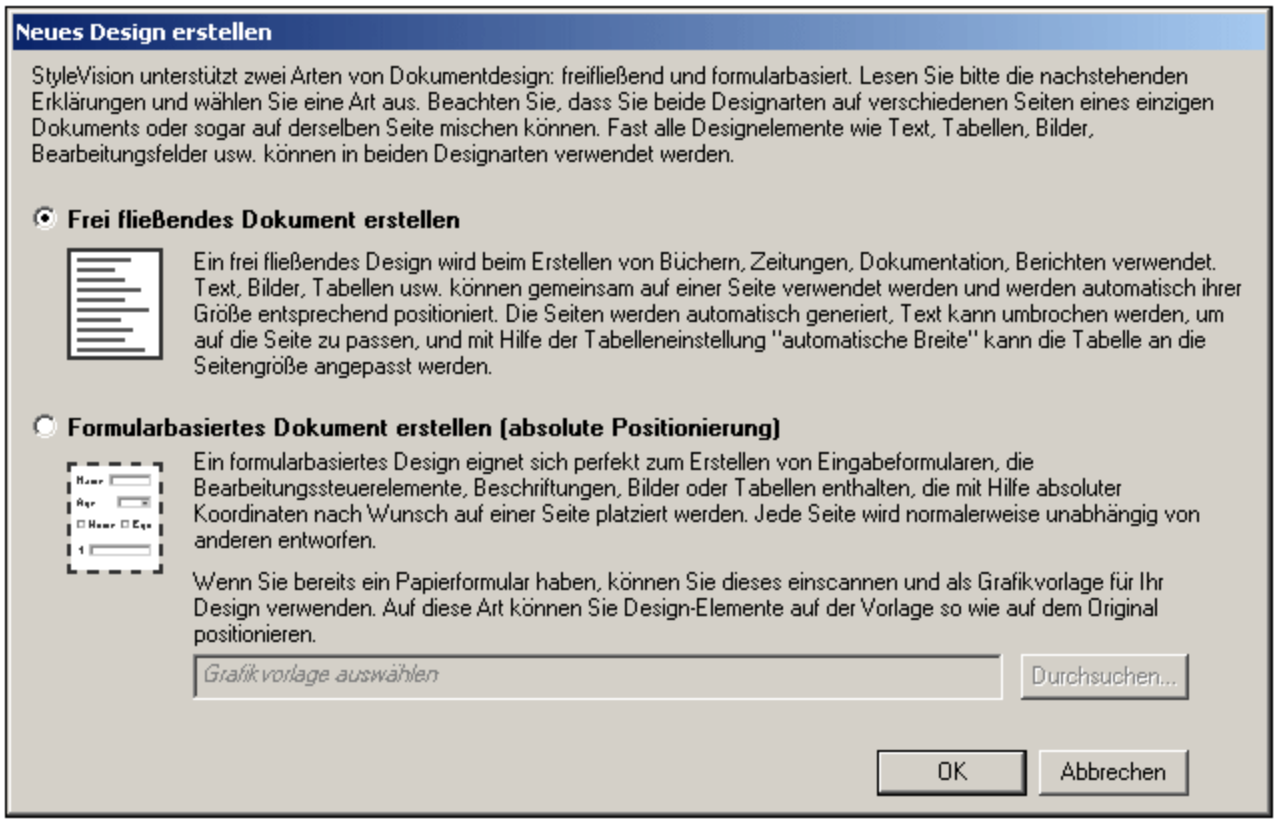
- Ein neues SPS basierend auf einem benutzerdefinierten Schema, das Sie Node für Node anhand einer HTML-Datei[408] erstellen (**Neu von HTML-Datei**). Das benutzerdefinierte Schema wird zu den Seitenleisten Design-Übersicht[36] und Schema-Struktur[38] hinzugefügt. In der Struktur der Schemaquellen hat es ein einziges Dokument-Element (Root-Element) und die HTML-Datei wird in die Design-Ansicht[29] geladen.
- Eine SPS-Datei kann anhand einer XSLT-für-HTML- oder einer XSLT-für-FO-Datei oder einer FO-Datei erstellt werden. Die Vorlagenstruktur und die Stile der XSLT-Datei werden im SPS erstellt. Sie können die SPS-Komponenten anschließend ändern und Inhalt sowie Formatierungen zur SPS-Datei hinzufügen. Nähere Informationen dazu finden Sie unter Neu von XSLT-Datei[379].
- Ein neues SPS, das den Inhalt eines MS Word-Dokuments als statischen Text enthält[115].
- Ein neues SPS, das den Inhalt eines MS Excel-Dokuments als statischen Text enthält[118].
- Ein neues leeres SPS (**Neu (leer)**). Es wird kein Schema zur Seitenleiste "Design-Übersicht" oder der Struktur der Schemaquellen hinzugefügt. In der Design-Ansicht[29] wird eine leere Hauptvorlage erstellt.

Anmerkung: Zum Definieren einer Datei oder Datenbankressource kann auch eine globale Ressource[362] verwendet werden.

Auswahl des Design-Typs

Nach Auswahl der (XSD- und XML-)-Dateiquellen, wird gegebenenfalls das Dialogfeld "Neues Design erstellen" angezeigt.

Im Dialogfeld "Neues Design erstellen" (*Abbildung unten*) werden Sie aufgefordert entweder: (i) ein frei fließendes Dokument oder (ii) ein formularbasiertes Dokument zu erstellen (in dem die Komponenten wie in einem Layout-Programm absolut positioniert werden).



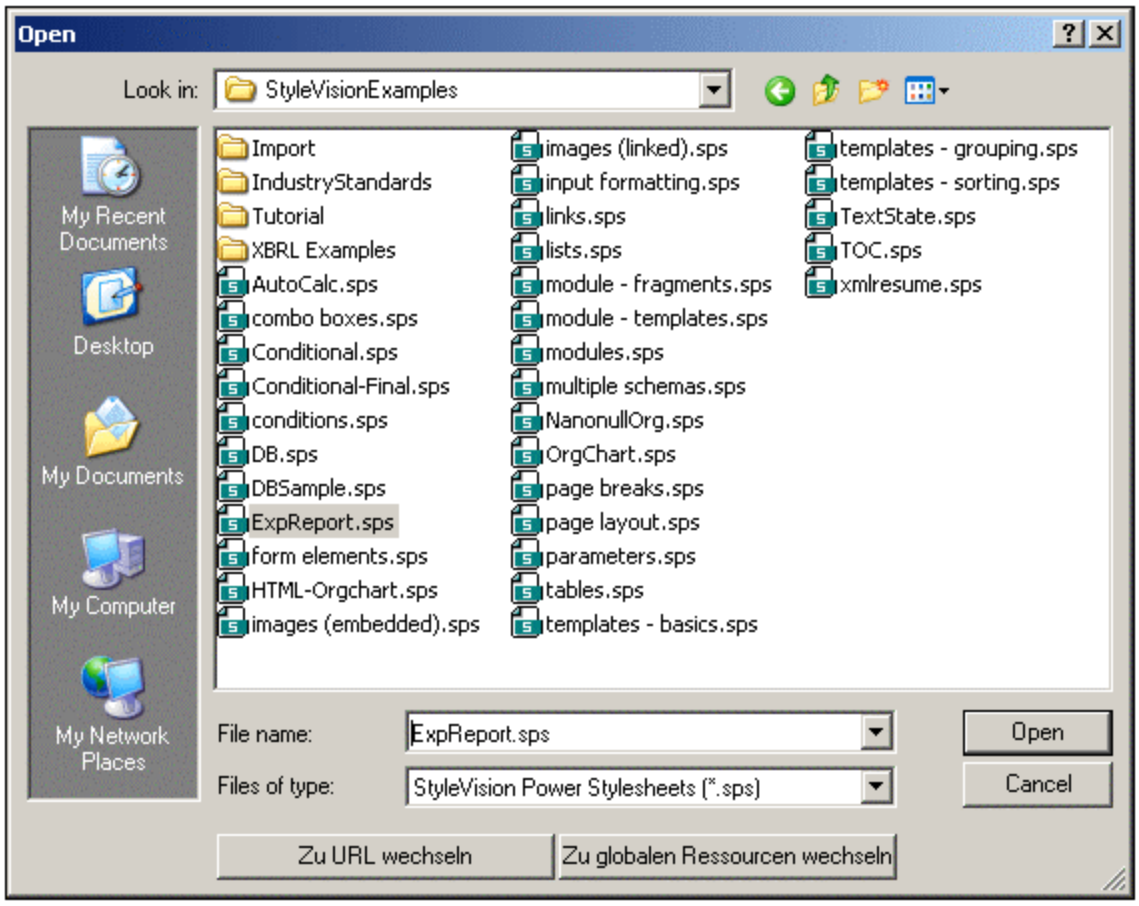
In einem frei fließenden Dokument ist das Dokument vom Layout her so angelegt, dass es den Maßen des Ausgabemediums entspricht (Papier oder Bildschirm). Elemente im Dokumentinhalt können nur relativ zueinander platziert werden, nicht aber an einer absoluten Position. Diese Art von Design eignet sich für Dokumente wie z.B. Berichte, Artikel und Bücher.

Ein einem formularbasierten Dokument wird ein einziger Layout Container[171] erstellt, in den Design-Komponenten absolut positioniert werden können. Die Dimensionen des Layout Containers sind benutzerdefiniert und Layout-Felder können innerhalb des Layout Containers absolut positioniert werden. Dokumentinhalt kann in einzelne Layout-Felder platziert werden. Wenn sich das Design Ihres SPS an einem spezifischen formularbasierten Design orientierten soll, können Sie ein Originalformular als Grafikvorlage[171] hinterlegen. Die Grafikvorlage kann dann als Hintergrundbild in den Layout Container inkludiert werden und dient als Hilfe für das Design Ihres Formulars; sie ist im Ausgabedokument nicht enthalten.

Auswählen 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 **Zu globalen Ressourcen wechseln**, um das entsprechende Dialogfeld aufzurufen.

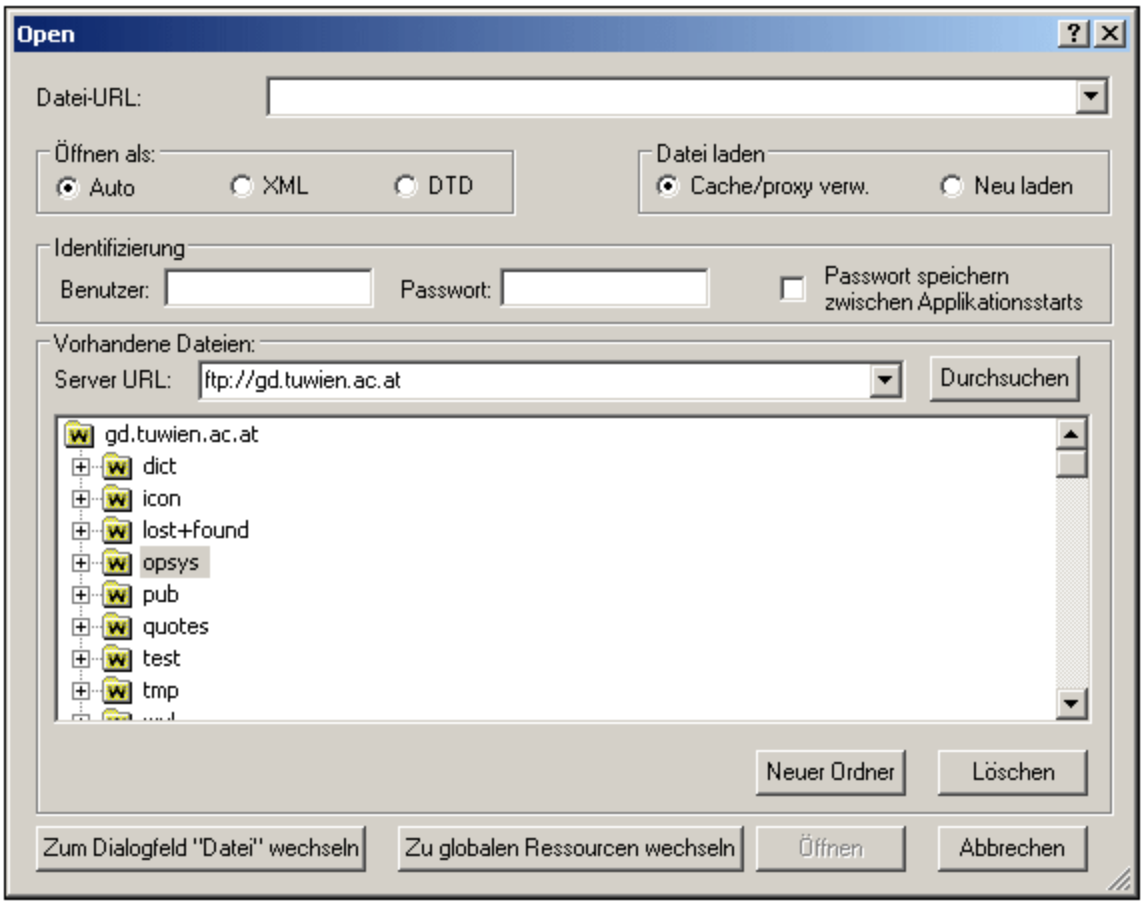
© 2019-2025 Altova GmbH



Auswählen von Dateien über URLs

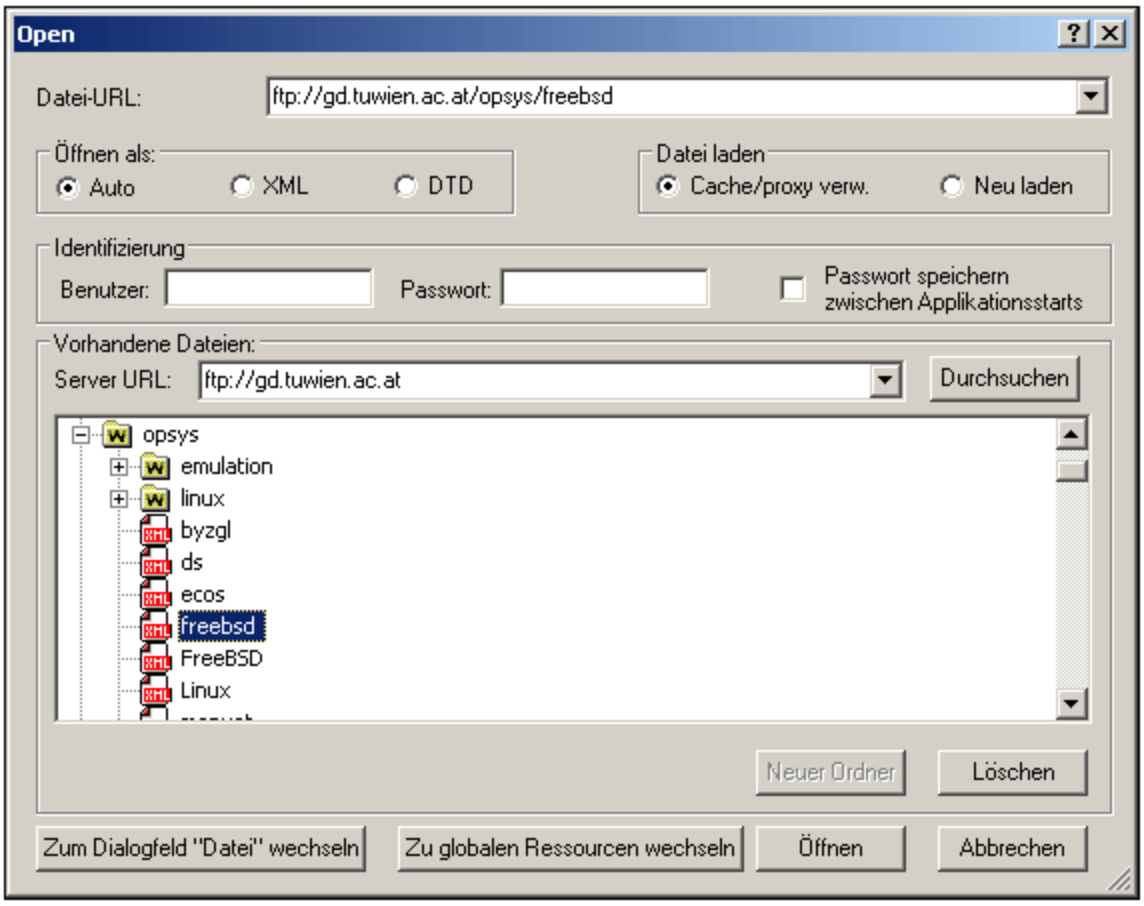
Um eine Datei über eine URL auszuwählen, gehen Sie folgendermaßen vor:

1. Klicken Sie auf den Befehl **Zu URL wechseln**. Daraufhin wechselt das Dialogfeld "Öffnen" in den URL-Modus (*Abbildung unten*).



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. Klicken Sie in die Ordnerstruktur, suchen Sie die gewünschte Datei und klicken Sie darauf.

© 2019-2025 Altova GmbH



Die Datei-URL wird im Feld "Datei-URL" angezeigt (*Abbildung oben*). Die Schaltfläche **Öffnen** wird erst zu diesem Zeitpunkt aktiv.

6. Klicken Sie auf **Öffnen** um die Datei zu laden. Die Datei, die Sie öffnen, erscheint im Hauptfenster.

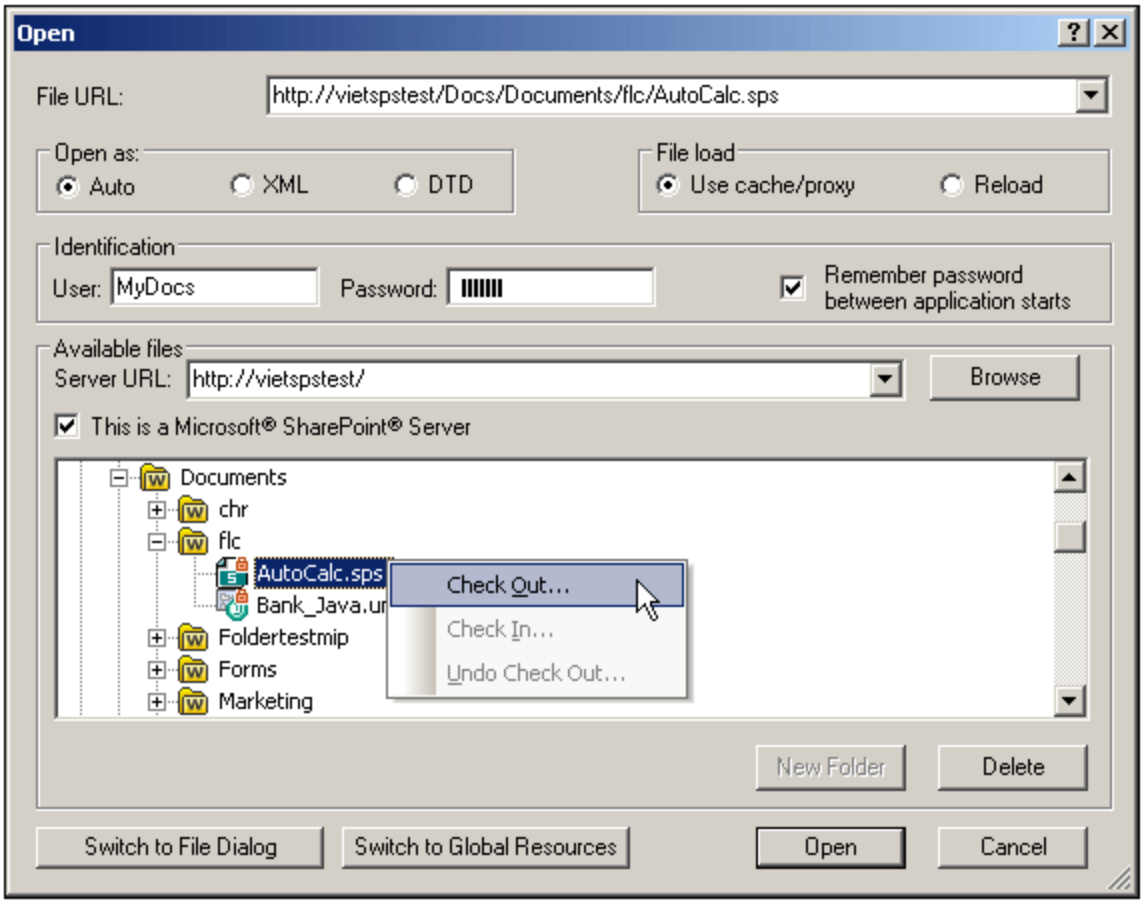
Anmerkung: Die Durchsuchen-Funktion steht nur auf Servern zur Verfügung, die WebDAV und Microsoft SharePoint Server unterstützen. Die unterstützten Protokolle sind FTP, HTTP und HTTPS.

Anmerkung: 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.
	Die Datei wurde von einem anderen Benutzer ausgecheckt und kann nicht ausgecheckt werden.
	Lokal ausgecheckt. Die Datei 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

© 2019-2025 Altova GmbH

"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.

11.4.2 Öffnen, Neu laden, Schließen, Alle schließen

Über den Befehl **Öffnen** (**Strg+O**) können Sie eine vorhandene SPS- oder PXF-Datei öffnen. Daraufhin wird das aus Windows bekannte Dialogfeld "Öffnen" 468 angezeigt, in dem Sie eine Datei mit der Erweiterung `.sps` öffnen können.

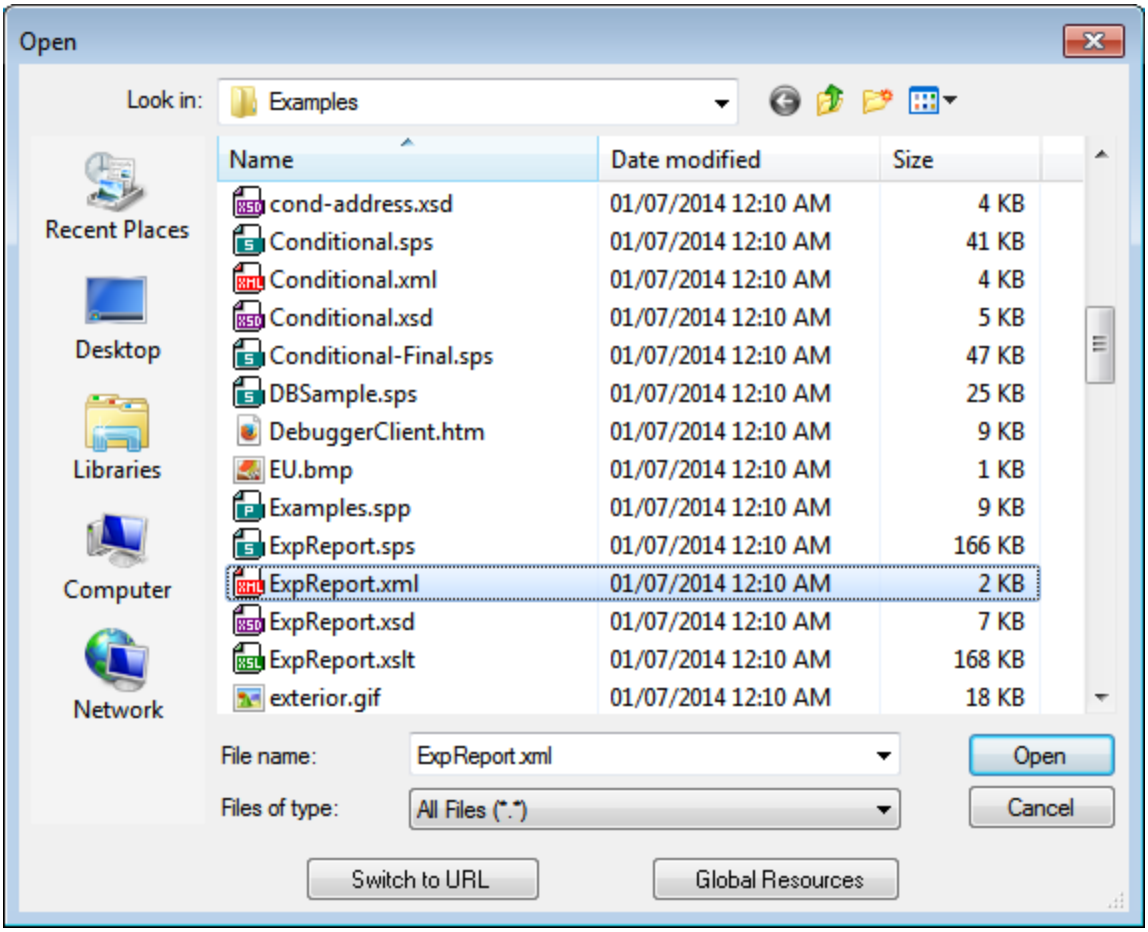
Mit dem Befehl **Neu laden** wird die SPS-Datei von der Festplatte neu geladen. Alle Änderungen, die seit der letzten Speicherung der Datei vorgenommen wurden, gehen verloren. Auch die XML-Arbeitsdatei wird neu geladen, sodass Sie die XML-Arbeitsdatei aktualisieren können, wenn sie extern geändert wurde.

Mit dem Befehl **Schließen** wird das gerade aktive SPS Dokument geschlossen. Beachten Sie: Während mehrere Dateien geöffnet sein können, ist nur eine davon aktiv. Das aktive Dokument kann auch durch Klicken auf die Schaltfläche **Schließen** rechts oben im Hauptfenster 28 geschlossen werden. Wenn das Dokument nicht gespeicherte Änderungen enthält, werden Sie aufgefordert, diese Änderungen zu speichern.

Mit dem Befehl **Alle schließen** werden alle geöffneten SPS-Dokumente geschlossen. Wenn ein offenes Dokument nicht gespeicherte Änderungen enthält, werden Sie aufgefordert, diese Änderungen zu speichern.

▼ 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.

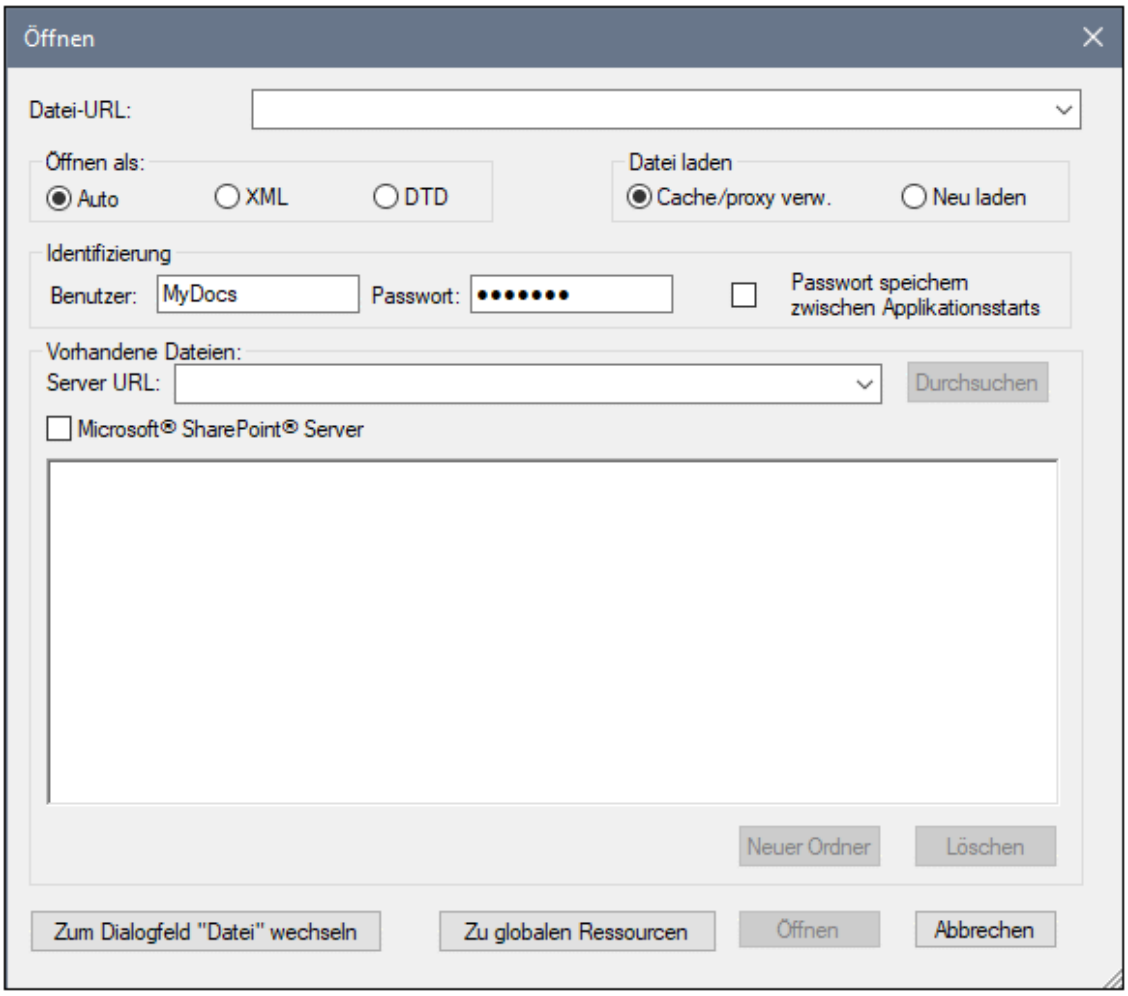


Auswählen von Dateien über URLs

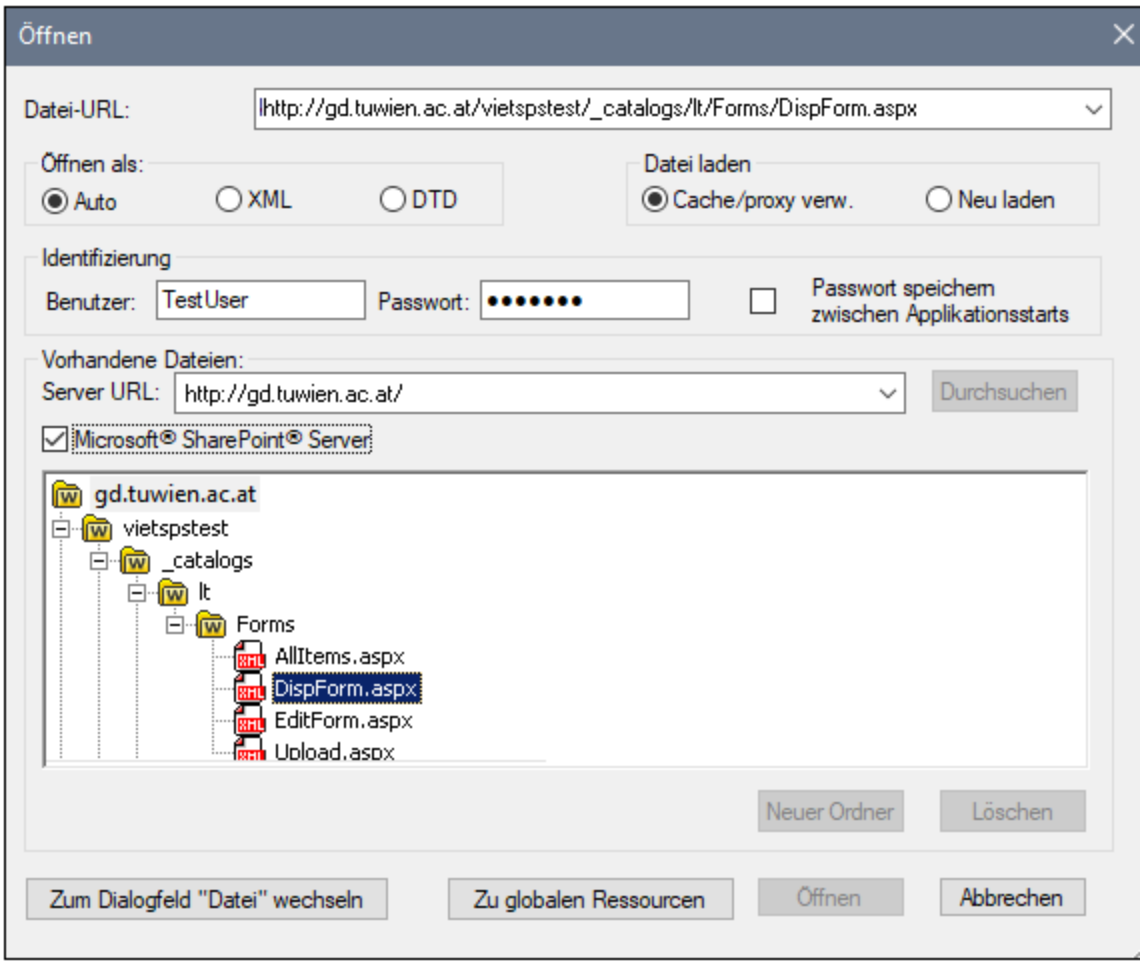
Um eine Datei über eine URL auszuwählen, gehen Sie folgendermaßen vor:

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"*).

© 2019-2025 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**.

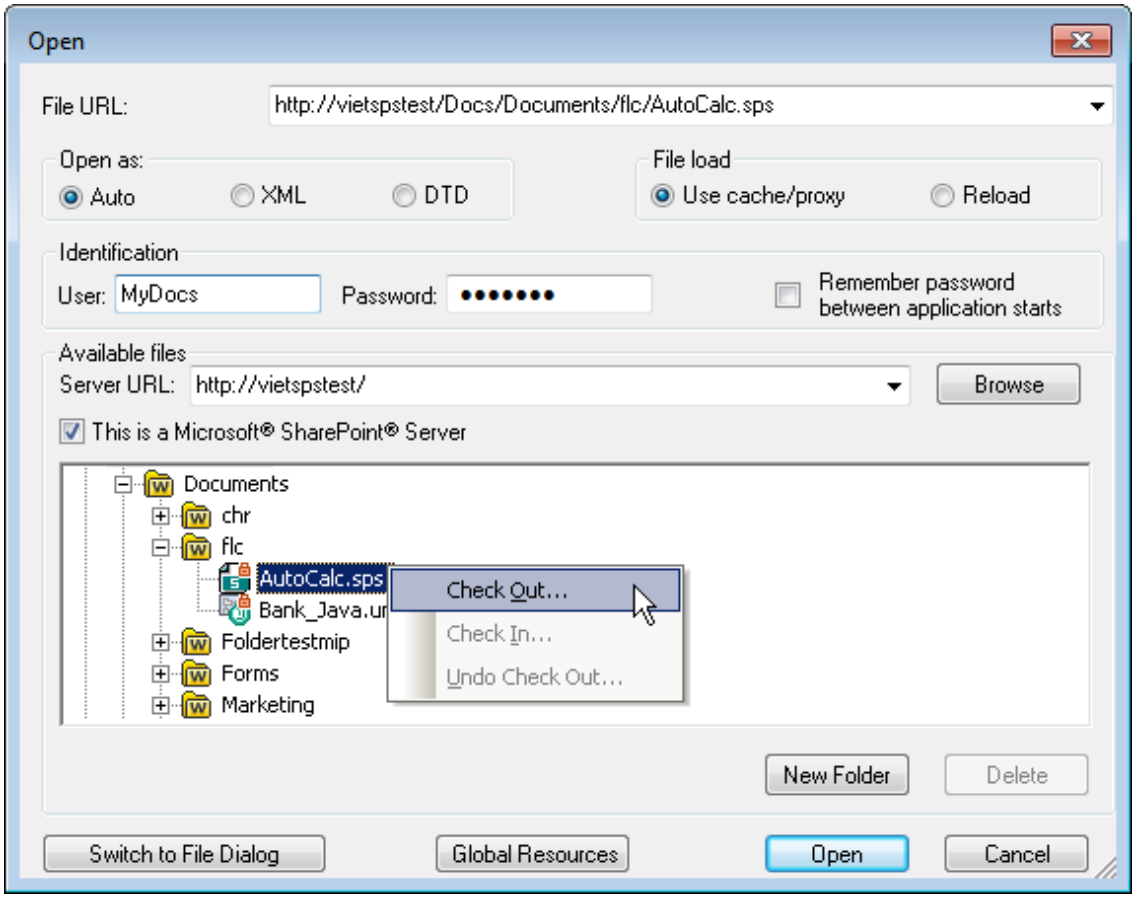
.

▼ Anmerkungen zu Microsoft® SharePoint® Server

Beachten Sie bei Dateien auf Microsoft® SharePoint® Servern die folgenden Punkte:

© 2019-2025 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 näher beschrieben. Eine allgemeine Beschreibung globaler Ressourcen finden Sie in der Dokumentation im Abschnitt.

11.4.3 Design speichern, Alles speichern

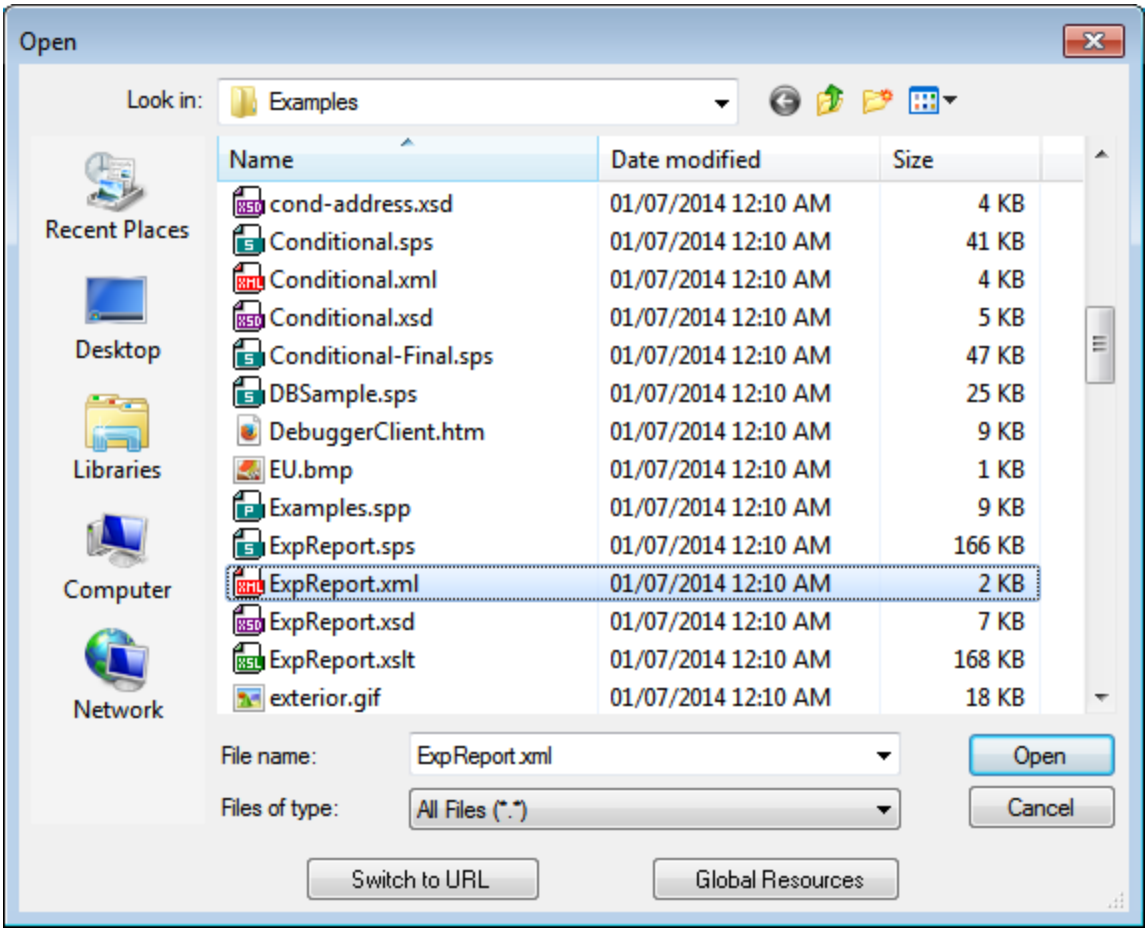
Mit dem Befehl **Design speichern (Strg+S)** wird das gerade offene Dokument als SPS Datei (mit der Dateierweiterung `.sps`) gespeichert.

Mit dem Befehl **Alles speichern (Strg+Umschalt+S)** werden alle offenen SPS Dokumente gespeichert.

▼ 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.

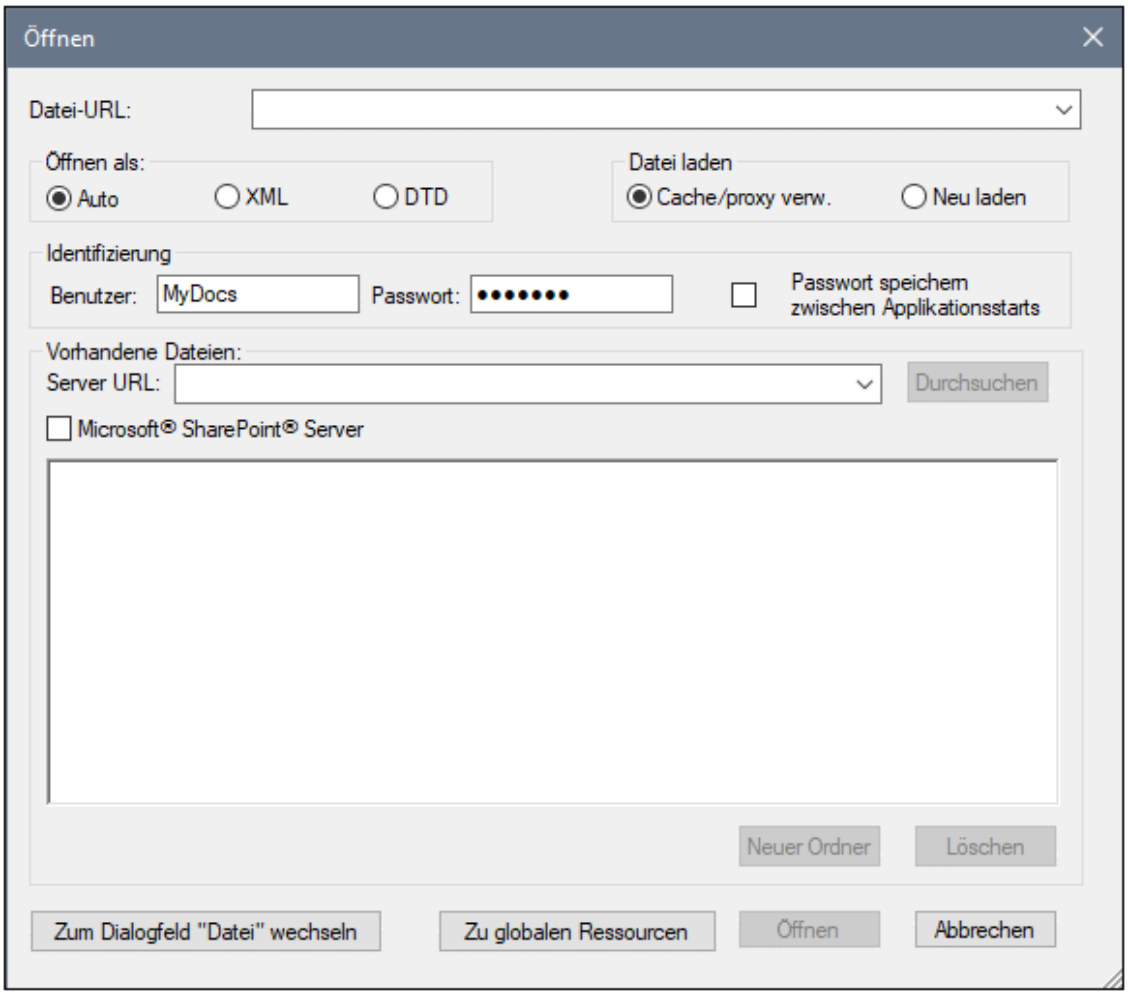
© 2019-2025 Altova GmbH



Auswählen von Dateien über URLs

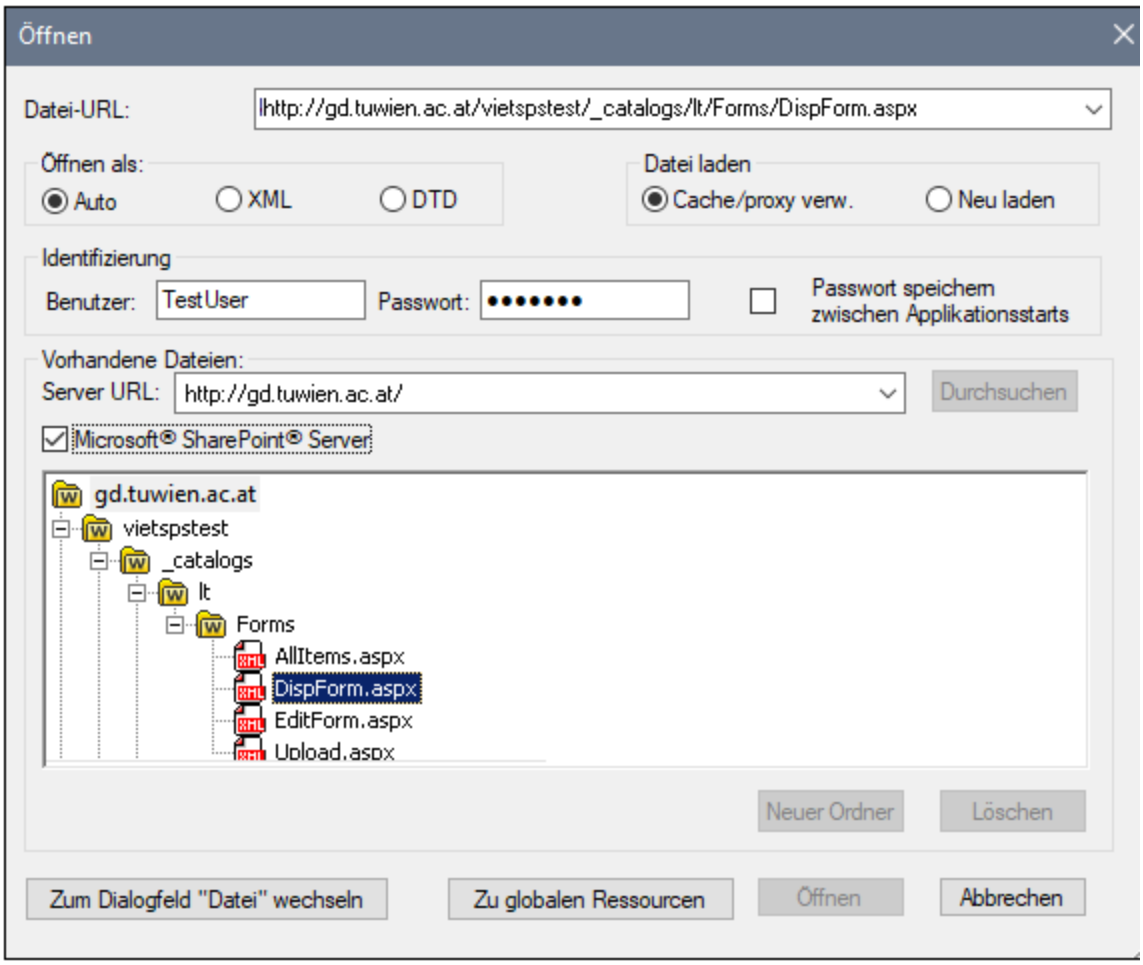
Um eine Datei über eine URL auszuwählen, gehen Sie folgendermaßen vor:

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.

© 2019-2025 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:

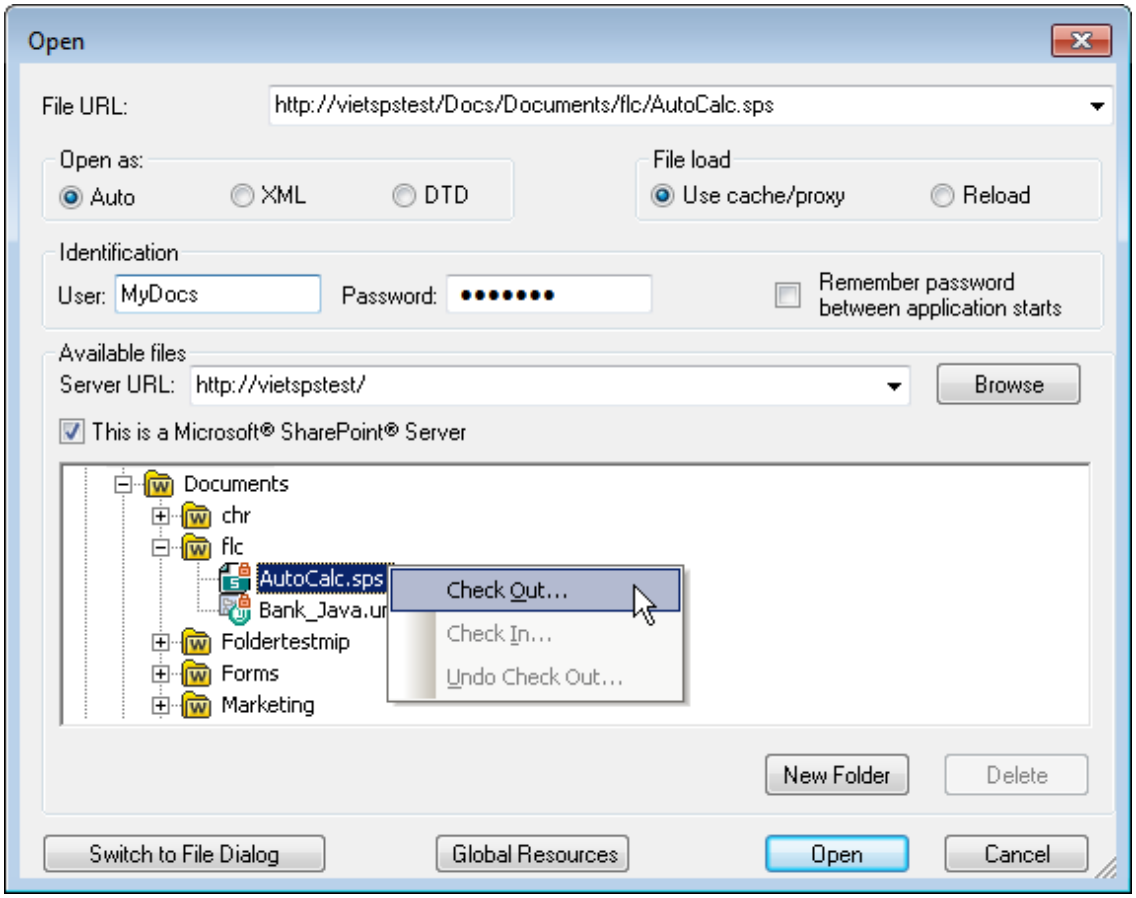
- Die Durchsuchen-Funktion steht nur auf Servern zur Verfügung, die WebDAV und Microsoft SharePoint Server unterstützen. Die unterstützten Protokolle sind FTP, HTTP und HTTPS.
- Damit Sie mehr Kontrolle über den Ladevorgang haben, können Sie auswählen, ob die Datei über den lokalen Cache-Speicher oder 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.

© 2019-2025 Altova GmbH

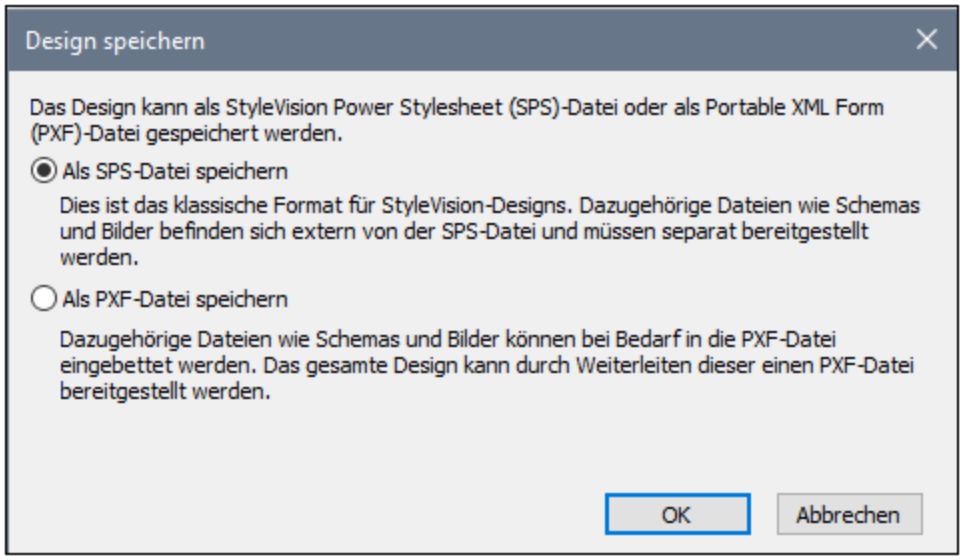
- Wenn eine Datei von einem anderen Benutzer ausgecheckt wurde, kann sie nicht auscheckt werden.
- Wenn eine Datei lokal von Ihnen ausgecheckt wurde, können Sie dies mit dem Kontextmenübefehl "Auschecken rückgängig" rückgängig machen. In diesem Fall wird die Datei dem Server unverändert zurückgegeben.
- 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 näher beschrieben. Eine allgemeine Beschreibung globaler Ressourcen finden Sie in der Dokumentation im Abschnitt.

11.4.4 Speichern unter

Mit dem Befehl **Speichern unter** kann das Design als (i) SPS-Datei oder als (ii) PXF (Portable XML Form)-Datei gespeichert werden. Bei Aufruf des Befehls wird das Dialogfeld "Speichern unten" (*Abbildung unten*) geöffnet. Wählen Sie das gewünschte Format aus und klicken Sie auf **OK**.



Das **SPS-Format** ist das Altova-Standardformat für StyleVision Designs. Das **PXF-Format** ist ein Altova-Format, mit dem alle mit dem Design in Zusammenhang stehenden Dateien (Schema-Dateien, XML-Dateien, Bilddateien, generierte XSLT-Stylehsheets, usw.) zusammen mit dem Design eingebettet werden können. Dieses Format eignet sich gut zum Transferieren aller Dateien, die zum Öffnen des Designs in der Authentic-Ansicht und/oder zum Generieren der HTML-Ausgabe anhand des Designs benötigt werden.

Als SPS-Datei speichern

Bei Aufruf der SPS-Option wird das aus Windows bekannte Dialogfeld "Speichern unter" angezeigt. Das Speichern funktioniert genau so, wie für den Befehl **Design speichern** 480 beschrieben. Der Vorteil des Befehls **Speichern unter** ist, dass Dateien, die bereits unter einem Dateinamen gespeichert wurden, unter einem anderen Dateinamen gespeichert werden können.

11.4.5 Als MobileTogether Design-Datei exportieren

Mit diesem Befehl wird anhand des aktiven SPS-Designs eine Altova MobileTogether Design-Datei erstellt. Eine MobileTogether Design-Datei dient zum Ausführen von Lösungen in der MobileTogether App für Mobilgeräte. So kann eine MobileTogether-Lösung etwa auf einem Mobilgerät wie z.B. einem Smartphone geöffnet werden, um den Inhalt einer Datenbank anzuzeigen und zu bearbeiten. MobileTogether-Lösungen werden in Altova MobileTogether Designer erstellt. Mit diesem Befehl können Sie ein SPS-Design in ein MobileTogether Design, das in MobileTogether Designer bearbeitet werden kann, konvertieren. Nähere Informationen dazu finden Sie auf der MobileTogether Webseite und in der MobileTogether Designer-Dokumentation. Die Konvertierungsoptionen stehen auf dem MobileTogether Design-Register des Dialogfelds "Optionen" (**Extras | Optionen** 552) zur Verfügung.

Anmerkung: Nicht für alle SPS-Designfunktionen gibt es Entsprechungen in MobileTogether Designs. Sie sollten daher die generierte Datei nach Ausführung dieses Befehls in MobileTogether Designer öffnen, um sie zu überprüfen und gegebenenfalls zu korrigieren. Von den folgenden Design-Funktionalitäten ist bekannt, dass sie nicht in MobileTogether Designs exportiert werden können: (i) Quellen, die über die erste im StyleVision Design aufgelistete, hinausgehen(eine Funktion der Enterprise Edition); (ii) globale Vorlagen 231; (ii) Module 216.

11.4.6 Generierte Dateien speichern

Wenn Sie den Befehl **Generierte Dateien speichern** aufrufen, erscheint ein Untermenü, das Optionen zum Speichern der folgenden Dateien enthält (*Abbildung unten*). Informationen darüber, im Zuge welcher Abläufe die Dateien generiert werden, finden Sie unter Allgemeine Arbeitsabläufe | Generierte Dateien 103.

Generierte XSLT-HTML Datei speichern

Mit dem Befehl "Generierte XSLT-HTML-Datei speichern" wird eine XSLT-Datei für die HTML-Ausgabe anhand Ihres SPS generiert. Mit Hilfe dieser XSLT-Datei können Sie anschließend ein XML-Dokument ins HTML-Format transformieren.

© 2019-2025 Altova GmbH

Generierte HTML-Datei(en) speichern

Mit dem Befehl "Generierte HTML-Datei(en) speichern" wird/werden eine bzw. mehrere HTML-Dateien generiert. Mehrere HTML-Dateien werden generiert, wenn im Design die Ausgabe mehrerer Dokumente[248] definiert wurde. Für diese Operation sind zwei Eingabedateien erforderlich:

- die XML-Arbeitsdatei, die der aktiven SPS-Datei zugewiesen ist. Wenn keine XML-Arbeitsdatei zugewiesen wurde, ist der Befehl **Generierte HTML-Datei speichern** deaktiviert.
- eine XSLT-Datei, die automatisch anhand der aktuell aktiven SPS-Datei generiert wird.

Generiertes benutzerdefiniertes Schema speichern

Dieser Befehl ist aktiv, wenn im SPS ein benutzerdefiniertes Schema verwendet wird. Das in der Seitenleiste "Schema-Struktur" erstellte Schema wird als XML-Schema mit der Erweiterung `.xsd` gespeichert.

Generierte benutzerdefinierte XML-Daten speichern

Die Daten in der importierten HTML-Datei, die dem benutzerdefinierten Schema entspricht, werden als XML-Datei gespeichert. Die entsprechenden Daten sind die Nodes im HTML-Dokument (in der Design-Ansicht), die als XML-Schema-Nodes erstellt wurden.

11.4.7 In FlowForce bereitstellen

Mit dem Befehl **In FlowForce bereitstellen** stellen Sie eine `.transformation` Datei auf Ihrem Altova FlowForce Server bereit. Die `.transformation` Datei enthält alle für die im SPS definierte Transformation erforderlichen Dateien und Informationen. Nachdem die `.transformation` Datei auf dem FlowForce Server bereitgestellt wurde, können Sie in Altova FlowForce Aufträge erstellen, die mit Hilfe der `.transformation` Datei gemäß den in der Auftragsdefinition festgelegten Triggern Transformationen generiert. Informationen zum Erstellen von FlowForce Aufträgen finden Sie in der Dokumentation zu FlowForce.

Eine `.transformation` Datei wird anhand einer Portable XML Format (PXF) Datei generiert. Der Befehl **In FlowForce bereitstellen** kann daher verwendet werden, wenn eine PXF-Datei aktiv ist. (Wenn eine SPS-Datei aktiv ist, ist der Befehl **In FlowForce bereitstellen** zwar aktiv, doch werden Sie bei Aufruf des Befehls aufgefordert, die SPS-Datei als PXF-Datei zu speichern. Um eine PXF-Datei anhand einer SPS-Datei zu erstellen, verwenden Sie den Befehl **Datei | Speichern unter** und wählen Sie PXF als das gewünschte Format aus.)

Beachten Sie die folgenden Punkte:

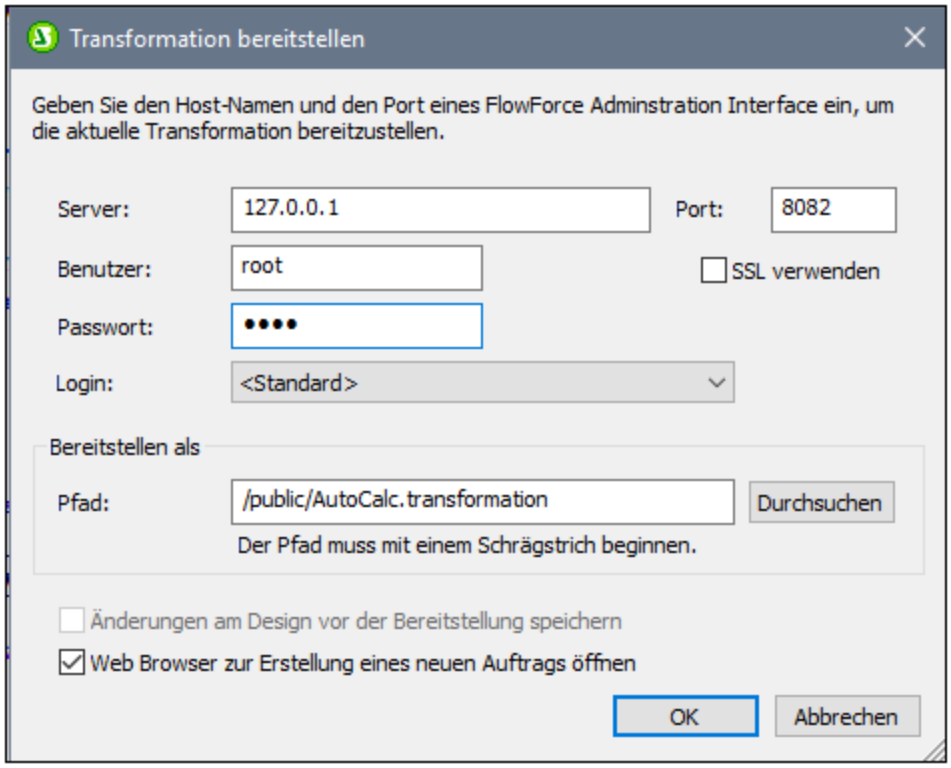
- Beim Speichern einer PXF-Datei steht eine Option zur Verfügung, mit der Sie externe Dateien (wie z.B. Bilddateien) darin inkludieren können. Wenn eine externe Datei nicht in der PXF-Datei enthalten ist, aber für die Transformation benötigt wird, so muss die externe Datei auf dem FlowForce Server gespeichert werden. Da die externen Dateien vom Arbeitsverzeichnis aus aufgerufen werden, sollten Sie relativ zu dem (in der FlowForce Auftragsdefinition definierten) Arbeitsverzeichnis gespeichert werden, so dass sie von Links aus dem Arbeitsverzeichnis korrekt aufgerufen werden können.
- Wenn ein FlowForce-Auftrag, für den eine StyleVision-Transformation benötigt wird, ausgeführt wird, wird der Auftrag an StyleVision Server übergeben, wo der Inhalt der PXF-Datei in das in den Auftragsparametern definierte Arbeitsverzeichnis extrahiert wird. Damit es bei der Extraktion zu keinen

Dateinamenkonflikten kommt, sollte das Arbeitsverzeichnis keine Datei enthalten, die denselben Namen wie eine Datei in der PXF-Datei enthält.

Stellen Sie vor Aufruf des Befehls **In FlowForce bereitstellen** sicher, dass Altova FlowForce Server und Altova StyleVision Server ordnungsgemäß lizenziert sind und ausgeführt werden. Nähere Informationen zum Einrichten von FlowForce Server finden Sie in der Dokumentation zu Altova FlowForce. (StyleVision Server ist zusammen mit FlowForce Server im selben Paket enthalten.)

Der Befehl "Bereitstellen"

Mit dem Befehl **In FlowForce bereitstellen** wird das Dialogfeld "Transformation bereitstellen" (*Abbildung unten*) aufgerufen.



In diesem Dialogfeld werden die folgenden Einstellungen vorgenommen:

- Die Adresse und Port-Nummer des FlowForce Web Servers (nicht des FlowForce Servers) sowie die Anmeldeinformationen (Benutzername und Passwort) für den FlowForce Server.
- Der Dateiname der Transformationsdatei und der Pfad auf dem FlowForce Server, wo diese gespeichert werden soll. Der Dateipfad muss mit einem Schrägstrich beginnen. Der Schrägstrich steht für das Root-Verzeichnis des FlowForce Servers.
- Fall seit der letzten Speicherung der Datei Änderungen am Design vorgenommen wurden, ist das Kontrollkästchen *Änderungen am Design vor der Bereitstellung speichern* aktiv. Aktivieren Sie das Kontrollkästchen, falls diese Änderungen gespeichert werden sollen. Deaktivieren Sie es, falls Sie dies nicht möchten.

© 2019-2025 Altova GmbH

- Um die Transformation über eine SSL-verschlüsselte Verbindung bereitzustellen, aktivieren Sie das Kontrollkästchen **SSL verwenden**. Dabei wird vorausgesetzt, dass FlowForce bereits so konfiguriert ist, dass SSL-Verbindungen zulässig sind. Nähere Informationen dazu finden Sie in der Dokumentation zu FlowForce Server.

Wenn Sie auf **OK** klicken, wird die `.transformation` Datei unter dem angegebenen Pfad auf dem FlowForce Server bereitgestellt. Wenn Sie das Kontrollkästchen *Web Browser zur Erstellung eines neuen Auftrags öffnen* (*siehe Abbildung oben*) aktivieren, wird die FlowForce Webserver-Schnittstelle in einem Webbrowser geöffnet und der im vorherigen Schritt erstellte Auftrag kann direkt im Browser bearbeitet werden.

Mehrere Versionen von StyleVision Server
Wenn auf dem Server, auf dem Sie die `.transformation`-Datei bereitstellen, mehrere Versionen von StyleVision Server unter Verwaltung von FlowForce Server ausgeführt werden (gibt nur für Windows Server), wird ein *StyleVision Server auswählen*-Dialogfeld angezeigt, in dem Sie aufgefordert werden, anzugeben, mit welcher Version von StyleVision Server dieses Mapping ausgeführt werden soll. Sie können Sie gewünschte Version manuell auswählen oder den Server die am besten geeignete Version automatisch auswählen lassen.

Dieses Dialogfeld wird angezeigt, wenn das FlowForce Server-Installationsverzeichnis für jede unter FlowForce Server-Verwaltung ausgeführte StyleVision Server-Version eine **`.tool`**-Datei enthält. Standardmäßig wird automatisch eine StyleVision Server **`.tool`**-Datei zu diesem Verzeichnis hinzugefügt, wenn Sie StyleVision Server im Rahmen der FlowForce Server-Installation installieren. Der Pfad,unter dem die **`.tool`**-Dateien in FlowForce gespeichert sind, lautet: **`C:\Programme\Altova\FlowForceServer2025\tools`**. Wenn Sie zusätzliche Versionen von StyleVision Server haben, die unter FlowForce Server-Verwaltung ausgeführt werden sollen, müssen deren **`.tool`**-Dateien eventuell manuell in das obige Verzeichnis kopiert werden. Die **`.tool`**-Datei von StyleVision Server befindet sich unter: **`C:\Programme\Altova\StyleVisionServer2025\etc`**.

Anmerkung: Informationen zum Arbeiten mit FlowForce Server finden Sie in der FlowForce Dokumentation.

11.4.8 Web Design

Der Befehl **Web Design** klappt ein Untermenü aus, das den Befehl **ASPX-Web-Applikation generieren** enthält. Der letztgenannte Befehl generiert im angegebenen Verzeichnisort alle Dateien, die benötigt werden um eine ASPX-Applikation zu starten. Ein Web-Browser liest die ASPX-Datei, die als Ausgabedatei erzeugt wird. C#-Code in diesem Dokument startet einen Prozess, durch den Daten in der Quelldatenbank oder im XML-Quelldokument mit Hilfe einer XSLT-Datei im ASPX-Paket dynamisch transformiert werden. Die ASPX-Datei (das Ausgabedokument des Transformationsprozesses) wird mit den aktuellsten Daten in der Quelldatenbank oder im XML-Quelldokument aktualisiert.

Für weitere Informationen siehe ASPX-Schnittstelle für Web-Applikationen 415.

11.4.9 Eigenschaften

Mit dem Befehl **Eigenschaften** wird das Dialogfeld "Eigenschaften" aufgerufen (*in der Abbildung unten sehen Sie das Dialogfeld in der Enterprise Edition*), in dem Sie verschiedene Eigenschaften für das aktive SPS definieren können.

Ausgabe

Auf dem Register "Ausgabe" können die folgenden Eigenschaften definiert werden:

- ***Ausgabekodierung:*** Im Bereich "Ausgabekodierung" können Sie die Kodierung Ihrer Ausgabedokumente auswählen. Wenn Sie die Kodierung in diesem Dialogfeld ändern, wird die Kodierung für das derzeit aktive SPS geändert. Sie können auch die applikationsweite für alle SPS-Dokumente definieren; dies erfolgt im Dialogfeld "Optionen" über das Register "Kodierung".

- ***HTML-Ausgabemodus:*** Sie können auswählen, ob das gesamte HTML-Dokument oder nur Child-Elemente des HTML-Elements `body` ausgegeben werden sollen. Die Child-Elemente werden parallel zueinander, d.h. auf derselben Ebene, ausgegeben und enthalten alle Nachfahren rekursiv. Infolgedessen können die Ausgabedokumente Fragmente des HTML-Codes sein..

- ***HTML-Ausgabemodus (DOCTYPE):*** Sie können auswählen, ob für die HTML-Ausgabe ein HTML 5, HTML 4.01 Transitional-Dokument oder ein XHTML 1.0 Transitional-Dokument generiert werden soll. Die Einstellung kann jederzeit beim Erstellen oder Bearbeiten des <%SPS%> Dokuments geändert werden.

- ***Internet Explorer-Kompatibilität und CSS-Unterstützung:*** Die CSS-Unterstützung in Internet Explorer-Versionen vor IE 6 war unvollständig und wurde in einigen Fällen nicht korrekt interpretiert. Die CSS-Unterstützung wurde in IE 6 verbessert und korrigiert und in IE 7, IE9 und höher weiter optimiert.

  In einem SPS können Sie den gewünschten Kompatibilitätsmodus im Dialogfeld "Eigenschaften" (*Abbildung oben*) auswählen. Zur Auswahl stehen entweder IE 5, IE 7 oder IE9. (Beachten Sie, dass IE9 oder höher installiert sein muss, damit der IE9 Kompatibilitätsmodus angewendet werden kann). Die entsprechende IE-Unterstützung steht sofort in der HTML-Vorschau zur Verfügung. Beachten Sie, dass bei neuen SPS-Dokumenten IE7-Kompatibilität ausgewählt ist. PS-Dokumente, die in früheren Versionen von Altova StyleVision erstellt wurden, können im gewünschten Kompatibilitätsmodus erneut gespeichert werden - (durch Auswahl der entsprechenden Kompatibilität im Dialogfeld Eigenschaften 489).

XSD/XSLT

Auf diesem Register (*in der Abbildung unten sehen Sie das Dialogfeld in der Enterprise Edition*) können Sie festlegen, welche XSD-Version für die Validierung von XML-Dateien und welche XSLT-Version im SPS verwendet werden soll.

StyleVision hat einen XSD 1.0 Validator und einen XSD 1.1 Validator. Sie haben die Wahl zwischen den folgenden Optionen:

- XSD 1.1 Validator verwenden, wenn das Attribut `/xs:schema/@vc:minVersion` des XSD-Dokuments auf `1.1` gesetzt ist; andernfalls XSD 1.0 Validator verwenden.
- Immer den XSD 1.1 Validator verwenden
- Immer den XSD 1.0 Validator verwenden

Auf dem Register "XSLT" kann die XSLT-Version für das aktive Dokument ausgewählt werden. Wenn Sie die Option mit der "*xsl:import-schema-Anweisung*" aktivieren, wird das `xsl:import-schema` Element der XSLT 2.0 und 3.0-Spezifikation in das von StyleVision generierte XSLT-Dokument inkludiert. Es wird empfohlen, dass Sie diese Option aktivieren, damit Datentypen aus dem Schema gelesen werden können, falls es kein `xsi:schemaLocation` Attribut im XML-Dokument gibt.

© 2019-2025 Altova GmbH

Einbettung von Bildern in die HTML-Ausgabe
Diagramme, Barcodes und Inline-Bilder können in die HTML-Ausgabe eingebettet werden. Wenn dieses Kontrollkästchen aktiviert ist, werden die Bilddaten in einen Base64-String konvertiert, der anhand des Data URI-Schemas im Attribut `src` des HTML `img`-Elements gespeichert wird. Diese Art der Einbettung steht nur zur Verfügung, wenn die XSLT-Version des SPS 2.0 oder 3.0 ist.

Pfade

In den Einstellungen auf diesem Register werden Standardpfade für verschiedene, von der SPS-Datei generierte Dateien und Pfade, die in der SPS-Datei gespeichert werden, angegeben (*in der Abbildung unten sehen Sie das Dialogfeld in der Enterprise Edition*).

Folgende Standardwerte werden auf dem Register "Pfade" gesetzt:

- Ob Vorschaudateien im Verzeichnis der SPS-Datei oder der XML-Arbeitsdatei der Hauptschemaquelle erstellt werden sollen.
- Wo zusätzlich generierte Dateien (Bilddateien, Barcode-Bilddateien, Diagrammbilddateien usw.) erstellt werden.
- Ob Dateipfade im SPS nur relativ sind, wenn sich das Zielverzeichnis in der Verzeichnisstruktur der SPS-Datei befindet, oder selbst dann relativ ist, wenn sich das Zielverzeichnis außerhalb der Verzeichnisstruktur der SPS-Datei befindet.

11.4.10 Druckvorschau, Drucken

Der Befehl **Druckvorschau** ist in der Authentic-Ansicht aktiviert. (*Die Authentic-Ansicht steht nur in der Enterprise und der Professional Edition zur Verfügung*).Er öffnet, wenn die Design-Ansicht aktiv ist, ein Fenster, das eine Vorschau des SPS- Designs (wenn die Design-Ansicht aktiv ist) oder der Authentic-Ansicht der XML-Arbeitsdatei (wenn die Authentic-Ansicht aktiv ist) enthält. In der Vorschau sehen Sie das Design mit oder ohne Tags, je nachdem, was am Bildschirm angezeigt wird.

Im Fenster "Druckvorschau" haben Sie über die Befehle der Symbolleiste am oberen Rand der Seite (*Abbildung oben*) und die Seitennavigationssymbole am unteren Rand der Seite die folgenden Möglichkeiten. Die Symbolleiste "Druckvorschau" enthält von links nach rechts die folgenden Befehle:

- Sie können die Seite über die Schaltfläche "Drucken" ausdrucken.
- Sie können Sie die Seitenausrichtung wählen: Hoch- oder Querformat.
- Sie können die Seiteneigenschaften durch Klicken auf die Schaltfläche **Seite einrichten** und Aufrufen des gleichnamigen Dialogfelds definieren.
- Sie können die Kopf- und Fußzeilen ein- und ausblenden und diese in den Ausdruck inkludieren oder davon ausnehmen,.
- Sie können die Seite in der Druckvorschau entweder in der vollen Seitenbreite oder -höhe anzeigen lassen, sodass die Seite entweder die gesamte Bildschirmbreite oder die gesamte Bildschirmhöhe einnimmt.

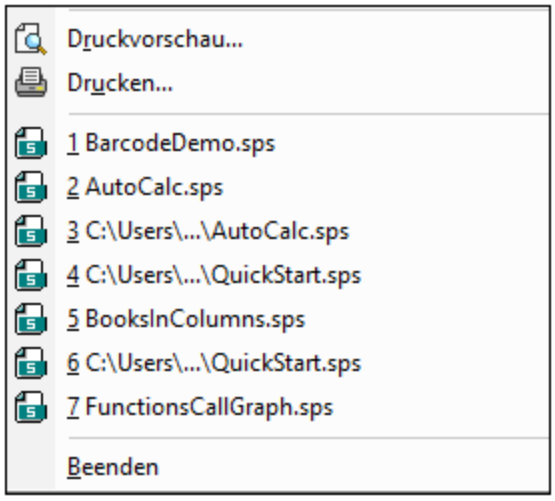
- Sie können einstellen, wie viele Seiten am Bildschirm angezeigt werden sollen.
- Sie können den Vergrößerungsfaktor der Seitenvorschau über die Schaltflächen "Vergrößern" und "Verkleinern" oder die Auswahlliste zur Auswahl des Zoom-Faktors ändern.

Der Befehl **Drucken** ist in der Authentic-Ansicht und auf den Ausgabevorschauregistern aktiv. Sie können damit je nachdem, wie die Seite für diese Ansicht konfiguriert ist, die ausgewählte Ansicht der XML-Arbeitsdatei drucken. Beachten Sie, dass Sie die Seiteneinrichtung für die Authentic-Ansicht im Dialogfeld "Seite einrichten", das über das Fenster "Druckvorschau" aufgerufen wird, bearbeiten können.

Anmerkung: Um Hintergrundfarben und Bilder in der Druckvorschau zu aktivieren, gehen Sie folgendermaßen vor: (i) Klicken Sie im Internet Explorer im Menü **Extras** auf **Internetoptionen** und 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**.

11.4.11 Zuletzt verwendete Dateien, Beenden

In der Liste der zuletzt verwendeten Dateien werden der Name und Pfad der zuletzt verwendeten Dateien angezeigt. Wenn Sie auf einen dieser Einträge klicken, wird die Datei auf einem neuen Register im Hauptfenster geöffnet.



Um diese Dateien über die **Tastatur** aufzurufen, drücken Sie **Alt+D**, um das Menü "Datei" zu öffnen und wählen Sie anschließend die Zahl für die gewünschte Datei aus; wenn Sie z.B. **1** drücken, wird die erste Datei in der Liste geöffnet, bei **2** die zweite, usw.

Mit dem Befehl **Beenden** wird StyleVision beendet. Enthält eine geöffnete Datei nicht gespeicherte Änderungen, erscheint eine entsprechende Meldung.

© 2019-2025 Altova GmbH

11.5 Menü "Bearbeiten"

Das Menü **Bearbeiten** enthält Befehle, die Ihnen bei der Bearbeitung von SPS helfen. Neben den Standardbearbeitungsbefehlen wie **Ausschneiden** (Umschalt+Entf oder Strg+X), **Kopieren** (Strg+C), **Einfügen** (Strg+V) und **Löschen** (Entf), die in diesem Abschnitt nicht beschrieben werden, stehen die folgenden Befehle zur Verfügung:

- Rückgängig, Wiederherstellen, Alles markieren 493: Dienen zum Rückgängigmachen bzw. Wiederherstellen vorhergehender Aktionen und zum Auswählen des gesamten Inhalts des SPS.
- Suchen, Weitersuchen, Ersetzen 493: Dienen zum Suchen von Text im SPS und in XSLT-Stylesheet-Vorschauen.
- Stylesheet Parameter 498: Dient zum Bearbeiten von global für das SPS deklarierten Parametern.
- Markup reduzieren/erweitern 499: Dient zum Reduzieren und Erweitern von Tags für SPS Design-Komponenten.

Die Befehle stehen auch über das Kontextmenü zur Verfügung, welches Sie durch Rechtsklick auf eine Komponente oder den Cursoreinfügepunkt aufrufen können. Zusätzlich dazu stehen einige Befehle auch als Tastaturkürzel und/oder Symbolleisten-Schaltflächen zur Verfügung. Beachten Sie jedoch, dass Befehle, die in einer bestimmten Dokumentenansicht oder an einer bestimmten Stelle nicht angewendet werden können, im Menü ausgegraut sind.

11.5.1 Rückgängig, Wiederherstellen, Alles markieren

Mit Hilfe des Befehls **Rückgängig (Strg+Z)** können Sie eine Änderung rückgängig machen. Die Anzahl der Schritte, die rückgängig gemacht werden können, ist unbegrenzt. Jede Aktion kann rückgängig gemacht werden. Sie können einen Befehl nach dem anderen rückgängig machen und zwar bis zur ersten Aktion nach dem Öffnen des Dokuments.

Mit dem Befehl **Wiederherstellen (Strg+Y)** können Sie beliebig viele zuvor rückgängig gemachte Befehle wiederherstellen. Mit Hilfe der Befehle "Rückgängig" und "Wiederherstellen" können Sie innerhalb des Befehlsverlaufs vor- und zurückgehen.

Mit dem Befehl **Alles markieren** wird der gesamte Inhalt des Design-Dokumentfensters markiert.

11.5.2 Suchen, Weitersuchen, Ersetzen

Über den Befehl **Suchen (Strg+F)** können Sie Textstrings in der Design-Ansicht, im JavaScript Editor sowie in XSLT-Stylesheets suchen.

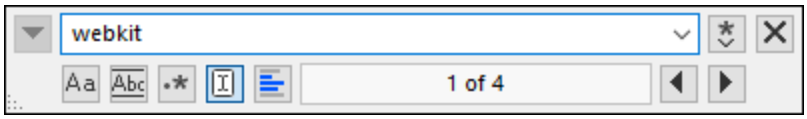
Welches "Suchen & Ersetzen"-Dialogfeld angezeigt wird, hängt davon ab, welche Ansicht gerade aktiv ist.

- Wenn die Design-Ansicht aktiv ist wird der Fokus bei Klick auf den Befehl **Suchen** oder **Ersetzen** auf die Seitenleiste "Suchen & Ersetzen" 53 gesetzt. Eine Beschreibung der Such- und Ersetzungsfunktionen der Design-Ansicht finden Sie im Kapitel Seitenleiste "Suchen & Ersetzen" 53.

- Die Such- und Ersetzungsfunktion im JavaScript Editorund XSLT Stylesheets ist in diesem Kapitel beschrieben.

XSLT Stylesheets und JavaScript Editor

Wenn Sie auf dem Register XSLT-für-HTML oder JavaScript Editor den Befehl **Suchen** auswählen, erscheint das folgende Dialogfeld:



Die folgenden Optionen stehen zur Auswahl:

- *GROSS/klein beachten*: Wenn die Schaltfläche aktiv ist, wird die Groß- und Kleinschreibung bei der Suche berücksichtigt (`Address` ist nicht gleich `address`).
- *Ganzes Wort*: Nur die exakte Wortentsprechung im Text wird gefunden. So wird z.B. bei Eingabe von `fit` bei Aktivierung der Option *ganzes Wort* nur das Wort `fit` gefunden; `fit` in `fitness` wird z.B. nicht gefunden.
- *Regular Expression*: Wenn die Option aktiv ist, wird der Suchbegriff als Regular Expression gelesen. Eine Beschreibung zur Verwendung von Regular Expressions finden Sie weiter unten unter *Regular Expressions* 495.
- *Anker suchen:* Bei Eingabe eines Suchbegriffs werden die Übereinstimmungen im Dokument markiert und eine dieser Übereinstimmungen wird als die aktuelle Auswahl markiert. Der Ein/Aus-Befehl *Anker suchen* legt fest, ob diese erste aktuelle Auswahl relativ zur Cursor-Position gemacht werden soll oder nicht. Wenn *Anker suchen* aktiv ist, so ist die erste gerade ausgewählte Übereinstimmung die nächste Übereinstimmung ab der aktuellen Cursorposition. Wenn *Anker suchen* deaktiviert ist, ist die erste aktuell ausgewählte Übereinstimmung, vom Anfang des Dokuments gesehen, die erste Übereinstimmung im Dokument.
- *In Auswahl suchen:* Wenn diese Option aktiv ist, wird der aktuell ausgewählte Text gesperrt und die Suche auf die Auswahl eingeschränkt. Andernfalls wird das gesamte Dokument durchsucht. Bevor Sie einen neuen Textbereich auswählen, heben Sie die aktuelle Auswahl durch Deaktivieren der Option *In Auswahl suchen* auf.
- *Ersetzen:* Klicken Sie auf dem Register JavaScript Editor links oben im Dialogfeld auf den Nach unten-Pfeil, um das *Ersetzen*-Feld zu öffnen. Hier können Sie den String eingeben, durch den der gefundene ersetzt werden soll.

HTML-Vorschau

Bei Klick auf den Befehl **Suchen** in der HTML-Vorschau wird ein einfaches Suchen und Ersetzen 53-Dialogfeld aufgerufen.

Beachten Sie dazu Folgendes:

- Um Ihre Eingabe mit ganzen Wörtern im XML-Dokument zu vergleichen, aktivieren Sie das Kontrollkästchen "Ganzes Wort". Bei Eingabe von soft wird z.B. nur das ganze Wort `soft` gesucht; das Fragment `soft` in `software` wird nicht gefunden.
- Um Ihre Eingabe mit Wortfragmenten zu vergleichen, lassen Sie das Kontrollkästchen "Ganzes Wort" deaktiviert. Auf diese Art können Sie sowohl nach `soft` als auch `software` suchen.
- Damit die Groß- und Kleinschreibung ignoriert wird, lassen Sie das Kontrollkästchen "GROSS/Klein beachten" deaktiviert. Dadurch finden Sie bei Eingabe von `soft` auch `Soft`.

© 2019-2025 Altova GmbH

Weitersuchen-Befehl

Mit dem Befehl **Weitersuchen (F3)** wird der Befehl Suchen wiederholt, um nach der nächsten Instanz des gesuchten Texts zu suchen. Eine Beschreibung zur Verwendung der Suchfunktion finden Sie unter Suchen 493.

Verwendung von Regular Expressions

Sie können zum Suchen eines Text-String Regular Expressions (regex) verwenden. Aktivieren Sie dazu zuerst die Option *Regular Expression* (*siehe oben*). Dadurch legen Sie fest, dass der Text im Suchbegrifffeld als Regular Expression ausgewertet werden soll. Geben Sie als Nächstes die Regular Expression in das Suchfeld ein. Hilfe zur Erstellung von Regular Expressions erhalten Sie durch Klick auf die Schaltfläche **Regular Expression Builder** rechts vom Suchfeld. Klicken Sie auf einen Eintrag im Builder, um die/das entsprechende(n) regex-Metazeichen in das Suchfeld einzugeben. Eine kurze Beschreibung zu Metazeichen finden Sie weiter unten.

Regular Expression-Metazeichen

In der nachstehenden Liste sehen Sie eine Auflistung von Regular Expression-Metazeichen.

`.`	Steht für jedes beliebige Zeichen. Dies ist ein Platzhalter für ein einzelnes Zeichen.
`(`	Markiert den Beginn eines markierten Ausdrucks.
`)`	Markiert das Ende eines markierten Ausdrucks.
`(abc)`	Die Metazeichen `(` und `)` markieren Beginn und Ende eines markierten Ausdrucks. Markierte Ausdrücke eignen sich dazu, eine gesuchte Region zu markieren ("sich diese zu merken"), um diese später referenzieren zu können (Rückreferenz). Es können bis zu neun Unterausdrücke markiert werden (und später im Feld "Suchen" oder "Ersetzen" rückreferenziert werden). So wird etwa mit `(the) \1` der String `the the` gefunden. Dieser Ausdruck erklärt sich folgendermaßen: Suche den String "the" (und merke ihn Dir als markierte Region), gefolgt von einem Leerzeichen, gefolgt von einer Rückreferenz auf die zuvor gesuchte markierte Region.
`\n`	`n` ist eine **Variable**, die Ganzzahlwerte von `1` bis `9` haben kann. Der Ausdruck bezieht sich auf die erste bis neunte markierte Region bei der Ersetzung. Lautet der Suchstring beispielsweise `Fred([1-9])XXX` und der Ersetzungsstring `Sam\1YYY`, so bedeutet dies, dass sich im Suchstring ein markierter Ausdruck befindet, der (implizit) mit der Zahl `1` indiziert ist; im Ersetzungsstring wird der markierte Ausdruck mit `\1` referenziert. Wenn der Such- und Ersetzungsbefehl auf `Fred2XXX` angewendet wird, würde `Sam2YYY` generiert.
`\<`	Steht für den Beginn eines Worts.
`\>`	Steht für das Ende eines Worts.
`\x`	Damit können Sie ein Zeichen `x` verwenden, das sonst eine spezielle Bedeutung hätte. So würde z.B. `\[` als `[` und nicht als Beginn einer Zeichenmenge interpretiert werden.
`[...]`	Kennzeichnet eine *Zeichenmenge*, `[abc]` z.B. steht für jedes der Zeichen `a, b` oder `c`. Sie können auch Bereiche angeben, z.B. `[a-z]` für alle Kleinbuchstaben.
`[^...]`	Die invertierte Zeichenmenge. `[^A-Za-z]` z.B. steht für jedes Zeichen mit Ausnahme alphabetischer Zeichen.

^	Steht für den Zeilenanfang (es sei denn dieses Zeichen wird innerhalb einer Menge verwendet, *siehe oben*).
$	Steht für das Zeilenende. Beispiel: `A+$` findet ein oder mehrere `A's` am Ende der Zeile.
*	Steht für 0 oder öfter. Mit `Sa*m` werden z.B. `Sm, Sam, Saam, Saaam` usw. gefunden.
+	Steht für 1 oder öfter. Mit `Sa+m` werden z.B. `Sam, Saam, Saaam` usw. gefunden.

Darstellung von Sonderzeichen
Beachten Sie die folgenden Ausdrücke.

\r	Wagenrücklauf (Carriage Return = CR). Zum Suchen oder Erstellen einer neuen Zeile können Sie entweder CR (**\r**) oder LF (**\n**) verwenden.
\n	Neue Zeile (LF). Zum Suchen oder Erstellen einer neuen Zeile können Sie entweder CR (**\r**) oder LF (**\n**) verwenden.
\t	Tabulatorzeichen
****	Verwenden Sie diese Zeichen, um Zeichen, die in Regular Expressions vorkommen, mit einem Escape zu versehen, z.B.: **\\\n**

Beispiele für Regular Expressions

In diesem Beispiel wird gezeigt, wie Sie Text mit Hilfe von Regular Expressions suchen und ersetzen können. In vielen Fällen ist das Suchen und Ersetzen von Text unkompliziert und es werden dafür keine Regular Expressions benötigt. Es gibt jedoch Fälle, in denen Sie Text auf eine Art und Weise bearbeiten müssen, wie es mit der normalen Such- und Ersetzungsoperation nicht möglich ist, z.B. wenn Sie eine XML-Datei mit mehreren tausend Zeilen haben, in der Sie bestimmte Elemente in einer einzigen Operation umbenennen müssen, ohne dabei den darin enthaltenen Inhalt zu ändern. Ein weiteres Beispiel: Sie müssen die Reihenfolge mehrerer Attribute eines Elements ändern. Hier sind Regular Expressions hilfreich, da Sie sich damit eine Menge Arbeit ersparen.

Beispiel 1: Umbenennen von Elementen

Der unten gezeigte XML-Beispielcode enthält eine Liste von Büchern. Angenommen, Sie möchten das Element `<Category>` jedes Buchs durch `<Genre>` ersetzen. Ein der Methoden, mit denen Sie dies erreichen, ist mit Hilfe von Regular Expressions.

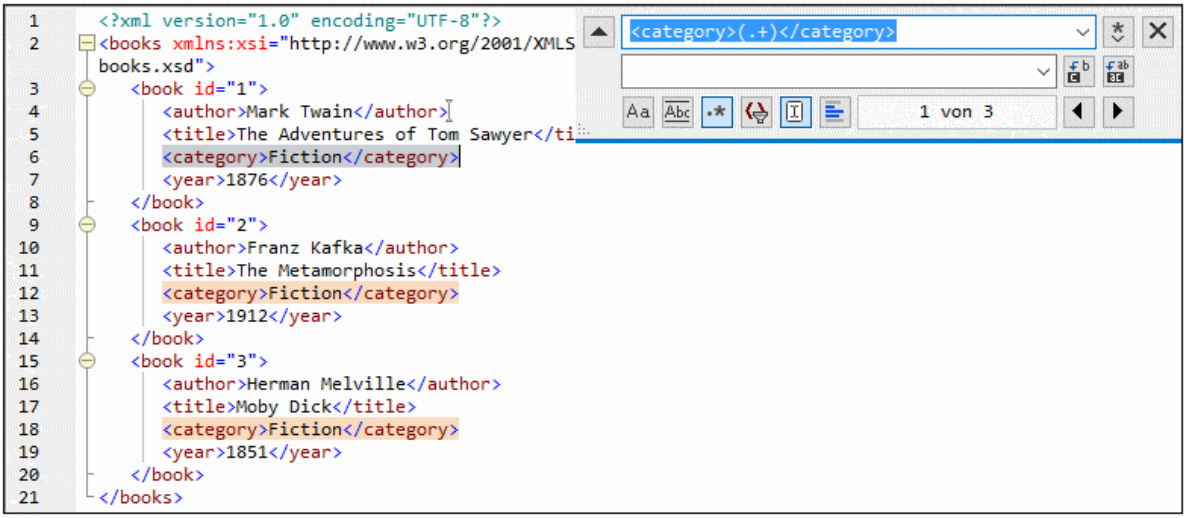
```
<?xml version="1.0" encoding="UTF-8"?>
<books xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="books.xsd">
   <book id="1">
      <author>Mark Twain</author>
      <title>The Adventures of Tom Sawyer</title>
      <category>Fiction</category>
      <year>1876</year>
   </book>
   <book id="2">
      <author>Franz Kafka</author>
      <title>The Metamorphosis</title>
      <category>Fiction</category>
      <year>1912</year>
```

 © 2019-2025 Altova GmbH

```
    </book>
    <book id="3">
        <author>Herman Melville</author>
        <title>Moby Dick</title>
        <category>Fiction</category>
        <year>1851</year>
    </book>
</books>
```

Gehen Sie dazu folgendermaßen vor:

1. Drücken Sie **Strg+H,** um das Such- und Ersetzungsdialogfeld aufzurufen.
2. Klicken Sie auf die Schaltfläche **Regular Expressions verwenden** .*.
3. Geben Sie in das Suchfeld den folgenden Text ein: `<category>(.+)</category>` . Mit dieser Regular Expression werden alle `category`-Elemente gefunden und markiert.



   Um den Text im Inhalt der einzelnen Elemente zu suchen (der im Vorhinein nicht bekannt ist), haben wir den getaggten Ausdruck `(.+)` verwendet. Der getaggte Ausdruck `(.+)` bedeutet, "finde eine oder mehrere Instanzen eines beliebigen Zeichens, welches `.+` entspricht, und merke Dir diese Übereinstimmung". Wie Sie im nächsten Schritt sehen, benötigen wird die Referenz auf den getaggten Ausdruck später.

4. Geben Sie in das Ersetzungsfeld den folgenden Text ein: `<genre>\1</genre>` . Mit dieser Regular Expression wird der Ersetzungstext definiert. Beachten Sie, dass darin eine Rückreferenz `\1` auf den vorhin im Suchfeld getaggten Ausdruck verwendet wird, d.h. `\1` bedeutet in diesem Zusammenhang "der Textinhalt der aktuellen <category>-Elementübereinstimmung".
5. Klicken Sie auf **Alle ersetzen** und überprüfen Sie die Ergebnisse. Alle `category`-Elemente wurden nun, wie geplant, in `genre` umbenannt.

Beispiel 2: Ändern der Reihenfolge von Attributen

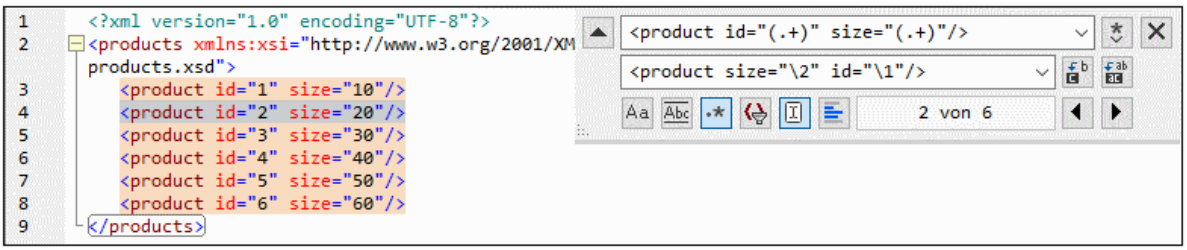
Der unten gezeigte XML-Beispielcode enthält eine Liste von Produkten. Jedes Produktelement hat zwei Attribute: `id` und `size`. Angenommen, Sie möchten die Reihenfolge der Attribute `id` und `size` in allen `product`-Elementen ändern (d.h. das Attribut `size` soll vor `id` stehen). Eine der Lösungsmethoden dafür ist mittels

Regular Expressions.

```
<?xml version="1.0" encoding="UTF-8"?>
<products xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="products.xsd">
   <product id="1" size="10"/>
   <product id="2" size="20"/>
   <product id="3" size="30"/>
   <product id="4" size="40"/>
   <product id="5" size="50"/>
   <product id="6" size="60"/>
</products>
```

Gehen Sie dazu folgendermaßen vor:

1. Drücken Sie **Strg+H,** um das Such- und Ersetzungsdialogfeld aufzurufen.
2. Klicken Sie auf die Schaltfläche **Regular Expressions verwenden** .
3. Geben Sie ins Suchfeld den folgenden Ausdruck ein: `<product id="(.+)" size="(.+)"/>` . Mit dieser Regular Expression wird ein product-Element im XML-Dokument gefunden. Beachten Sie, dass zwei Mal ein getaggter Ausdruck `(.+)` verwendet wird, um den Wert der einzelnen Attribute (der im Vorhinein nicht bekannt ist) zu finden. Mit dem getaggten Ausdruck `(.+)` wird der Wert jedes einzelnen Attributs gefunden (Der Wert wird als eine oder mehrere Instanzen eines beliebigen Zeichens angenommen, also `.+`).
4. Geben Sie in das Ersetzungsfeld den folgenden Ausdruck ein: `<product size="\2" id="\1"/>` . Diese Regular Expression enthält den Ersetzungstext für jedes gefundene product-Element. Beachten Sie, dass darin zwei Referenzen `\1` und `\2` verwendet werden. Diese entsprechen den getaggten Ausdrücken aus dem Suchfeld. Anders ausgedrückt, bedeutet `\1` "der Wert des Attributs `id`" und `\2` "der Wert des Attributs `size`".

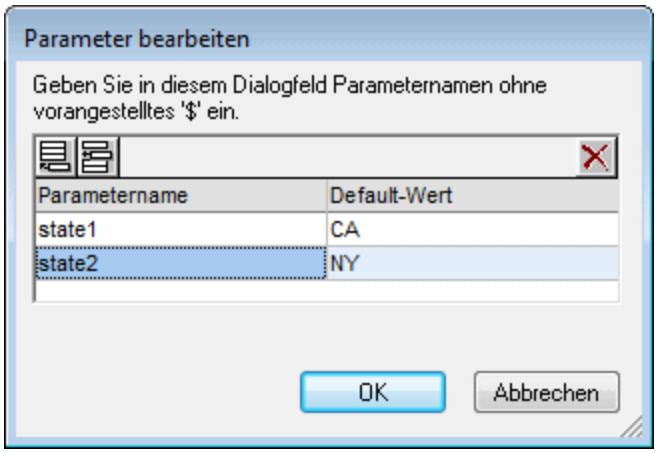


6. Klicken Sie auf **Alle ersetzen** und überprüfen Sie die Ergebnisse. Alle `product`-Elemente wurden nun aktualisiert, so dass das Attribut `size` jetzt vor dem Attribut `id` steht.

11.5.3 Stylesheet-Parameter

Mit Hilfe des Befehls **Stylesheet-Parameter** können Sie Parameter und ihre Standardwerte deklarieren und bearbeiten. Der Befehl steht sowohl in der Design-Dokument-Ansicht als auch in der Authentic Editor-Ansicht zur Verfügung. Bei Aufruf dieses Befehls wird das Dialogfeld "Parameter bearbeiten" (*Abbildung unten*) angezeigt.

© 2019-2025 Altova GmbH



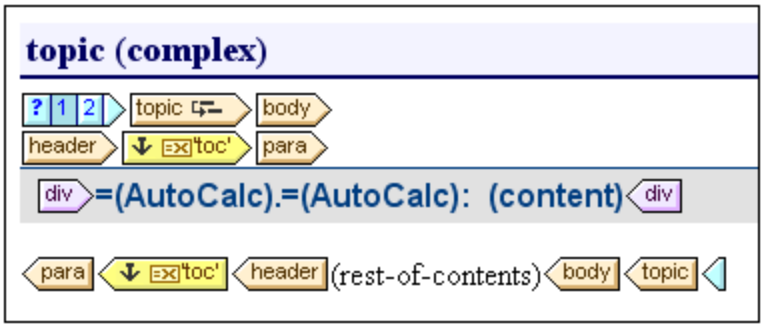
Beachten Sie bitte die folgenden Punkte:

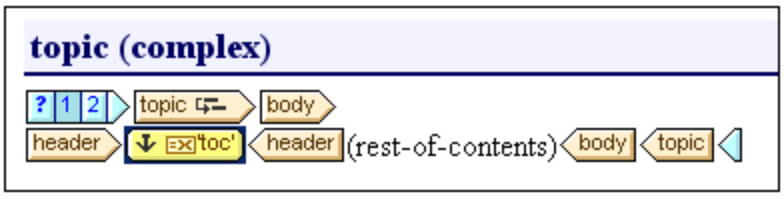
- Sie können Parameter für das gesamte Stylesheet einfügen, anhängen, bearbeiten und löschen.
- Parameternamen müssen mit einem Buchstaben beginnen und können die Zeichen `A` bis `Z`, `a` bis `z`, `0` bis `9` und den Unterstrich enthalten.
- Das Dialogfeld "Parameter bearbeiten" enthält alle benutzerdefinierten Parameter in einem SPS.
- Parameter können auch in der Seitenleiste "Design-Übersicht" [36] deklariert werden .

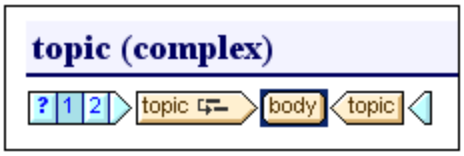
11.5.4 Markup reduzieren/erweitern

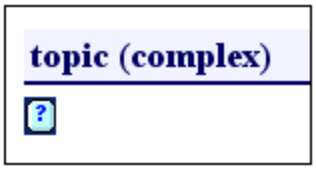
Der Befehl **Markup reduzieren/erweitern** ist ein Ein/Aus-Befehl, mit dem der ausgewählte Tag reduziert bzw. erweitert wird. Er kann auf jede Art von Tag angewendet werden: Node, vordefiniertes Format, SPS Verfahren usw. Um einen Tag zu reduzieren bzw. zu erweitern, doppelklicken Sie auf den Tag; Sie können auch auf den End-Tag eines erweiterten Tags doppelklicken, um diesen Tag zu reduzieren.

In der Abbildung unten sehen Sie, wie eine Reihe von Tags reduziert wird. Doppelklicken Sie auf einen reduzierten Tag, um ihn zu erweitern.







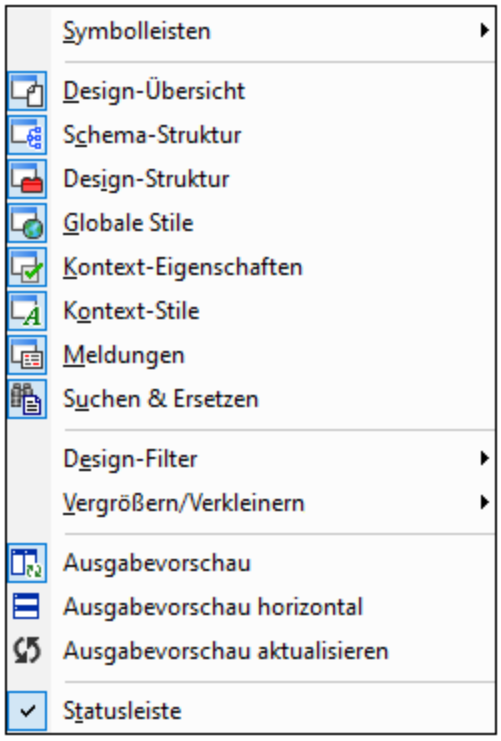


Durch Reduzieren von Tags können Sie die Anzeige Ihren Bedürfnissen bei der Bearbeitung entsprechend anpassen.

© 2019-2025 Altova GmbH

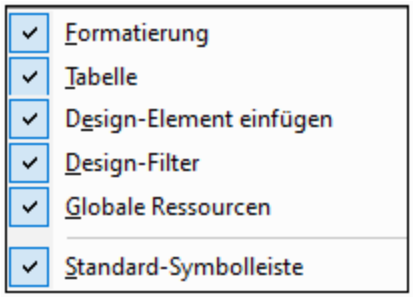
11.6 Menü "Ansicht"

Über das Menü **Ansicht** (*Abbildung unten*) können Sie das Aussehen der Benutzeroberfläche ändern und die diversen Komponenten der Benutzeroberfläche ein- und ausblenden. Sie können einzelne Symbolleisten, Design-Seitenleisten, Design-Filter und die Statusleiste ein- und ausblenden.



11.6.1 Symbolleisten und Statusleiste

Wenn Sie den Cursor über den Eintrag **Symbolleisten** platzieren, erscheint ein Untermenü (*Abbildung unten),* über das Sie die Anzeige der verschiedenen Symbolleisten ein- und ausblenden können.



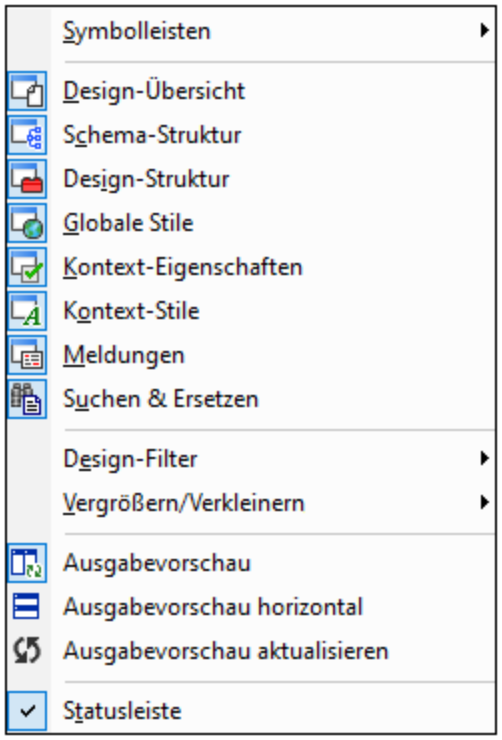
Wenn eine Symbolleiste mit einem Häkchen versehen ist, wird sie angezeigt. In der Abbildung oben werden alle Symbolleisten angezeigt. Um eine Symbolleiste ein- bzw. auszublenden, klicken Sie auf den entsprechenden Eintrag für die Symbolleiste. Eine ausführliche Beschreibung der Symbolleisten finden Sie im Abschnitt Referenz | Symbolleisten 457.

Statusleiste

Um die Statusleiste am unteren Rand des Applikationsfensters ein- oder auszublenden, klicken Sie auf den Eintrag "Statusleiste".

11.6.2 Design-Seitenleisten

Das Menü **Ansicht** enthält Befehle zum Ein- und Ausblenden der einzelnen Seitenleisten (*Abbildung unten*).



Wenn eine Seitenleiste eingeblendet ist (Die Schaltfläche des Befehls wird umrandet angezeigt), wird Sie in der Design-Ansicht angezeigt. Klicken Sie auf eine Seitenleiste, um sie je nach Wunsch ein- oder auszublenden. Eine ausgeblendete Seitenleiste kann auf diese Art auch wieder eingeblendet werden. Die für eine Seitenleiste vorgenommene Ansichtseinstellung ist ansichtsspezifisch: Eine Einstellung, die in einer bestimmten Ansicht (Design-Ansicht, Ausgabeansicht, kein Dokument geöffnet) definiert ist, wird so lange für dieser Ansichtskategorie beibehalten, bis sie geändert wird.

© 2019-2025 Altova GmbH

11.6.3 Design-Filter, Vergrößern/Verkleinern

Design-Filter

Das Menü **Design-Filter** enthält ein Untermenü mit Befehlen, mit denen Sie die in der Design-Ansicht angezeigten Vorlagen filtern können. Dies erweist sich als nützlich, wenn Ihr Design sehr lang ist oder mehrere Vorlagen enthält. Mit Hilfe von Design-Filtern können Sie festlegen, welche Arten von Vorlagen angezeigt werden sollen. Die folgenden Filteroptionen stehen zur Verfügung:

Schaltfläche	Befehl	Beschreibung
	Immer nur eine Vorlage anzeigen	Zeigt nur die ausgewählte Vorlage an. Platzieren Sie den Cursor in eine Vorlage und klicken Sie, um nur diese Vorlage anzuzeigen.
	Alle Vorlagentypen anzeigen	Zeigt alle Vorlagen im SPS an (Hauptvorlage, globale Vorlagen, Design-Fragmente und Layout-Vorlagen) .
	Importierte Vorlagen anzeigen	Blendet importierte Vorlagen ein und aus.
	Hauptvorlage ein-/ausblenden	Blendet die Hauptvorlage ein und aus.
	Globale Vorlagen ein-/ausblenden	Blendet die globalen Vorlagen ein und aus.
	Design-Fragmente ein-/ausblenden	Blendet die Design-Fragmente ein und aus.

Beachten Sie, dass diese Befehle auch als Schaltflächen in der Symbolleiste Design-Filter[464] zur Verfügung stehen.

Vergrößern/Verkleinern

Mit dem Befehl **Vergrößern/Verkleinern** können Sie einen Zoom-Faktor im Untermenü auswählen. Sie können aber auch hinein- oder herauszoomen, indem Sie den Zoom-Faktor in der Auswahlliste (in der Standard-Symbolleiste) ändern oder die **Strg-Taste** drücken und mit der Maus scrollen.

11.6.4 Ausgabevorschauen

Es gibt drei Ausgabevorschau-Befehle:

- **Ausgabevorschau**: Damit schalten Sie die Ausgabevorschau ein/aus.
- **Ausgabevorschau horizontal**: Dieser Befehl ist aktiviert, wenn die Ausgabevorschau aktiviert wurde. Mit diesem Befehl wechseln Sie von einer vertikal geteilten Ansicht in eine horizontal geteilte Ansicht.

Wenn die Ausgabevorschau horizontal geteilt angezeigt wird, ändert sich der Befehl automatisch in **Ausgabevorschau vertikal**, wodurch Sie die Ansicht wieder vertikal geteilt anzeigen lassen können.

- **Ausgabevorschau aktualisieren**, wodurch die Ausgabevorschau aktualisiert wird, sodass alle Änderungen am Design sofort angezeigt werden.

© 2019-2025 Altova GmbH

11.7 Menü "Einfügen"

Das Menü **Einfügen** enthält Befehle, über die Sie die verschiedensten Design-Komponenten in das SPS einfügen können. Einige dieser Befehle stehen als Symbolleisten-Schaltflächen[457] zur Verfügung. Des Weiteren stehen Befehle aus dem Menü "Einfügen" auch über Kontextmenüs zur Verfügung, die angezeigt werden, wenn Sie im SPS Design mit der rechten Maustaste auf eine Auswahl oder einen Cursoreinfügepunkt klicken. Befehle, die an dieser Stelle im SPS nicht verfügbar sind, erscheinen im Kontextmenü ausgegraut.

Anmerkung: Da die Befehle zum Einfügen zum Erstellen des SPS dienen, stehen Sie nur in der Design-Ansicht zur Verfügung.

11.7.1 Inhalt

Mit dem Befehl **Inhalt** wird ein Platzhalter `(Inhalt)` am Cursoreinfügepunkt eingefügt. Ein Platzhalter `(Inhalt)` kann innerhalb von zwei Nodetypen eingefügt werden: **Element** und **Attribut** und gibt an, dass alle Child-Nodes des aktuellen Node verarbeitet werden.

- Wenn es sich beim aktuellen Node um einen Element-Node handelt, werden die Child-Element-Nodes und Text-Nodes des Node verarbeitet. Für die Verarbeitung von Child-Element-Nodes werden globale Vorlagen verwendet, falls vorhanden. Andernfalls wird die vordefinierte Vorlagenregel für Elemente verwendet. Für die Verarbeitung von Text-Nodes wird die vordefinierte Vorlagenregel für Text-Nodes verwendet, sodass der Text ausgegeben wird. Bei Verwendung der vordefinierten Vorlagenregel für Elemente wird der Text aller untergeordneten Text-Nodes ausgegeben. Beachten Sie, dass Attributwerte bei Verwendung des Platzhalters (`Inhalt`) nicht ausgegeben werden, es sei denn, für den Parent Node des Attributs oder einen seiner übergeordneten Nodes wurde eine globale Vorlage definiert oder einer seiner übergeordneten Nodes und das Attribut werden mit Hilfe des Platzhalters `(Inhalt)` oder einer beliebigen anderen Komponente zur Darstellung von Inhalt explizit ausgegeben.
- Wenn es sich beim aktuellen Node um einen Attribut-Node handelt, wird die vordefinierte Vorlagenregel für den Child-Text-Node des Attributs verwendet. Diese Vorlage kopiert den Text des Text-Node in die Ausgabe, sodass dadurch der Wert des Attributs ausgegeben wird.

Der Platzhalter `(Inhalt)` kann auch für einen Node eingefügt werden, indem der Cursor innerhalb der Node-Tags platziert wird, Sie auf die rechte Maustaste klicken und den Befehl **Einfügen | Inhalt** wählen. Oder Sie klicken in der Symbolleiste "Design-Element einfügen"[462] auf die Schaltfläche **Inhalt einfügen** und anschließend auf die Stelle im Design, an der das Element eingefügt werden soll.

Zuweisen von Stilen zum Inhalt

Der Platzhalter `(Inhalt)` kann formatiert werden, indem Sie ihn auswählen und ein vordefiniertes Format und/oder Eigenschaften in der Seitenleiste "Stil" verwenden. Diese Formatierung ist im Design sichtbar und wird in der Ausgabe auf den Inhalt des Node angewendet.

Ersetzen von Inhalt

Wenn ein anderer Node aus der Schema-Struktur in einen Node gezogen wird, der einen Platzhalter `(Inhalt)` enthält, so wird der vorhandene Platzhalter `(Inhalt)` durch den neuen Node ersetzt.

Löschen von Inhalt

Der Platzhalter `(Inhalt)` kann gelöscht werden, indem Sie ihn auswählen und die **Entf**-Taste auf der Tastatur drücken.

Anmerkung: Sie können eine **leere Vorlagenregel** erstellen, indem Sie den Platzhalter `(Inhalt)` eines Node löschen. Eine leere Vorlagenregel eignet sich, um zu definieren, dass auf einen bestimmten Node keine Vorlage angewendet werden soll, d.h. dass keine Ausgabe erzeugt werden soll.

11.7.2 Restlicher Inhalt

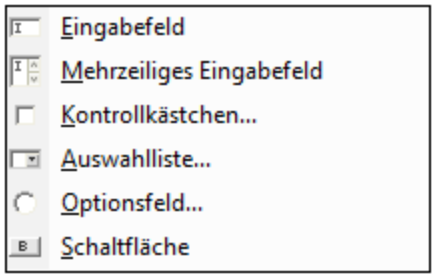
Mit dem Befehl **Restlicher Inhalt** wird der Platzhalter `(restlicher Inhalt)` für diesen Node eingefügt. Dieser Platzhalter steht für den Inhalt **nicht verwendeter Child Nodes** des aktuellen Node; entspricht der XSLT Regel `xsl:apply-templates`, die auf nicht verwendete Elemente und Text-Nodes des aktuellen Elements angewendet wird. Beachten Sie, dass Vorlagen bei Child-Attributen nicht angewendet werden. Der Platzhalter `(restlicher Inhalt)` kann auch für ein Element eingefügt werden, in dem der Cursor innerhalb der Element-Tags platziert wird und durch Klick auf die rechte Maustaste aus dem Kontextmenü der Befehl **Restlichen Inhalt einfügen** ausgewählt wird.

Verwenden Sie den Platzhalter `(restlicher Inhalt)` in Situationen, wenn ein Child-Element auf eine bestimmte Art verarbeitet und Vorlagen auf die gleichrangigen Elemente dieses Elements angewendet werden sollen. Es ist wichtig, dass Vorlagen auf gleichrangige Elemente angewendet werden, um zu vermeiden, dass die gleichrangigen Elemente eventuell nicht verarbeitet werden. Auf diese Art können Sie auch Elemente weiter unten in der Dokument-Hierarchie verarbeiten.

Der Platzhalter `(restlicher Inhalt)` kann gelöscht werden, indem Sie ihn auswählen und auf der Tastatur die **Entf**-Taste drücken.

11.7.3 Formular-Steuerelemente

Wenn Sie die Maus über den Befehl **Formularsteuerelemente einfügen** platzieren, wird ein Untermenü mit den Befehlen zum Einfügen der verschiedenen Formular-Steuerelemente (Dateneingabeelemente[158]) (*siehe Abbildung unten*) angezeigt.



Eine Anleitung zum Erstellen der einzelnen Formular-Steuerelemente finden Sie im Abschnitt Verwenden von Dateneingabeelementen[158]. Nach Erstellung eines Formular-Steuerelements können Sie seine Eigenschaften

 © 2019-2025 Altova GmbH

bearbeiten. Wählen Sie das Element dazu aus und bearbeiten Sie die entsprechende Eigenschaft in der Seitenleiste "Eigenschaften"[49].

Formular-Steuerelemente können auch ins Design eingefügt werden, indem Sie mit der rechten Maustaste auf den Einfügepunkt klicken und den Befehl **Einfügen | Formularsteuerelemente** auswählen oder in der Symbolleiste "Design-Element einfügen"[462] auf die entsprechende Schaltfläche klicken und anschließend auf die Stelle im Design, an der das Steuerelement eingefügt werden soll.

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-Ansicht als Komponenten des Textinhalts angezeigt werden.

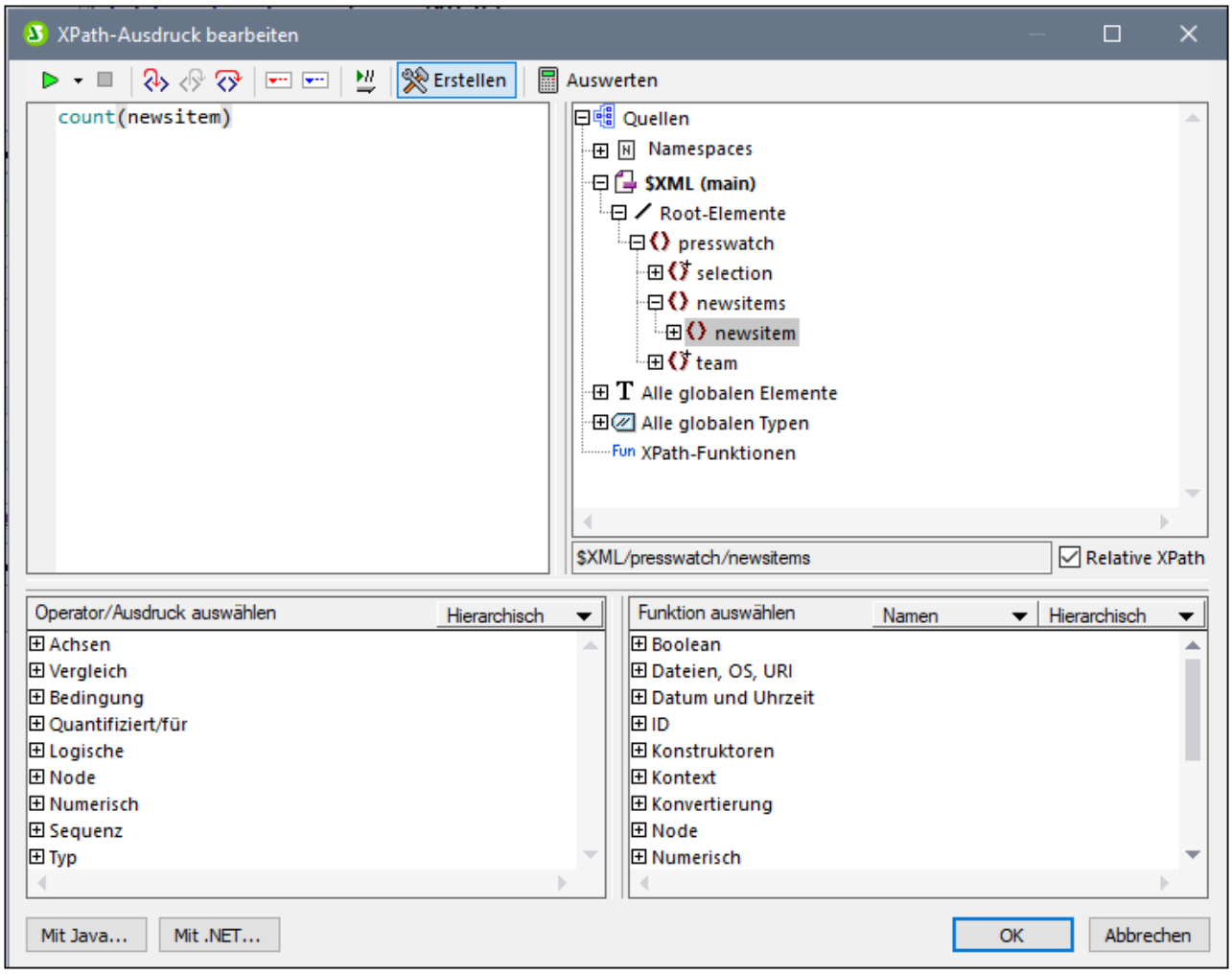
11.7.4 Automatische Berechnung

Bei einer **automatischen Berechnung** wird ein XPath-Ausdruck zum Berechnen eines Werts verwendet. Dieser Wert wird an der Stelle angezeigt, an der die automatische Berechnung eingefügt wird. Eine automatische Berechnung kann als Textwert, Eingabefeld oder mehrzeiliges Eingabefeld in das SPS eingefügt werden. Platzieren Sie den Cursor an die gewünschte Stelle und klicken Sie entweder auf die rechte Maustaste oder wählen Sie den Befehl über des Menü **Einfügen** aus. Wenn Sie den Cursor über **Einfügen | Automatische Berechnung** platzieren, wird ein Menü angezeigt (*Abbildung unten*), über das Sie auswählen können, wie die automatische Berechnung eingefügt werden soll. Alternativ dazu können Sie die Schaltfläche "Automatische Berechnung " in der Symbolleiste "Design-Element einfügen"[462] auswählen.

Der Wert der automatischen Berechnung wird im Ausgabedokument angezeigt.

Der XPath-Ausdruck für die automatische Berechnung

Wenn Sie auswählen, wie die automatische Berechnung dargestellt werden soll, wird das Dialogfeld "XPath-Ausdruck bearbeiten[440]" (*Abbildung unten*) angezeigt.



Der Kontext-Node für den erstellten Ausdruck erscheint in der Schemastruktur im linken Bereich markiert. Sie können den XPath-Ausdruck direkt in das Textfeld eingeben oder auf einen Eintrag in jedem beliebigen der drei Fensterbereiche doppelklicken um ihn einzufügen. Nodes, die aus der Schemastruktur im linken Bereich eingefügt werden, werden relativ zum Kontext-Node (wenn das Kontrollkästchen *Relativer XPath* aktiviert ist) oder als absoluter Ausdruck beginnend mit den Dokument-Node eingefügt (wenn das Kontrollkästchen *Relativer XPath* deaktiviert ist).

Nachdem Sie den XPath-Ausdruck fertiggestellt haben, klicken Sie auf OK, um die automatische Berechnung in das Dokument einzufügen.

11.7.5 Absatz, Spezieller Absatz

Mit dem Befehl **Absatz** § wird ein HTML-Absatzelement `<p>` rund um die ausgewählte Komponente eingefügt. Eine Komponente gilt als ausgewählt, wenn der gesamte Node ausgewählt ist (indem Sie auf einen seiner Tags klicken) oder wenn der statische Text ausgewählt ist. Wenn der Cursor in den statischen Text platziert wird, wird an dieser Stelle das Absatzelement (Start- und End-Tag) eingefügt. Ein Absatz kann auch mittels der Schaltfläche **Absatz einfügen** aus der Symbolleiste "Design-Element einfügen" 462 eingefügt

werden.

Mit dem Befehl **Spezieller Absatz** können Sie dem ausgewählten Node ein vordefiniertes Format zuweisen. Die verfügbaren vordefinierten Formate können auch über die Auswahlliste in der Symbolleiste ausgewählt werden.

Jede Absatzart verfügt über eigene Formatierungsfunktionen. Beachten Sie, dass Sie mit Hilfe der Formatart `pre` Zeilenschaltungen als Zeilenschaltungen ausgeben können, anstatt dass diese zu Leerzeichen normalisiert werden.

11.7.6 Bild

Mit dem Befehl **Bild** wird das Dialogfeld "Bild einfügen" (*siehe Abbildungen unten*) aufgerufen, in dem Sie festlegen können, welches Bild eingefügt werden soll. Dasselbe Dialogfeld lässt sich auch über die Symbolleiste "Design-Elemente einfügen" 462 über die Schaltfläche **Bild einfügen** aufrufen.

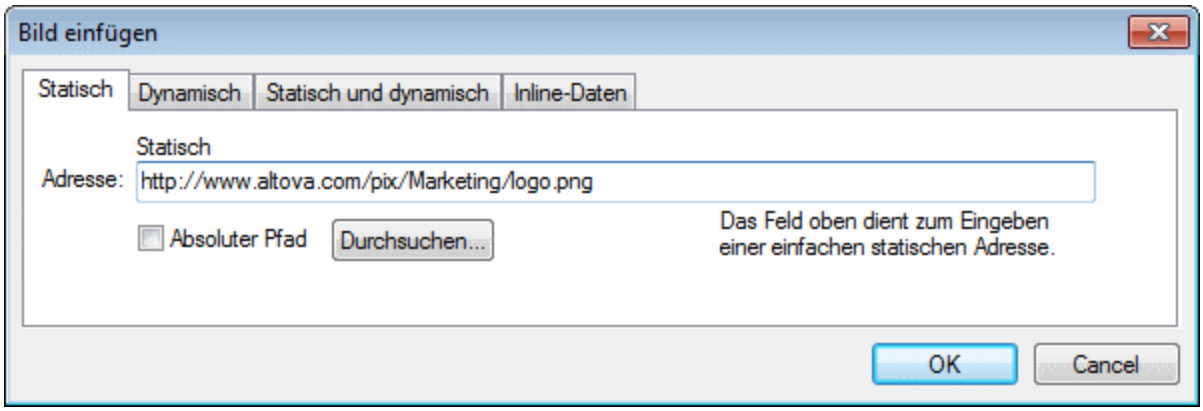
Das Dialogfeld "Bild einfügen" verfügt über vier Register. Auf jedem davon steht eine andere Methode zur Verfügung, um den Bildpfad zu definieren:

- *Statisch:* Dient zur direkten Eingabe der Bild-URI
- *Dynamisch:* Zur Ermittlung der Bild-URI anhand des XML-Dokuments oder zur Generierung der URI mit Hilfe eines XPath-Ausdrucks
- *Statisch und dynamisch:* Dient zum Kombinieren der statischen und dynamischen Methode
- *Inline-Daten:* Dient zur Auswahl eines Bilds, das in einer XML-Datei als Base-16- oder Base-64-kodierter Text gespeichert ist.

Im Folgenden werden die Register näher beschrieben.

Statisch

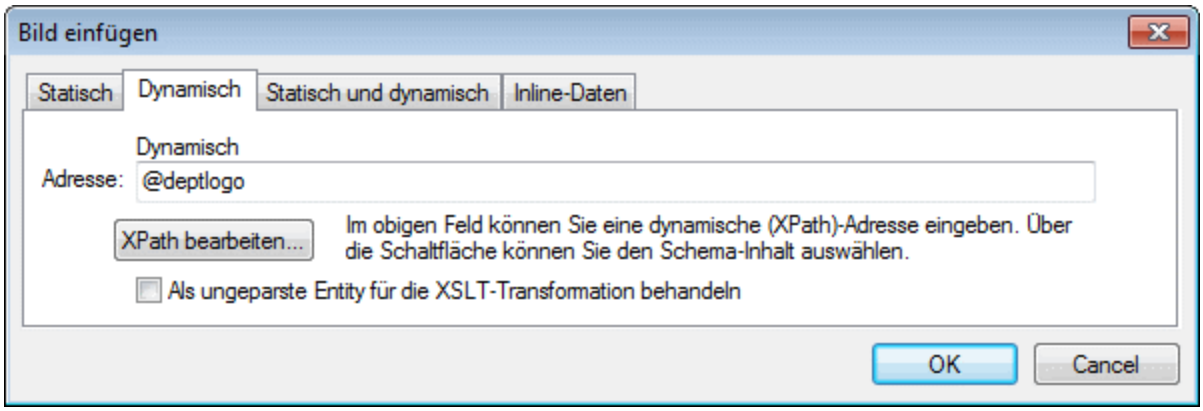
Die Bild-URI wird direkt in das Feld *Adresse* eingegeben (*siehe Abbildung unten*). In der Abbildung unten lautet die Bild-URI: `http://www.altova.com/pix/Marketing/logo.png`.



Sie können angeben, ob die URI absolut (Kontrollkästchen *Absolut* ist aktiviert) oder relativ (Kontrollkästchen *Absolut* ist deaktiviert) sein soll. Falls eine relative URI eingegeben wird, wird sie relativ zum Pfad der SPS-Datei aufgelöst. Um die (absolute oder relative) URI automatisch einzugeben, klicken Sie auf die Schaltfläche **Durchsuchen** und navigieren Sie zur gewünschten Bilddatei.

Dynamisch

Die Bild-URI wird durch einen XPath-Ausdruck zurückgegeben. Der XPath-Ausdruck in der Abbildung unten lautet `@deptlogo`. Dabei wird von der Annahme ausgegangen, dass die Bild-URI im Attribut `deptlogo` des Kontext-Node gespeichert ist. Der Kontext-Node ist der Node, in dem das Bild erstellt wird.

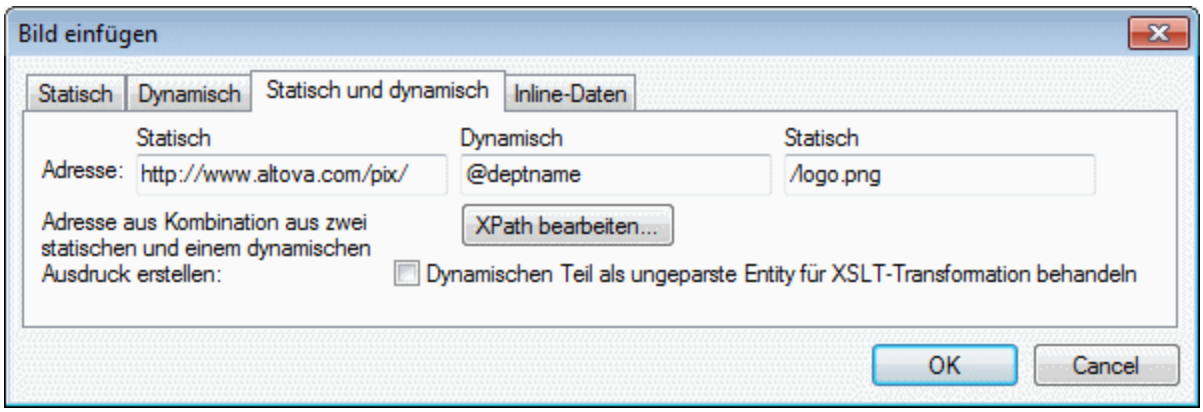


Klicken Sie auf die Schaltfläche **XPath bearbeiten,** um das Dialogfeld XPath-Ausdruck bearbeiten[452] aufzurufen. Der Kontext-Node erscheint in der Schema-Struktur des Dialogfelds markiert.

Wenn das SPS auf einer DTD basiert und darin **ungeparste Entities** verwendet werden, kann eine ungeparste Entity, die die Bild-URI referenziert, verwendet werden. Aktivieren Sie zuerst das Kontrollkästchen *Als ungeparste Entity behandeln* und geben Sie anschließend den XPath-Ausdruck zur Auswahl des Nodes, der die ungeparste Entity enthält, ein. Nähere Informationen zur Verwendung von ungeparsten Entities finden Sie unter URIs von ungeparsten Entities[377].

Statisch und dynamisch

Hierbei werden zum Generieren der URI sowohl statische als auch dynamische Methoden verwendet.

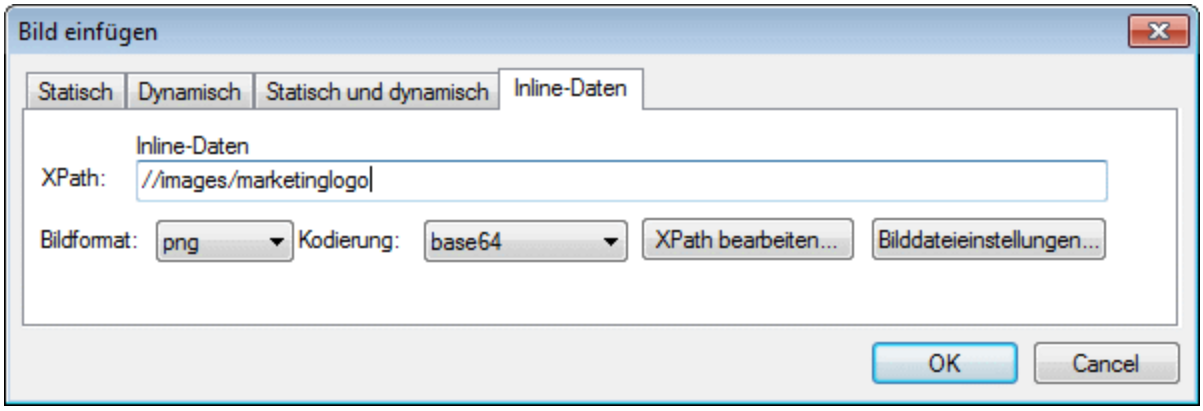


Wenn das Attribut `deptname` des Kontext-Node einen Wert `Marketing` hat, dann lautet die in der Abbildung oben zusammengesetzte Bild-URI: `http://www.altova.com/pix/Marketing/logo.png`. Beachten Sie, dass Sie für den dynamischen Teil das Dialogfeld XPath-Ausdruck bearbeiten[452] verwenden können.

© 2019-2025 Altova GmbH

Inline-Daten

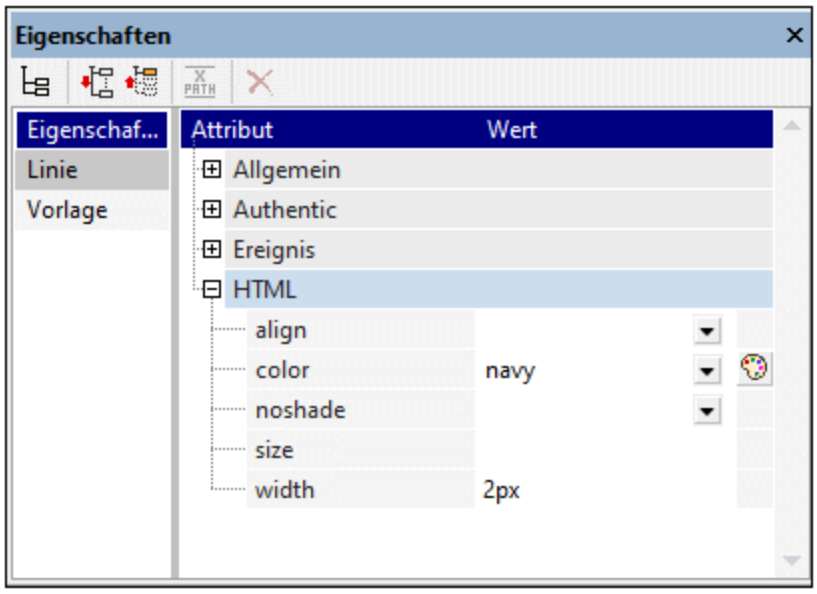
Ein Bild kann in einer XML-Datei als in Base-16 oder Base-64 kodierter Text gespeichert werden. Der im Dialogfeld "Bild einfügen" (*siehe Abbildung oben*) definierte XPath-Ausdruck wählt den Node aus, der den kodierten Text enthält. Über die Auswahlliste "Kodierung" wird die in der XML-Quelldatei verwendete Kodierung angegeben, damit StyleVision den kodierten Text korrekt lesen kann und die Auswahlliste "Bildformat" gibt an, in welchem Format die Bilddatei generiert werden muss. (Anhand der kodierten Textdaten wird eine Bilddatei generiert und diese Datei wird anschließend im Ausgabedokument verwendet.)



Über das (über die Schaltfläche **Bilddateieinstellungen** aufgerufene) Dialogfeld "Bilddateieinstellungen" können Sie der Bilddatei, die erstellt werden soll, einen Namen geben. Sie können auch wählen, dass kein Name angegeben wird und StyleVision einen Namen generiert.

11.7.7 Horizontale Linie

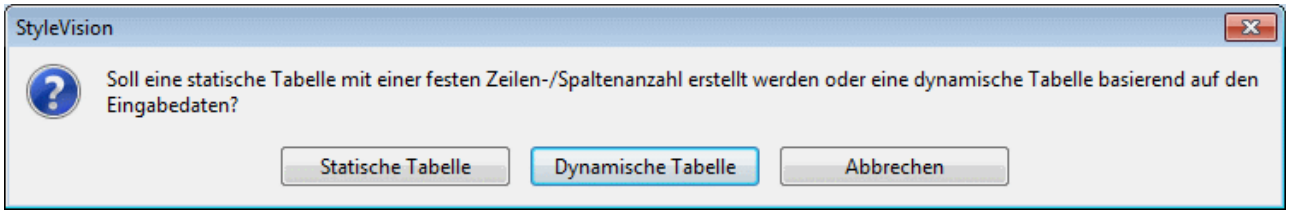
Mit Hilfe des Befehls **Horizontale Linie** wird eine horizontale Linie am Cursoreinfügepunkt eingefügt. Dieser Befehl steht nicht zur Verfügung, wenn eine SPS Komponente ausgewählt ist. Um Eigenschaften für die horizontale Linie zu definieren, wählen Sie die Linie im Design aus und wählen Sie in der Seitenleiste "Eigenschaften" den Eintrag *Linie* aus und geben Sie in der Gruppe *HTML* bestimmte Werte für Eigenschaften ein (*siehe Abbildung unten*).



Sie können die folgenden Eigenschaften für die Linie definieren: `color` (Farbe), `size` (Dicke), `width` (Breite im Design), `alignment` (Ausrichtung) und die Eigenschaft `noshade` (kein Schatten).

11.7.8 Tabelle

Mit dem Befehl **Tabelle einfügen** wird das Dialogfeld "Tabelle einfügen" aufgerufen (*Abbildung unten*).



Wählen Sie die entsprechende Schaltfläche, je nachdem, ob Sie eine statische oder eine dynamische Tabelle erstellen möchten. Eine Anleitung zu diesen Varianten finden Sie in den Abschnitten: Statische SPS-Tabellen[128] und Dynamische SPS-Tabellen[129].

Beachten Sie, dass Tabellen auch mit dem Menübefehl **Tabelle | Tabelle einfügen** und über die Schaltfläche **Tabelle einfügen** (aus der Symbolleiste "Design-Elemente einfügen" erstellt werden können.

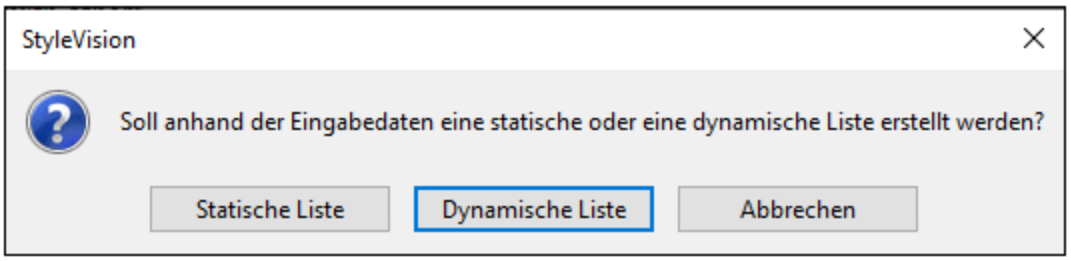
© 2019-2025 Altova GmbH

11.7.9 Nummerierung und Aufzählungszeichen

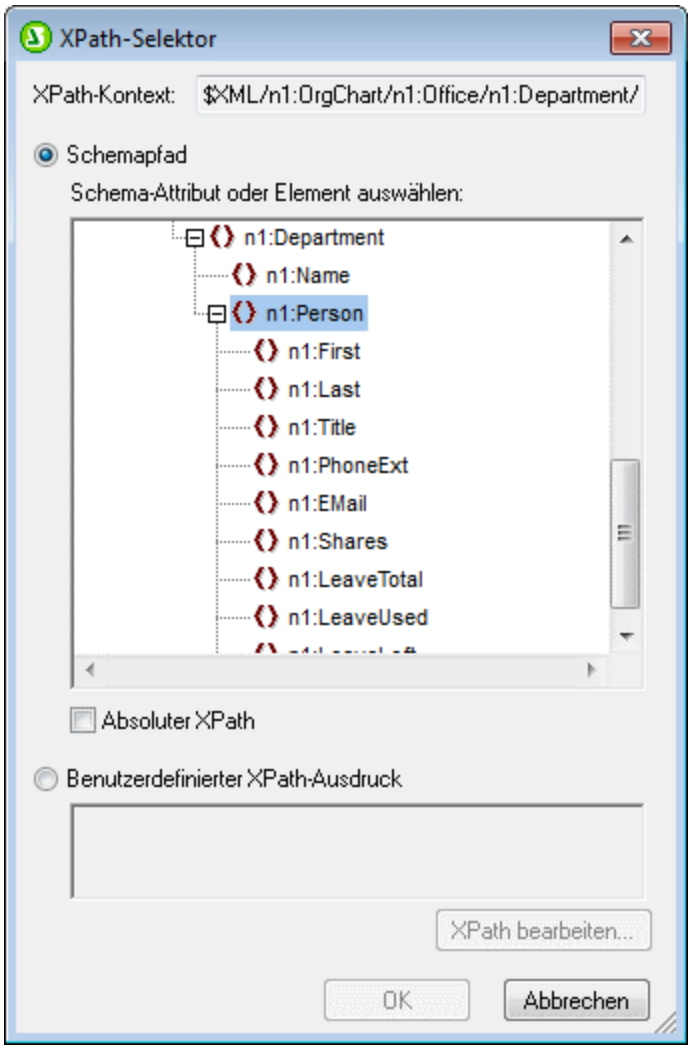
Mit Hilfe des Befehls **Nummerierung und Aufzählungszeichen** können Sie statische oder dynamische Listen erstellen. Die Einträge einer statischen Liste werden in das SPS eingegeben, während die Einträge dynamischer Listen aus gleichrangigen Nodes im XML-Dokument stammen.

Um eine dynamische Liste zu erstellen, gehen Sie folgendermaßen vor:

1. Platzieren Sie den Cursor an die Stelle im Design, an der die statische Liste erstellt werden soll und wählen Sie den Menübefehl **Nummerierung und Aufzählungszeichen**. Daraufhin wird ein Dialogfeld angezeigt, in dem Sie gefragt werden, ob Sie eine statische oder eine dynamische Liste erstellen möchten (*Abbildung unten*).



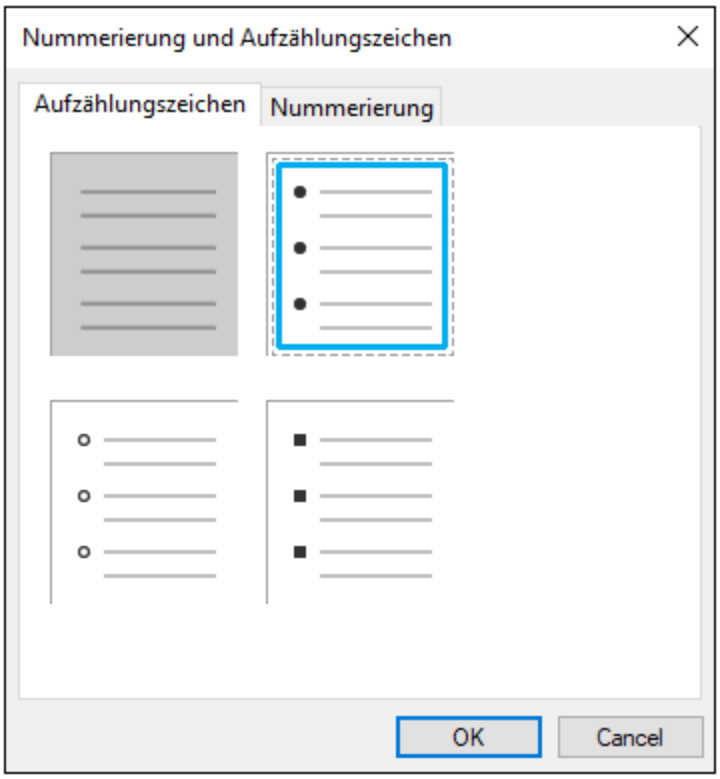
2. Wenn Sie auf **Statische Liste** klicken wird das in Schritt 3 beschriebene Dialogfeld angezeigt. Wenn Sie auf **Dynamische Liste** klicken, wird das Dialogfeld "XPath-Selektor" (*Abbildung unten)* angezeigt.
3. Beachten Sie, dass der XPath-Kontext im Dialogfeld "XPath-Selektor" der Kontext an der Einfügestelle ist und dass dieser Kontext im Dialogfeld nicht geändert werden kann. Wählen Sie den Node aus, der als dynamische Liste erstellt werden soll. In der Abbildung unten ist der Kontext-Node `n1:Department` und der Node `n1:Person` wurde als der Node ausgewählt, der als Liste erstellt werden soll. D.h. der Inhalt der einzelnen `n1:Person` Nodes wird in Form von Listeneinträgen erstellt.



Wenn Sie die Option "Benutzerdefinierter XPath-Ausdruck" auswählen, können Sie einen XPath-Ausdruck eingeben, der den Node auswählt, der als dynamische Tabelle erstellt werden soll. Wenn Sie auf **OK** klicken, wird das im nächsten Schritt beschriebene Dialogfeld "Nummerierung und Aufzählungszeichen" angezeigt.

4. Wählen Sie hier den gewünschten Listentyp aus. Sie haben die Wahl zwischen einer Liste mit Aufzählungszeichen (mit den Aufzählungszeichen Punkte, Kreise oder Quadrate) oder einer nummerierten Liste. Klicken Sie auf **OK**, um die Liste mit der auswählten Variante zu erstellen).

© 2019-2025 Altova GmbH



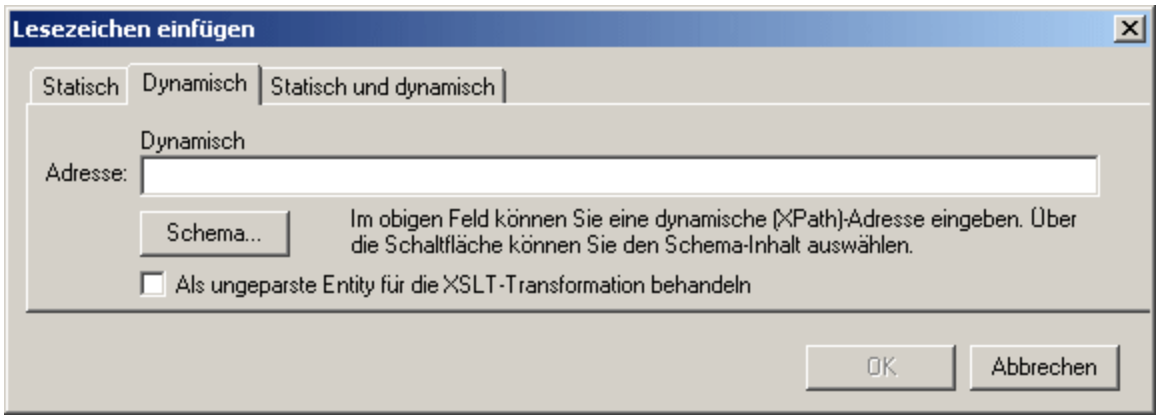
Anmerkung: Sie können die statische Liste auch durch Platzieren des Cursors an die gewünschte Stelle im Design und Klicken auf die Schaltfläche "Nummerierung und Aufzählungszeichen" in der Symbolleiste "Design-Element einfügen" 462 erstellen. Eine dynamische Liste kann auch durch Ziehen eines Node aus der Schema-Struktur in das Design erstellt werden.

11.7.10 Lesezeichen

Mit Hilfe des Befehls **Lesezeichen** können Sie ein Lesezeichen (Anchor) an jeder beliebigen Stelle im SPS einfügen. Ein Lesezeichen kann durch einen Hyperlink 517 referenziert werden.

Um ein Lesezeichen einzufügen, gehen Sie folgendermaßen vor:

1. Platzieren Sie den Cursor an die Stelle, an der das Lesezeichen erstellt werden soll.
2. Wählen Sie den Befehl **Einfügen | Lesezeichen** oder klicken Sie auf die rechte Maustaste und wählen Sie den Befehl **Einfügen | Lesezeichen**.



3. Wählen Sie im Dialogfeld "Lesezeichen einfügen" [320], je nachdem, ob der Name des Lesezeichens statisch sein soll (Register "Statisch"), dynamisch aus dem XML-Dokument abgerufen werden ("Dynamisch") oder sowohl aus statischen als auch dynamischen Teilen bestehen soll (Statisch und dynamisch), ein Register aus). In der Abbildung oben wird ein dynamisches Lesezeichen erstellt, dessen Name eine eindeutige ID für die einzelnen `Name`-Child-Elemente des Kontekt-Node ist.
4. Klicken Sie auf OK. Das Lesezeichen wird definiert.

Hinweis:

Ein Lesezeichen wird für eine einzelne fixe Stelle im Design angelegt. Wenn sich das Element, in dem sich das Lesenzeichen befindet, wiederholt, so wird in jeder Instanz des sich wiederholenden Elements ein Lesezeichen erstellt. Wenn Sie dem Lesenzeichen einen statischen Namen geben, so erhält jedes dieser Lesezeichen denselben Namen. Daher empfiehlt es sich, dem Lesenzeichen in solchen Fällen (bei sich wiederholenden Elementen) einen dynamischen Namen zu geben. Dabei kann es sich z.B. um den Namen eines Child-Elements des Kontextnode (also des Elements in dem das Lesezeichen erstellt wird) handeln. Wenn der für den dynamischen Namen ausgewählte Node eventuell in mehreren Instanzen denselben Namen haben könnte, können Sie mit Hilfe der `generate-id( )`-Funktion zur Generierung des Namens (*siehe Abbildung oben*) sicherstellen, dass das Lesezeichen einen eindeutigen Namen hat. Zur Referenzierung eines solchen Lesezeichens kann dieselbe ID auch als `href`-Wert eines Hyperlink [325] generiert werden. Verwenden Sie in diesem Fall unbedingt den Fragment-Identifier `#` vor der `generate-id()` Funktion. Der XPath-Ausdruck würde lauten: `concat('#', generate-id(nodeXXX))`.

Nachdem Sie das Lesezeichen erstellt haben, können Sie es bearbeiten. Klicken Sie dazu mit der rechten Maustaste auf das Lesezeichen und wählen Sie im Kontextmenü den Befehl **Lesezeichennamen bearbeiten**. Alternativ dazu können Sie auch in der Eigenschaftsleiste auf der Seite in der Eigenschaftsgruppe *Lesezeichen* auf die Schaltfläche **Bearbeiten** des Namensattributs des Lesezeichens klicken und die erforderlichen Änderungen vornehmen.

Löschen eines Lesezeichens

Um ein Lesezeichen zu löschen, wählen Sie es im Design aus und drücken Sie die **Entf**-Taste.

© 2019-2025 Altova GmbH

11.7.11 Hyperlink

Mit Hilfe des Befehls **Hyperlink** können Sie einen Link einfügen, der von einer beliebigen Stelle im Ausgabedokument (HTML) auf einen Anker innerhalb des Ausgabedokuments oder ein externes Dokument oder Dokumentfragment verweist.

Um einen Hyperlink einzufügen, gehen Sie folgendermaßen vor:

1. Ein Hyperlink kann rund um eine vorhandene Design-Komponente angelegt oder an jeder beliebigen Stelle des Dokuments (mit dem darauf folgenden Linktext) eingefügt werden. Wählen Sie die SPS Komponente oder das Textfragment aus, die/das zu einem Hyperlink gemacht werden soll oder platzieren Sie den Cursor an die Stelle, an der der Link eingefügt werden soll.
2. Klicken Sie in der Symbolleiste auf das Hyperlink-Symbol oder wählen Sie den Befehl **Einfügen | Hyperlink** oder klicken Sie auf die rechte Maustaste und wählen Sie den Befehl **Einfügen | Hyperlink** (wenn keine Design-Komponenten ausgewählt ist) oder **Einschließen in | Hyperlink** (wenn eine Design-Komponente ausgewählt ist). Ein Hyperlink kann auch mittels der Schaltfläche **Hyperlink einfügen** aus der Symbolleiste "Design-Element einfügen" 462 eingefügt werden.
3. Definieren Sie im Dialogfeld "Hyperlink einfügen" 322, das angezeigt wird, das Dokument oder Dokumentfragment, auf das der Link verweisen soll. Wenn Sie einen Link zu einem Dokumentfragment erstellen (d.h. zu einem Lesezeichen innerhalb eines Dokuments), inkludieren Sie das #-Symbol. Die URI für den Hyperlink wird in einer der folgenden Formen definiert:

   - als statische Adresse (wird direkt eingegeben; Sie können über die Durchsuchen-Schaltfläche eine HTML-Datei , oder über die Lesezeichen-Schaltfläche ein Fragment im aktuellen Dokument auswählen). Ein Beispiel wäre: `http://www.altova.com` (statische Webseiten URI); `U:\documentation\index.html` (über die Schaltfläche "Durchsuchen"); oder `#top_of_page` (über die Schaltfläche "Lesezeichen").
   - als dynamische Adresse (welche aus einem von Ihnen definierten Node im XML-Dokument stammt). Ein Beispiel wäre ein Node wie z.B. `//otherdocs/doc1`. Wenn der Name des Lesezeichens mit der `generate-id()` Funktion generiert wurde, dann sollte das `href` Attribut des Hyperlinks mit derselben `generate-id()` Funktion generiert werden. Informationen dazu finden Sie unter Definieren von Hyperlinks 325.
   - als Kombination aus statischem und dynamischem Text für eine Adresse (Sie geben den statischen Text und den XML-Dokument-Node an). Ein Beispiel wäre `www.altova.com -- department/name -- #intropara`.
4. Klicken Sie auf OK. Der Hyperlink wird erstellt.

Anmerkung: Sie können den XPath-Ausdruck bei Angabe des Node für die Definition eines dynamischen Hyperlinks als absoluten XPath-Ausdruck angeben, indem Sie das Kontrollkästchen "*Absoluter Pfad*" aktivieren. Wenn dieses Kästchen deaktiviert ist, wird der XPath-Ausdruck relativ zu dem Kontext-Node, d.h. dem Node, in den der Hyperlink eingefügt wird, angegeben.

Verwendung von ungeparsten Entities

Falls Sie eine DTD als Schema verwenden, können Sie für den dynamischen Teil einer Hyperlink-Adresse die URI verwenden, die in der DTD für eine ungeparste Entity deklariert wurde. Nähere Informationen zur Verwendung von ungeparsten Entities finden Sie unter Verwendung von URIs für ungeparste Entities 377.

Bearbeiten eines Hyperlink

Sie können die `href` eines Hyperlink bearbeiten, nachdem Sie sie erstellt haben. Klicken Sie dazu mit der rechten Maustaste auf den Hyperlink und wählen Sie den Befehl **URL bearbeiten.**. Alternativ dazu können Sie auch in der Seitenleiste "Eigenschaften" in der Eigenschaftsgruppe für den Link auf die **Bearbeiten**-Schaltfläche des URL-Attributs klicken und die nötigen Änderungen vornehmen.

Löschen eines Hyperlink

Um einen Hyperlink zu löschen, wählen Sie ihn im Design aus und drücken Sie die **Entf**-Taste.

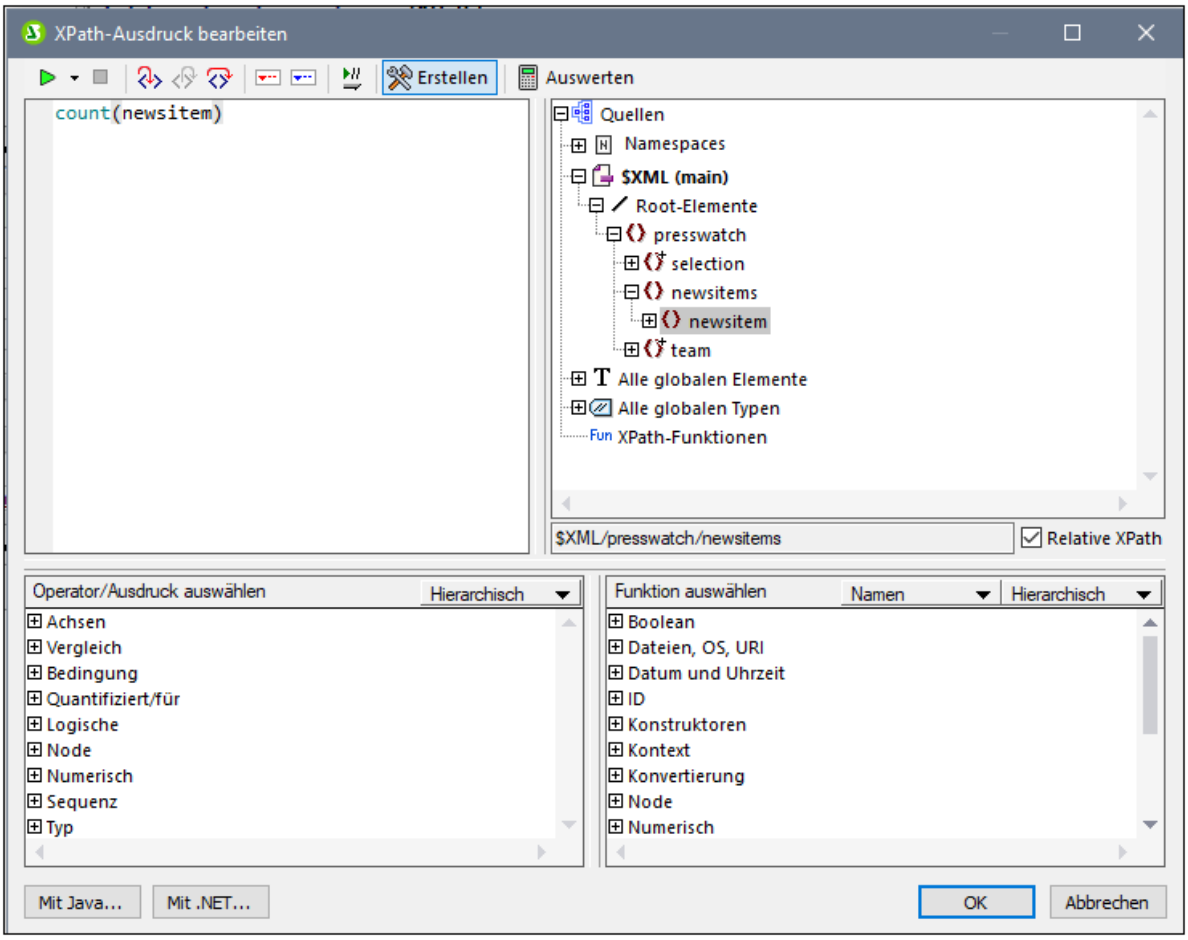
11.7.12 Bedingung, Ausgabe-basierte Bedingung

Mit Hilfe des Befehls **Bedingung** können Sie eine Bedingung am Cursoreinfügepunkt oder rund um eine Auswahl einfügen. Eine Bedingung besteht aus einer oder mehreren Verzweigungen, wobei jede Verzweigung bestimmte Verarbeitungsregeln enthält. Auf diese Art können Sie für verschiedene Verzweigungen unterschiedliche Verarbeitungsregeln definieren. Wenn z.B. der Inhalt eines Node der String `Stop` ist, kann dies in dieser Verzweigung überprüft werden und es kann festgelegt werden, dass der Inhalt der Node rot angezeigt werden soll; in einer zweiten Verzweigung können Sie überprüfen, ob der Inhalt eines Node der String `Go` ist. Ist dies der Fall, wird der Inhalt des Node grün angezeigt; in einer dritten Verzweigung kann definiert werden, dass der Inhalt eines Node, wenn es sich weder um den String `Stop` noch um den String `Go` handelt, schwarz angezeigt werden soll.

Um eine Bedingung einzufügen, gehen Sie folgendermaßen vor:

1. Platzieren Sie den Cursor an die gewünschte Stelle im Design oder wählen Sie die Komponente aus, um die die Bedingung eingefügt werden soll.
2. Wählen Sie den Menübefehl **Einfügen | Bedingung** oder klicken Sie auf die rechte Maustaste und wählen Sie den Kontextmenübefehl **Einfügen | Bedingung**.
3. Geben Sie im Dialogfeld XPath-Ausdruck bearbeiten[440], das nun angezeigt wird (*Abbildung unten*), den XPath-Ausdruck ein.

© 2019-2025 Altova GmbH



Der Kontext-Node für den erstellten Ausdruck erscheint im Schemastrukturfenster auf der linken Seite markiert. Sie können den XPath-Ausdruck direkt in das Textfeld eingeben oder auf einen Eintrag (in einem der drei Fensterbereiche) doppelklicken, um ihn einzufügen. Nodes, die aus der Schemastruktur im linken Bereich eingefügt werden, werden relativ zum Kontext-Node (wenn das Kontrollkästchen *Relativer XPath* aktiviert ist) oder als absoluter Ausdruck beginnend mit den Dokument-Node eingefügt (wenn das Kontrollkästchen *Relativer XPath* deaktiviert ist).

4. Klicken Sie auf **OK**, um den Ausdruck fertig zu stellen. Die Bedingung wird mit einer Verzweigung eingefügt. Diese Verzweigung enthält den soeben erstellten XPath-Ausdruck.

Bearbeiten von XPath-Ausdrücken von Verzweigungen

Um den XPath-Ausdruck einer Verzweigung zu bearbeiten, wählen Sie die Verzweigung in der Design-Ansicht aus. Wählen Sie anschließend in der Seitenleiste "Eigenschaften" den Befehl `Bedingungsverzweigung | when`. Klicken Sie auf die **Bearbeiten**-Schaltfläche ... für den XPath-Eintrag. Daraufhin wird das Dialogfeld "XPath-Ausdruck bearbeiten" (*Abbildung unten*) angezeigt, in dem Sie den XPath-Ausdruck bearbeiten können. Klicken Sie abschließend auf `OK`.

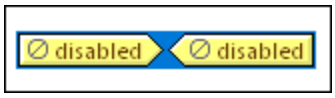
Hinzufügen von Verzweigungen, Ändern der Reihenfolge von Verzweigungen und Löschen von Verzweigungen

Um neue Verzweigungen hinzuzufügen, die Reihenfolge von Verzweigungen zu ändern und Verzweigungen zu löschen, rechtsklicken Sie auf die gewünschte Verzweigung und wählen Sie den gewünschten Befehl im

Kontextmenü aus.

11.7.13 Deaktiviert

Mit dem Befehl **Deaktiviert** wird eine Disabled-Komponente an der Cursorposition eingefügt (*Abbildung unten*). (Um die Disabled-Komponente rund um ausgewählten Inhalt einzufügen, verwenden Sie den Befehl **Einschließen in | Deaktiviert** 530).



Inhalt innerhalb einer Disabled-Komponente wird in der Ausgabe ignoriert. Sie können dadurch Inhalt, der ignoriert werden soll, zur Disabled-Komponente hinzufügen. Mit Hilfe der Disabled-Komponente kann Inhalt somit auskommentiert werden.

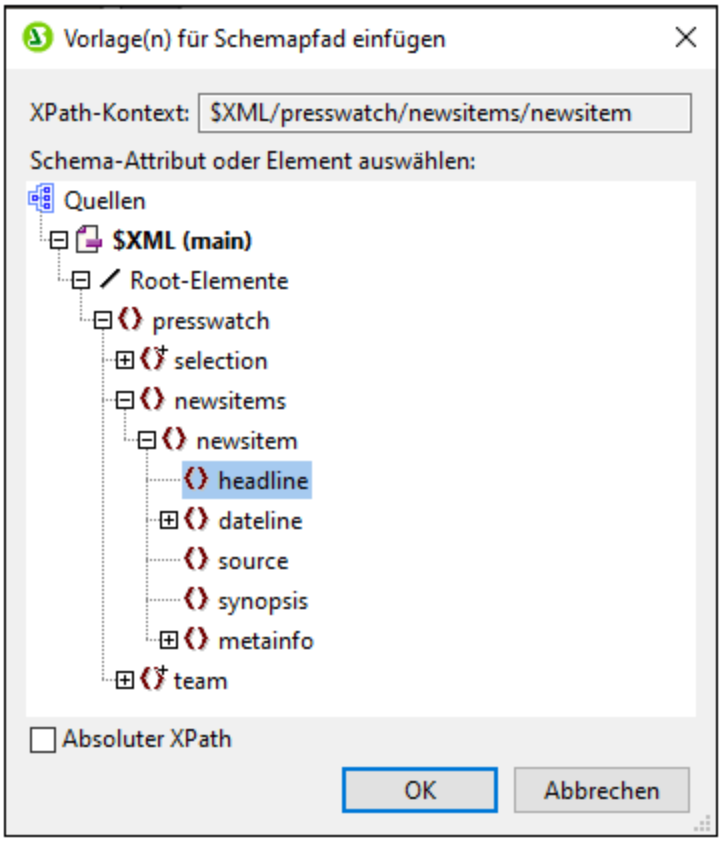
Um deaktivierten Inhalt wieder zu aktivieren, entfernen Sie einfach die Disabled-Tags aus dem Inhalt. Klicken Sie dazu mit der rechten Maustaste auf die Disabled-Komponente und wählen Sie den Befehl **Nur Tag entfernen**.

11.7.14 Vorlage

Mit dem Befehl **Vorlage** wird am Cursoreinfügepunkt eine leere Vorlage für den ausgewählten Schema-Node eingefügt. Gehen Sie dabei folgendermaßen vor:

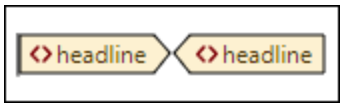
1. Platzieren Sie den Cursor an die Stelle im Design, an der Sie die Vorlage einfügen möchten.
2. Klicken Sie auf den Befehl **Einfügen | Vorlage**. Daraufhin wird das Dialogfeld "Vorlage einfügen" (*Abbildung unten*) angezeigt.

© 2019-2025 Altova GmbH



3. Das XPath-Kontextfeld enthält den Kontext-Node des Cursor-Einfügepunkts. Dieser Node bildet den Kontext-Node für die Vorlage, wenn diese erstellt wird. Wählen Sie im Dialogfeld den Node, für den Sie die Vorlage erstellen möchten. In der Abbildung oben wurde der Node `headline` ausgewählt. Daher wird eine Vorlage für den Node `headline` im Kontext des Elements `newsitem` erstellt.
4. Kicken Sie zum Fertigstellen auf **OK**.

Daraufhin wird eine leere Vorlage für den ausgewählten Node erstellt. In der Abbildung unten wurde eine leere Vorlage für den Node `headline` erstellt.



11.7.15 Benutzerdefinierte Vorlage

Mit dem Befehl **Benutzerdefinierte Vorlage** wird am Cursoreinfügepunkt eine leere Vorlage eingefügt, die einen vom Benutzer mit einem XPath-Ausdruck definierten Node auswählt. Gehen Sie beim Einfügen einer benutzerdefinierten Vorlage folgendermaßen vor:

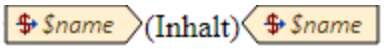
1. Platzieren Sie den Cursor an die Stelle im Design, an der die Vorlage eingefügt werden soll.
2. Klicken Sie auf den Befehl **Einfügen | Benutzerdefinierte Vorlage.** Daraufhin wird das Dialogfeld XPath-Ausdruck bearbeiten[440] angezeigt.
3. Geben Sie den XPath-Ausdruck zur Auswahl des gewünschten Node ein. Beachten Sie dabei einige Punkte: (i) Der XPath-Ausdruck wird im Kontext des Node ausgewertet, in dem die benutzerdefinierte Vorlage erstellt wird; (ii) Der XPath-Ausdruck kann jeden beliebigen Node überall im Dokument sowie in einem anderen XML-Dokument auswählen.
4. Klicken Sie nach Eingabe des XPath-Ausdrucks zum Fertigstellen auf **OK**.

Daraufhin wird eine leere benutzerdefinierte Vorlage für den Ziel-Node erstellt.

Nähere Informationen finden Sie im Abschnitt, SPS-Datei: Inhalt | Benutzerdefinierte Vorlagen[120].

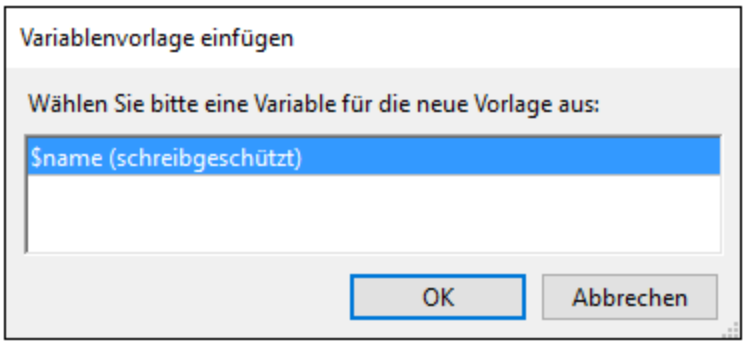
11.7.16 Variablenvorlage

Eine **Variablenvorlage** ist eine Vorlage für eine Variable, die standardmäßig den Inhalt der Variablen ausgibt. Diese Vorlage wird mit dem Befehl **Einfügen | Variablenvorlage** oder **Einschließen in | Variable** eingefügt. Daraufhin wird an der Cursorposition eine Vorlage für eine im SPS definierte Variable eingefügt. Die Variablenvorlage (*Abbildung unten*) enthält standardmäßig einen Platzhalter `Inhalt` zur Ausgabe des Inhalts der Variablen. Sie können je nach Wunsch zusätzlichen Inhalt (sowohl statischen als auch dynamischen) in die Variablenvorlage einfügen oder die Vorlage wie jede andere Vorlage bearbeiten. Eine Variablenvorlage wird durch ein Dollarsymbol im Start- und End-Tag gekennzeichnet.



Um eine Variablenvorlage einzufügen, gehen Sie folgendermaßen vor:

1. Platzieren Sie den Cursor an die Stelle im Design, an der die Vorlage eingefügt werden soll.
2. Klicken Sie auf die rechten Maustaste und wählen Sie den Befehl **Einfügen | Variablenvorlage**. Daraufhin wird das Dialogfeld "Variablenvorlage einfügen" angezeigt (*Abbildung unten*).



 © 2019-2025 Altova GmbH

3. Das Dialogfeld enthält eine Liste aller im SPS definierten benutzerdefinierten Parameter und Variablen[282]. Wählen Sie die Variable aus, für die Sie eine Variablenvorlage hinzufügen möchten.
4. Klicken Sie zum Fertigstellen auf **OK**.

11.7.17 Design Fragment

Wenn Sie die Maus über den Befehl "Design Fragment" positionieren, erscheint ein Untermenü mit allen Design Fragmenten, die derzeit im Design vorhanden sind. Wenn Sie in diesem Menü auf ein Design Fragment klicken, wird es am Cursoreinfügepunkt eingefügt.

11.7.18 Layout Container, Layout-Feld, Linie

Mit dem Befehl **Einfügen | Layout Container** können Sie einen Layout Container an jeder beliebigen Stelle im Design einfügen. Ein Layout-Feld oder eine Linie kann in einen Layout Container eingefügt werden. Diese beiden Befehle sind nur aktiv, wenn ein Layout Container ausgewählt ist.

Layout Container, Layout-Felder und Linien können auch über die entsprechenden Schaltflächen in der Symbolleiste "Design-Elemente einfügen"[462] eingefügt werden. Um eines dieser Elemente über die Symbolleisten-Schaltflächen einzufügen, müssen Sie zuerst auf die entsprechende Schaltfläche klicken und anschließend auf die gewünschte Stelle im Design.

Eine ausführliche Beschreibung von Layout-Modulen und ihrer Verwendung im Design finden Sie im Abschnitt Layout-Module[171].

11.7.19 Inhaltsverzeichnis

Wenn Sie den Mauszeiger über den Befehl **Inhaltsverzeichnis** platzieren, wird ein Untermenü angezeigt, das Befehle zum Einfügen verschiedener Befehle zur Erstellung einer Inhaltsverzeichnis-Vorlage, von Inhaltsverzeichnismarkierungen und einer Design-Dokumentstruktur für das Inhaltsverzeichnis enthält.

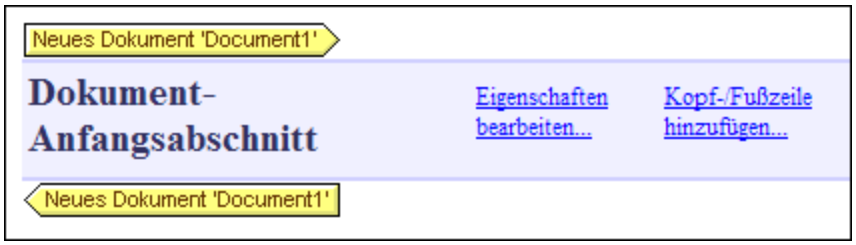
Dies ist die Liste der Befehle. Nähere Informationen zur Verwendung der Befehle finden Sie, wenn Sie auf die entsprechenden Links klicken.

- Inhaltsverzeichnis einfügen[301]
- Inhaltsverz.-Markierung[298]
- Inhaltsverz.-Markierung (Assistent)[293]
- Inhaltsverzeichnis-Referenz[304]
- Inhaltsverz.-Referenz | Eingabetext / Punktierte Linie / Seitenverweis[305]
- Hierarchische Nummerierung[305]
- Sequenzielle Nummerierung[305]
- Ebene[295]
- Ebenenreferenz[303]
- Vorlage dient als Ebene[295]

Anmerkung: Diese Befehle stehen auch im Kontextmenü zur Verfügung. Je nachdem, wo Sie mit der rechten Maustaste in das Design klicken, stehen andere Befehle im Kontextmenü zur Verfügung.

11.7.20 Neues Dokument

Der Befehl **Neues Dokument einfügen** fügt eine Vorlage für ein neues Dokument (*Abbildung unten*) am Cursoreinfügepunkt ein.



Die Vorlage für ein neues Dokument enthält einen leeren Dokument-Anfangsabschnitt. Sie können nun Inhalte in den Dokument-Anfangsabschnitt eingeben. Falls erwünscht, können mit Hilfe des Befehls **Einfügen | Seite/Spalte/Dokumentabschnitt einfügen** weitere Dokumentabschnitte an den Dokument-Anfangsabschnitt angehängt werden.

Eine Vorlage für ein neues Dokument erstellt ein neues Dokument in der Ausgabe. Daraus folgt, dass die Ausgabe aus mehreren Ausgabedokumenten besteht.

Eine detaillierte Beschreibung der Arbeit mit mehreren Ausgabedokumenten finden Sie im Abschnitt Mehrfach-Dokumentausgabe[248].

11.7.21 Benutzerdefiniertes Element

Wenn Sie die Maus über den Befehl **Einfügen | Benutzerdefiniertes Element** platzieren, wird ein Untermenü angezeigt, das Befehle zum Einfügen eines benutzerdefinierten Elements[123] oder eines benutzerdefinierten XML-Textblocks[124] enthält. Eine Beschreibung zur Verwendung dieser beiden Komponenten finden Sie im Abschnitt SPS-Datei: Inhalt | Benutzerdefinierte Elemente, XML-Textblöcke[123].

© 2019-2025 Altova GmbH

11.8 Menü "Einschließen in"

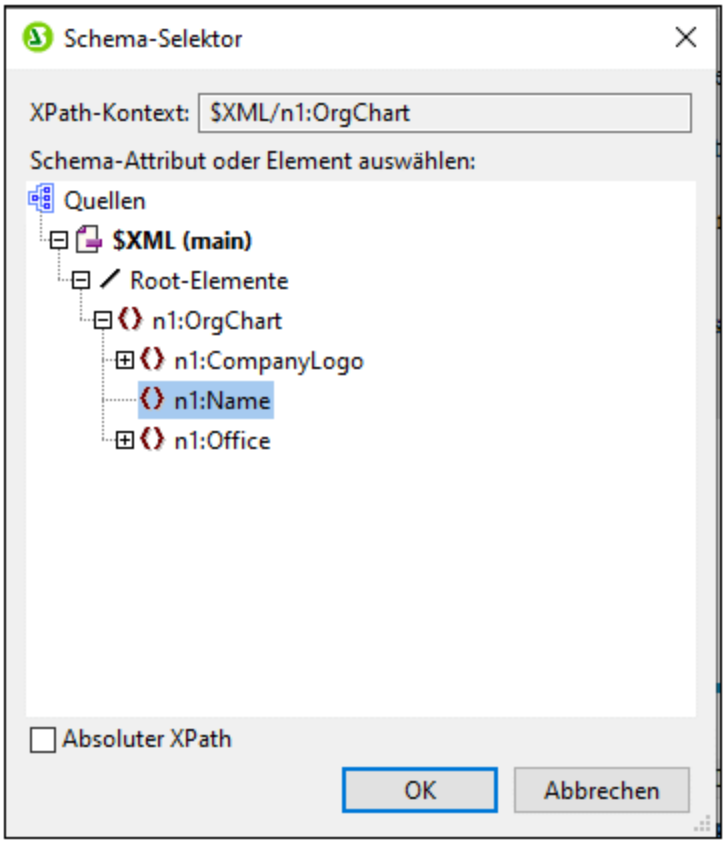
Das Menü **Einschließen in** enthält Befehle, über die Sie eine Auswahl im Design in die verschiedensten Design-Komponenten einschließen können. Einige dieser Befehle stehen in Form von Symbolleisten-Schaltflächen[457] zur Verfügung, über die Sie die Komponente in das Design einfügen können (die entsprechenden Befehle stehen auch im Menü **Einfügen**[505] zur Verfügung). Zusätzlich dazu stehen **Einschließen in**-Menübefehle auch über Kontextmenüs zur Verfügung, die angezeigt werden, wenn Sie mit der rechten Maustaste auf eine Auswahl klicken. Befehle in den Menüs und Kontextmenüs, die an dieser Stelle im SPS nicht zur Verfügung stehen, sind deaktiviert.

Anmerkung: Da die **Einschließen in**-Befehle zur Erstellung des SPS dienen, stehen Sie nur in der Design-Ansicht zur Verfügung.

11.8.1 Vorlage

Mit dem Befehl **Einschließen in | Vorlage** wird die ausgewählte Design-Komponente oder der ausgewählte Text in eine Vorlage für den ausgewählten Schema-Struktur-Node eingeschlossen. Gehen Sie dabei folgendermaßen vor:

1. Wählen Sie die gewünschte Design-Komponente oder den Text aus.
2. Wählen Sie den Befehl **Einschließen in | Vorlage** aus. Daraufhin wird das Dialogfeld "Schema-Selektor" angezeigt (*Abbildung unten*).



3. Das Feld "XPath-Kontext" enthält den Kontext-Node der Auswahl und dient als Kontext-Node der Vorlage, wenn diese erstellt ist. Wählen Sie den Node, für den Sie die Vorlage erstellen möchten, aus. In der Abbildung oben ist der Node `n1:Name` als Node ausgewählt, für den die Vorlage erstellt wird.
4. Klicken Sie zum Fertigstellen auf **OK**.

Rund um die Auswahl wird eine Vorlage für den ausgewählten Node erstellt.

11.8.2 Benutzerdefinierte Vorlage

Mit dem Befehl **Einschließen in | Benutzerdefinierte Vorlage** wird die Auswahl in eine Vorlage für einen Node eingeschlossen, den der Benutzer mittels eines XPath-Ausdrucks definiert. Um eine benutzerdefinierte Vorlage einzufügen, gehen Sie folgendermaßen vor:

1. Wählen Sie die gewünschte Design-Komponente aus.
2. Wählen Sie den Befehl **Einschließen in | Benutzerdefinierte Vorlage** aus. Daraufhin wird das Dialogfeld XPath-Ausdruck bearbeiten 440 angezeigt
3. Geben Sie den XPath-Ausdruck zur Auswahl des gewünschten Node ein. Beachten Sie dabei einige Punkte: (i) Der XPath-Ausdruck wird im Kontext des Node ausgewertet, in dem die benutzerdefinierte Vorlage erstellt wird; (ii) Der XPath-Ausdruck kann jeden beliebigen Node überall im Dokument und in einem anderen XML-Dokument auswählen.
2. Klicken Sie nach Eingabe des XPath-Ausdrucks auf **OK.**

Daraufhin wird rund um die Auswahl eine benutzerdefinierte Vorlage für den Ziel-Node erstellt.

Nähere Informationen dazu finden Sie im Abschnitt SPS-Datei: Struktur | Vorlagen und Design-Fragmente | Benutzerdefinierte Vorlagen 235.

11.8.3 Variablenvorlagen

Mit dem Befehl **Einschließen in | Variablenvorlage** wird die Auswahl in eine Vorlage für eine Variable eingeschlossen, die im SPS Design definiert wurde.

1. Wählen Sie die Komponente im Design aus, die in eine Variablenvorlage eingeschlossen werden soll.
2. Klicken Sie auf den Befehl **Einschließen in | Variablenvorlage**. Daraufhin wird das Dialogfeld In Variablenvorlage einschließen 238 angezeigt.
3. Wählen Sie aus der Liste im Dialog die gewünschte Variable aus.
4. Kicken Sie zum Fertigstellen auf **OK**.

Daraufhin wird die Auswahl in eine Variablenvorlage eingeschlossen.

Nähere Informationen dazu finden Sie im Abschnitt SPS-Datei: Inhalt | Benutzerdefinierte Vorlagen und Design-Fragmente | Variablenvorlagen 238.

© 2019-2025 Altova GmbH

11.8.4 Absatz, Spezieller Absatz

Mit dem Befehl **Absatz** wird ein HTML-Absatzelement `<p>` rund um die ausgewählte Komponente eingefügt. Eine Komponente gilt in diesem Zusammenhang als ausgewählt, wenn der gesamte Node ausgewählt ist (durch Klicken auf einen seiner Tags) oder wenn statischer Text ausgewählt ist. Wenn der Cursor innerhalb von statischen Text platziert wird, wird das Absatzelement (Start- und End-Tags) an dieser Stelle eingefügt. Ein Absatz kann auch über die Schaltfläche **Absatz einfügen** in der Symbolleiste "Design-Element einfügen" 462 eingefügt werden.

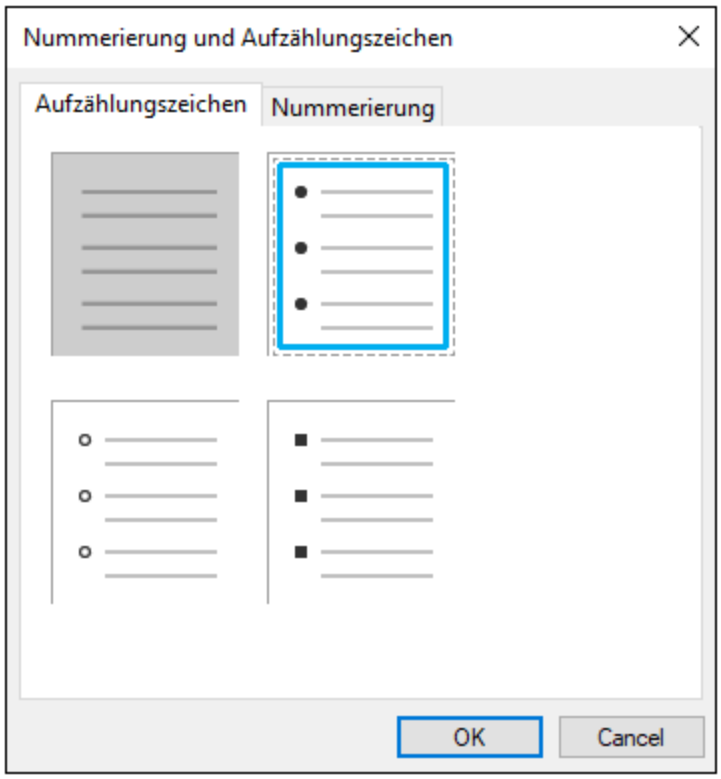
Mit dem Befehl **Speziellen Absatz einfügen** können Sie dem ausgewählten Node ein vordefiniertes Format zuweisen. Das verfügbare vordefinierte Format kann auch über die Auswahlliste in der Symbolleiste ausgewählt werden.

Jede Absatzart hat eigene Formatierungsfeatures, die nutzbringend eingesetzt werden können. Beachten Sie, dass mit dem Formattyp `pre` im Ausgabedokument Zeilenumbrüche ausgegeben werden können, anstatt dass diese zu Leerzeichen normalisiert werden.

11.8.5 Nummerierung und Aufzählungszeichen

Mit dem Befehl **Einschließen in | Nummerierung und Aufzählungszeichen** wird rund um die Auswahl eine statische Liste mit Listeneinträgen erstellt. Wenn die Auswahl einen CR-LF, Carriage-Return und/oder Linefeed enthält (eingefügt durch Drücken der **Eingabetaste**), werden für jedes Textfragment separate (durch CR-LF getrennte) Listeneinträge erstellt.

Bei Auswahl dieses Befehls wird das Dialogfeld "Nummerierung und Aufzählungszeichen" angezeigt (*Abbildung unten*).



Wählen Sie die gewünschte Art von Aufzählungszeichen aus und klicken Sie auf **OK**. Daraufhin wird eine Liste erstellt. Die Anzahl der Listeneinträge entspricht der Anzahl der CR-LF-Zeichen (Wagenrücklauf und/oder Zeilenschaltung) in der Auswahl. Durch Drücken der **Eingabetaste** können Sie mehr Einträge zur Liste hinzufügen.

Anmerkung: Sie können dasselbe Ergebnis auch durch Platzieren des Cursors an die gewünschte Stelle im Design und Klicken auf die Schaltfläche "Nummerierung und Aufzählungszeichen" in der Symbolleiste "Design-Element einfügen" 462 erstellen.

11.8.6 Lesezeichen und Hyperlinks

Die Befehle **Einschließen in | Lesezeichen** und **Einschließen in | Hyperlink** sind aktiviert, wenn im SPS-Design Text oder eine Komponente ausgewählt ist. Mit Hilfe dieser Befehle kann rund um die Auswahl ein Lesezeichen bzw. ein Hyperlink erstellt werden. Nähere Informationen zum Erstellen und zur Verwendung von Lesezeichen und Hyperlinks finden Sie im Abschnitt Komplexe Funktionen | Inhaltsverzeichnis, Referenzieren, Lesezeichen 320.

11.8.7 Bedingung, ausgabebasierte Bedingung

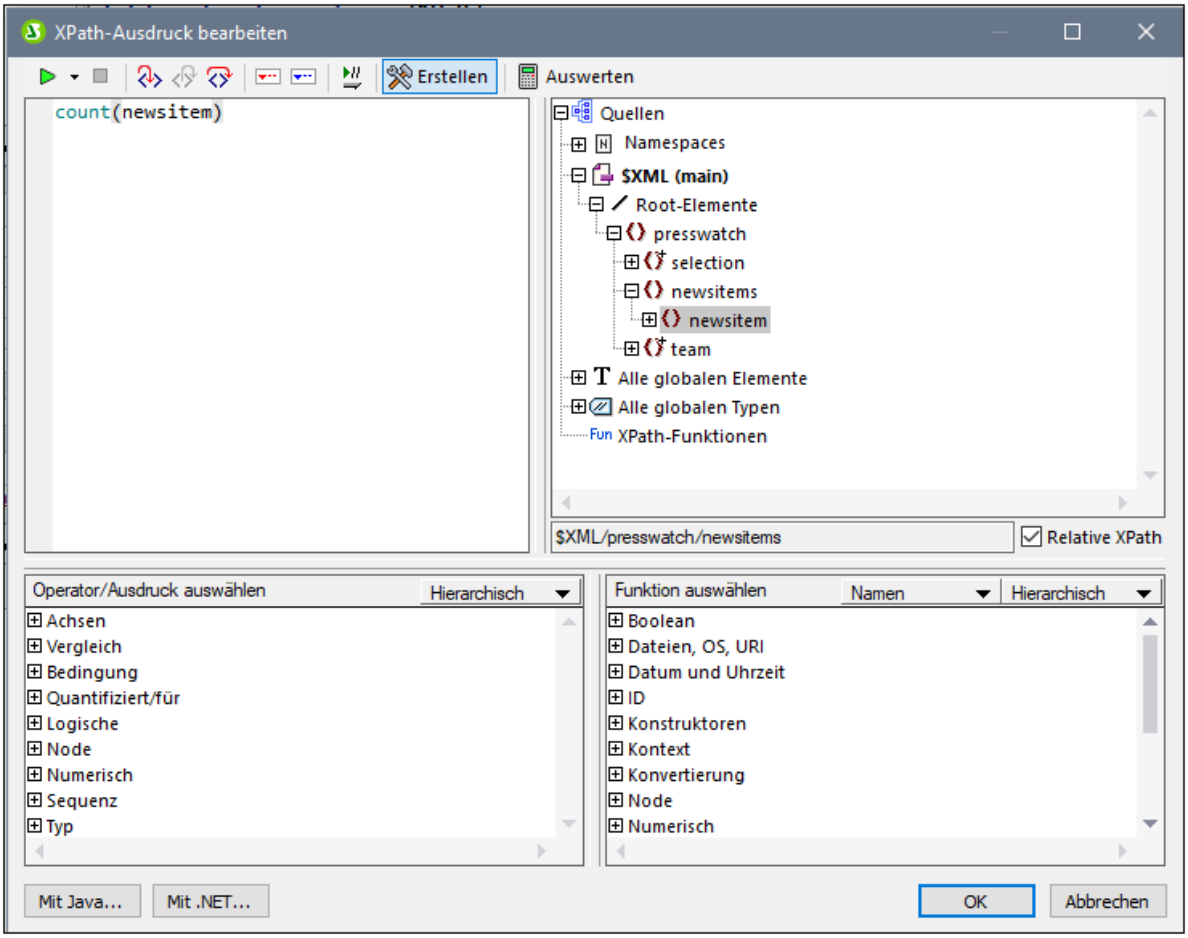
Mit Hilfe des Befehls **Bedingung** können Sie eine Bedingung am Cursoreinfügepunkt oder rund um eine Auswahl einfügen. Eine Bedingung besteht aus einer oder mehreren Verzweigungen, wobei jede Verzweigung bestimmte Verarbeitungsregeln enthält. Auf diese Art können Sie für verschiedene Verzweigungen unterschiedliche Verarbeitungsregeln definieren. Wenn z.B. der Inhalt eines Node der String `Stop` ist, kann dies

 © 2019-2025 Altova GmbH

in dieser Verzweigung überprüft werden und es kann festgelegt werden, dass der Inhalt der Node rot angezeigt werden soll; in einer zweiten Verzweigung können Sie überprüfen, ob der Inhalt eines Node der String `Go` ist. Ist dies der Fall, wird der Inhalt des Node grün angezeigt; in einer dritten Verzweigung kann definiert werden, dass der Inhalt eines Node, wenn es sich weder um den String `Stop` noch um den String `Go` handelt, schwarz angezeigt werden soll.

Um eine Bedingung einzufügen, gehen Sie folgendermaßen vor:

1. Platzieren Sie den Cursor an die gewünschte Stelle im Design oder wählen Sie die Komponente aus, um die die Bedingung eingefügt werden soll.
2. Wählen Sie den Menübefehl **Einfügen | Bedingung** oder klicken Sie auf die rechte Maustaste und wählen Sie den Kontextmenübefehl **Einfügen | Bedingung**.
3. Geben Sie im Dialogfeld XPath-Ausdruck bearbeiten[440], das nun angezeigt wird (*Abbildung unten*), den XPath-Ausdruck ein.



Der Kontext-Node für den erstellten Ausdruck erscheint im Schemastrukturfenster auf der linken Seite markiert. Sie können den XPath-Ausdruck direkt in das Textfeld eingeben oder auf einen Eintrag (in einem der drei Fensterbereiche) doppelklicken, um ihn einzufügen. Nodes, die aus der Schemastruktur im linken Bereich eingefügt werden, werden relativ zum Kontext-Node (wenn das Kontrollkästchen *Relativer XPath* aktiviert ist) oder als absoluter Ausdruck beginnend mit den Dokument-Node eingefügt (wenn das Kontrollkästchen *Relativer XPath* deaktiviert ist).

4. Klicken Sie auf **OK**, um den Ausdruck fertig zu stellen. Die Bedingung wird mit einer Verzweigung

eingefügt. Diese Verzweigung enthält den soeben erstellten XPath-Ausdruck.

Bearbeiten von XPath-Ausdrücken von Verzweigungen

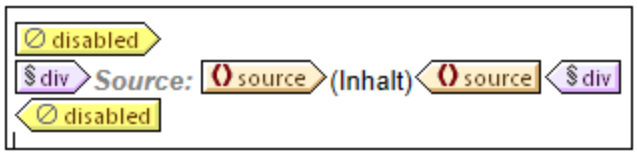
Um den XPath-Ausdruck einer Verzweigung zu bearbeiten, wählen Sie die Verzweigung in der Design-Ansicht aus. Wählen Sie anschließend in der Seitenleiste "Eigenschaften" den Befehl `Bedingungsverzweigung | when`. Klicken Sie auf die **Bearbeiten**-Schaltfläche ... für den XPath-Eintrag. Daraufhin wird das Dialogfeld "XPath-Ausdruck bearbeiten" (*Abbildung unten*) angezeigt, in dem Sie den XPath-Ausdruck bearbeiten können. Klicken Sie abschließend auf `OK`.

Hinzufügen von Verzweigungen, Ändern der Reihenfolge von Verzweigungen und Löschen von Verzweigungen

Um neue Verzweigungen hinzuzufügen, die Reihenfolge von Verzweigungen zu ändern und Verzweigungen zu löschen, rechtsklicken Sie auf die gewünschte Verzweigung und wählen Sie den gewünschten Befehl im Kontextmenü aus.

11.8.8 Deaktiviert

Mit dem Befehl **Deaktiviert** wird ausgewählter Inhalt (einschließlich Design-Komponenten) in eine Disabled-Komponente eingeschlossen (*siehe Abbildung unten*).



Inhalt innerhalb einer Disabled-Komponente wird in der Ausgabe ignoriert. Sie können dadurch Inhalt, der ignoriert werden soll, in eine Disabled-Komponente einschließen.

Um deaktivierten Inhalt wieder zu aktivieren, entfernen Sie einfach die Disabled-Tags aus dem Inhalt. Klicken Sie dazu mit der rechten Maustaste auf die Disabled-Komponente und wählen Sie den Befehl **Nur Tag entfernen**.

11.8.9 Inhaltsverz.-Markierungen und Inhaltsverz.-Ebenen

Wenn eine Komponente im Design ausgewählt ist, kann sie in eine oder mehrere Inhaltsverzeichnis-Komponenten eingeschlossen werden. Im Folgenden sehen Sie die Liste der Inhaltsverzeichnis-Befehle. Nähere Informationen zu ihrer Verwendung finden Sie unter den entsprechenden Links.

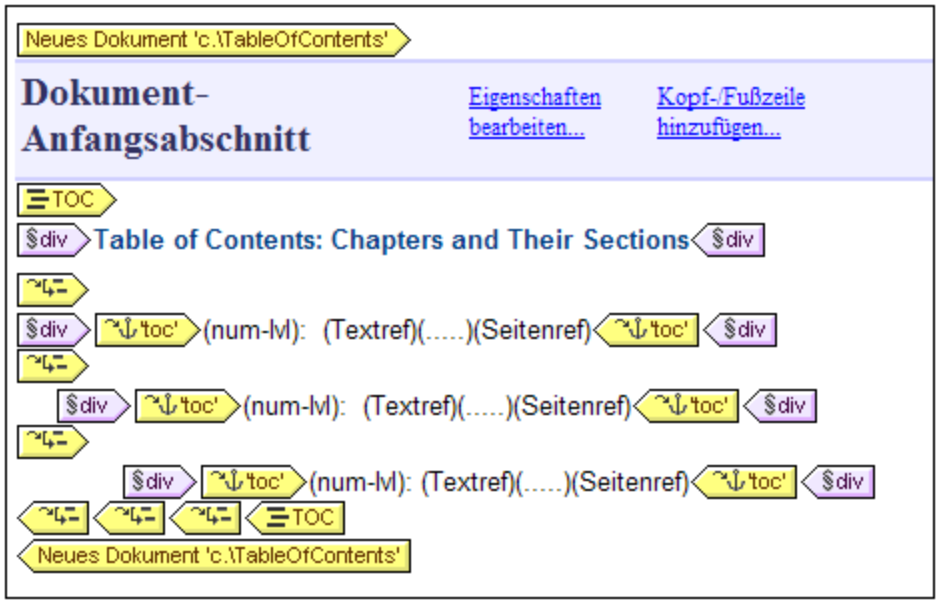
- Inhaltsverz.-Markierung 298
- Inhaltsverz.-Markierung (Assistent) 293
- Ebene 295
- Ebenenreferenz 303

© 2019-2025 Altova GmbH

Anmerkung: Diese Befehle stehen auch, je nachdem, an welche Stelle im Design Sie klicken, in Kontextmenüs zur Verfügung.

11.8.10 Neues Dokument

Der Befehl **Einschließen in Neues Dokument** schließt die aktuelle Markierung in eine Vorlage für ein neues Dokument (*Abbildung unten*) ein.



Die Vorlage für ein neues Dokument enthält einen leeren Dokument-Anfangsabschnitt. Sie können nun Inhalte in den Dokument-Anfangsabschnitt eingeben. Falls erwünscht, können mit Hilfe des Befehls **Einfügen | Seite/Spalte/Dokumentabschnitt einfügen** weitere Dokumentabschnitte an den Dokument-Anfangsabschnitt angehängt werden.

Eine Vorlage für ein neues Dokument erstellt ein neues Dokument in der Ausgabe. Daraus folgt, dass die Ausgabe aus mehreren Ausgabedokumenten besteht.

Eine detaillierte Beschreibung der Arbeit mit mehreren Ausgabedokumenten finden Sie im Abschnitt Mehrfach-Dokumentausgabe 248.

11.8.11 Benutzerdefiniertes Element

Mit dem Befehl **Einschließen in | Benutzerdefiniertes Element** wird rund um die Auswahl im Design ein benutzerdefiniertes Element 123 erstellt. Eine Anleitung zur Verwendung von benutzerdefinierten Elementen finden Sie im Abschnitt SPS-Datei: Inhalt | Benutzerdefinierte Elemente 123.

11.9 Menü "Tabelle"

Im Menü **Tabelle** stehen Ihnen Befehle zum Einfügen einer statischen oder dynamischen Tabelle und zum Ändern der Struktur von statischen und dynamischen Tabellen zur Verfügung. Sie können die Tabellenstruktur durch Anhängen, Einfügen, Löschen, Zusammenfügen und Aufteilen von Zeilen und Spalten bearbeiten. Die Eigenschaften der Tabelle sowie die einzelnen Spalten, Zeilen und Zellen werden mit Hilfe von CSS-Stilen[136] und HTML-Eigenschaften für Tabellen und ihre Unterkomponenten[136] definiert.

Die Tabellenbefehle stehen im Menü "Tabelle" zur Verfügung (*siehe Liste unten*) und in Form von Symbolen in der Symbolleiste "Tabelle"[460]. Welche Tabellenbefehle jeweils verfügbar sind, hängt von der Position des Cursors ab. Eine Tabelle kann überall im SPS durch Auswahl des Befehls Tabelle einfügen[532] eingefügt werden. Um die Tabellenstruktur zu bearbeiten, platzieren Sie den Cursor in die entsprechende Zelle, Spalte oder Zeile und wählen Sie den gewünschten Bearbeitungsbefehl aus. Um eine Formatierungseigenschaft zu bearbeiten, setzen Sie den Cursor in die entsprechende Zelle, Spalte, Zeile oder Tabelle und definieren Sie in der Seitenleiste "Stil"[136] und/oder der Seitenleiste "Eigenschaften"[136] die gewünschte Eigenschaft für diese Tabellenkomponente.

Die folgenden Befehle stehen im Menü "Tabelle" zur Verfügung:

- Tabelle einfügen, Tabelle löschen[532]
- Kopf- / Fußzeilen für Tabellen einfügen[533]
- Zeile/Spalte anhängen/einfügen[533]
- Zeile, Spalte löschen[534]
- Zelle links, rechts, unten, oben verbinden[534]
- Zelle horizontal, vertikal teilen[535]
- Zellenumrandung anzeigen / Tabellen-Markup-Code anzeigen[535]
- Tabelleneigenschaften[535]
- Vertikale Ausrichtung des Zelleninhalts[537]

Kopf- und Fußzeile

Wenn Sie eine dynamische Tabelle erstellen, können Sie festlegen, ob eine Kopf- und/oder Fußzeile hinzugefügt werden soll. (Fußzeilen sind nur dann zulässig, wenn die Tabelle von oben nach unten wächst). Sie können in einer statischen Tabelle eine Kopf- und Fußzeile erstellen, indem Sie manuell eine oberste bzw. unterste Zeile einfügen. Die Struktur von Kopf- und Fußzeilen kann sowohl in dynamischen als auch statischen Tabellen durch Aufteilen und Zusammenfügen von Zellen geändert werden.

Navigieren in Tabellen
Verwenden Sie zum Navigieren in Tabellenzellen die Tabulator- und die Pfeiltasten.

Hinzufügen von Zelleninhalt
Als Inhalt einer Zelle kann jede beliebige SPS Komponentenart eingefügt werden. Die Komponente könnte mit Hilfe der Standard-Formatierungstools formatiert werden.

11.9.1 Tabelle einfügen, Tabelle löschen

Mit Hilfe des Befehls **Tabelle einfügen** wird eine leere Tabelle auf dem Design-Register eingefügt. Wenn Sie diesen Befehl auswählen, wird ein Dialogfeld geöffnet, in dem Sie auswählen können, ob Sie eine statische oder dynamische Tabelle erstellen möchten.

© 2019-2025 Altova GmbH

- Wenn Sie eine statische Tabelle erstellen, werden Sie in einem Dialogfeld nach der Größe der Tabelle gefragt (wie viele Zeilen und Spalten sie enthalten soll).
- Wenn Sie eine dynamische Tabelle erstellen, wird das Dialogfeld "XPath-Selektor" angezeigt, in dem Sie den Node auswählen können, der als dynamische Tabelle erstellt werden soll. Wenn Sie auf **OK** klicken, wird das Dialogfeld "Dynamische Tabelle" angezeigt, in dem Sie die Child Nodes auswählen können, die als Felder der einzelnen Tabelleneinträge angezeigt werden sollen. Nähere Informationen dazu finden Sie unter Erstellen dynamischer Tabellen 129

Sie können die Struktur der Tabelle anschließend durch Anhängen, Einfügen und Löschen von Zeilen und/oder Spalten ändern.

Mit Hilfe des Befehls **Tabelle löschen** können Sie die statische oder dynamische Tabelle, in der sich der Cursor befindet, löschen.

11.9.2 Kopf-/Fußzeilen für Tabelle einfügen

Tabellenüberschriftszeilen können in Form einer Kopfzeile (oberhalb des Tabellenkörpers) oder als Überschriftsspalte (links vom Tabellenkörper, obwohl eine Überschriftsspalte im Markup-Code auch in den Tabellenkörper platziert werden kann) angezeigt werden. Auf ähnliche Weise können auch Tabellenfußzeilen als Fußzeile(unterhalb des Tabellenkörpers) oder als Fußzeilenspalte (rechts vom Tabellenkörper - allerdings kann eine Fußzeile im Markup-Code auch in den Tabellenkörper platzier werden) angezeigt werden.

Hinweis: Da Tabellenkopfzeilen in der HTML-Ausgabe in `th` Elemente eingeschlossen werden, werden sie fett angezeigt (da das `th` Element automatisch fett formatiert wird).

Mit den Befehlen "Tabellenkopfzeile hinzufügen" und "Tabellenfußzeile hinzufügen" fügen Sie Tabellenkopf- und Fußzeilen in Form von Spalten und Zeilen hinzu. Hierfür stehen die folgenden Befehle zur Verfügung:

Spalte für Tabellenüberschrift hinzufügen :	Fügt eine Tabellenüberschriftsspalte links vom Tabellenkörper hinzu.
Spalte für Tabellenfußzeile hinzufügen :	Fügt rechts vom Tabellenkörper eine Spalte für die Tabellenfußzeile hinzu.
Tabellenkopfzeile hinzufügen :	Fügt oberhalb des Tabellenkörpers eine Kopfzeile hinzu.
Tabellenfußzeile hinzufügen:	Fügt unterhalb des Tabellenkörpers eine Fußzeile hinzu.

11.9.3 Zeile/Spalten anhängen, einfügen

Mit Hilfe des Befehls **Zeile anhängen** können Sie eine Zeile an die statische oder dynamische Tabelle, in der sich der Cursor befindet, anhängen.

Mit Hilfe des Befehls **Zeile einfügen** wird eine Zeile oberhalb der Zeile, in der sich der Cursor befindet, eingefügt. Dieser Befehl ist sowohl für statische als auch dynamische Tabellen verfügbar.

Mit Hilfe des Befehls **Spalte anhängen** können Sie eine Spalte an die statische oder dynamische Tabelle, in der sich der Cursor befindet, anhängen.

Mit Hilfe des Befehls **Spalte einfügen** wird eine Spalte links von der Spalte, in der sich der Cursor befindet, eingefügt. Dieser Befehl ist sowohl für statische als auch dynamische Tabellen verfügbar.

11.9.4 Zeile, Spalte löschen

Mit Hilfe des Befehls **Zeile löschen** wird die Zeile, in der sich der Cursor befindet, gelöscht. Dieser Befehl ist sowohl für statische als auch dynamische Tabellen verfügbar.

Mit Hilfe des Befehls **Spalte löschen** wird die Spalte, in der sich der Cursor befindet, gelöscht. Dieser Befehl ist sowohl für statische als auch dynamische Tabellen verfügbar.

11.9.5 Zelle links, rechts, unten, oben verbinden

Mit Hilfe des Befehls **Zelle links verbinden** wird die Zelle, in der sich der Cursor befindet, mit der links daneben liegenden Zelle verbunden. Der Inhalt der beiden Zellen wird in der neuen Zelle zusammengefügt. Alle Eigenschaftswerte der linken Zelle gehen auf die neue Zelle über. Dieser Befehl steht sowohl bei statischen als auch dynamischen Tabellen zur Verfügung.

Mit Hilfe des Befehls **Zelle rechts verbinden** wird die Zelle, in der sich der Cursor befindet, mit der rechts daneben liegenden Zelle verbunden. Der Inhalt der beiden Zellen wird in der neuen Zelle zusammengefügt. Alle Eigenschaftswerte der linken Zelle gehen auf die neue Zelle über. Dieser Befehl steht sowohl bei statischen als auch dynamischen Tabellen zur Verfügung.

Mit Hilfe des Befehls **Zelle unten verbinden** wird die Zelle, in der sich der Cursor befindet, mit der darunter liegenden Zelle verbunden. Der Inhalt der beiden Zellen wird in der neuen Zelle zusammengefügt. Alle Eigenschaftswerte der oberen Zelle gehen auf die neue Zelle über. Dieser Befehl steht sowohl bei statischen als auch dynamischen Tabellen zur Verfügung.

Mit Hilfe des Befehls **Zelle oben verbinden** wird die Zelle, in der sich der Cursor befindet, mit der darüber liegenden Zelle verbunden. Der Inhalt der beiden Zellen wird in der neuen Zelle zusammengefügt. Alle Eigenschaftswerte der oberen Zelle gehen auf die neue Zelle über. Dieser Befehl steht sowohl bei statischen als auch dynamischen Tabellen zur Verfügung.

© 2019-2025 Altova GmbH

11.9.6 Zelle horizontal oder vertikal teilen

Mit Hilfe des Befehls **Zelle horizontal teilen** wird rechts von der Zelle, in der sich der Cursor befindet, eine neue Zelle erstellt. Der Inhalt der ursprünglichen Zelle bleibt in der ursprünglichen Zelle. Alle Eigenschaften der ursprünglichen Zelle werden für die neue Zelle übernommen. Dieser Befehl steht sowohl bei statischen als auch dynamischen Tabellen zur Verfügung.

Mit Hilfe des Befehls **Zelle vertikal teilen** wird unterhalb der Zelle, in der sich der Cursor befindet, eine neue Zelle erstellt. Der Inhalt der ursprünglichen Zelle bleibt in der oberen Zelle. Alle Eigenschaften der ursprünglichen Zelle werden für die neue Zelle übernommen. Dieser Befehl steht sowohl bei statischen als auch dynamischen Tabellen zur Verfügung.

11.9.7 Zellumrandung / Tabellen-Markup-Code anzeigen

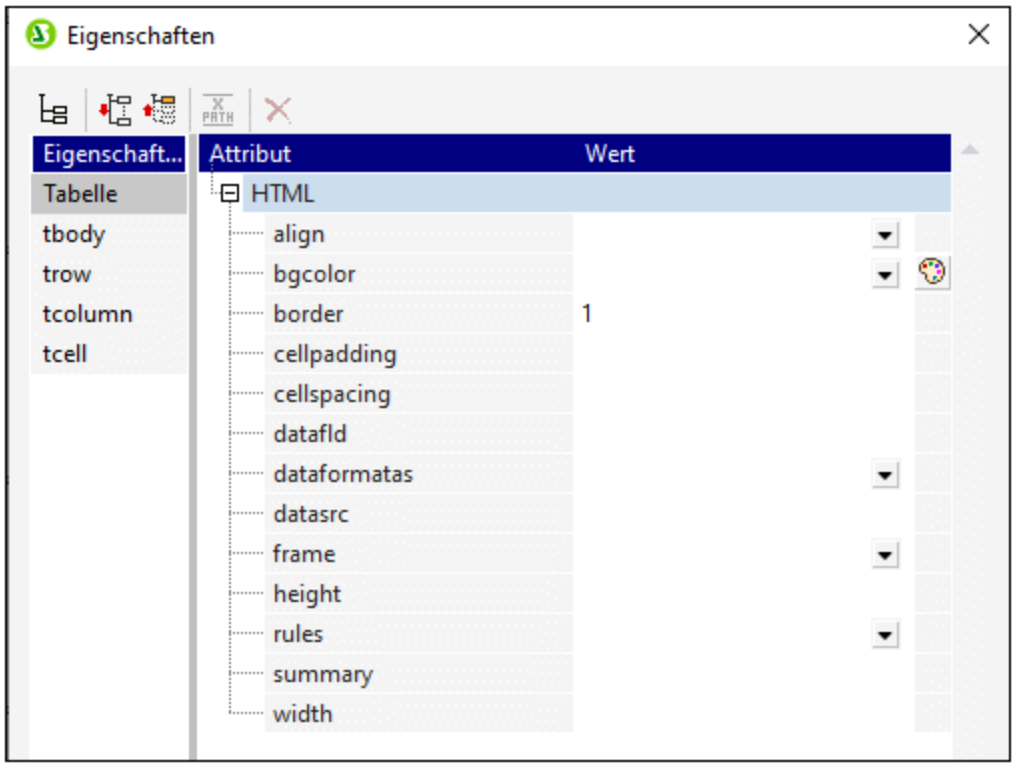
Mit Hilfe der Befehle **Zellumrandung anzeigen** und **Tabellen-Markup-Code anzeigen** können Sie die Umrandung von Tabellen bzw. den Markup-Code von Tabellenspalten und -zeilen anzeigen. Wenn diese beiden Optionen aktiviert sind, kommt die Struktur der Tabelle deutlicher zu Tage. Wenn Sie diese Optionen deaktivieren, wird die Tabelle exakter visualisiert.

Mit dem Befehl **Zellumrandung anzeigen** können Sie bei Tabellen, die einen Zellumrandungswert 0 haben, die Tabellenumrandungen ein und ausblenden.

Mit dem Befehl **Tabellen-Markup-Code anzeigen** können die blauen Spalten- und Zeilenmarkierungen ein- und ausgeblendet werden.

11.9.8 Tabelleneigenschaften

Der Befehl **Tabelleneigenschaften** ist aktiv, wenn der Cursor sich innerhalb einer statischen oder dynamischen Tabelle befindet[126]. Bei Aufruf des Befehls wird die Seitenleiste "Eigenschaften" angezeigt, wobei die Komponente "Tabelle" ausgewählt ist (*Abbildung unten*).



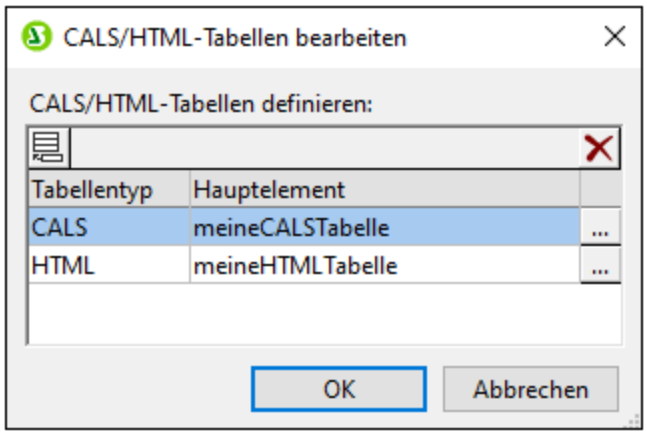
Sie können Die Eigenschaften der Tabelle nun bearbeiten. Klicken Sie anschließend auf OK.

11.9.9 CALS/HTML-Tabellen bearbeiten

Wenn das zugrunde liegende Schema des SPS ein Element mit einer Substruktur hat, die dem CALS- oder dem HTML-Tabellenmodell entspricht, können Sie dieses XML-Element in den Ausgaben automatisch als Tabelle generieren. Dazu müssen Sie angeben, welches SPS-Schema-Element welchem CALS- oder HTML-Tabellenelement entspricht. Der Tabellen-Markup-Code in den Ausgabeformaten wird anschließend direkt vom XML-Dokument abgeleitet.

Bei Auswahl des Befehls **CALS/HTML-Tabellen bearbeiten** wird das Dialogfeld "CALS/HTML-Tabellen" (*Abbildung unten*) aufgerufen. Wählen Sie im Dialogfeld den Namen des SPS-Schema-Elements, das einem CALS- oder HTML-Tabellenelement entspricht, aus. Sie können im Dialogfeld grundlegende Tabellenformatierungsstile definieren und später im Design Stile hinzufügen.

© 2019-2025 Altova GmbH



Nähere Informationen zu CALS/HTML-Tabellen finden Sie im Abschnitt Tabellen[126].

11.9.10 Vertikale Ausrichtung des Zelleninhalts

In der Symbolleiste "Tabelle" stehen Befehle zur Definition der vertikalen Ausrichtung von Zelleninhalt zur Verfügung. Setzen Sie den Cursor an eine beliebige Stelle in die Zelle und klicken Sie auf das gewünschte Symbol.

Bündig oben richtet den Zelleninhalt vertikal am oberen Rand der Zelle aus.

Vertikal zentrieren richtet den Zelleninhalt vertikal in der Mitte der Zelle aus

Bündig unten richtet den Zelleninhalt vertikal am unteren Rand der Zelle aus.

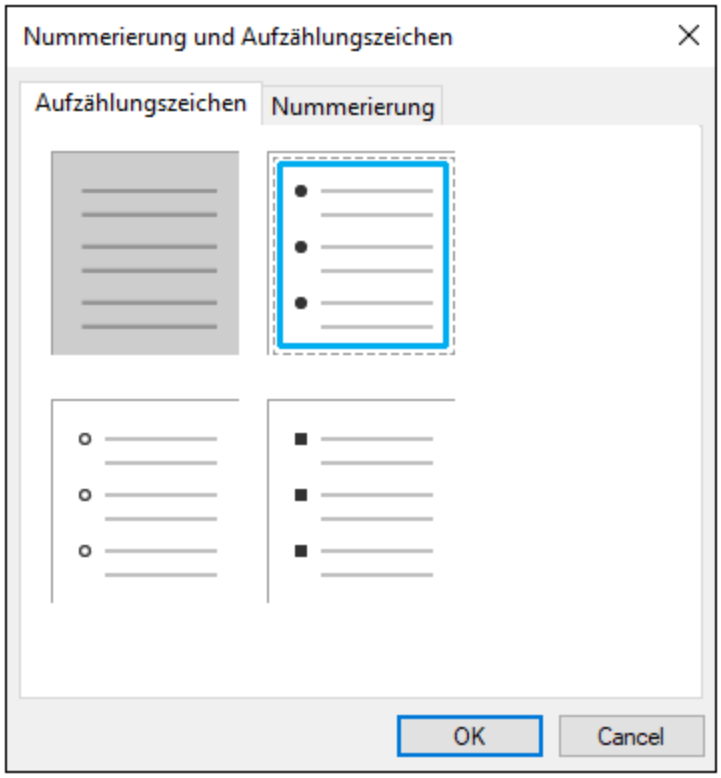
11.10 Menü "Eigenschaften"

Das Menü **Eigenschaften** enthält Befehle zum Einfügen von Listen und Definieren von Datentyp-Formaten für die Funktion Werteformatierung[333]. Die Beschreibung der Befehle ist in die folgenden Unterabschnitte gegliedert:

- der Befehl Nummerierung und Aufzählungszeichen[538] zum Einfügen von Listen.
- der Befehl Vordefinierte Formatierungsstrings[538] zum Definieren von numerischen Datentypformaten für ein bestimmtes SPS.

11.10.1 Nummerierung und Aufzählungszeichen bearbeiten

Mit Hilfe des Befehls **Nummerierung und Aufzählungszeichen bearbeiten** können Sie eine Liste an der Cursorposition einfügen. Wenn Sie auf den Befehl klicken, wird das Dialogfeld "Nummerierung und Aufzählungszeichen" (Abbildung unten) angezeigt, in welchem Sie den Listenstil auswählen können; bei nummerierten Listen können Sie auch die Anfangsnummer definieren.



11.10.2 Vordefinierte Wertformatierungsstrings

Jedem Platzhalter `(Inhalt)`, jedem Eingabefeld, jeder automatischen Berechnung vom Datentyp `numeric`, `date`, `time`, `dateTime` oder `duration` kann über das Dialogfeld Werteformatierung[333] ein benutzerdefiniertes Format zugewiesen werden. Im Dialogfeld "Werteformatierung" können Sie ein Format

© 2019-2025 Altova GmbH

entweder direkt erstellen oder es aus einer Dropdown-Liste vordefinierter Formate auswählen. Diese Liste besteht aus zwei Arten von vordefinierten Formaten:

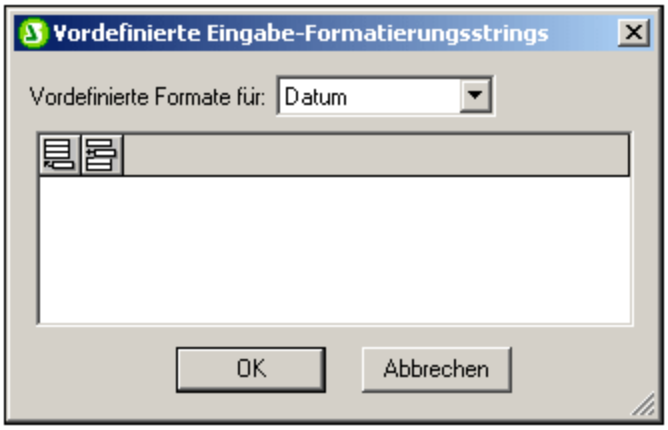
In der Dropdown-Liste stehen zwei Arten von vordefinierten Formaten zur Verfügung:

- vordefinierte Formate, die im Lieferumfang von StyleVision enthalten sind, und
- vordefinierte Formate, die vom Benutzer mit dem Befehl **vordefinierte Wertformatierungsstrings** (diesem Befehl) definiert werden. Wenn ein Benutzer vordefinierte Werteformate erstellt, werden diese für die aktuell geöffnete SPS-Datei erstellt - und nicht für die gesamte Applikation. Nachdem die benutzerdefinierten Wertformate erstellt wurden, muss die SPS-Datei gespeichert werden, damit die Formate beim nächsten Öffnen der Datei zur Verfügung stehen.

Erstellen eines vordefinierten Wertformatierungsstrings

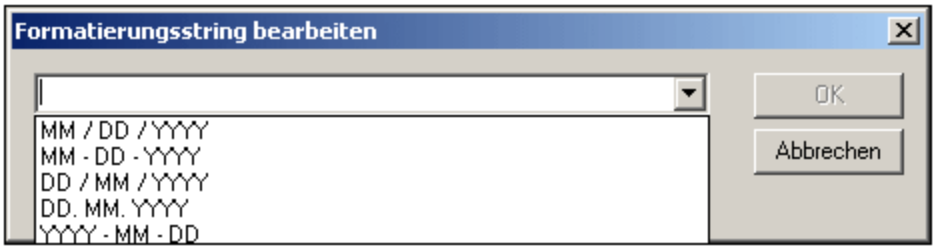
Ein vordefinierter Wertformatierungsstring ist Datentyp-spezifisch. Um einen vordefinierten Wertformatierungsstring zu erstellen, gehen Sie folgendermaßen vor:

1. Klicken Sie auf **Eigenschaften | Vordefinierte Wertformatierungsstrings...**. Daraufhin wird das folgende Dialogfeld angezeigt:



2. Wählen Sie aus der Dropdown-Auswahlliste einen Datentyp aus und klicken Sie anschließend je nach Bedarf auf das Symbol "**Anhängen**" oder "**Einfügen**". Daraufhin wird das Dialogfeld "Formatierungsstring bearbeiten" angezeigt:

Wenn Sie in der Auswahlliste auf den Nach-unten-Pfeil klicken, erscheint eine Dropdown-Liste mit den mit StyleVision installierten vordefinierten Formaten für diesen Datentyp (siehe Abbildung unten).



Sie können nun aus der Liste entweder ein Format auswählen und es ändern oder Sie geben das Format direkt in das Eingabefeld ein. Die Syntax für die Formatierungsdefinition wird unter Werteformatierung[333] erläutert. Wenn Sie Hilfe zur Syntax benötigen, klicken Sie auf die Schaltflächen **Feld einfügen** und **Feldoptionen**.

3. Nachdem Sie ein Format definiert haben, klicken Sie auf OK, um die SPS-Datei zu speichern. Der Formatierungsstring wird zur Liste der vordefinierten Formate für diesen Datentyp hinzugefügt und im Dialogfeld "Werteformatierung" (der aktuellen SPS-Datei) als Option angezeigt, wenn das ausgewählte Element den entsprechenden Datentyp hat.

Beachten Sie die folgenden Punkte:

- Sie können beliebig viele benutzerdefinierte Formatierungsstrings für verschiedene Datentypen hinzufügen.
- Die Reihenfolge der Formatierungsstrings im Dialogfeld "Vordefinierte Formatierungsstrings" legt fest, in welcher Reihenfolge diese Formatierungsstrings im Dialogfeld "Werteformatierung" angezeigt werden. Die benutzerdefinierten Formatierungsstrings werden oberhalb der bereits bereit gestellten vordefinierten Formate angezeigt.
- Um einen benutzerdefinierten Formatierungsstring zu bearbeiten, doppelklicken Sie im Dialogfeld "Vordefinierte Formatierungsstrings" auf den Eintrag.
- Zum Löschen eines benutzerdefinierten Formatierungsstrings, wählen Sie ihn aus und klicken Sie auf die Löschen-Schaltfläche im Dialogfeld "Vordefinierte Wertformatierungsstrings".

© 2019-2025 Altova GmbH

11.11 Menü "Extras"

Das Menü **Extras** enthält den Befehl zur Rechtschreibprüfung und Befehle zum Anpassen von StyleVision.

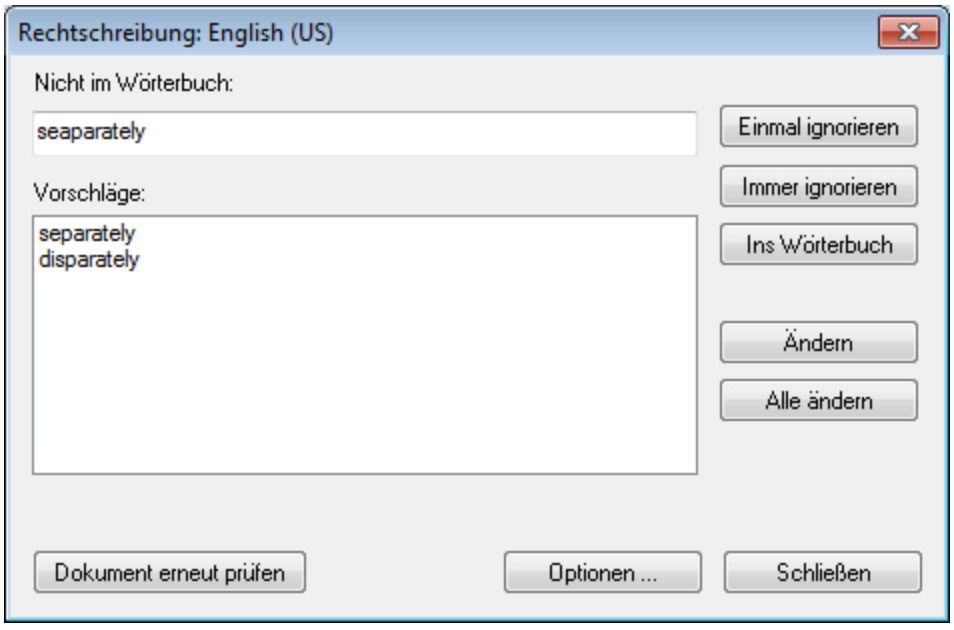
Die Beschreibung zu den Befehlen des Menüs "Extras" ist in die folgenden Unterabschnitte gegliedert:

- Rechtschreibung 541
- Rechtschreiboptionen 542
- XML-Schema-Manager 197
- Anpassen 547

11.11.1 Rechtschreibung

Mit dem Befehl **Rechtschreibung** wird - je nachdem, welches Dokument aktiv ist - eine Rechtschreibprüfung am SPS (in der Design-Ansicht) durchgeführt. Sie können die zu verwendende Sprache für die Rechtschreibprüfung aus den integrierten Sprachwörterbüchern auswählen (*siehe Anmerkung unten*).

Anmerkung: Die Auswahl der integrierten Wörterbücher, die mit Altova Software verfügbar sind, ist nicht Ausdruck einer Präferenz für bestimmte Sprachen durch Altova, sondern ist größtenteils auf der Verfügbarkeit der Wörterbücher begründet, die mit kommerzieller Software im Rahmen von Lizenzen wie z.B. MPL-, LGPL- oder BSD-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 https://www.altova.com/de/dictionaries zur Verfügung. 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.

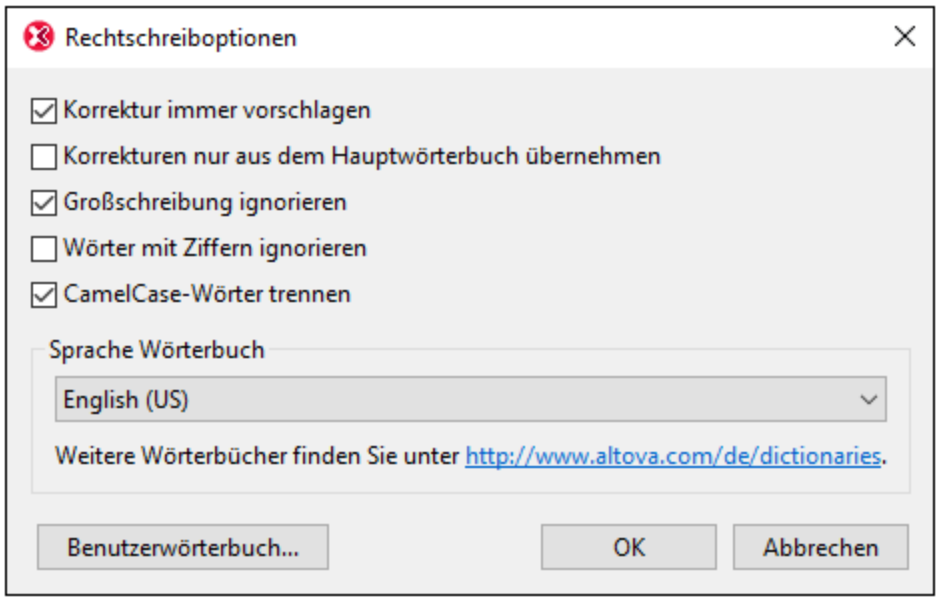


Anschließend können Sie einen Eintrag in der Liste der Vorschläge auswählen und auf **Ändern** oder **Alle ändern** klicken, um entweder eine oder alle Instanzen des markierten Worts zu ändern. (Wenn Sie in der Liste "Vorschläge" auf ein Wort doppelklicken, wird das unbekannte Wort durch dieses ersetzt.) Alternativ dazu können Sie *diese Instanz* des unbekannten Worts ignorieren (**Einmal ignorieren**) oder *alle Instanzen* dieses unbekannten Worts ignorieren (**Immer ignorieren**); oder Sie können dieses unbekannte Wort zum (Benutzer)-Wörterbuch hinzufügen (**Ins Wörterbuch**). Wenn das unbekannte Wort zum Wörterbuch hinzugefügt wird, wird das Wort von der Rechtschreibprüfung als korrekt betrachtet und die Rechtschreibprüfung im Dokument wird fortgesetzt. Sie können des Dokument von Anfang an erneut überprüfen (**Dokument erneut prüfen**) oder das Dialogfeld jederzeit mit dem Befehl **Schließen** schließen.

Über die Schaltfläche **Optionen** wird das Dialogfeld Rechtschreibung Optionen 542 geöffnet. Hier können Sie Optionen für die Rechtschreibprüfung definieren.

11.11.2 Rechtschreiboptionen

Mit Hilfe des Befehls **Rechtschreibung - Optionen** wird ein Dialogfeld (siehe unten) geöffnet, in dem Sie Optionen für die Rechtschreibprüfung definieren können.



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:

 © 2019-2025 Altova GmbH

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.

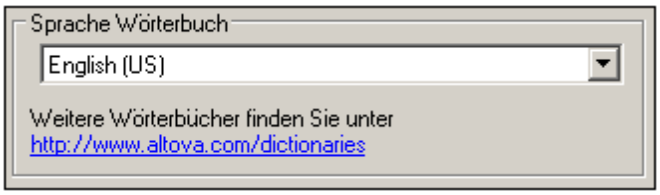
Alle installierten Wörterbücher stehen für alle Benutzer auf dem Rechner und alle Altova-Produktversionen (ob 32-Bit oder 64-Bit) zur Verfügung

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.

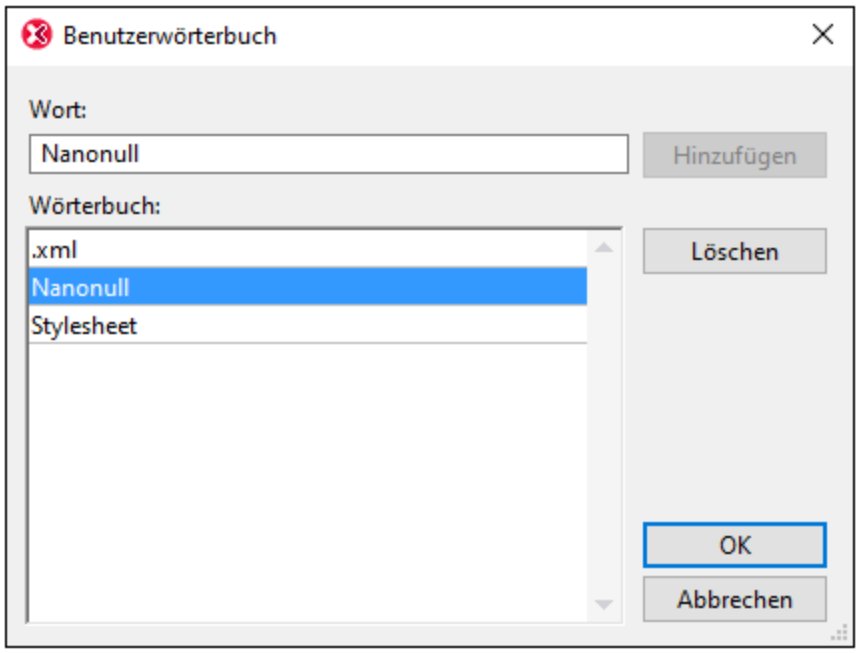


Anmerkung: Es bleibt Ihnen überlassen, ob Sie mit den für das jeweilige Wörterbuch geltenden Lizenzbedingungen einverstanden sind und ob das Wörterbuch sich für die Verwendung mit der auf Ihrem Rechner installierten Software eignet.

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*).

© 2019-2025 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.

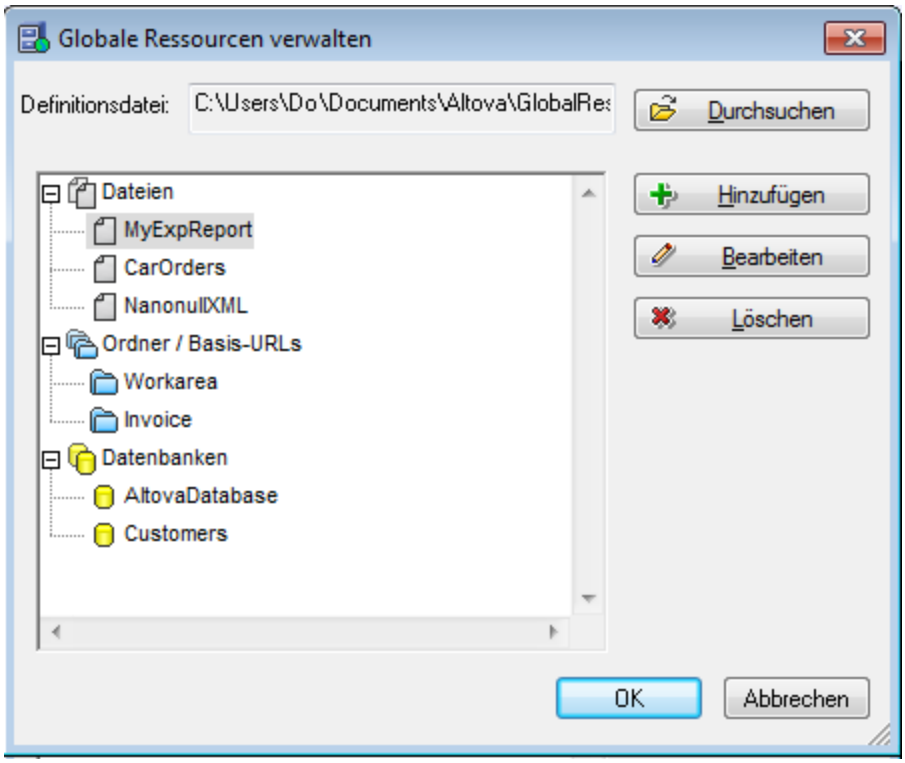
Wörter können auch während einer Rechtschreibprüfung zum Benutzerwörterbuch hinzugefügt werden. Wenn bei der Rechtschreibprüfung ein unbekanntes Wort gefunden wird, wird das Dialogfeld Rechtschreibprüfung 541 aufgerufen und Sie werden aufgefordert zwischen verschiedenen Aktionen zu wählen. Wenn Sie auf 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`

11.11.3 Globale Ressourcen

Mlt dem Befehl **Globale Ressourcen** wird das Dialogfeld "Globale Ressourcen" aufgerufen (*Abbildung unten*), in dem Sie Folgendes tun können:

- Angabe der XML-Datei für globale Ressourcen.
- Hinzufügen von globalen Ressourcen (oder Aliassen) vom Typ Datei, Ordner und Datenbank
- Angabe verschiedener Konfigurationen für die einzelnen globalen Ressourcen (Aliasse). Jede Konfiguration wird einer bestimmten Ressource zugeordnet.

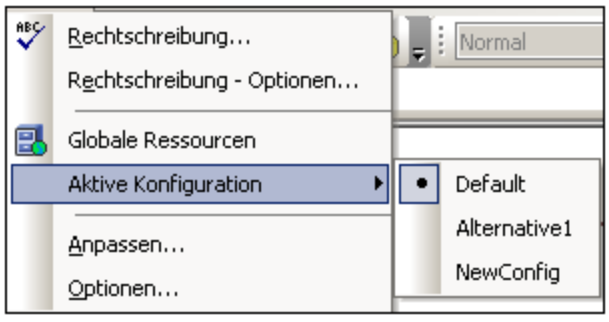


Eine Beschreibung, wie man globale Ressourcen definiert, finden Sie im Abschnitt Definieren globaler Ressourcen[362].

Anmerkung: Das Dialogfeld "Globale Ressourcen" kann auch über die Symbolleiste "Globale Ressourcen"[501] aufgerufen werden (**Ansicht | Symbolleisten | Globale Ressourcen**).

11.11.4 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[362] 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.

© 2019-2025 Altova GmbH

Anmerkung: Die aktive Konfiguration kann auch über die Symbolleiste "Globale Ressourcen" 501 ausgewählt werden (**Ansicht | Symbolleisten | Globale Ressourcen**).

11.11.5 XML-Schema-Manager

Mit diesem Befehl wird das Dialogfeld "XML-Schema-Manager" aufgerufen, in dem Sie Ihre XML-Schema-Pakete verwalten können.

Um ein XML-Schema zu installieren, aktivieren Sie das Kontrollkästchen neben dem gewünschten Schema und klicken Sie auf **Anwenden**. Des Weiteren können Sie Schemas deinstallieren, aktualisieren, überprüfen, ob neue Schemas hinzugekommen sind, und alle Ihre Schemas zentral verwalten.

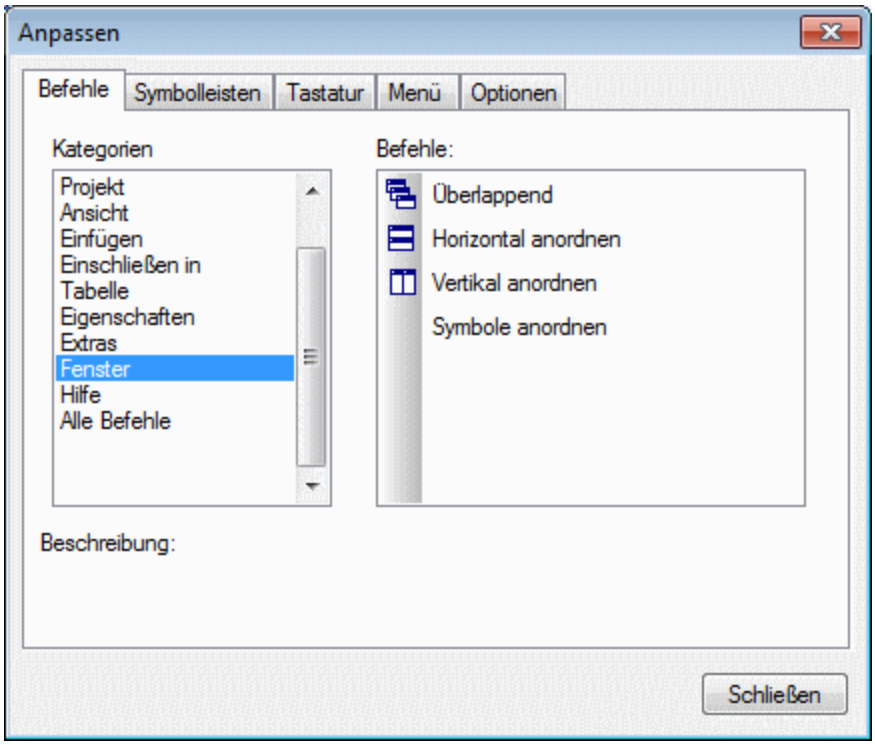
Nähere Informationen dazu finden Sie unter Schema-Manager 197.

11.11.6 Anpassen

Mit Hilfe des Befehls "Anpassen" können Sie StyleVision Ihren persönlichen Bedürfnissen entsprechend anpassen.

Register "Befehle"

Auf dem Register **Befehle** des Dialogfelds "Anpassen" können Sie einzelne Befehle in die Menüleiste und die Symbolleiste platzieren.



Um einen Befehl zur Menüleiste oder Symbolleiste hinzuzufügen, wählen Sie den Befehl auf dem Register "Befehle" im Bereich "Befehle" aus und ziehen Sie ihn in die Menü- oder die Symbolleiste. Sobald sich der Cursor über einer gültigen Position befindet, wird das I-Zeichen angezeigt und Sie können die Maustaste nun loslassen. Wenn der Befehl nicht an diese Stelle gezogen werden kann, erscheint ein Häkchen. Wenn Sie die Maustaste loslassen, wird der Befehl als Symbol erstellt, wenn der Befehl bereits mit einem Symbol verknüpft ist. Andernfalls wird der Befehl als Text erstellt. Nachdem ein Befehl zur Menü- oder Symbolleiste hinzugefügt wurde, können Sie die Darstellung durch Rechtsklick und Auswahl der gewünschten Aktion ändern.

Zum Löschen eines Elements in der Menü- oder Symbolleiste, klicken Sie bei geöffnetem Dialogfeld "Anpassen" mit der rechten Maustaste auf das gewünschte Element und wählen Sie den Befehl "Löschen".

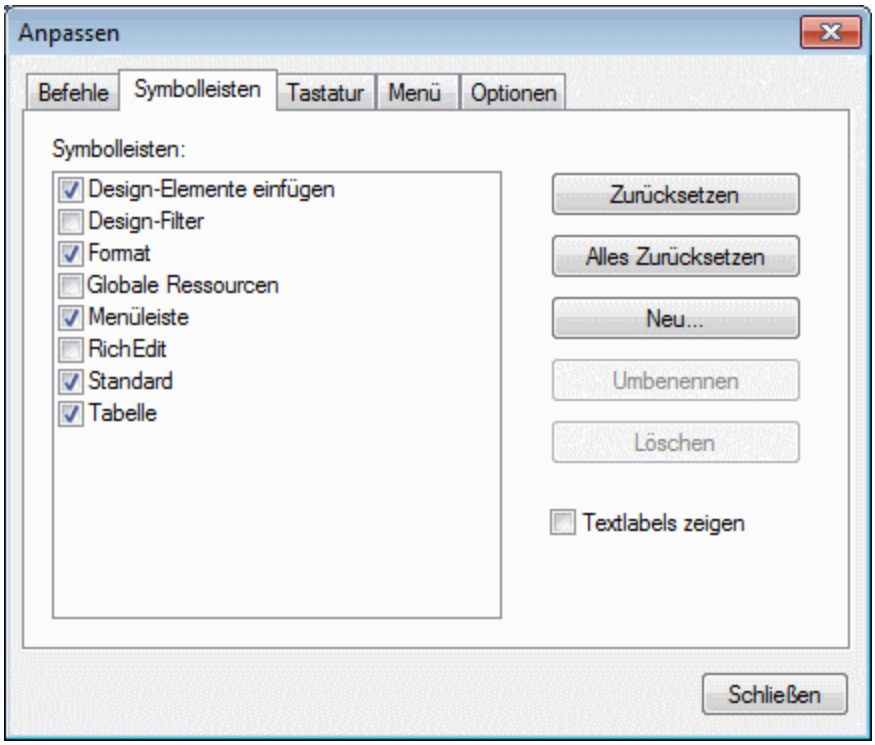
Anmerkung:

- Die oben beschriebene Anpassung wird auf die Applikation angewendet und zwar egal, ob ein Dokument in StyleVision geöffnet ist oder nicht.
- Um Menüs und Symbolleisten in den Original-Installationszustand zu versetzen, gehen Sie zum Register "Symbolleiste" und klicken Sie auf die entsprechende "Zurücksetzen"-Schaltfläche.

Register "Symbolleisten"

Über das Register **Symbolleisten** haben Sie die Möglichkeit, bestimmte Symbolleisten zu aktivieren oder zu deaktivieren, Textlabels für Symbolleisten-Elemente anzuzeigen und bei der Menüleiste und den Symbolleisten den Installationszustand wiederherzustellen.

© 2019-2025 Altova GmbH



Auf der Benutzeroberfläche von StyleVision werden eine fixe Menüleiste sowie mehrere optionale Symbolleisten (Design Filter, Formatierung, Standard, Tabellen und Inhaltsverzeichnis) angezeigt.

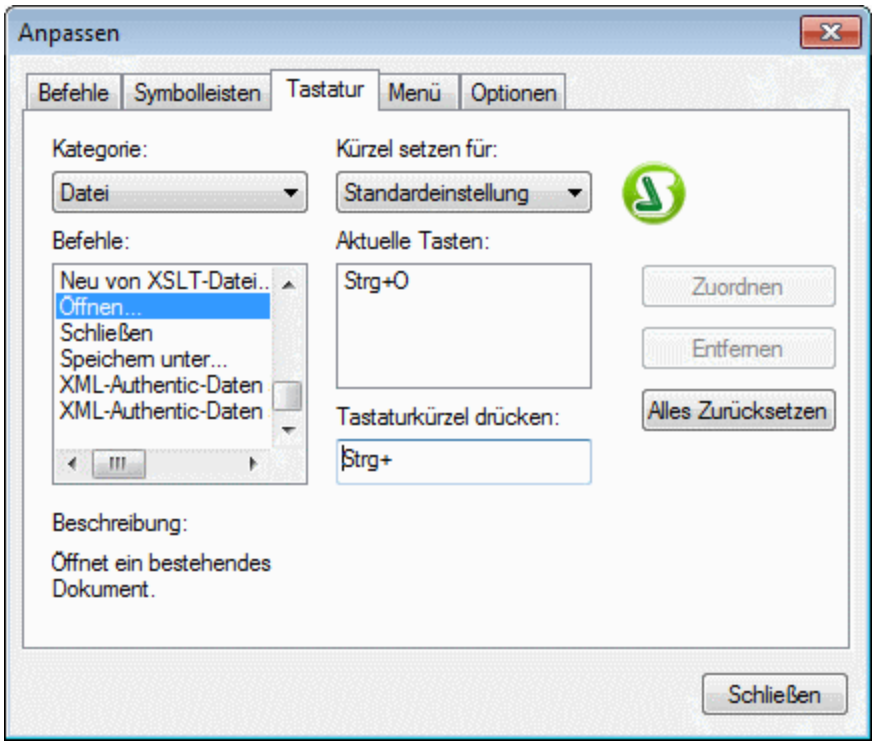
Jede Symbolleiste kann in Befehlsgruppen unterteilt werden. Über das Register "Befehle" können Befehle zu einer Symbolleiste hinzugefügt werden. Außerdem können Sie Symbolleisten aus der angedockten Position an eine beliebige Stelle auf dem Bildschirm ziehen. Wenn Sie auf die (maximierte oder minimierte) Titelleiste klicken, wird diese an- bzw. abgedockt.

Im Dialogfeld "Anpassen" können Sie Symbolleisten durch Aktivieren bzw. Deaktivieren des Kontrollkästchens ein- oder ausblenden. Wenn eine Symbolleiste (auf dem Register "Symbolleisten") ausgewählt ist, können Sie die Textlabels zu den Schaltflächen dieser Symbolleiste durch Aktivieren des Kontrollkästchens **Textlabels zeigen** einblenden. Außerdem können Sie eine ausgewählte Symbolleiste in den Original-Installationszustand zurücksetzen, indem Sie auf die Schaltfläche **Zurücksetzen** klicken. Wenn Sie auf die Schaltfläche **Alles zurücksetzen** klicken, werden alle Symbolleisten und die Menüleiste zurückgesetzt.

Anmerkung zur Menüleiste
Sie können Befehle zur Menüleiste hinzufügen und Schaltflächen daraus löschen: siehe Befehle oben. Um die Menüleiste in den Original-Installationszustand zurückzusetzen, wählen Sie die Menüleiste im Dialogfeld "Anpassen" auf dem Register "Symbolleisten" aus und klicken Sie auf die Schaltfläche **Zurücksetzen**. (Wenn Sie auf die Schaltfläche **Alles zurücksetzen** klicken, werden auch die Symbolleisten zurückgesetzt.)

Register "Tastatur"

Über das Register **Tastatur** können Sie Tastaturkürzel für jeden beliebigen StyleVision-Befehl definieren oder ändern.



So weisen Sie einem Befehl ein Tastaturkürzel zu

1. Wählen Sie über die Auswahlliste "Kategorie" die Kategorie, in der sich der Befehl befindet, aus.
2. Wählen Sie in der Liste der Befehle den gewünschten Befehl aus.
3. Klicken Sie in das Feld "Kürzel setzen für" und drücken Sie die Tastaturkürzel, mit denen dieser Befehl aktiviert werden soll. Das Tastaturkürzel wird sofort im Feld "Tastaturkürzel drücken" angezeigt. Wenn dieses Tastaturkürzel bereits einem anderen Befehl zugewiesen ist, so wird dieser Befehl unterhalb des Eingabefelds angezeigt. (In der Abbildung oben ist z.B. Strg+C bereits dem Befehl "Kopieren" zugeordnet, das Kürzel kann daher dem Befehl "Öffnen" nicht zugewiesen werden.) Um die Eingabe 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 permanent zuzuweisen. Das Tastaturkürzel wird nun im Textfeld "Aktuelle Tasten" angezeigt.

So heben Sie die Zuweisung eines Tastaturkürzels auf (oder löschen es)

1. Wählen Sie den Befehl, für den das Tastaturkürzel gelöscht werden soll.
2. Klicken Sie auf das Tastaturkürzel, das Sie im Listefeld der aktuellen Tasten löschen möchten.
3. Klicken Sie auf die Schaltfläche **Entfernen** (die jetzt aktiv ist).

So setzen Sie alle Tastaturkürzelzuweisungen zurück

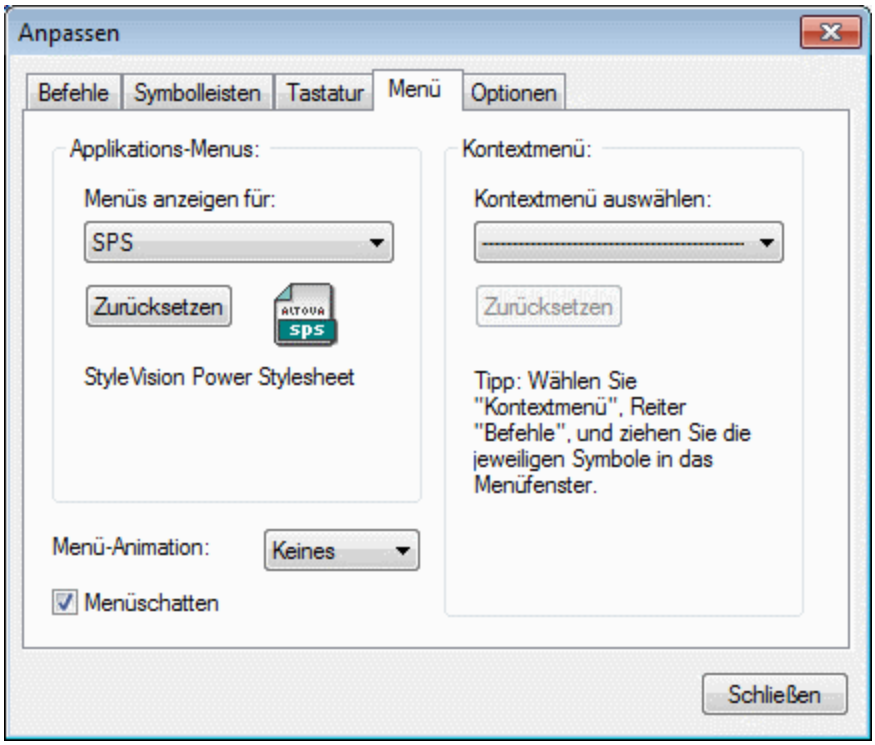
1. Klicken Sie auf die Schaltfläche **Alles zurücksetzen**, um den Originalzustand wieder herzustellen. Daraufhin erscheint ein Dialogfeld, in dem Sie gefragt werden, ob Sie wirklich alle Tastenzuweisungen zurücksetzen möchten.
2. Klicken Sie auf "Ja", um alle Tastenzuweisungen zurückzusetzen.

Kürzel setzen für
Derzeit steht keine Funktion zur Verfügung.

© 2019-2025 Altova GmbH

Register "Menü"

Über das Register **Menü** können Sie die Hauptmenüleiste sowie die Kontextmenüs anpassen. Es gibt zwei Varianten der Hauptmenüleiste: *Standard* (wird angezeigt, wenn kein Dokument geöffnet ist) und *SPS* (wird angezeigt, wenn ein SPS-Dokument geöffnet ist).



So passen Sie ein Menü an

1. Wählen Sie die gewünschte Menüleiste aus *(in der Abbildung oben SPS)*.
2. Klicken Sie auf das Register **Befehle** und ziehen Sie die Befehle in die Menüleiste Ihrer Wahl.

So löschen Sie Befehle aus einem Menü

1. Wählen Sie auf dem Register "Menü" im Bereich "Applikations-Menüs" entweder die Option *Standardmenü* (damit werden die Menüs, die verfügbar sind, wenn kein Dokument geöffnet ist, angezeigt) oder die Option *SPS* aus (damit werden die Menüs angezeigt, die verfügbar sind, wenn ein oder mehrere Dokumente geöffnet sind).
2. Während das Dialogfeld "Anpassen" geöffnet ist, wählen Sie (i) das Menü, das Sie aus der Applikationsmenüleiste löschen möchten oder (ii) den Befehl, den Sie aus einem dieser Menüs löschen möchten, aus.
3. (i) Ziehen Sie das Menü bzw. den Menübefehl entweder mit der Maus aus der Menüleiste bzw. dem Menü heraus oder (ii) klicken Sie mit der rechten Maustaste auf das Menü oder den Menübefehl und wählen Sie den Befehl **Löschen**.

So setzen Sie eine der Menüleisten zurück

1. Wählen Sie den Menüeintrag in der Auswahlliste des Bereichs "Applikations-Menüs" aus.
2. Klicken Sie auf die Schaltfläche **Zurücksetzen** unterhalb des Menünamens. Daraufhin werden Sie gefragt, ob Sie die Menüleiste wirklich zurücksetzen möchten.

So passen Sie ein Kontextmenü an

1. Wählen Sie das Kontextmenü in der Auswahlliste aus.
2. Klicken Sie auf das Register **Befehle** und ziehen Sie die Befehle in das gerade geöffnete Kontextmenü.

So löschen Sie Befehle aus dem Kontextmenü

1. Klicken Sie mit der rechten Maustaste auf das Symbol für den Befehl und
2. Wählen Sie im Popup-Menü die Option **Löschen** aus
   oder
2. Ziehen Sie den Befehl aus dem Kontextmenü heraus und lassen Sie die Maustaste los, sobald unterhalb des Mauszeigers das Häkchensymbol angezeigt wird.

So setzen Sie ein Kontextmenü zurück

1. Wählen Sie das Kontextmenü in der Auswahlliste aus und
2. Klicken Sie auf die Schaltfläche **Zurücksetzen** unterhalb des Kontextmenünamens. Daraufhin werden Sie gefragt, ob Sie das Kontextmenü wirklich zurücksetzen möchten.

So schließen Sie ein Kontextmenüfenster

- Klicken Sie auf das Schließen-Symbol rechts oben in der Titelleiste oder
- Klicken Sie auf die Schließen-Schaltfläche des Dialogfelds "Anpassen".

Menüanimationen
Über die Option "Menü-Animation" wird festgelegt, wie ein Menü angezeigt wird, wenn Sie darauf klicken. Wählen Sie eine Option aus der Dropdown-Liste der Menüanimationen aus.

Menüschatten
Wenn die Menüs mit einem Schatten angezeigt werden sollen, aktivieren Sie diese Option. Alle Menüs werden daraufhin mit einem Schatten angezeigt.

Register "Optionen"

Über das Register **Optionen** können Sie zusätzliche Funktionen der Symbolleiste anpassen.

Die Tooltipps für Schaltflächen der Symbolleiste werden angezeigt, wenn die Option "Tooltip in Symbolleiste einblenden" aktiv ist. Diese Option hat eine Unteroption, ob Tastaturkürzel (falls verfügbar) in den Tooltipps angezeigt werden sollen oder nicht.

11.11.7 Symbolleisten und Fenster wiederherstellen

Mit diesem Befehl wird der ursprüngliche Standardzustand von Symbolleisten, Fenster, Eingabehilfen und anderen GUI-Komponenten wiederhergestellt. Damit die Änderungen wirksam werden, müssen Sie StyleVision neu starten.

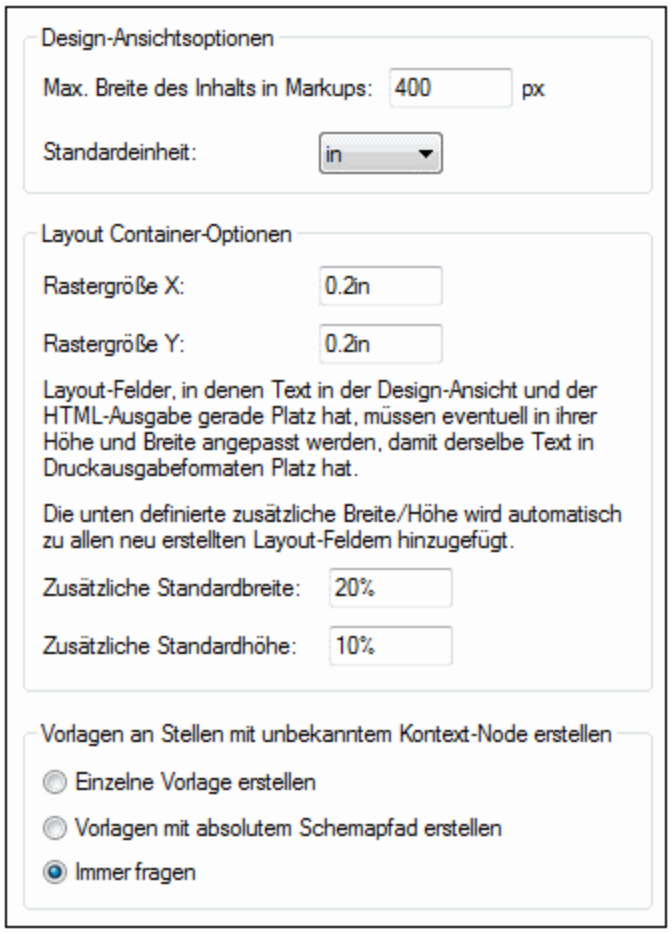
11.11.8 Optionen

Mit dem Befehl **Optionen** wird ein Dialogfeld geöffnet, in welchem Sie die Kodierung für die HTML-Ausgabedatei festlegen können (*Abbildung unten*).

© 2019-2025 Altova GmbH

Design-Optionen

Auf dem Register "Design" (*Abbildung unten*) können Sie die allgemeinen Optionen für Designs für die gesamte Applikation definieren.



Die folgenden Optionen stehen zur Verfügung:

- Maximale Breite von Markup-Tags (in Pixeln): Geben Sie die positive Ganzzahl für die gewünschte Pixelanzahl ein.
- Rastergröße von Layout Containern in absoluten Längeneinheiten. Die angegebenen Längenwerte sind die Abstände zwischen zwei Punkten auf der entsprechenden Rasterachse.
- Zusätzliche Standardbreite und Höhe von Layout-Feldern. Diese zusätzlichen Höhen- und Breitenangaben werden zu allen Layout-Felder hinzugefügt, damit größer dargestellter Text in Druckformaten Platz hat. Diese Werte können als Prozentwerte oder in Form von absoluten Längeneinheiten definiert werden.
- Das Standardverhalten, wenn eine Node-Vorlage an einer Stelle erstellt wird, an der der Kontext-Node nicht bekannt ist. Diese Option gilt normalerweise für benutzerdefinierte Vorlagen, in denen die Vorlage für Elemente erstellt wurde, die nicht in Kontext in der Schemaquelle des Designs platziert werden können. Wenn ein Node in einer solchen benutzerdefinierten Vorlage erstellt wird, kann er (i) nur mit seinem Namen oder (ii) mit dem vollständigen Pfad von der Schema-Root zum Node erstellt werden. Sie können eine dieser Optionen als Standardverhalten definieren oder festlegen, dass Sie jedes Mal, wenn diese Situation auftritt, gefragt werden. Die Standardeinstellung für diese Option ist *Immer*

fragen.

Vorschauen

Auf dem Register "Vorschauen" (*Abbildung unten*) können Sie Vorschauoptionen auswählen.

- *HTML-Ausgabe:* Wählen Sie aus, ob als Browser für die HTML-Vorschauen Edge/Webview2 oder Internet Explorer verwendet werden soll.
- *PDF-Ausgabe:* Wählen Sie aus, welcher PDF-Reader für die PDF-Vorschau verwendet werden soll: (i) PDF.js (ein Open-Source PDF-Ansichtsprogramm für Browser) in Edge; (ii) Edge Native (der integrierte PDF-Reader von Microsoft Edge) oder (iii) Adobe Acrobat Reader.
- *Einstellungen für geteilte Vorschau:* Sie können eine Verzögerung für die bei automatischer Aktualisierung generierten Ausgabevorschauen definieren. Die Ansicht wird automatisch aktualisiert, wenn das Design geändert wird. Wenn Sie z.B. eine Verzögerung von Null Millisekunden definieren, wird die Ausgabevorschau bei jeder Änderung des Designs mit einer Verzögerung von Null neu generiert. Wenn ein kurzes Intervall für die Verzögerung definiert wurde, wird die Ausgabe in kurzen Intervallen und somit sehr oft neu generiert. Dies führt nicht nur zu einer langsamen Generierung der Ausgabevorschau, sondern auch zu Zwischenausgaben wie Fehlermeldungen, die auftreten, wenn ein Änderungsvorgang temporär ein ungültiges Design oder ungültigen Daten zur Folge hat. Eine lange Verzögerung könnte wiederum zu langen Wartezeiten für die Ausgabevorschau führen. Wählen Sie je nach der Größe Ihres Designs und Ihrer Daten einen optimalen Wert aus. Beachten Sie, dass die in dieser Einstellung definierte Verzögerung nicht angewendet wird, wenn Sie das erste Mal in die geteilte Vorschau wechseln oder die Ausgabevorschau durch Klick auf die Schaltfläche **Aktualisieren** manuell aktualisieren. In diesen beiden Fällen, wird die Ausgabe sofort generiert.

Schema-Optionen

In der Schemastruktur können Elemente und Attribute alphabetisch in aufsteigender Reihenfolge aufgelistete werden. Aktivieren Sie dazu auf dem Register "Schema-Optionen" die entsprechenden Kontrollkästchen. Standardmäßig werden Attribute alphabetisch und Elemente in der Reihenfolge, wie sie in der Schemastruktur erscheinen, aufgelistet, sofern dies möglich ist.

Standardkodierung

Um die Standardkodierung der HTML-Ausgabedatei zu definieren, öffnen Sie das Dropdown-Menü der Auswahlliste, wählen Sie die gewünschte Option aus der Liste der Kodierungsoptionen aus und klicken Sie auf **OK**. In jedem neuen SPS, das von nun an erstellt wird, wird die auf diesem Register verwendete HTML-Ausgabekodierung verwendet.

In der XSLT-für-HTML-Datei sind die Ausgabe-Kodierungs-Informationen an den folgenden Stellen registriert:

- Im Attribut `encoding` des Stylesheet-Elements `xsl:output`:
  ```
  <xsl:output version="1.0" encoding="UTF-8" indent="no" omit-xml-declaration="no"
  media-type="text/html" />
  ```
- Im Attribut `charset` des Elements `content-type meta` im HTML Header:
  ```
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  ```

Anmerkung: Diese Einstellungen sind die Standard-Kodierungen, die für neue SPSs verwendet werden. Sie können die Kodierung des gerade offenen SPS über dieses Dialogfeld nicht ändern. Um die Kodierung eines geöffneten SPS zu ändern, verwenden Sie den Befehl Datei | Eigenschaften 489.

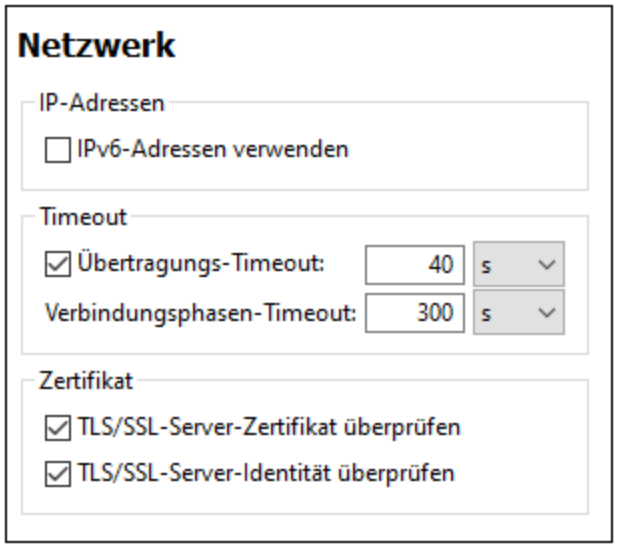
© 2019-2025 Altova GmbH

XSL-Optionen

In den Metadaten von HTML- und PDF-Ausgabedateien wird standardmäßig die Zeile 'Generiert von StyleVision' erzeugt. Bei gegen Gebühr erworbenen Versionen haben Sie die Option, die Generierung dieser Zeile zu deaktivieren.

Netzwerk-Optionen

Im Abschnitt **Netzwerk** (*Abbildung unten*) können Sie wichtige Netzwerkeinstellungen konfigurieren.



IP-Adressen

Wenn Host-Namen in gemischten IPv4/IPv6-Netzwerken zu mehr als einer Adresse aufgelöst werden, werden bei Auswahl dieser Option die IPv6-Adressen verwendet. Wenn die Option in solchen Umgebungen nicht aktiviert ist und IPv4-Adressen zur Verfügung stehen, werden IPv4-Adressen verwendet.

Timeout

- *Übertragungs-Timeout:* Wenn bei der Übertragung zweier beliebiger aufeinander folgender Datenpakete einer Übertragung (bei Sendung oder Empfang) dieses Limit erreicht wird, wird die gesamte Übertragung abgebrochen. Die Werte können in Sekunden [s] oder Millisekunden [ms] angegeben werden, wobei der Standardwert 40 Sekunden beträgt. Wenn die Option nicht aktiviert ist, gibt es keinen Grenzwert, ab dem eine Übertragung abgebrochen wird.
- *Verbindungsphasen-Timeout:* Dies ist das Zeitlimit, innerhalb dessen die Verbindung (inklusive Sicherheitshandshake) hergestellt worden sein muss. Die Werte können in Sekunden [s] oder Millisekunden [ms] angegeben werden, wobei der Standardwert 300 Sekunden beträgt. Dieses Timeout kann nicht deaktiviert werden.

Zertifikat

- *TLS/SSL-Server-Zertifikat überprüfen* Wenn diese Option aktiviert ist, wird die Authentizität des Server-Zertifikats überprüft, indem die digitale Signaturkette überprüft wird, bis ein vertrauenswürdiges Root-Zertifikat erreicht wird. Diese Option ist standardmäßig aktiviert. Wenn diese Option nicht aktiviert wird, ist die Kommunikation nicht sicher. Angriffe (z.B. ein Man-in-the-Middle-Angriff) würden nicht erkannt. Beachten Sie, dass mit dieser Option nicht überprüft wird, ob das Zertifikat tatsächlich das Zertifikat für den Server, mit dem kommuniziert wird, ist. Um eine umfassende Sicherheit zu

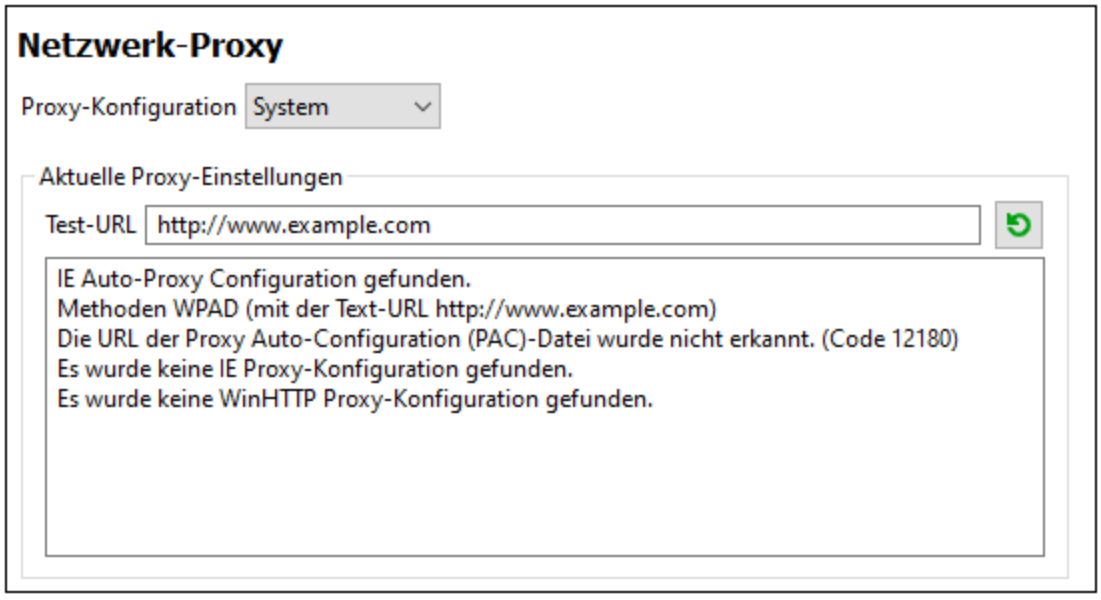
gewährleisten, müssen sowohl das Zertifikat als auch die Identität überprüft werden (*siehe nächste Option*).

- *TLS/SSL-Server-Identität überprüfen* Wenn diese Option ausgewählt ist, wird überprüft, ob das Server-Zertifikat zu dem Server gehört, mit dem kommuniziert werden soll. Dazu wird überprüft, ob der Server-Name in der URL mit dem Namen im Zertifikat übereinstimmt. Diese Option ist standardmäßig aktiviert. Wenn diese Option nicht aktiviert ist, wird die Identität des Servers nicht überprüft. Beachten Sie, dass das Zertifikat des Servers mit dieser Option nicht überprüft wird. Um eine umfassende Sicherheit zu gewährleisten, müssen sowohl das Zertifikat als auch die Identität überprüft werden (*siehe vorhergehende Option*).

Netzwerk-Proxy-Optionen

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`**

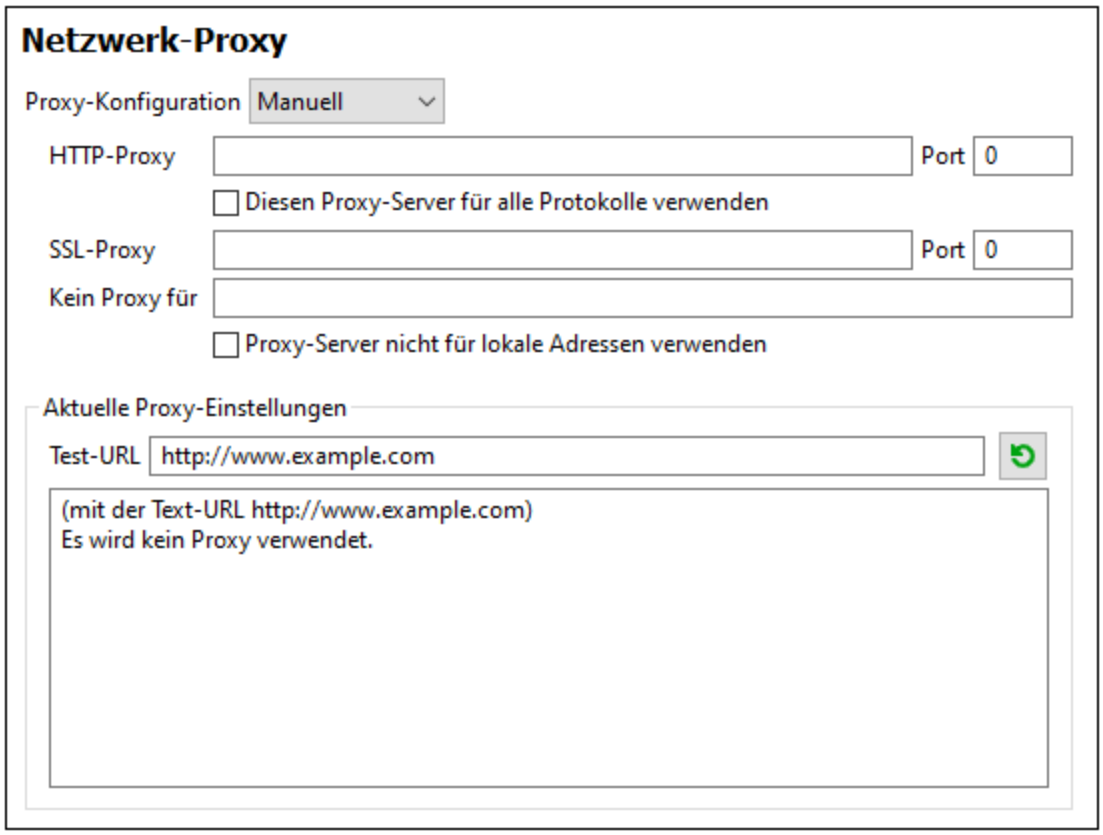
© 2019-2025 Altova GmbH

(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 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.

Anmerkung: Wenn ein Proxy-Server definiert wurde und Sie eine Transformation auf Altova FlowForce Server bereitstellen möchten, müssen Sie die Option *Proxy-Server nicht für lokale Adressen verwenden* aktivieren

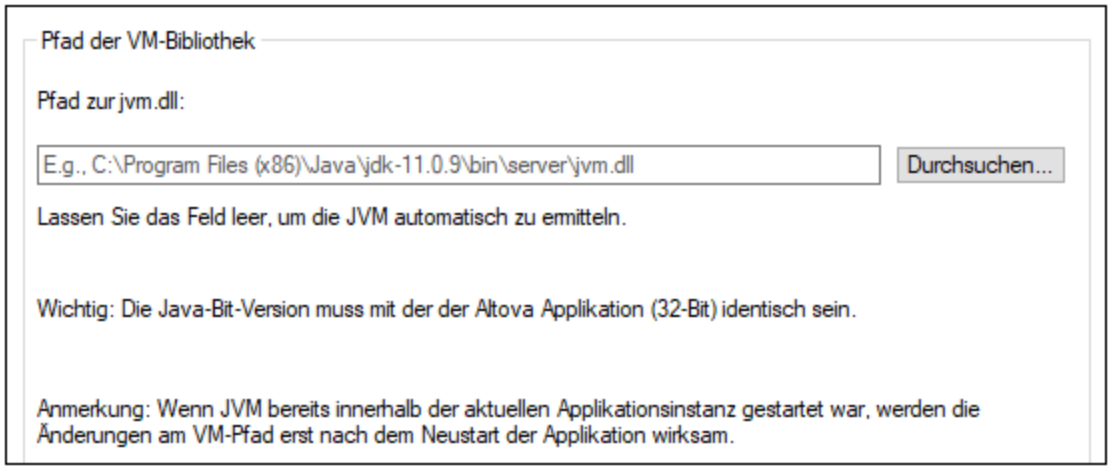
Aktuelle Proxy-Einstellungen
Stellt ein ausführliches Protokoll der Proxy-Ermittlung bereit. Es kann über die Schaltfläche **Aktualisieren** rechts vom Feld *Test-URL* aktualisiert werden (z.B. bei Wechsel zu einer anderen Test-URL oder wenn die Proxy-Einstellungen geändert wurden).

- *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*).

Java-Optionen

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. StyleVision 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 StyleVision 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

 © 2019-2025 Altova GmbH

Verzeichnis, in dem JDK installiert ist) verweisen.
- Die StyleVision-Plattform (32-Bit, 64-Bit) muss mit der des JDK identisch sein.
- Nachdem Sie den Java VM-Pfad geändert haben, müssen Sie StyleVision eventuell neu starten, damit die neuen Einstellungen wirksam werden.

Hilfe

StyleVision bietet eine Hilfe (Benutzerhandbuch) in zwei Formaten:

- eine Online-Hilfe im HTML-Format. Diese steht auf der Altova-Website zur Verfügung. Um die Online-Hilfe aufrufen zu können, benötigen Sie Internet-Zugriff.
- eine Hilfedatei im PDF-Format, die bei der Installation von StyleVision auf Ihrem Rechner installiert wird. Sie hat den Namen `StyleVision.pdf` und befindet sich im Applikationsordner (im Ordner "Programme"). Wenn Sie keinen Internet-Zugriff haben, können Sie immer diese lokal gespeicherte Hilfedatei öffnen.

Über die Option Hilfe (*Abbildung unten*) können Sie auswählen, welches der beiden Formate geöffnet werden soll, wenn Sie im Menü **Hilfe** auf den Befehl **Hilfe (F1)** klicken.

Sie können diese Option jederzeit ändern. Über die Links in diesem Abschnitt (*siehe Abbildung oben*) können Sie das entsprechende Hilfeformat öffnen.

11.12 Menü "Fenster"

Das **Menü "Fenster"** enthält Befehle, mit denen festgelegt wird, wie StyleVision Fenster auf der Benutzeroberfläche angezeigt werden sollen (überlappend, nebeneinander oder maximiert). Um ein Fenster zu maximieren, klicken Sie auf die Maximieren-Schaltfläche dieses Fensters.

Außerdem werden alle aktuell geöffneten Dokumentfenster in diesem Menü nach Dokumentnamen aufgelistet. Das aktive Fenster ist mit einem Häkchen versehen. Um ein anderes Fenster zum aktiven zu machen, klicken Sie auf den Namen des gewünschten Fensters.

Dialogfeld "Fenster"
Unterhalb der Liste der geöffneten Fenster befindet sich der Eintrag "Fenster". Wenn Sie darauf klicken, wird das Dialogfeld "Fenster" geöffnet. In diesem Dialogfeld sehen Sie eine Liste aller offenen Fenster sowie Befehle, die auf das/die ausgewählte(n) Fenster angewendet werden können. (Ein Fenster wird durch Klicken auf seinen Namen ausgewählt.)

Warnung: Um das Dialogfeld "Fenster" zu schließen, klicken Sie auf OK; klicken Sie **nicht** auf die Schaltfläche "Fenster schließen", da damit das/die aktuell ausgewählte(n) Fenster geschlossen werden.

© 2019-2025 Altova GmbH

11.13 Menü "Hilfe"

Im Menü **Hilfe** finden Sie Befehle zum Aufrufen der Bildschirmhilfe zu StyleVision, Befehle zum Abrufen von Informationen über StyleVision und Links zu den Support-Seiten auf der Altova Website. Außerdem enthält das Menü **Hilfe** das Registrierungsdialogfeld[561], über das Sie nach Erwerb des Produkts den Lizenz-Keycode eingeben können.

Die Beschreibung der Befehle des Menüs **Hilfe** ist in die folgenden Unterabschnitte gegliedert:

- Hilfe[561]
- Registrierung, Bestellformular[561]
- Weitere Befehle[565]

11.13.1 Hilfe

Mit dem Befehl **Hilfe (F1)** wird die Hilfe-Dokumentation (das Benutzerhandbuch) der Applikation geöffnet. Standardmäßig wird die Online-Hilfe im HTML-Format auf der Altova Website aufgerufen.

Falls Sie keinen Internet-Zugriff haben oder die Online-Hilfe aus einem anderen Grund nicht aufrufen möchten, können Sie die lokal gespeicherte Version des Benutzerhandbuchs verwenden. Dabei handelt es sich um eine PDF-Datei namens `StyleVision.pdf`, die sich im Applikationsordner (im Ordner "Programme") befindet.

Im Abschnitt "Hilfe" des Dialogfelds "Optionen" (Menübefehl **Extras | Optionen**) können Sie das gewünschte Standardformat wechseln (Online-Hilfe oder lokale PDF-Datei).

11.13.2 Aktivierung, Bestellformular, Registrieren, Updates

Software-Aktivierung

Lizenzieren Ihres Produkts
Nachdem Sie Ihre Altova-Software heruntergeladen haben, können Sie sie entweder mit Hilfe eines 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. Klicken Sie darauf, um Ihre Lizenz abzurufen. Wenn Sie auf diese Schaltfläche klicken, wird ein Hash Ihrer Rechner-ID erzeugt und über HTTPS an Altova gesendet. Die Lizenzinformationen werden per HTTP-Response an den Rechner zurückgesendet. Wenn die Lizenz erfolgreich erstellt wurde, wird in Ihrer Altova-Applikation ein entsprechendes Dialogfeld angezeigt. Wenn Sie in diesem Dialogfeld auf **OK** klicken, wird die Software für einen Zeitraum von 30 Tagen **auf diesem bestimmten Rechner 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.

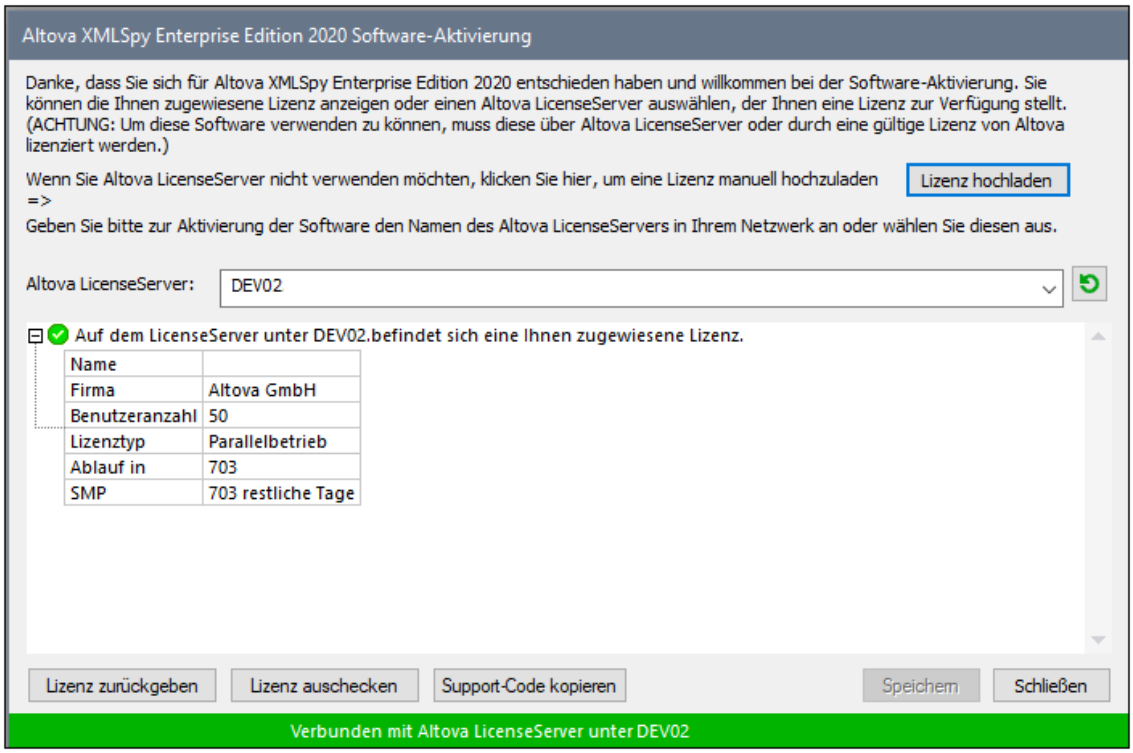
Aktivieren Ihrer Software
Sie können die Software durch Registrieren der Lizenz im Dialogfeld "Software-Aktivierung" oder durch Lizenzierung über Altova LicenseServer (*nähere Informationen siehe unten*) aktivieren.

- *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

 © 2019-2025 Altova GmbH

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.



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 es sich bei der ausgecheckten Lizenz um eine Einzelplatzlizenz oder Parallellizenz handelt, wird sie auf dem Rechner ausgecheckt und steht dem Benutzer, der die Lizenz ausgecheckt hat, zur Verfügung. Wenn es sich bei der Lizenz um eine Named User-Lizenz handelt, wird die Lizenz an das Windows-Konto des jeweiligen Benutzers (Named User) ausgecheckt. Lizenz Check-outs funktionieren auf einer virtuellen Maschine, nicht aber auf einem virtuellen Desktop (in einer VDI). Anmerkung:Wenn eine Named User-Lizenz ausgecheckt wird, werden die Daten zur Identifikation des Check-outs im Profil des Benutzers gespeichert. Damit Lizenz-Check-outs funktionieren, muss das Profil des Benutzers auf dem lokalen Rechner, der offline verwendet werden soll, gespeichert sein. Wenn das Profil des Benutzers nicht lokal (z.B. auf einem freigegebenen Laufwerk) gespeichert ist, wird der Check-out als ungültig gemeldet, sobald der Benutzer versucht, die Altova-Applikation zu verwenden.

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

 © 2019-2025 Altova GmbH

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.

11.13.3 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 können und ein Formular, um unsere Mitarbeiter vom technischen Support zu kontaktieren.

- 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.

- StyleVision im Internet

  Der Befehl StyleVision im Internet ist ein Link zur Altova Website im Internet. Hier erfahren Sie mehr über StyleVision und verwandte Technologien und Produkte auf der Altova Website.

- Über StyleVision

  Mit dem Befehl Über StyleVision wird das Willkommensfenster und die Versionsnummer Ihres Produkts angezeigt. Wenn Sie die 64-Bit-Verson von StyleVision verwenden, wird dies durch das Suffix (x64) nach dem Applikationsnamen angezeigt. Die 32-Bit-Version hat kein Suffix.

12 Anhänge

Diese Anhänge enthalten (i) Informationen über die in StyleVision verwendeten XSLT-Prozessoren; (ii) Informationen über die Konvertierung von Datenbanktypen in XML-Schema-Datentypen; (iii) technische Informationen über StyleVision; und (iv) Lizenzinformationen zu StyleVision. Die Anhänge sind in die unten aufgeführten Abschnitte gegliedert.

XSLT-Prozessorinformationen 567
Enthält implementierungsspezifische Informationen zu den Altova XSLT-Prozessoren, welche von StyleVision zur Generierung der Ausgabe verwendet werden.

- Altova XSLT 1.0-Prozessor
- Altova XSLT 2.0-Prozessor
- Altova XSLT 3.0-Prozessor
- XSLT- und XPath/XQuery-Funktionen

Technische Daten 683
Enthält technische Informationen zu StyleVision.

- Betriebssystem und Arbeitsspeicheranforderungen
- Altova XML Parser
- Altova XSLT- und XQuery-Prozessoren
- Unicode-Unterstützung
- Internet-Verwendung

Lizenzinformationen 686
Enthält Informationen über Vertrieb und Lizenzierung von StyleVision.

- Electronic Software Distribution
- Lizenzüberwachung
- Copyright
- Endbenutzerlizenzvereinbarung

© 2019-2025 Altova GmbH

12.1 Informationen zum XSLT- und XQuery-Prozessor

Der XSLT- und der XQuery-Prozessor von StyleVision hält sich genau an die W3C-Spezifikationen und ist daher strenger als die früheren Altova-Prozessoren, wie z.B. die in frühere Versionen von XMLSpy integrierten . Infolgedessen werden auch leichte Fehler, die von früheren Prozessoren ignoriert wurden, von StyleVision als Fehler gekennzeichnet.

Zum Beispiel:

- Wenn das Ergebnis eines Pfad-Operators sowohl Nodes als auch Nicht-Nodes enthält, wird ein Typfehler (`err:XPTY0018`) ausgegeben.
- Wenn `E1` in einem Pfadausdruck `E1/E2` nicht zu einer Node-Sequenz ausgewertet wird, wird ein Typfehler (`err:XPTY0019`) ausgegeben.

Ändern Sie bei Auftreten eines solchen Fehlers je nach Bedarf, entweder das XSLT/XQuery-Dokument oder das Instanzdokument.

In diesem Abschnitt sind implementierungsspezifische Funktionalitäten der Prozessoren geordnet nach Spezifikation beschrieben:

- XSLT 1.0 567
- XSLT 2.0 567
- XSLT 3.0 569
- XQuery 1.0 571
- XQuery 3.1 574

12.1.1 XSLT 1.0

Der XSLT 1.0-Prozessor von StyleVision entspricht der XSLT 1.0 Recommendation vom 16. November 1999 und der XPath 1.0 Recommendation vom 16. November 1999 des World Wide Web Consortium (W3C). Beachten Sie die folgenden Informationen zur Implementierung.

Anmerkungen zur Implementierung

Wenn das `method`-Attribut von `xsl:output` auf HTML gesetzt ist oder wenn standardmäßig die HTML-Ausgabe ausgewählt ist, werden Sonderzeichen in der XML- oder XSLT-Datei als HTML-Zeichenreferenzen in das HTML-Ausgabedokument eingefügt. So wird z.B. das Zeichen U+00A0 (die hexadezimale Zeichenreferenz für ein geschütztes Leerzeichen) entweder als Zeichenreferenz (` ` or ` `) oder als Entity-Referenz ` ` in den HTML-Code eingefügt.

12.1.2 XSLT 2.0

In diesem Abschnitt:

- Prozessorkonformität 568
- Rückwärtskompatibilität 568

- Namespaces 568
- Schema-Fähigkeit 569
- Implementierungsspezifisches Verhalten 569

Standardkonformität

Der XSLT 2.0-Prozessor von StyleVision entspricht der XSLT 2.0 Recommendation vom 23. Jänner 2007 und der XPath 2.0 Recommendation vom 14. Dezember 2010 des World Wide Web Consortium (W3C).

Rückwärtskompatibilität

Der XSLT 2.0-Prozessor ist rückwärtskompatibel. Normalerweise kommt die Rückwärtskompatibilität des XSLT 2.0.-Prozessors nur dann zum Einsatz, wenn Sie den XSLT 2.0-Prozessor zur Verarbeitung eines XSLT 1.0 Stylesheets oder einer XSLT 1.0-Anweisung verwenden. Beachten Sie, dass sich das Ergebnis des XSLT 1.0-Prozessors und des rückwärtskompatiblen XSLT 2.0.Prozessors unter Umständen unterscheiden kann.

Namespaces

In Ihrem XSLT 2.0 Stylesheet sollten die folgenden Namespaces deklariert sein, damit Sie die in XSLT 2.0 verfügbaren Typ-Konstruktoren und Funktionen verwenden können. Normalerweise werden die unten aufgelisteten Präfixe verwendet; bei Bedarf können Sie auch andere Präfixe verwenden.

Namespace Name	Präfix	Namespace URI
XML Schema-Typen	`xs:`	`http://www.w3.org/2001/XMLSchema`
XPath 2.0-Funktionen	`fn:`	`http://www.w3.org/2005/xpath-functions`

Normalerweise werden diese Namespaces im Element `xsl:stylesheet` oder `xsl:transform` deklariert, wie unten gezeigt:

```
<xsl:stylesheet version="2.0"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:fn="http://www.w3.org/2005/xpath-functions"
   ...
</xsl:stylesheet>
```

Beachten Sie die folgenden Punkte:

- Der XSLT 2.0-Prozessor verwendet als **Standard-Funktions-Namespace** den Namespace für XPath 2.0- und XQuery 1.0-Funktionen (siehe Tabelle oben). Sie können daher XPath 2.0- und XSLT 2.0-Funktionen in Ihrem Stylesheet ohne Präfix verwenden. Wenn Sie den Namespace für XPath 2.0-Funktionen in Ihrem Stylesheet mit einem Präfix deklarieren, können Sie zusätzlich dazu das in der Deklaration zugewiesene Präfix verwenden.
- Bei Verwendung von Typ-Konstruktoren und Typen aus dem XML Schema-Namespace, muss bei Aufruf des Typ-Konstruktors (z.B. `xs:date`) das in der jeweiligen Namespace-Deklaration verwendeten Präfix verwendet werden.
- Einige XPath 2.0-Funktionen haben denselben Namen wie XML Schema-Datentypen. So gibt es z.B. für die XPath-Funktionen `fn:string` und `fn:boolean` XML-Schema-Datentypen mit demselben lokalen Namen: `xs:string` und `xs:boolean`. Wenn Sie daher den XPath-Ausdruck `string('Hello')` verwenden, wird der Ausdruck als `fn:string('Hello')` ausgewertet und nicht als `xs:string('Hello')`.

© 2019-2025 Altova GmbH

Schemafähigkeit

Der XSLT 2.0-Prozessor ist schemafähig. Sie können daher benutzerdefinierte Schematypen und die `xsl:validate`-Anweisung verwenden.

Implementierungsspezifisches Verhalten

Im Folgenden finden Sie eine Beschreibung, wie der XSLT 2.0-Prozessor implementierungsspezifische Aspekte von bestimmten XSLT 2.0-Funktionen behandelt.

`xsl:result-document`
Zusätzlich werden die folgenden Kodierungen unterstützt: `x-base16tobinary` und `x-base64tobinary`.

`function-available`
Die Funktion überprüft, ob in-scope-Funktionen (XSLT, XPath und Erweiterungsfunktionen) verfügbar sind.

`unparsed-text`
Das Attribut `href` akzeptiert (i) relative Pfade für Dateien im Basis-URI-Ordner und (ii) absolute Pfade mit oder ohne das `file://`-Protokoll. Zusätzlich werden die folgenden (Altova-spezifischen) Kodierungen unterstützt: `binarytobase16` und `binarytobase64`. Beispiel: `xs:base64Binary(unparsed-text('chart.png', 'x-binarytobase64'))`.

`unparsed-text-available`
Das Attribut `href` akzeptiert (i) relative Pfade für Dateien im Basis-URI-Ordner und (ii) absolute Pfade mit oder ohne das `file://`-Protokoll. Zusätzlich werden die folgenden (Altova-spezifischen) Kodierungen unterstützt: `binarytobase16` und `binarytobase64`.

Anmerkung: Die folgenden Kodierungswerte, die in früheren Versionen von AltovaXML, dem Vorgängerprodukt von RaptorXML, verwendet wurden, werden nun nicht mehr verwendet:`base16tobinary`, `base64tobinary`, `binarytobase16` und `binarytobase64`.

12.1.3 XSLT 3.0

Der XSLT 3.0-Prozessor von StyleVision entspricht der XSLT 3.0 Recommendation vom 8. Juni 2017 und der XPath 3.1 Recommendation vom 21. März 2017 des World Wide Web Consortium (W3C).

Der XSLT 3.0-Prozessor hat dieselben implementierungsspezifischen Eigenschaften wie der XSLT 2.0-Prozessor567. Zusätzlich dazu unterstützt er eine Reihe neuer XSLT 3.0-Funktionen: XPath- und XQuery 3.1-Funktionen und Operatoren und die XPath 3.1-Spezifikation.

Anmerkung: Die optionale Streaming-Funktion wird derzeit nicht unterstützt., d.h. das gesamte Dokument wird unabhängig vom Wert des Attributs `streamable` in den Arbeitsspeicher geladen. Falls genügend Arbeitsspeicher zur Verfügung steht, (i) wird das gesamte Dokument - ohne Streaming - verarbeitet (ii) werden "guaranteed-streamable" Konstrukte korrekt verarbeitet, so als würde bei der Ausführung Streaming verwendet und (iii) Streaming-Fehler würden nicht erkannt werden. In 64-Bit-Apps sollte die Nicht-Streaming-Verarbeitung kein Problem darstellen. Falls zu wenig Arbeitsspeicher zur Verfügung steht, können Sie mehr Speicherplatz zum System hinzufügen.

Namespaces

In Ihrem XSLT 3.0 Stylesheet sollten die folgenden Namespaces deklariert sein, damit Sie alle in XSLT 3.0 verfügbaren Typ-Konstruktoren und Funktionen verwenden können. Normalerweise werden die unten aufgelisteten Präfixe verwendet; bei Bedarf können Sie auch andere Präfixe verwenden.

Namespace Name	Präfix	Namespace URI
XML Schema-Typen	`xs:`	`http://www.w3.org/2001/XMLSchema`
XPath/XQuery 3.1-Funktionen	`fn:`	`http://www.w3.org/2005/xpath-functions`
Mathematische Funktionen	`math:`	`http://www.w3.org/2005/xpath-functions/math`
Zuordnungsfunktionen	`map:`	`http://www.w3.org/2005/xpath-functions/map`
Array-Funktionen	`array:`	`http://www.w3.org/2005/xpath-functions/array`
XQuery-, XSLT- und XPath-Fehlercodes	`err:`	`http://www.w3.org/2005/xpath-functions/xqt-errors`
Serialisierungsfunktionen	`Ausgabe`	`http://www.w3.org/2010/xslt-xquery-serialization`

Normalerweise werden diese Namespaces im Element `xsl:stylesheet` oder `xsl:transform` deklariert, wie unten gezeigt:

```
<xsl:stylesheet version="3.0"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:fn="http://www.w3.org/2005/xpath-functions"
   ...
</xsl:stylesheet>
```

Beachten Sie die folgenden Punkte:

- Der XSLT 3.0-Prozessor verwendet als **Standard-Funktions-Namespace** den Namespace für XPath- und XQuery-Funktionen und Operatoren 3.1 (siehe Tabelle oben). Sie können daher die Funktionen dieses Namespace in Ihrem Stylesheet ohne Präfix verwenden. Wenn Sie den Namespace für Funktionen in Ihrem Stylesheet mit einem Präfix deklarieren, können Sie zusätzlich dazu das in der Deklaration zugewiesene Präfix verwenden.
- Bei Verwendung von Typ-Konstruktoren und Typen aus dem XML Schema-Namespace, muss bei Aufruf des Typ-Konstruktors (z.B. `xs:date`) das in der jeweiligen Namespace-Deklaration verwendeten Präfix verwendet werden.
- Einige XPath/XQuery-Funktionen haben denselben Namen wie XML Schema-Datentypen. So gibt es z.B. für die XPath-Funktionen `fn:string` und `fn:boolean` XML-Schema-Datentypen mit demselben lokalen Namen: `xs:string` und `xs:boolean`. Wenn Sie daher den XPath-Ausdruck `string('Hello')` verwenden, wird der Ausdruck als `fn:string('Hello')` ausgewertet und nicht als `xs:string('Hello')`.

© 2019-2025 Altova GmbH

12.1.4 XQuery 1.0

In diesem Abschnitt:

- Prozessorkonformität 571
- Schema-Fähigkeit 571
- Kodierung 571
- Namespaces 568
- XML-Quelle und Validierung 572
- Statische und dynamische Typüberprüfung 572
- Bibliotheksmodule 572
- Externe Funktionen 573
- Collations 573
- Präzision von numerischen Daten 573
- Unterstützung für XQuery-Anweisungen 573
- Implementierungsspezifisches Verhalten 573

Standardkonformität

Der XQuery 1.0-Prozessor von StyleVision entspricht der XQuery 1.0 Recommendation vom 14. Dezember 2010 des W3C. Der Query-Standard stellt bei vielen Funktionen frei, wie viele diese zu implementieren sind. Im Folgenden finden Sie eine Liste, wie der Altova XQuery 1.0-Prozessor diese Funktionen implementiert.

Schema-Fähigkeit

Der Altova XQuery 1.0-Prozessor ist **schemafähig.**

Kodierung

Die UTF-8 und die UTF-16 Zeichen-Kodierungen werden unterstützt.

Namespaces

Die folgenden Namespace-URIs und die damit verknüpften Bindings sind vordefiniert.

Namespace Name	Präfix	Namespace URI
XML Schema-Typen	`xs:`	`http://www.w3.org/2001/XMLSchema`
Schema-Instanz	`xsi:`	`http://www.w3.org/2001/XMLSchema-instance`
Vordefinierte Funktionen	`fn:`	`http://www.w3.org/2005/xpath-functions`
Lokale Funktionen	`local:`	`http://www.w3.org/2005/xquery-local-functions`

Beachten Sie die folgenden Punkte:

- Der Altova XQuery 1.0-Prozessor ist so konfiguriert, dass die oben aufgelisteten Präfixe an die entsprechenden Namespaces gebunden sind.
- Da der oben angeführte Namespace für vordefinierte Funktionen (siehe `fn:`) der Standard-Funktions-Namespace in XQuery ist, muss beim Aufruf von vordefinierten Funktionen das Präfix `fn:` nicht

verwendet werden (`string("Hello")` ruft z.B. die Funktion `fn:string` auf). Das Präfix `fn:` kann jedoch verwendet werden, um eine vordefinierte Funktion aufzurufen, ohne die Namespace im Abfrage-Prolog deklarieren zu müssen (z.B.: `fn:string("Hello")`).

- Sie können den Standard-Funktions-Namespace durch Deklarierung des `default function namespace`-Ausdrucks im Abfrageprolog ändern.
- Bei Verwendung von Typen aus dem XML Schema-Namespace kann das Präfix `xs:` verwendet werden, ohne dass Sie den Namespace explizit deklarieren müssen und dieses Präfix im Abfrageprolog daran binden müssen. (Beispiel: `xs:date` und `xs:yearMonthDuration`.) Wenn Sie ein anderes Präfix für den XML-Schema-Namespace verwenden möchten, muss dieses im Abfrageprolog explizit deklariert werden. (Beispiel: `declare namespace alt = "http://www.w3.org/2001/XMLSchema"; alt:date("2004-10-04")`.)
- Beachten Sie, dass die Datentypen `untypedAtomic`, `dayTimeDuration` und `yearMonthDuration` mit den Candidate Recommendations vom 23 January 2007 aus dem XPath Datentypen-Namespace in den XML-Schema Namespace verschoben wurden, d.h. `xs:yearMonthDuration`.

Wenn Namespaces für Funktionen, Typ-Konstruktoren, Node Tests usw. falsch zugewiesen wurden, wird ein Fehler ausgegeben. Beachten Sie jedoch, dass einige Funktionen denselben Namen wie Schema-Datentypen haben, z.B. `fn:string` und `fn:boolean`. (Sowohl `xs:string` als auch `xs:boolean` ist definiert.) Das Namespace-Präfix legt fest, ob die Funktion oder der Typ-Konstruktor verwendet wird.

XML-Quelldokument und Validierung

XML-Dokumente, die bei der Ausführung eines XQuery-Dokuments mit dem Altova XQuery 1.0-Prozessor verwendet werden, müssen wohlgeformt sein. Sie müssen jedoch nicht gemäß einem XML-Schema gültig sein. Wenn die Datei nicht gültig ist, wird die ungültige Datei ohne Schemainformationen geladen. Wenn die XML-Datei mit einem externen Schema verknüpft ist und gemäß diesem Schema gültig ist, werden für die XML-Daten nachträglich Validierungsinformationen generiert und für die Auswertung der Abfrage verwendet.

Statische und dynamische Typüberprüfung

In der statischen Analysephase werden Aspekte der Abfrage überprüft wie z.B. die Syntax, ob externe Referenzen (z.B. für Module) vorhanden sind, ob aufgerufene Funktionen und Variablen definiert sind, usw. Wenn in dieser Phase ein Fehler gefunden wird, wird eine Meldung ausgegeben und die Ausführung wird gestoppt.

Die dynamische Typ-Überprüfung wird zur Laufzeit durchgeführt, während die Abfrage ausgeführt wird. Wenn ein Typ mit den Anforderungen einer Operation nicht kompatibel ist, wird ein Fehler ausgegeben. So gibt z.B. der Ausdruck `xs:string("1") + 1` einen Fehler zurück, weil die Operation "Addition" nicht an einem Operanden vom Typ `xs:string` ausgeführt werden kann.

Bibliotheksmodule

Bibliotheksmodule dienen zum Speichern von Funktionen und Variablen, damit diese wiederverwendet werden können. Der Altova XQuery 1.0-Prozessor unterstützt Module, die in einer **einzigen externen XQuery-Datei** gespeichert sind. Eine solche Moduldatei muss im Prolog eine `module`-Deklaration enthalten, in der ein Target Namespace zugewiesen wird. Hier ein Beispielmodul:

```
module namespace libns="urn:module-library";
declare variable $libns:company := "Altova";
declare function libns:webaddress() { "http://www.altova.com" };
```

Alle im Modul deklarierten Funktionen und Variablen gehören zu dem mit dem Modul verknüpften Namespace. Das Modul wird durch Import in eine XQuery-Datei mittels der `import module`-Anweisung im Abfrageprolog

 © 2019-2025 Altova GmbH

verwendet. Die `import module`-Anweisung importiert nur Funktionen und Variablen, die direkt in der Bibliotheksmodul-Datei deklariert sind:

```
import module namespace modlib = "urn:module-library" at "modulefilename.xq";
if         ($modlib:company = "Altova")
then        modlib:webaddress()
else        error("No match found.")
```

External functions

Externe Funktionen, d.h. diejenigen Funktionen, die das Schlüsselwort `external` verwenden, werden nicht unterstützt:

```
declare function hoo($param as xs:integer) as xs:string external;
```

Collations

Die Standard-Collation ist die Unicode Codepoint Collation, die Strings auf Basis ihrer Unicode-Codepunkte vergleicht. Andere unterstützte Collations sind die hier[576] aufgelisteten ICU-Collations. Um eine bestimmte Collation zu verwenden, geben Sie ihre in der Liste der unterstützten Collations[576] angeführte URI an. String-Vergleiche, wie die Funktionen `fn:max` und `fn:min` werden anhand der angegebenen Collation durchgeführt. Wenn die Collation-Option nicht definiert ist, wird die Standard-Unicode Codepoint Collation verwendet.

Präzision von numerischen Typen

- Der Datentyp `xs:integer` hat eine beliebige Präzision, d.h. er kann beliebig viele Stellen haben.
- Der Datentyp `xs:decimal` kann nach dem Dezimalpunkt maximal 20 Stellen haben.
- Die Datentypen `xs:float` und `xs:double` sind auf 15 Stellen beschränkt.

Unterstützung für XQuery-Anweisungen

Die `Pragma`-Anweisung wird nicht unterstützt. Gegebenenfalls wird sie ignoriert und der Fallback-Ausdruck wird evaluiert.

Implementierungsspezifisches Verhalten

Im Folgenden wird beschrieben, wie der XQuery- und der XQuery Update 1.0-Prozessor implementierungsspezifische Aspekte bestimmter Funktionen behandeln.

`unparsed-text`
Das Attribut `href` akzeptiert (i) relative Pfade für Dateien im Basis-URI-Ordner und (ii) absolute Pfade mit oder ohne das `file://`-Protokoll. Zusätzlich werden die folgenden (Altova-spezifischen) Kodierungen unterstützt: `binarytobase16` und `binarytobase64`. Beispiel: `xs:base64Binary(unparsed-text('chart.png', 'x-binarytobase64'))`.

`unparsed-text-available`
Das Attribut `href` akzeptiert (i) relative Pfade für Dateien im Basis-URI-Ordner und (ii) absolute Pfade mit oder ohne das `file://`-Protokoll. Zusätzlich werden die folgenden (Altova-spezifischen) Kodierungen unterstützt: `binarytobase16` und `binarytobase64`.

Anmerkung: Die folgenden Kodierungswerte, die in früheren Versionen von AltovaXML, dem Vorgängerprodukt von RaptorXML, verwendet wurden, werden nun nicht mehr verwendet:`base16tobinary`, `base64tobinary`,

`binarytobase16` und `binarytobase64`.

12.1.5 XQuery 3.1

Der XQuery 3.1-Prozessor von StyleVision entspricht der XQuery 3.1 Recommendation vom 21. März 2017 des World Wide Web Consortium (W3C) und unterstützt XPath- und XQuery-Funktionen 3.1. Die XQuery 3.1-Spezifikation ist eine Übermenge der 3.0-Spezifikation. Daher unterstützt der XQuery 3.1-Prozessor XQuery 3.0-Funktionen..

Namespaces

In Ihrem XQuery 3.1 Stylesheet sollten die folgenden Namespaces deklariert sein, damit Sie die in XQuery 3.1 verfügbaren Typ-Konstruktoren und Funktionen verwenden können. Normalerweise werden die unten aufgelisteten Präfixe verwendet; bei Bedarf können Sie auch andere Präfixe verwenden.

Namespace Name	Präfix	Namespace URI
XML Schema-Typen	`xs:`	`http://www.w3.org/2001/XMLSchema`
XPath/XQuery 3.1-Funktionen	`fn:`	`http://www.w3.org/2005/xpath-functions`
Mathematische Funktionen	`math:`	`http://www.w3.org/2005/xpath-functions/math`
Zuordnungsfunktionen	`map:`	`http://www.w3.org/2005/xpath-functions/map`
Array-Funktionen	`array:`	`http://www.w3.org/2005/xpath-functions/array`
XQuery-, XSLT- und XPath-Fehlercodes	`err:`	`http://www.w3.org/2005/xpath-functions/xqt-errors`
Serialisierungsfunktionen	`output`	`http://www.w3.org/2010/xslt-xquery-serialization`

Beachten Sie die folgenden Punkte:

- Der Altova XQuery 3.1-Prozessor ist so konfiguriert, dass die oben aufgelisteten Präfixe an die entsprechenden Namespaces gebunden sind.
- Da der oben angeführte Namespace für vordefinierte Funktionen (siehe `fn:`) der Standard-Funktions-Namespace in XQuery ist, muss beim Aufruf von vordefinierten Funktionen das Präfix `fn:` nicht verwendet werden (`string("Hello")` ruft z.B. die Funktion `fn:string` auf). Das Präfix `fn:` kann jedoch verwendet werden, um eine vordefinierte Funktion aufzurufen, ohne die Namespace im Abfrage-Prolog deklarieren zu müssen (z.B.: `fn:string("Hello")`).
- Sie können den Standard-Funktions-Namespace durch Deklarierung des `default function namespace`-Ausdrucks im Abfrageprolog ändern.
- Bei Verwendung von Typen aus dem XML Schema-Namespace kann das Präfix `xs:` verwendet werden, ohne dass Sie den Namespace explizit deklarieren müssen und dieses Präfix im Abfrageprolog daran binden müssen. (Beispiel: `xs:date` und `xs:yearMonthDuration`.) Wenn Sie ein anderes Präfix für den XML-Schema-Namespace verwenden möchten, muss dieses im Abfrageprolog explizit deklariert werden. (Beispiel: `declare namespace alt = "http://www.w3.org/2001/XMLSchema"; alt:date("2004-10-04")`.)

© 2019-2025 Altova GmbH

Wenn Namespaces für Funktionen, Typ-Konstruktoren, Node Tests usw. falsch zugewiesen wurden, wird ein Fehler ausgegeben. Beachten Sie jedoch, dass einige Funktionen denselben Namen wie Schema-Datentypen haben, z.B. `fn:string` und `fn:boolean`. (Sowohl `xs:string` als auch `xs:boolean` ist definiert.) Das Namespace-Präfix legt fest, ob die Funktion oder der Typ-Konstruktor verwendet wird.

Implementierungsspezifisches Verhalten

Die implementierungsspezifischen Eigenschaften sind dieselben wie für XQuery 1.0 571.

Zusätzlich dazu kann die Altova-spezifische Kodierung `x-base64tobinary` verwendet werden, um ein Binärdokument wie z.B. ein Bild zu erzeugen.

12.2 XSLT- und XPath/XQuery-Funktionen

Dieser Abschnitt enthält eine Liste von Altova-Erweiterungsfunktionen und anderen Erweiterungsfunktionen, die in XPath und/oder XQuery-Ausdrücken verwendet werden können. Itova-Erweiterungsfunktionen können mit dem XSLT- und XQuery-Prozessor von Altova verwendet werden und bieten zusätzliche Funktionalitäten zu den in den W3C-Standards definierten Funktionsbibliotheken.

In diesem Abschnitt werden hauptsächlich XPath/XQuery-Erweiterungsfunktionen, die von Altova für zusätzliche Operationen sowie andere Erweiterungsfunktionen[657] erstellt wurden, beschrieben. Diese Erweiterungsfunktionen[577] können vom Altova-XSLT- und XQuery-Prozessor gemäß den in diesem Abschnitt beschriebenen Regeln verarbeitet werden. Informationen zu den regulären XPath/XQuery-Funktionen finden Sie in der Altova XPath/XQuery-Funktionsreferenz

Allgemeine Punkte

Beachten Sie bitte die folgenden allgemeinen Punkte:

- Funktionen aus den in den W3C-Spezifikationen definierten core-Funktionsbibliotheken können ohne Präfix aufgerufen werden, da der Altova-XSLT- und XQuery-Prozessor Funktionen, die kein Präfix haben, als Funktionen des in der XPath/XQuery Functions-Spezifikation Standard-Funktions-Namespace `http://www.w3.org/2005/xpath-functions` liest. Wenn dieser Namespace in einem XSLT- oder XQuery-Dokument explizit deklariert ist, kann das in der Namespace-Deklaration definierte Präfix optional auch in Funktionsnamen verwendet werden.
- Wenn bei einer Funktion eine Sequenz von einem Datenelement als Argument erwartet wird und eine Sequenz von mehr als einem Datenelement gesendet wird, wird ein Fehler zurückgegeben.
- Alle String-Vergleiche werden unter Verwendung der Unicode Codepoint Collation ausgeführt.
- Ergebnisse, bei denen es sich um QNames handelt, werden in der Form `[prefix:]localname` serialisiert.

Präzision von xs:decimal

Die Präzision bezieht sich auf die Anzahl der Stellen in einer Zahl. Laut Spezifikation sind mindestens 18 Stellen erforderlich. Bei Divisionen, bei denen ein Ergebnis vom Typ `xs:decimal` erzeugt wird, beträgt die Präzision 19 Kommastellen ohne Runden.

Implizite Zeitzone

Beim Vergleich zweier `date`, `time`, oder `dateTime`-Werte muss die Zeitzone der verglichenen Werte bekannt sein. Wenn die Zeitzone in einem solchen Wert nicht explizit angegeben ist, wird die implizite Zeitzone verwendet. Als implizite Zeitzone wird die der Systemuhr verwendet. Der Wert kann mit Hilfe der Funktion `implicit-timezone()` überprüft werden.

Collations

Die Standard-Collation ist die Unicode Codepoint Collation, die Strings auf Basis ihrer Unicode-Codepunkte vergleicht. Der Prozessor verwendet den Unicode Collation-Algorithmus. Andere unterstützte Collations sind die hier aufgelisteten ICU-Collations. Um eine bestimmte Collation zu verwenden, geben Sie Ihre URI an (siehe Tabelle unten). String-Vergleiche, wie die Funktionen `fn:max` und `fn:min` werden anhand der angegebenen Collation durchgeführt. Wenn die Collation-Option nicht definiert ist, wird die Standard-Unicode Codepoint Collation verwendet.

© 2019-2025 Altova GmbH

Sprache	URIs
da: Dänisch	da_DK
de: Deutsch	de_AT, de_BE, de_CH, de_DE, de_LI, de_LU
en: Englisch	en_AS, en_AU, en_BB, en_BE, en_BM, en_BW, en_BZ, en_CA, en_GB, en_GU, en_HK, en_IE, en_IN, en_JM, en_MH, en_MP, en_MT, en_MU, en_NA, en_NZ, en_PH, en_PK, en_SG, en_TT, en_UM, en_US, en_VI, en_ZA, en_ZW
es: Spanisch	es_419, es_AR, es_BO, es_CL, es_CO, es_CR, es_DO, es_EC, es_ES, es_GQ, es_GT, es_HN, es_MX, es_NI, es_PA, es_PE, es_PR, es_PY, es_SV, es_US, es_UY, es_VE
fr: Französisch	fr_BE, fr_BF, fr_BI, fr_BJ, fr_BL, fr_CA, fr_CD, fr_CF, fr_CG, fr_CH, fr_CI, fr_CM, fr_DJ, fr_FR, fr_GA, fr_GN, fr_GP, fr_GQ, fr_KM, fr_LU, fr_MC, fr_MF, fr_MG, fr_ML, fr_MQ, fr_NE, fr_RE, fr_RW, fr_SN, fr_TD, fr_TG
it: Italienisch	it_CH, it_IT
ja: Japanisch	ja_JP
nb: Norwegisch (Bokmal)	nb_NO
nl: Holländisch	nl_AW, nl_BE, nl_NL
nn: Norwegisch (Nynorsk)	nn_NO
pt: Portugiesisch	pt_AO, pt_BR, pt_GW, pt_MZ, pt_PT, pt_ST
ru: Russisch	ru_MD, ru_RU, ru_UA
sv: Schwedisch	sv_FI, sv_SE

Namespace-Achse
Die Namespace-Achse wird in XPath 2.0 nicht mehr verwendet, wird aber weiterhin unterstützt. Um Namespace-Informationen mit XPath 2.0-Mechanismen aufzurufen, verwenden Sie die Funktionen `in-scope-prefixes()`, `namespace-uri()` und `namespace-uri-for-prefix()`.

12.2.1 Altova-Erweiterungsfunktionen

Altova-Erweiterungsfunktionen können in XPath/XQuery-Ausdrücken verwendet werden. Dadurch stehen neben den Funktionen in der Standardbibliothek der XPath-, XQuery- und XSLT-Funktionen zusätzliche Funktionen zur Verfügung. Die Altova-Erweiterungsfunktionen befinden sich im **Altova-Erweiterungsfunktions-Namespace, `http://www.altova.com/xslt-extensions`** und sind in diesem Abschnitt mit dem Präfix **`altova:`**, das als an diesen Namespace gebunden angenommen wird, gekennzeichnet. Beachten Sie, dass manche Funktionen in zukünftigen Versionen Ihres Produkts eventuell nicht mehr unterstützt werden oder dass sich das Verhalten einzelner Funktionen ändern kann. Um zu sehen, welche Altova-Erweiterungsfunktionen unterstützt werden, lesen Sie bitte die Dokumentation zur jeweiligen Release.

Die in der "XPath/XQuery Functions"-Spezifikation des W3C definierten Funktionen können (i) in einem XSLT-Kontext in XPath-Ausdrücken und (ii) in einem XQuery-Dokument in XQuery-Ausdrücken verwendet werden. In dieser Dokumentation sind die Funktionen, die im Zusammenhang mit XPath in XSLT verwendet werden können, mit einem **XP**-Symbol und Funktionen, die im Zusammenhang mit XQuery verwendet werden können, mit einem **XQ**-Symbol markiert; sie fungieren als XQuery-Funktionen. In den XSLT-Spezifikationen des W3C (nicht in den "XPath/XQuery Functions"-Spezifikationen) sind außerdem Funktionen definiert, die in XSLT-Dokumenten in XPath-Ausdrücken verwendet werden können. Diese Funktionen sind mit dem Symbol **XSLT** gekennzeichnet und werden als XSLT-Funktionen bezeichnet. In welcher XPath/XQuery- und XSLT-Version eine Funktion verwendet werden kann, wird in der Beschreibung der Funktion (*siehe Symbole unten)* angegeben. Funktionen aus der XPath/XQuery- und XSLT-Funktionsbibliothek werden ohne Präfix aufgelistet. Erweiterungsfunktionen aus anderen Bibliotheken wie z.B. Altova-Erweiterungsfunktionen werden mit einem Präfix angegeben.

XPath-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	**XP1** **XP2** **XP3.1**
XSLT-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	**XSLT1** **XSLT2** **XSLT3**
XQuery-Funktionen (in XQuery-Ausdrücken in XQuery verwendet):	**XQ1** **XQ3.1**

Verwendung von Altova-Erweiterungsfunktionen

Um Altova-Erweiterungsfunktionen verwenden zu können, müssen Sie den Altova-Erweiterungsfunktions-Namespace deklarieren (*erster markierter Bereich im Codefragment unten*) und die Erweiterungsfunktionen anschließend so verwenden, dass sie als zu diesem Namespace gehörig aufgelöst werden (*siehe zweiter markierter Bereich*). Im Beispiel unten wird die Altova-Erweiterungsfunktion **age** verwendet.

```
<xsl:stylesheet version="2.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:fn="http://www.w3.org/2005/xpath-functions"
    xmlns:altova="http://www.altova.com/xslt-extensions">
    <xsl:output method="text" encoding="ISO-8859-1"/>
    <xsl:template match="Persons">
        <xsl:for-each select="Person">
            <xsl:value-of select="concat(Name, ': ')"/>
            <xsl:value-of select="altova:age(xs:date(BirthDate))"/>
            <xsl:value-of select="' years&#x0A;'"/>
        </xsl:for-each>
    </xsl:template>
</xsl:stylesheet>
```

XSLT-Funktionen 579

XSLT-Funktionen können in XPath-Ausdrücken nur im XSLT-Kontext verwendet werden (ähnlich wie die XSLT 2.0-Funktionen `current-group()` oder `key()`). Diese Funktionen sind nicht für Nicht-XSLT-Kontext gedacht und funktionieren in einem solchen Kontext (z.B. in einem XQuery-Kontext) nicht. Beachten Sie, dass XSLT-Funktionen für XBRL nur mit Altova Produkten verwendet werden können, die XBRL unterstützen.

© 2019-2025 Altova GmbH

XPath/XQuery-Funktionen

XPath/XQuery-Funktionen können sowohl in XPath-Ausdrücken im XSLT-Kontext als auch in XQuery-Ausdrücken verwendet werden.

- Datum/Uhrzeit 582
- Standort 600
- Bildbezogene 612
- Numerisch 617
- Schema 619
- Sequenz 639
- String 648
- Verschiedenes 654

12.2.1.1 XSLT-Funktionen

XSLT-Erweiterungsfunktionen können in XPath-Ausdrücken in einem XSLT-Kontext verwendet werden. In einem Nicht-XSLT-Kontext (z.B. in einem XQuery-Kontext) funktionieren sie nicht.

Anmerkung zur Benennung von Funktionen und zur Anwendbarkeit der Sprache

Altova-Erweiterungsfunktionen können in XPath/XQuery-Ausdrücken verwendet werden. Dadurch stehen neben den Funktionen in der Standardbibliothek der XPath-, XQuery- und XSLT-Funktionen zusätzliche Funktionen zur Verfügung. Die Altova-Erweiterungsfunktionen befinden sich im **Altova-Erweiterungsfunktions-Namespace,** `http://www.altova.com/xslt-extensions` und sind in diesem Abschnitt mit dem Präfix `altova:`, das als an diesen Namespace gebunden angenommen wird, gekennzeichnet. Beachten Sie, dass manche Funktionen in zukünftigen Versionen Ihres Produkts eventuell nicht mehr unterstützt werden oder dass sich das Verhalten einzelner Funktionen ändern kann. Um zu sehen, welche Altova-Erweiterungsfunktionen unterstützt werden, lesen Sie bitte die Dokumentation zur jeweiligen Release.

XPath-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	XP1 XP2 XP3.1
XSLT-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	XSLT1 XSLT2 XSLT3
XQuery-Funktionen (in XQuery-Ausdrücken in XQuery verwendet):	XQ1 XQ3.1

Allgemeine Funktionen

▼ distinct-nodes [altova:]

altova:distinct-nodes(*node()****) als node()*** XSLT1 XSLT2 XSLT3

Erhält eine Gruppe von einem oder mehreren Nodes als Input und gibt dieselbe Gruppe ohne Nodes mit doppelt vorhandenen Werten zurück. Der Vergleich wird mittels der XPath/XQuery-Funktion `fn:deep-equal` durchgeführt.

⊟ *Beispiele*

- **altova:distinct-nodes**(country) gibt alle Child `country` Nodes ohne diejenigen mit doppelt vorhandenen Werten zurück.

evaluate [altova:]

altova:evaluate(XPathExpression *as xs:string*[, **ValueOf$p1, ... ValueOf$pN]) XSLT1 XSLT2 XSLT3**

Erhält einen XPath-Ausdruck als obligatorisches Argument, der als String übergeben wird, und gibt das Resultat des ausgewerteten Ausdrucks zurück. Beispiel: **altova:evaluate('//Name[1]')** gibt den Inhalt des ersten `Name` Elements im Dokument zurück. Beachten Sie, dass der Ausdruck `//Name[1]` durch Einschließen in einfache Anführungszeichen als String übergeben wird.

Die Funktion `altova:evaluate` kann zusätzliche (optionale) Argumente erhalten. Diese Argumente sind die Werte der einzelnen im Geltungsbereich befindlichen Variablen und haben die Namen `p1`, `p2`, `p3`... `pN`. Beachten Sie zur Verwendung die folgenden Punkte: (i) Die Variablennamen müssen die Form `pX` haben, wobei `X` eine Ganzzahl ist; (ii) die Argumente der Funktion `altova:evaluate` (*siehe Signatur oben*) liefern vom zweiten Argument an die Werte der Variablen, wobei die Reihenfolge der Argumente der numerisch geordneten Variablensequenz entspricht: `p1` bis `pN`. Das zweite Argument wird der Wert der Variablen `p1`, das dritte Argument der der Variablen `p2`, usw.; (iii) Die Werte der Variablen müssen vom Typ `item*` sein

Beispiel

```
<xsl:variable name="xpath" select="'$p3, $p2, $p1'" />
<xsl:value-of select="altova:evaluate($xpath, 10, 20, 'hi')" />
```

gibt aus `"hi 20 10"`

Beachten Sie im obigen Beispiel folgende Punkte:

- Das zweite Argument des Ausdrucks `altova:evaluate` ist der der Variablen `$p1` zugewiesene Wert, das dritte Argument ist das der Variablen `$p2` zugewiesene usw.
- Beachten Sie, dass das vierte Argument der Funktion ein String-Wert ist. Als String-Wert wird dieser innerhalb von Anführungszeichen gesetzt.
- Das `select` Attribut des Elements `xs:variable` liefert den XPath-Ausdruck. Da dieser Ausdruck den Typ `xs:string`, haben muss, wird er in einfache Anführungszeichen gesetzt.

Weitere Beispiele für die Verwendung der Variablen

- ```
  <xsl:variable name="xpath" select="'$p1'" />
  <xsl:value-of select="altova:evaluate($xpath, //Name[1])" />
  ```
  Gibt den Wert des ersten `Name` *Elements zurück.*

- ```
  <xsl:variable name="xpath" select="'$p1'" />
  <xsl:value-of select="altova:evaluate($xpath, '//Name[1]')" />
  ```
  Gibt "`//Name[1]`*" aus*

Die `altova:evaluate()` Erweiterungsfunktion ist in Situationen nützlich, in denen ein XPath-Ausdruck im XSLT-Stylesheet einen oder mehrere Teile enthält, die dynamisch ausgewertet werden müssen. Angenommen ein Benutzer gibt seinen Request für das Sortierkriterium ein und das Sortierkriterium ist im Attribut `UserReq/@sortkey` gespeichert. Im Stylesheet könnten Sie den folgenden Ausdruck haben: `<xsl:sort select="altova:evaluate(../UserReq/@sortkey)" order="ascending"/>` Die `altova:evaluate()` Funktion liest das `sortkey` Attribut des `UserReq` Child-Elements des Parent des Kontext-Node. Angenommen der Wert des `sortkey` Attributs ist `Price`, dann wird von der `altova:evaluate()` Funktion `Price` zurückgegeben und wird zum Wert des `select` Attributs:

© 2019-2025 Altova GmbH

`<xsl:sort select="Price" order="ascending"/>`. Wenn diese `sort` Anweisung im Kontext eines Elements namens `Order` vorkommt, dann werden die `Order` Elemente nach den Werten Ihrer `Price` Children sortiert. Alternativ dazu, wenn der Wert von `@sortkey` z.B. `Date` ist, werden die `Order` Elemente nach den Werten ihrer `Date` Children sortiert. Das Sortierkriterium für `Order` wird also zur Laufzeit aus dem `sortkey` Attribut ausgewählt. Diese hätte man mit einem Ausdruck wie dem folgenden nicht bewerkstelligen können: `<xsl:sort select="../UserReq/@sortkey" order="ascending"/>`. Im oben gezeigten Beispiel wäre das Sortierkriterium das `sortkey` Attribut selbst, nicht `Price` oder `Date` (oder jeder beliebige andere Inhalt von `sortkey`)

Hinweis:
Der statische Kontext enthält Namespaces, Typen und Funktionen - aber keine Variablen - aus der aufrufenden Umgebung. Die Basis-URI und der Standard-Namespace werden vererbt.

Weitere Beispiele

- Statische Variablen: `<xsl:value-of select="$i3, $i2, $i1" />`
  Gibt die Werte von drei Variablen aus.

- Dynamischer XPath-Ausdruck mit dynamischen Variablen:
  ```
  <xsl:variable name="xpath" select="'$p3, $p2, $p1'" />
  <xsl:value-of select="altova:evaluate($xpath, 10, 20, 30)" />
  ```
  Gibt "30 20 10" aus

- Dynamischer XPath-Ausdruck ohne dynamische Variable:
  ```
  <xsl:variable name="xpath" select="'$p3, $p2, $p1'" />
  <xsl:value-of select="altova:evaluate($xpath)" />
  ```
  Gibt einen Fehler aus.: Es wurde keine Variable für `$p3` definiert.

encode-for-rtf [altova:]

altova:encode-for-rtf(input *als xs:string*, **preserveallwhitespace** *als xs:boolean*, **preservenewlines** *als xs:boolean*) **als xs:string** XSLT2 XSLT3

Konvertiert den Input-String in Code für RTF. Whitespaces und neue Zeilen werden gemäß dem für die entsprechenden Parameter definierten Booleschen Wert beibehalten.

[Nach oben 579]

XBRL-Funktionen

Altova XBRL-Funktionen können nur mit Editionen von Altova-Produkten verwendet werden, die XBRL unterstützen.

xbrl-footnotes [altova:]

altova:xbrl-footnotes(*node()***)** als **node()*** XSLT2 XSLT3

Erhält einen Node als Input-Argument und gibt die durch den Input-Node referenzierte Gruppe der XBRL-Fußnoten-Nodes zurück.

▼ xbrl-labels [altova:]

altova:xbrl-labels(*xs:QName, xs:string***)** als **node()*** XSLT2 XSLT3
Erhält zwei Input-Argumente: einen Node-Namen und den Pfad der Taxonomiedatei, die den Node enthält. Die Funktion gibt die XBRL Labels zurück, die mit dem Input-Node verknüpft sind.

[Nach oben 579]

12.2.1.2 XPath/XQuery-Funktionen: Datum und Uhrzeit

Die Datums- und Uhrzeit-Erweiterungsfunktionen von Altova können im Zusammenhang mit XPath- und XQuery-Ausdrücken verwendet werden und bieten zusätzliche Funktionalitäten für die Verarbeitung von Daten, die in Form von XML-Schema-Datums- und Uhrzeit-Datentypen zur Verfügung stehen. Die Funktionen in diesem Abschnitt können mit dem **XPath 3.0** - und dem **XQuery 3.0** -Prozessor von Altova verwendet werden. Sie stehen in verschiedenen XPath/XQuery-Kontexten zur Verfügung.

Anmerkung zur Benennung von Funktionen und zur Anwendbarkeit der Sprache

Altova-Erweiterungsfunktionen können in XPath/XQuery-Ausdrücken verwendet werden. Dadurch stehen neben den Funktionen in der Standardbibliothek der XPath-, XQuery- und XSLT-Funktionen zusätzliche Funktionen zur Verfügung. Die Altova-Erweiterungsfunktionen befinden sich im **Altova-Erweiterungsfunktions-Namespace, `http://www.altova.com/xslt-extensions`** und sind in diesem Abschnitt mit dem Präfix **`altova:`**, das als an diesen Namespace gebunden angenommen wird, gekennzeichnet. Beachten Sie, dass manche Funktionen in zukünftigen Versionen Ihres Produkts eventuell nicht mehr unterstützt werden oder dass sich das Verhalten einzelner Funktionen ändern kann. Um zu sehen, welche Altova-Erweiterungsfunktionen unterstützt werden, lesen Sie bitte die Dokumentation zur jeweiligen Release.

XPath-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	XP1 XP2 XP3.1
XSLT-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	XSLT1 XSLT2 XSLT3
XQuery-Funktionen (in XQuery-Ausdrücken in XQuery verwendet):	XQ1 XQ3.1

▼ Nach Funktionalität gruppiert

- Hinzufügen einer Zeitdauer zu xs:dateTime und Rückgabe von xs:dateTime 583
- Hinzufügen einer Zeitdauer zu xs:date und Rückgabe von xs:date 585
- Hinzufügen einer Zeitdauer zu xs:time und Rückgabe von xs:time 587
- Formatieren und Abrufen einer Zeitdauer 586
- Entfernen der Zeitzone aus Funktionen, die das aktuelle Datum/die aktuelle Uhrzeit generieren 588
- Rückgabe von Tagen, Stunden, Minuten und Sekunden anhand einer Zeitdauer 589
- Rückgabe des Wochentags anhand des Datums als Ganzzahl 591
- Rückgabe eines Wochentags als Ganzzahl anhand eines Datums 591
- Erstellen des Datums, der Uhrzeit oder des Zeitdauertyps anhand der lexikalischen Komponenten der einzelnen Typen 594
- Konstruieren des Typs "Datum", "Datum und Uhrzeit" oder "Uhrzeit" anhand eines String Input 595
- Funktionen zur Berechnung des Alters 597
- Epochen-Zeit (Unix-Zeit)-Funktionen 598

 © 2019-2025 Altova GmbH

▼ in alphabetischer Reihenfolge

altova:add-days-to-date 585
altova:add-days-to-dateTime 583
altova:add-hours-to-dateTime 583
altova:add-hours-to-time 587
altova:add-minutes-to-dateTime 583
altova:add-minutes-to-time 587
altova:add-months-to-date 585
altova:add-months-to-dateTime 583
altova:add-seconds-to-dateTime 583
altova:add-seconds-to-time 587
altova:add-years-to-date 585
altova:add-years-to-dateTime 583
altova:age 597
altova:age-details 597
altova:build-date 594
altova:build-duration 594
altova:build-time 594
altova:current-dateTime-no-TZ 588
altova:current-date-no-TZ 588
altova:current-time-no-TZ 588
altova:date-no-TZ 588
altova:dateTime-from-epoch 598
altova:dateTime-from-epoch-no-TZ 598
altova:dateTime-no-TZ 588
altova:days-in-month 589
altova:epoch-from-dateTime 598
altova:hours-from-dateTimeDuration-accumulated 589
altova:minutes-from-dateTimeDuration-accumulated 589
altova:seconds-from-dateTimeDuration-accumulated 589
altova:format-duration 586
altova:parse-date 595
altova:parse-dateTime 595
altova:parse-duration 586
altova:parse-time 595
altova:time-no-TZ 588
altova:weekday-from-date 591
altova:weekday-from-dateTime 591
altova:weeknumber-from-date 592
altova:weeknumber-from-dateTime 592

[Nach oben 582]

Hinzufügen einer Zeitdauer zu xs:dateTime XP3.1 XQ3.1

Mit diesen Funktionen werden Zeitdauerwerte zu **xs:dateTime** hinzugefügt, bevor **xs:dateTime** zurückgegeben wird. Der Typ `xs:dateTime` hat das Format `JJJJ-MM-TTZhh:mm:ss.sss`. Es handelt sich hierbei um eine Verkettung des `xs:date` und `xs:time` Formats, getrennt durch den Buchstaben `Z`. Ein Zeitzonensuffix (wie z.B. **+01:00**) ist optional.

▼ add-years-to-dateTime [altova:]

altova:add-years-to-dateTime(DateTime *als xs:dateTime*, **Years** *als xs:integer*) **als**

xs:dateTime XP3.1 XQ3.1
Fügt eine Zeitdauer in Jahren zu einem `xs:dateTime` Wert (*siehe Beispiele unten*) hinzu. Beim zweiten Argument handelt es sich um die Anzahl der Jahre, die zu dem im ersten Parameter angegebenen `xs:dateTime` Wert hinzugefügt werden sollen. Das Ergebnis ist vom Typ `xs:dateTime`.

Beispiele

- **altova:add-years-to-dateTime**(xs:dateTime("2014-01-15T14:00:00"), 10) gibt 2024-01-15T14:00:00 zurück
- **altova:add-years-to-dateTime**(xs:dateTime("2014-01-15T14:00:00"), -4) gibt 2010-01-15T14:00:00 zurück

add-months-to-dateTime [altova:]

altova:add-months-to-dateTime(DateTime *als xs:dateTime*, **Months** *als xs:integer*) als **xs:dateTime** XP3.1 XQ3.1
Fügt eine Zeitdauer in Monaten zu einem `xs:dateTime` Wert (*siehe Beispiele unten*) hinzu. Beim zweiten Argument handelt es sich um die Anzahl der Monate, die zu dem im ersten Argument angegebenen `xs:dateTime` Wert hinzugefügt werden sollen. Das Ergebnis ist vom Typ `xs:dateTime`.

Beispiele

- **altova:add-months-to-dateTime**(xs:dateTime("2014-01-15T14:00:00"), 10) gibt 2014-11-15T14:00:00 zurück
- **altova:add-months-to-dateTime**(xs:dateTime("2014-01-15T14:00:00"), -2) gibt 2013-11-15T14:00:00 zurück

add-days-to-dateTime [altova:]

altova:add-days-to-dateTime(DateTime *als xs:dateTime*, **Days** *als xs:integer*) als **xs:dateTime** XP3.1 XQ3.1
Fügt eine Zeitdauer in Tagen zu einem `xs:dateTime` Wert (*siehe Beispiel unten*) hinzu. Beim zweiten Argument handelt es sich um die Anzahl der Tage, die zu dem im ersten Argument angegebenen `xs:dateTime` Wert hinzugefügt werden sollen. Das Ergebnis ist vom Typ `xs:dateTime`.

Beispiele

- **altova:add-days-to-dateTime**(xs:dateTime("2014-01-15T14:00:00"), 10) gibt 2014-01-25T14:00:00 zurück
- **altova:add-days-to-dateTime**(xs:dateTime("2014-01-15T14:00:00"), -8) gibt 2014-01-07T14:00:00 zurück

add-hours-to-dateTime [altova:]

altova:add-hours-to-dateTime(DateTime *als xs:dateTime*, **Hours** *als xs:integer*) als **xs:dateTime** XP3.1 XQ3.1
Fügt eine Zeitdauer in Stunden zu einem `xs:dateTime` Wert (*siehe Beispiel unten*) hinzu. Beim zweiten Argument handelt es sich um die Anzahl der Stunden, die zu dem im ersten Argument angegebenen `xs:dateTime` Wert hinzugefügt werden sollen. Das Ergebnis ist vom Typ `xs:dateTime`.

Beispiele

- **altova:add-hours-to-dateTime**(xs:dateTime("2014-01-15T13:00:00"), 10) gibt 2014-01-

© 2019-2025 Altova GmbH

15T23:00:00 zurück
- **altova:add-hours-to-dateTime**(xs:dateTime("2014-01-15T13:00:00"), -8) gibt 2014-01-15T05:00:00 zurück

▼ add-minutes-to-dateTime [altova:]

altova:add-minutes-to-dateTime(DateTime *als xs:dateTime*, **Minutes** *als xs:integer***)** als **xs:dateTime** XP3.1 XQ3.1
Fügt eine Zeitdauer in Minuten zu einem xs:dateTime Wert (*siehe Beispiele unten)* hinzu. Beim zweiten Argument handelt es sich um die Anzahl der Minuten, die zu dem im ersten Argument angegebenen xs:dateTime Wert hinzugefügt werden sollen. Das Ergebnis ist vom Typ xs:dateTime.

Beispiele

- **altova:add-minutes-to-dateTime**(xs:dateTime("2014-01-15T14:10:00"), 45) gibt 2014-01-15T14:55:00 zurück
- **altova:add-minutes-to-dateTime**(xs:dateTime("2014-01-15T14:10:00"), -5) gibt 2014-01-15T14:05:00 zurück

▼ add-seconds-to-dateTime [altova:]

altova:add-seconds-to-dateTime(DateTime *als xs:dateTime*, **Seconds** *als xs:integer***)** als **xs:dateTime** XP3.1 XQ3.1
Fügt eine Zeitdauer in Sekunden zu einem xs:dateTime Wert (*siehe Beispiele unten*) hinzu. Beim zweiten Argument handelt es sich um die Anzahl der Sekunden, die zu dem im ersten Argument angegebenen xs:dateTime Wert hinzugefügt werden sollen. Das Ergebnis ist vom Typ xs:dateTime.

Beispiele

- **altova:add-seconds-to-dateTime**(xs:dateTime("2014-01-15T14:00:10"), 20) gibt 2014-01-15T14:00:30 zurück
- **altova:add-seconds-to-dateTime**(xs:dateTime("2014-01-15T14:00:10"), -5) gibt 2014-01-15T14:00:05 zurück

[Nach oben 582]

Hinzufügen einer Zeitdauer zu xs:date XP3.1 XQ3.1

Mit diesen Funktionen werden Zeitdauerwerte zu **xs:date** hinzugefügt, bevor **xs:date** zurückgegeben wird. Der Typ xs:date hat das Format JJJJ-MM-TT.

▼ add-years-to-date [altova:]

altova:add-years-to-date(Date *als xs:date*, **Years** *als xs:integer***)** als **xs:date** XP3.1 XQ3.1
Fügt eine Zeitdauer in Jahren zu einem Datumswert hinzu. Beim zweiten Argument handelt es sich um die Anzahl der Jahre, die zu dem im ersten Argument angegebenen xs:date Wert hinzugefügt werden sollen. Das Ergebnis ist vom Typ xs:date.

Beispiele

- **altova:add-years-to-date**(xs:date("2014-01-15"), 10) gibt 2024-01-15 zurück

- **altova:add-years-to-date**(xs:date("2014-01-15"), -4) gibt 2010-01-15 zurück

▼ add-months-to-date [altova:]

altova:add-months-to-date(Date *als xs:date*, **Months** *als xs:integer***)** als **xs:date** **XP3.1** **XQ3.1**
Fügt eine Zeitdauer in Monaten zu einem Datumswert hinzu. Beim zweiten Argument handelt es sich um die Anzahl der Monate, die zu dem im ersten Argument angegebenen xs:date Wert hinzugefügt werden sollen. Das Ergebnis ist vom Typ xs:date.

▣ *Beispiele*

- **altova:add-months-to-date**(xs:date("2014-01-15"), 10) gibt 2014-11-15 zurück
- **altova:add-months-to-date**(xs:date("2014-01-15"), -2) gibt 2013-11-15 zurück

▼ add-days-to-date [altova:]

altova:add-days-to-date(Date *als xs:date*, **Days** *als xs:integer***)** als **xs:date** **XP3.1** **XQ3.1**
Fügt eine Zeitdauer in Tagen zu einem Datumswert hinzu. Beim zweiten Argument handelt es sich um die Anzahl der Tage, die zu dem im ersten Argument angegebenen xs:date Wert hinzugefügt werden sollen. Das Ergebnis ist vom Typ xs:date.

▣ *Beispiele*

- **altova:add-days-to-date**(xs:date("2014-01-15"), 10) gibt 2014-01-25 zurück
- **altova:add-days-to-date**(xs:date("2014-01-15"), -8) gibt 2014-01-07 zurück

[Nach oben 582]

Formatieren und Abrufen einer Zeitdauer **XP3.1** **XQ3.1**

Mit diesen Funktionen wird ein Input **xs:duration**- oder **xs:string**-Wert geparst und ein **xs:string**- bzw. **xs:duration**-Wert zurückgegeben.

▼ format-duration [altova:]

altova:format-duration(Duration *als xs:duration*, **Picture** *als xs:string***)** als **xs:string** **XP3.1** **XQ3.1**
Formatiert eine Zeitdauer, die als erstes Argument bereitgestellt wird, gemäß einem Muster-String, der als zweites Argument bereitgestellt wird. Die Ausgabe ist ein Textstring, der dem Muster-String entsprechend formatiert ist.

▣ *Beispiele*

- **altova:format-duration**(xs:duration("P2DT2H53M11.7S"), "Days:[D01] Hours:[H01] Minutes:[m01] Seconds:[s01] Fractions:[f0]") gibt "Days:02 Hours:02 Minutes:53 Seconds:11 Fractions:7" gibt
- **altova:format-duration**(xs:duration("P3M2DT2H53M11.7S"), "Months:[M01] Days:[D01] Hours:[H01] Minutes:[m01]") gibt "Months:03 Days:02 Hours:02 Minutes:53" gibt

▼ parse-duration [altova:]

© 2019-2025 Altova GmbH

altova:parse-duration(InputString *als xs:string*, **Picture** *als xs:string***)** als **xs:duration** XP3.1 XQ3.1
Erhält einen Pattern-String als erstes Argument und eine Muster-String als zweites Argument. Der Input-Strings wird auf Basis des Muster-Strings geparst und ein `xs:duration` wird zurückgegeben.

Beispiele

- **altova:parse-duration**("Days:02 Hours:02 Minutes:53 Seconds:11 Fractions:7"), "Days:[D01] Hours:[H01] Minutes:[m01] Seconds:[s01] Fractions:[f0]") gibt "P2DT2H53M11.7S" zurück
- **altova:parse-duration**("Months:03 Days:02 Hours:02 Minutes:53 Seconds:11 Fractions:7", "Months:[M01] Days:[D01] Hours:[H01] Minutes:[m01]") gibt "P3M2DT2H53M" zurück

[Nach oben 582]

Hinzufügen einer Zeitdauer zu xs:time XP3.1 XQ3.1

Diese Funktionen fügen einen Zeitdauerwert zu **xs:time** hinzu und geben **xs:time** zurück. Der Typ `xs:time` entspricht in seiner lexikalischen Form `hh:mm:ss.sss`. Eine optionale Zeitzone kann angehängt werden. Der Buchstabe `Z` steht für Coordinated Universal Time (UTC). Alle anderen Zeitzonen werden in Form des Zeitunterschieds zur UTC im Format `+hh:mm`, oder `-hh:mm` dargestellt. Wenn kein Wert für die Zeitzone vorhanden ist, wird sie als unbekannt und nicht als UTC angenommen.

add-hours-to-time [altova:]

altova:add-hours-to-time(Time *als xs:time*, **Hours** *als xs:integer***)** als **xs:time** XP3.1 XQ3.1
Fügt eine Zeitdauer in Stunden zu einem Uhrzeitwert hinzu. Beim zweiten Argument handelt es sich um die Anzahl der Stunden, die zu dem im ersten Argument angegebenen `xs:time` Wert hinzugefügt werden sollen. Das Ergebnis ist vom Typ `xs:time`.

Beispiele

- **altova:add-hours-to-time**(xs:time("11:00:00"), 10) gibt 21:00:00 zurück
- **altova:add-hours-to-time**(xs:time("11:00:00"), -7) gibt 04:00:00 zurück

add-minutes-to-time [altova:]

altova:add-minutes-to-time(Time *als xs:time*, **Minutes** *als xs:integer***)** als **xs:time** XP3.1 XQ3.1
Fügt eine Zeitdauer in Minuten zu einem `xs:time` Wert hinzu. Beim zweiten Argument handelt es sich um die Anzahl der Minuten, die zu dem im ersten Argument angegebenen `xs:time` Wert hinzugefügt werden sollen. Das Ergebnis ist vom Typ `xs:time`.

Beispiele

- **altova:add-minutes-to-time**(xs:time("14:10:00"), 45) gibt 14:55:00 zurück
- **altova:add-minutes-to-time**(xs:time("14:10:00"), -5) gibt 14:05:00 zurück

add-seconds-to-time [altova:]

altova:add-seconds-to-time(Time *as xs:time*, **Minutes** *als xs:integer***)** als **xs:time** XP3.1 XQ3.1
Fügt eine Zeitdauer in Sekunden zu einem Uhrzeitwert hinzu. Beim zweiten Argument handelt es sich um

die Anzahl der Sekunden, die zu dem im ersten Argument angegebenen `xs:time` Wert hinzugefügt werden sollen. Das Ergebnis ist vom Typ `xs:time`. Die Seconds Komponenten kann sich im Bereich von `0` bis `59.999` befinden.

Beispiele

- **`altova:add-seconds-to-time`**`(xs:time("14:00:00"), 20)` gibt `14:00:20` zurück
- **`altova:add-seconds-to-time`**`(xs:time("14:00:00"), 20.895)` gibt `14:00:20.895` zurück

[Nach oben 582]

Entfernen der Zeitzone aus date/time-Datentypen XP3.1 XQ3.1

Diese Funktionen entfernen die Zeitzone aus den aktuellen **`xs:dateTime`**, **`xs:date`** bzw. **`xs:time`** Werten. Beachten Sie, dass im Unterschied zu `xs:dateTime` bei `xs:dateTimeStamp` die Zeitzone erforderlich ist (während sie im ersteren Fall optional ist). Das Format eines `xs:dateTimeStamp` Werts lautet daher: `JJJJ-MM-TTZhh:mm:ss.sss±hh:mm`. oder `JJJJ-MM-TTZhh:mm:ss.sssZ`. Wenn das Datum und die Uhrzeit von der Systemuhr als `xs:dateTimeStamp` ausgelesen wird, können Sie die Zeitzone, falls erforderlich, mit der Funktion `current-dateTime-no-TZ()` entfernen.

current-date-no-TZ [altova:]

`altova:current-date-no-TZ()` als **`xs:date`** XP3.1 XQ3.1

Die Funktion hat kein Argument. Sie entfernt die Zeitzone aus dem `current-date()` Wert (welcher das aktuelle Datum laut Systemuhr ist) und gibt einen `xs:date` Wert zurück.

Beispiele

Wenn das aktuelle Datum **`2014-01-15+01:00`** lautet:

- **`altova:current-date-no-TZ`**`()` gibt `2014-01-15` zurück

current-dateTime-no-TZ [altova:]

`altova:current-dateTime-no-TZ()` als **`xs:dateTime`** XP3.1 XQ3.1

Die Funktion hat kein Argument. Sie entfernt die Zeitzone aus dem `current-dateTime()` Wert (welcher das aktuelle Datum und die aktuelle Uhrzeit laut Systemuhr ist) und gibt einen `xs:dateTime` Wert zurück.

Beispiele

Wenn der aktuelle Datums- und Uhrzeitwert **`2014-01-15T14:00:00+01:00`** lautet:

- **`altova:current-dateTime-no-TZ`**`()` gibt `2014-01-15T14:00:00` zurück

current-time-no-TZ [altova:]

`altova:current-time-no-TZ()` als **`xs:time`** XP3.1 XQ3.1

Die Funktion hat kein Argument. Sie entfernt die Zeitzone aus dem `current-time()` Wert (welcher die aktuelle Uhrzeit laut Systemuhr ist) und gibt einen `xs:time` Wert zurück.

Beispiele

Wenn der aktuelle Uhrzeitwert **`14:00:00+01:00`** lautet:

© 2019-2025 Altova GmbH

- `altova:current-time-no-TZ()` gibt `14:00:00` zurück

▼ date-no-TZ [altova:]

`altova:date-no-TZ(InputDate` *`as xs:date`***`)`** als **`xs:date`** XP3.1 XQ3.1
Diese Funktion verwendet ein `xs:date` Argument, entfernt den Zeitzonenteil daraus und gibt einen `xs:date` Wert zurück. Beachten Sie, dass das Datum nicht geändert wird..

Beispiele

- `altova:date-no-TZ(xs:date("2014-01-15+01:00"))` gibt `2014-01-15` zurück

▼ dateTime-no-TZ [altova:]

`altova:dateTime-no-TZ(InputDateTime` *`als xs:dateTime`***`)`** als **`xs:dateTime`** XP3.1 XQ3.1
Diese Funktion verwendet ein `xs:dateTime` Argument, entfernt den Zeitzonenteil daraus und gibt einen `xs:dateTime` Wert zurück. Beachten Sie, dass weder Datum noch Uhrzeit geändert werden.

Beispiele

- `altova:dateTime-no-TZ(xs:date("2014-01-15T14:00:00+01:00"))` gibt `2014-01-15T14:00:00` zurück

▼ time-no-TZ [altova:]

`altova:time-no-TZ(InputTime` *`als xs:time`***`)`** als **`xs:time`** XP3.1 XQ3.1
Diese Funktion verwendet ein `xs:time` Argument, entfernt den Zeitzonenteil daraus und gibt einen `xs:time` Wert zurück. Beachten Sie, dass die Uhrzeit nicht geändert wird.

Beispiele

- `altova:time-no-TZ(xs:time("14:00:00+01:00"))` gibt `14:00:00` zurück

[Nach oben 582]

Rückgabe der Anzahl von Tagen, Stunden, Minuten, Sekunden anhand einer Zeitdauer XP3.1 XQ3.1

Diese Funktionen geben die Anzahl der Tage in einem Monat bzw. die Anzahl der Stunden, Minuten und Sekunden anhand einer Zeitdauer zurück.

▼ days-in-month [altova:]

`altova:days-in-month(Year` *`als xs:integer`***`, Month`** *`als xs:integer`***`)`** als **`xs:integer`** XP3.1 XQ3.1
Gibt die Anzahl der Tage im angegebenen Monat zurück. Der Monat wird mit Hilfe der Argumente `Year` und `Month` angegeben.

Beispiele

- `altova:days-in-month(2018, 10)` gibt `31` zurück
- `altova:days-in-month(2018, 2)` gibt `28` zurück
- `altova:days-in-month(2020, 2)` gibt `29` zurück

hours-from-dayTimeDuration-accumulated

altova:hours-from-dayTimeDuration-accumulated(DayAndTime *als xs:duration***)** als **xs:integer**
XP3.1 **XQ3.1**
Gibt die Gesamtzahl der Stunden in der durch das Argument `DayAndTime` bereitgestellten Zeitdauer (vom Typ `xs:duration`) zurück. Die Stunden in den Komponenten `Day` und `Time` werden addiert, um ein Ergebnis in Form einer Ganzzahl zu erhalten. Nur für volle 60 Minuten wird eine neue Stunde berechnet. Negative Zeitdauerergebnisse ergeben einen negativen Stundenwert.

Beispiele

- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("P5D")) gibt 120 zurück, d.h. die Gesamtanzahl der Stunden in 5 Tagen.
- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("P5DT2H")) gibt 122 zurück, d.h. die Gesamtzahl der Stunden in 5 Tagen plus 2 Stunden.
- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("P5DT2H60M")) gibt 123 zurück, d.h. die Gesamtzahl der Stunden in 5 Tagen plus 2 Stunden und 60 Minuten.
- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("P5DT2H119M")) gibt 123 zurück, d.h. die Gesamtzahl der Stunden in 5 Tagen plus 2 Stunden und 119 Minuten.
- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("P5DT2H120M")) gibt 124 zurück, d.h. die Gesamtzahl der Stunden in 5 Tagen plus 2 Stunden und 120 Minuten.
- **altova:hours-from-dayTimeDuration-accumulated**(xs:duration("-P5DT2H")) gibt -122 zurück.

minutes-from-dayTimeDuration-accumulated

altova:minutes-from-dayTimeDuration-accumulated(DayAndTime *als xs:duration***)** als **xs:integer** **XP3.1** **XQ3.1**
Gibt die Gesamtzahl der Minuten in der durch das Argument `DayAndTime` bereitgestellten Zeitdauer (vom Typ `xs:duration`) zurück. Die Minuten in den Komponenten `Day` und `Time` werden addiert, um ein Ergebnis in Form einer Ganzzahl zu erhalten. Negative Zeitdauerergebnisse ergeben einen negativen Minutenwert.

Beispiele

- **altova:minutes-from-dayTimeDuration-accumulated**(xs:duration("PT60M")) gibt 60 zurück.
- **altova:minutes-from-dayTimeDuration-accumulated**(xs:duration("PT1H")) gibt 60 zurück, d.h. die Gesamtzahl der Minuten in 1 Stunde.
- **altova:minutes-from-dayTimeDuration-accumulated**(xs:duration("PT1H40M") gibt 100 zurück.
- **altova:minutes-from-dayTimeDuration-accumulated**(xs:duration("P1D")) gibt 1440 zurück, d.h. die Gesamtzahl der Minuten an einem Tag.
- **altova:minutes-from-dayTimeDuration-accumulated**(xs:duration("-P1DT60M")) gibt -1500 zurück.

seconds-from-dayTimeDuration-accumulated

altova:seconds-from-dayTimeDuration-accumulated(DayAndTime *als xs:duration***)** als **xs:integer** **XP3.1** **XQ3.1**

© 2019-2025 Altova GmbH

Gibt die Gesamtzahl der Sekunden in der durch das Argument `DayAndTime` bereitgestellten Zeitdauer (vom Typ `xs:duration`) zurück. Die Sekunden in den Komponenten `Day` und `Time` werden addiert, um ein Ergebnis in Form einer Ganzzahl zu erhalten. Negative Zeitdauerergebnisse ergeben einen negativen Sekundenwert.

Examples

- **altova:seconds-from-dayTimeDuration-accumulated**(xs:duration("PT1M")) gibt 60 zurück, d.h. die Gesamtzahl der Sekunden in 1 Minute.
- **altova:seconds-from-dayTimeDuration-accumulated**(xs:duration("PT1H")) gibt 3600 zurück, d.h. die Gesamtzahl der Sekunden in 1 Stunde.
- **altova:seconds-from-dayTimeDuration-accumulated**(xs:duration("PT1H2M")) gibt 3720 zurück.
- **altova:seconds-from-dayTimeDuration-accumulated**(xs:duration("P1D"))gibt 86400 zurück, d.h. die Gesamtzahl der Sekunden an 1 Tag.
- **altova:seconds-from-dayTimeDuration-accumulated**(xs:duration("-P1DT1M")) gibt -86460 zurück.

Rückgabe des Wochentages anhand von xs:dateTime oder xs:date XP3.1 XQ3.1

Diese Funktionen geben anhand des `xs:dateTime` oder `xs:date` Werts den Wochentag in Form einer Ganzzahl zurück. Die Tage der Woche sind (im amerikanischen Format) von `1` bis `7` nummeriert, wobei `Sonntag=1`. Im europäischen Format beginnt die Woche am Montag (`=1`). Das amerikanische Format, in dem `Sonntag=1`, kann mittels der Ganzzahl `0` definiert werden, wenn das Format mittels einer Ganzzahl angegeben werden kann.

weekday-from-dateTime [altova:]

altova:weekday-from-dateTime(DateTime *als xs:dateTime*) **als xs:integer** XP3.1 XQ3.1

Erhält ein Datum mit einer Uhrzeit als einziges Argument und gibt den Tag der Woche dieses Datums in Form einer Ganzzahl zurück. Die Wochentage sind beginnend mit `Sonntag=1` nummeriert. Wenn das europäische Format benötigt wird (wo `Montag=1`), verwenden Sie die andere Signatur dieser Funktion (*siehe nächste Signatur unten*).

Beispiele

- **altova:weekday-from-dateTime**(xs:dateTime("2014-02-03T09:00:00")) gibt 2 zurück, wobei 2 für Montag steht.

altova:weekday-from-dateTime(DateTime *als xs:dateTime*, **Format** *als xs:integer*) **als xs:integer** XP3.1 XQ3.1

Erhält ein Datum mit einer Uhrzeit als erstes Argument und gibt den Tag der Woche dieses Datums in Form einer Ganzzahl zurück. Wenn das zweite (Integer)-Argument `0` ist, werden die Wochentage beginnend mit `Sonntag=1` von `1` bis `7` nummeriert. Wenn das zweite Argument eine andere Ganzzahl als `0` ist, so ist `Montag=1`. Wenn es kein zweites Argument gibt, wird die Funktion gelesen, als ob sie die andere Signatur dieser Funktion hätte (*siehe vorherige Signatur*).

Beispiele

- **altova:weekday-from-dateTime**(xs:dateTime("2014-02-03T09:00:00"), 1) gibt 1 zurück, wobei 1 für Montag steht
- **altova:weekday-from-dateTime**(xs:dateTime("2014-02-03T09:00:00"), 4) gibt 1 zurück, wobei 1 für Montag steht

- **altova:weekday-from-dateTime**(xs:dateTime("2014-02-03T09:00:00"), 0) gibt 2 zurück, wobei 2 für Montag steht.

▼ weekday-from-date [altova:]

altova:weekday-from-date(Date *als xs:date*) **als xs:integer** XP3.1 XQ3.1
Erhält ein Datum als einziges Argument und gibt den Tag der Woche dieses Datums in Form einer Ganzzahl zurück. Die Wochentage sind beginnend mit Sonntag=1 nummeriert. Wenn das europäische Format benötigt wird (wo Montag=1), verwenden Sie die andere Signatur dieser Funktion (*siehe nächste Signatur unten*).

Beispiele

- **altova:weekday-from-date**(xs:date("2014-02-03+01:00")) gibt 2 zurück, d.h. dies wäre ein Montag.

altova:weekday-from-date(Date *als xs:date*, **Format** *als xs:integer*) **als xs:integer** XP3.1 XQ3.1
Erhält ein Datum als erstes Argument und gibt den Tag der Woche dieses Datums in Form einer Ganzzahl zurück. Wenn das zweite Argument (Format) 0 ist, werden die Wochentage beginnend mit Sonntag=1 von 1 bis 7 nummeriert. Wenn das zweite Argument eine andere Ganzzahl als 0 ist, so ist Montag=1. Wenn es kein zweites Argument gibt, wird die Funktion gelesen, als ob sie die andere Signatur dieser Funktion hätte (*siehe vorherige Signatur*).

Beispiele

- **altova:weekday-from-date**(xs:date("2014-02-03"), 1) gibt 1 zurück, wobei 1 für Montag steht
- **altova:weekday-from-date**(xs:date("2014-02-03"), 4) gibt 1 zurück, wobei 1 für Montag steht
- **altova:weekday-from-date**(xs:date("2014-02-03"), 0) gibt 2 zurück, wobei 2 für Montag steht.

[Nach oben 582]

Rückgabe der Wochennummer anhand von xs:dateTime oder xs:date XP2 XQ1 XP3.1 XQ3.1

Diese Funktionen geben anhand von xs:dateTime oder xs:date die Wochennummer als Ganzzahl zurück. Die Wochennummer steht in den Kalenderformaten US, ISO/European und Islamic zur Verfügung. Die Wochennummerierung unterscheidet sich in diesen Kalenderformaten, da die Woche in diesen Formaten an unterschiedlichen Tagen beginnt (Im Format US am Sonntag, im Format ISO/European am Montag und im Format Islamic am Samstag).

▼ weeknumber-from-date [altova:]

altova:weeknumber-from-date(Date *als xs:date*, **Calendar** *als xs:integer*) als **xs:integer** XP2 XQ1 XP3.1 XQ3.1
Gibt die Wochennummer des bereitgestellten **Date** Arguments als Ganzzahl zurück. Das zweite Argument (**Calendar**) definiert das zu verwendende Kalendersystem.
Unterstützte **Calendar** Werte sind:

© 2019-2025 Altova GmbH

- `0` = **`US-Kalender`** (Woche beginnt am Sonntag)
- `1` = **`ISO-Standard, Europäischer Kalender`** (Woche beginnt am Montag)
- `2` = **`Islamischer Kalender`** (Woche beginnt am Samstag)

Der Standardwert ist `0`.

Beispiele

- **`altova:weeknumber-from-date`**`(xs:date("2014-03-23"), 0)` gibt `13` zurück
- **`altova:weeknumber-from-date`**`(xs:date("2014-03-23"), 1)` gibt `12` zurück
- **`altova:weeknumber-from-date`**`(xs:date("2014-03-23"), 2)` gibt `13` zurück
- **`altova:weeknumber-from-date`**`(xs:date("2014-03-23") )` gibt `13` zurück

Der Tag des Datums in den obigen Beispielen (`2014-03-23`) ist ein Sonntag. Daher ist der US- und der islamische Kalender dem europäischen Kalender an diesem Tag eine Woche voraus.

weeknumber-from-dateTime [altova:]

`altova:weeknumber-from-dateTime(DateTime` *als xs:dateTime*, **`Calendar`** *als xs:integer*`)` **`als xs:integer`** **XP2** **XQ1** **XP3.1** **XQ3.1**

Gibt die Wochennummer des bereitgestellten **`DateTime`** Arguments als Ganzzahl zurück. Das zweite Argument (**`Calendar`**) definiert das zu verwendende Kalendersystem.
Unterstützte **`Calendar`** Werte sind:

- `0` = **`US-Kalender`** (*Woche beginnt am Sonntag*)
- `1` = **`ISO-Standard, Europäischer Kalender`** (*Woche beginnt am Montag*)
- `2` = **`Islamischer Kalender`** (*Woche beginnt am Samstag*)

Der Standardwert ist `0`.

Beispiele

- **`altova:weeknumber-from-dateTime`**`(xs:dateTime("2014-03-23T00:00:00"), 0)` gibt `13` zurück
- **`altova:weeknumber-from-dateTime`**`(xs:dateTime("2014-03-23T00:00:00"), 1)` gibt `12` zurück
- **`altova:weeknumber-from-dateTime`**`(xs:dateTime("2014-03-23T00:00:00"), 2)` gibt `13` zurück
- **`altova:weeknumber-from-dateTime`**`(xs:dateTime("2014-03-23T00:00:00") )` gibt `13` zurück

Der Tag des Datums- und Uhrzeitwerts in den obigen Beispielen (`2014-03-23T00:00:00`) ist ein Sonntag. Daher ist der US- und der islamische Kalender dem europäischen Kalender an diesem Tag eine Woche voraus.

[Nach oben 582]

Erstellen des Datums-, Uhrzeit- oder Zeitdauer-Datentyps anhand der lexikalischen Komponenten der einzelnen Typen XP3.1 XQ3.1

Die Funktionen erhalten die lexikalischen Komponenten des `xs:date`, `xs:time` oder `xs:duration`-Datentyps als Input-Argumente und kombinieren diese zum entsprechenden Datentyp.

build-date [altova:]

altova:build-date(Year *als xs:integer*, **Month** *als xs:integer*, **Date** *als xs:integer*) als **xs:date** XP3.1 XQ3.1

Das erste, zweite und dritte Argument steht für das Jahr, bzw. den Monat bzw. das Datum. Sie werden zu einem Wert vom Typ `xs:date` kombiniert. Die Werte der Ganzzahlen müssen sich innerhalb des korrekten Bereichs dieses jeweiligen Datumsteils befinden. So sollte z.B. das zweite Argument nicht größer als `12` sein.

Beispiele

- **altova:build-date**(2014, 2, 03) gibt 2014-02-03 zurück

build-time [altova:]

altova:build-time(Hours *als xs:integer*, **Minutes** *als xs:integer*, **Seconds** *als xs:integer*) als **xs:time** XP3.1 XQ3.1

Das erste, zweite und dritte Argument steht für die Stunde (`0` bis `23`), bzw. die Minuten (`0` bis `59`) bzw. die Sekunden (`0` bis `59`). Sie werden zu einem Wert von Typ `xs:time` kombiniert. Die Werte der Ganzzahlen müssen sich innerhalb des korrekten Bereichs dieses jeweiligen Uhrzeitteils befinden. So sollte z.B. der zweite Parameter nicht größer als `59` sein. Um eine Zeitzone zum Wert hinzuzufügen, verwenden Sie die andere Signatur der Funktion (*siehe nächste Signatur)*.

Beispiele

- **altova:build-time**(23, 4, 57) gibt 23:04:57 zurück

altova:build-time(Hours *als xs:integer*, **Minutes** *als xs:integer*, **Seconds** *als xs:integer*, **TimeZone** *als xs:string*) als **xs:time** XP3.1 XQ3.1

Das erste, zweite und dritte Argument steht für die Stunde (`0` bis `23`), bzw. die Minuten (`0` bis `59`) bzw. die Sekunden (`0` bis `59`). Das vierte Argument ist ein String, der den Zeitzonenteil des Werts liefert. Die vier Argumente werden zu einem Wert vom Typ `xs:time` kombiniert. Die Werte der Ganzzahlen müssen sich innerhalb des korrekten Bereichs dieses jeweiligen Uhrzeitteils befinden. So sollte z.B. das zweite Argument (Minuten) nicht größer als `59` sein.

Beispiele

- **altova:build-time**(23, 4, 57, '+1') gibt 23:04:57+01:00 zurück

build-duration [altova:]

altova:build-duration(Years *als xs:integer*, **Months** *als xs:integer*) als **xs:yearMonthDuration** XP3.1 XQ3.1

Setzt aus zwei Argumenten einen Wert vom Typ `xs:yearMonthDuration` zusammen. Das erste Argument liefert den Jahr-Teil des Zeitdauerwerts, während das zweite Argument den Monat-Teil liefert. Wenn der zweite Parameter (Monate) größer oder gleich `12` ist, so wird die Ganzzahl durch 12 dividiert. Der Quotient wird zum ersten Argument hinzugefügt, um den Jahr-Teil des Zeitdauerwerts zu liefern,

© 2019-2025 Altova GmbH

während der Rest (der Division) den Monat-Teil liefert. Eine Beschreibung zur Erstellung einer Zeitdauer vom Typ `xs:dayTimeDuration` finden Sie in der nächsten Signatur.

Beispiele

- **altova:build-duration**(2, 10) gibt P2Y10M zurück
- **altova:build-duration**(14, 27) gibt P16Y3M zurück
- **altova:build-duration**(2, 24) gibt P4Y zurück

altova:build-duration(Days *als xs:integer*, **Hours** *als xs:integer*, **Minutes** *als xs:integer*, **Seconds** *als xs:integer*) als **xs:dayTimeDuration** **XP3.1** **XQ3.1**

Kombiniert vier Argumente zu einem Wert vom Typ `xs:dayTimeDuration`. Das erste Argument liefert den Tage-Teil, das zweite die Stunden, das dritte die Minuten und das vierte die Sekunden des Zeitdauerwerts. Die einzelnen Uhrzeitparameter werden in den entsprechenden Wert für die nächsthöhere Einheit konvertiert und das Ergebnis wird zur Berechnung der Gesamtdauer weitergegeben. So werden z.B. 72 Sekunden in `1M(inute)12S(ekunden)` konvertiert. Dieser Wert wird zur Berechnung der Gesamtdauer weitergegeben. Um eine Zeitdauer vom Typ `xs:yearMonthDuration` zu berechnen, verwenden Sie die vorherige Signatur.

Beispiele

- **altova:build-duration**(2, 10, 3, 56) gibt P2DT10H3M56S zurück
- **altova:build-duration**(1, 0, 100, 0) gibt P1DT1H40M zurück
- **altova:build-duration**(1, 0, 0, 3600) gibt P1DT1H zurück

[Nach oben 582]

Konstruieren von Datum, Datum und Uhrzeit und Zeit-Datentypen anhand des String-Input XP2 XQ1 XP3.1 XQ3.1

Diese Funktionen erhalten Strings als Argumente und konstruieren anhand dieser die Datentypen `xs:date`, `xs:dateTime` oder `xs:time`. Der String wird anhand eines bereitgestellten Pattern-Arguments nach Komponenten des Datentyps analysiert.

parse-date [altova:]

altova:parse-date(Date *als xs:string*, **DatePattern** *als xs:string*) als **xs:date** **XP2** **XQ1** **XP3.1** **XQ3.1**

Gibt den Input-String **Date** als **xs:date** Wert zurück. Das zweite Argument **DatePattern** definiert das Pattern (die Komponentensequenz) des Input-String. **DatePattern** wird durch die unten aufgelisteten Komponenten-Specifier beschrieben. Als Komponententrennzeichen kann jedes beliebige Zeichen verwendet werden. Siehe Beispiele unten.

D	Datum
M	Monat
Y	Jahr

Das Pattern in **DatePattern** muss mit dem Pattern in **Date** übereinstimmen. Da die Ausgabe vom Typ **xs:date** ist, hat sie immer das lexikalische Format **YYYY-MM-DD**.

Beispiele

- **altova:parse-date**(xs:string("09-12-2014"), "[D]-[M]-[Y]") gibt 2014-12-09 zurück
- **altova:parse-date**(xs:string("09-12-2014"), "[M]-[D]-[Y]") gibt 2014-09-12 zurück

- **altova:parse-date**("06/03/2014", "[M]/[D]/[Y]") gibt 2014-06-03 zurück
- **altova:parse-date**("06 03 2014", "[M] [D] [Y]") gibt 2014-06-03 zurück
- **altova:parse-date**("6 3 2014", "[M] [D] [Y]") gibt 2014-06-03 zurück

parse-dateTime [altova:]

altova:parse-dateTime(DateTime *als xs:string*, **DateTimePattern** *als xs:string*) als **xs:dateTime** **XP2** **XQ1** **XP3.1** **XQ3.1**

Gibt den Input-String **DateTime** als **xs:dateTime** Wert zurück. Das zweite Argument **DateTimePattern** definiert das Pattern (die Komponentensequenz) des Input-String. **DateTimePattern** wird durch die unten aufgelisteten Komponenten-Specifier beschrieben. Als Komponententrennzeichen kann jedes beliebige Zeichen verwendet werden. Siehe Beispiele unten.

D	Datum
M	Monat
Y	Jahr
H	Stunde
m	Minuten
s	Sekunden

Das Pattern in **DateTimePattern** muss mit dem Pattern in **DateTime** übereinstimmen. Da die Ausgabe vom Typ **xs:dateTime** ist, hat sie immer das lexikalische Format **YYYY-MM-DDTHH:mm:ss**.

Beispiele

- **altova:parse-dateTime**(xs:string("09-12-2014 13:56:24"), "[M]-[D]-[Y] [H]:[m]:[s]") gibt 2014-09-12T13:56:24 zurück
- **altova:parse-dateTime**("time=13:56:24; date=09-12-2014", "time=[H]:[m]:[s]; date=[D]-[M]-[Y]") gibt 2014-12-09T13:56:24 zurück

parse-time [altova:]

altova:parse-time(Time *als xs:string*, **TimePattern** *als xs:string*) als **xs:time** **XP2** **XQ1** **XP3.1** **XQ3.1**

Gibt den Input-String **Time** als **xs:time** Wert zurück. Das zweite Argument **TimePattern** definiert das Pattern (die Komponentensequenz) des Input-String. **TimePattern** wird durch die unten aufgelisteten Komponenten-Specifier beschrieben. Als Komponententrennzeichen kann jedes beliebige Zeichen verwendet werden. Siehe Beispiele unten.

H	Stunde
m	Minuten
s	Sekunden

Das Pattern in **TimePattern** muss mit dem Pattern in **Time** übereinstimmen. Da die Ausgabe vom Typ **xs:Time** ist, hat sie immer das lexikalische Format **HH:mm:ss**.

Beispiele

- **altova:parse-time**(xs:string("13:56:24"), "[H]:[m]:[s]") gibt 13:56:24 zurück

© 2019-2025 Altova GmbH

- **altova:parse-time**("13-56-24", "[H]-[m]") gibt 13:56:00 zurück
- **altova:parse-time**("time=13h56m24s", "time=[H]h[m]m[s]s") gibt 13:56:24 zurück
- **altova:parse-time**("time=24s56m13h", "time=[s]s[m]m[H]h") gibt 13:56:24 zurück

[Nach oben 582]

Funktionen zur Berechnung des Alters XP3.1 XQ3.1

Diese Funktionen geben das Alter berechnet (i) anhand von einem Input-Argument und dem aktuellen Datum oder (ii) anhand zweier Input-Argumentdaten zurück. Die Funktion **altova:age** gibt das Alter in Jahren zurück, die Funktion **altova:age-details** gibt das Alter als Sequenz von drei Ganzzahlen zurück, die die Jahre, Monate und Tage des Alters angeben.

age [altova:]

altova:age(StartDate *als xs:date*) als **xs:integer** XP3.1 XQ3.1
Gibt eine Ganzzahl zurück, die das Alter eines Objekts in *Jahren* angibt. Berechnet wird das Alter anhand des durch das Argument gelieferten Startdatums endend mit dem aktuellen Datum (laut Systemuhr). Wenn das Input-Argument eines Datums größer oder gleich einem Jahr in der Zukunft ist, ist der Rückgabewert negativ.

Beispiele

Wenn das aktuelle Datum **2014-01-15** lautet:

- **altova:age**(xs:date("2013-01-15")) gibt 1 zurück
- **altova:age**(xs:date("2013-01-16")) gibt 0 zurück
- **altova:age**(xs:date("2015-01-15")) gibt -1 zurück
- **altova:age**(xs:date("2015-01-14")) gibt 0 zurück

altova:age(StartDate *als xs:date*, **EndDate** *als xs:date*) als **xs:integer** XP3.1 XQ3.1
Gibt eine Ganzzahl zurück, die das Alter eines Objekts in *Jahren* angibt. Berechnet wird das Alter anhand des durch das erste Argument gelieferten Startdatums endend mit dem als zweites Datum gelieferten Enddatum. Wenn das erste Argument ein Jahr oder mehr nach dem zweiten Argument liegt, ist der Rückgabewert negativ.

Beispiele

Wenn das aktuelle Datum **2014-01-15** lautet:

- **altova:age**(xs:date("2000-01-15"), xs:date("2010-01-15")) gibt 10 zurück
- **altova:age**(xs:date("2000-01-15"), current-date()) gibt 14 zurück, wenn das aktuelle Datum 2014-01-15 ist
- **altova:age**(xs:date("2014-01-15"), xs:date("2010-01-15")) gibt -4 zurück

age-details [altova:]

altova:age-details(InputDate *als xs:date*) als (**xs:integer**)* XP3.1 XQ3.1
Gibt drei Ganzzahlen zurück. Dabei handelt es sich um die Jahre, Monate bzw. Tage zwischen dem als

Argument angegebenen Datum und dem aktuellen Datum (laut Systemuhr). Die Summe der zurückgegebenen `years+months+days` gibt zusammen die Gesamtzeitdifferenz zwischen den beiden Datumswerten (dem Input-Datum und dem aktuellen Datum) an. Das Input-Datum hat eventuell einen Wert, der vor oder nach dem aktuellen Datum liegt, doch wird dies nicht aus dem Vorzeichen der Rückgabewerte ersichtlich; die Rückgabewerte sind immer positiv.

Beispiele

Wenn das aktuelle Datum **2014-01-15** lautet:

- `altova:age-details(xs:date("2014-01-16"))` gibt `(0 0 1)` zurück
- `altova:age-details(xs:date("2014-01-14"))` gibt `(0 0 1)` zurück
- `altova:age-details(xs:date("2013-01-16"))` gibt `(1 0 1)` zurück
- `altova:age-details(current-date())` gibt `(0 0 0)` zurück

altova:age-details(Date-1 *als xs:date*, **Date-2** *als xs:date***)** als **(xs:integer)*** **XP3.1** **XQ3.1**
Gibt drei Ganzzahlen zurück. Dabei handelt es sich um die Jahre, Monate bzw. Tage zwischen den beiden Argumentdaten. Die Summe der zurückgegebenen `years+months+days` gibt zusammen die Gesamtzeitdifferenz zwischen den beiden Input-Datumswerten an. Es ist unerheblich, ob das frühere oder spätere Datum als erstes Argument angegeben wird. Die Rückgabewerte geben nicht an, ob das Input-Datum vor oder nach dem aktuellen Datum liegt. Die Rückgabewerte sind immer positiv.

Beispiele

- `altova:age-details(xs:date("2014-01-16"), xs:date("2014-01-15"))` gibt `(0 0 1)` zurück
- `altova:age-details(xs:date("2014-01-15"), xs:date("2014-01-16"))` gibt `(0 0 1)` zurück

[Nach oben 582]

Epochen-Zeit (Unix-Zeit)-Funktionen XP3.1 XQ3.1

Die Epochenzeit ist ein auf Unix-Systemen verwendetes Zeitsystem. Darin wird jeder Zeitpunkt als Anzahl der Sekunden seit 00:00:00 UTC des 1. Januar 1970 definiert. Diese Epochenzeitfunktionen konvertieren **xs:dateTime**-Werte in Epochenzeitwerte und umgekehrt.

dateTime-from-epoch [altova:]

altova:dateTime-from-epoch(Epoch *als xs:decimal* **als xs:dateTime** **XP3.1** **XQ3.1**
Die Epochenzeit ist ein auf Unix-Systemen verwendetes Zeitsystem. Darin wird jeder Zeitpunkt als Anzahl der Sekunden seit 00:00:00 UTC des 1. Januar 1970 definiert. Die Funktion **dateTime-from-epoch** gibt das **xs:dateTime**-Äquivalent einer Epochenzeit zurück, passt die lokale Zeitzone an und inkludiert die Zeitzoneninformation im Ergebnis.

Die Funktion erhält ein `xs:decimal`-Argument und gibt einen `xs:dateTime`-Wert, der einen **zz**-Teil (Zeitzone) enthält, zurück. Das Ergebnis wird durch Berechnung des UTC **dateTime**-Äquivalents der Epochenzeit und Hinzufügen der (anhand der Systemuhr ermittelten) lokalen Zeitzone ermittelt. Wenn die Funktion z.B. auf einem Rechner, der in der Zeitzone +01:00 (relativ zur UTC) konfiguriert wurde, ausgeführt wird, so wird nach Berechnung des UTC-**dateTime**-Äquivalents eine Stunde zum Ergebnis addiert. Auch die Zeitzoneninformation, die einen optionalen lexikalischen Bestandteil des **xs:dateTime**-Ergebnisses bildet, wird im **dateTime**-Ergebnis ausgegeben. Vergleichen Sie dieses Ergebnis mit dem von **dateTime-from-epoch-no-TZ** und auch der Funktion **epoch-from-dateTime**.

Beispiele

© 2019-2025 Altova GmbH

In den Beispielen unten wird eine lokale Zeitzone von UTC +01:00 angenommen. Das UTC **dateTime**-Äquivalent der angegebenen Epochenzeit wird folglich um eine Stunde erhöht. Die Zeitzone wird im Ergebnis ausgegeben.

- **altova:dateTime-from-epoch**(34) gibt 1970-01-01T01:00:34+01:00 zurück.
- **altova:dateTime-from-epoch**(62) gibt 1970-01-01T01:01:02+01:00 zurück.

dateTime-from-epoch-no-TZ [altova:]

altova:dateTime-from-epoch-no-TZ(Epoch *als xs:decimal* **als xs:dateTime** XP3.1 XQ3.1
Die Epochenzeit ist ein auf Unix-Systemen verwendetes Zeitsystem. Darin wird jeder Zeitpunkt als Anzahl der Sekunden seit 00:00:00 UTC des 1. Januar 1970 definiert. Die Funktion **dateTime-from-epoch-no-TZ** gibt das **xs:dateTime**-Äquivalent einer Epochenzeit zurück, passt es an die lokale Zeitzone an, inkludiert die Zeitzoneninformation jedoch nicht im Ergebnis.

Die Funktion erhält ein xs:decimal-Argument und gibt einen xs:dateTime-Wert, der keinen **zz**-Teil (Zeitzone) enthält, zurück. Das Ergebnis wird durch Berechnung des UTC **dateTime**-Äquivalents der Epochenzeit und Hinzufügen der (anhand der Systemuhr ermittelten) lokalen Zeitzone ermittelt. Wenn die Funktion z.B. auf einem Rechner, der in der Zeitzone +01:00 (relativ zur UTC) konfiguriert wurde, ausgeführt wird, so wird nach Berechnung des UTC-**dateTime**-Äquivalents eine Stunde zum Ergebnis addiert. Die Zeitzoneninformation, die einen optionalen lexikalischen Bestandteil des **xs:dateTime**-Ergebnisses bildet, wird nicht im **dateTime**-Ergebnis ausgegeben. Vergleichen Sie dieses Ergebnis mit dem von **dateTime-from-epoch** und auch der Funktion **epoch-from-dateTime**.

Beispiele

In den Beispielen unten wird eine lokale Zeitzone von UTC +01:00 angenommen. Das UTC **dateTime**-Äquivalent der angegebenen Epochenzeit wird folglich um eine Stunde erhöht. Die Zeitzone wird nicht im Ergebnis ausgegeben.

- **altova:dateTime-from-epoch**(34) gibt 1970-01-01T01:00:34 zurück.
- **altova:dateTime-from-epoch**(62) gibt 1970-01-01T01:01:02 zurück.

epoch-from-dateTime [altova:]

altova:epoch-from-dateTime(dateTimeValue *als xs:dateTime*) **als xs:decimal** XP3.1 XQ3.1
Die Epochenzeit ist ein auf Unix-Systemen verwendetes Zeitsystem. Darin wird jeder Zeitpunkt als Anzahl der Sekunden seit 00:00:00 UTC des 1. Januar 1970 definiert. Die Funktion **epoch-from-dateTime** gibt das Epochenzeitäquivalent von **xs:dateTime** zurück, welches als Argument der Funktion bereitgestellt wird. Beachten Sie, dass Sie den **xs:dateTime**-Wert eventuell explizit konstruieren müssen. Der angegebene **xs:dateTime**-Wert kann den optionalen **zz** (Zeitzonen)-Wert enthalten, muss ihn aber nicht enthalten.

Unabhängig davon, ob der Zeitzonenteil als Bestandteil des Arguments angegeben wird oder nicht, wird der (anhand der Systemuhr ermittelte) lokale Zeitzonenunterschied vom angegebenen **dateTimeValue**-Argument subtrahiert. Dadurch wird das UTC-Zeit-Äquivalent erzeugt, anhand dessen die entsprechende Epochenzeit berechnet wird. Wenn die Funktion z.B. auf einem Rechner, der für die Zeitzone +01:00 (relativ zur UTC) konfiguriert wurde, ausgeführt wird, so wird vor Berechnung des Epochenzeitwerts eine Stunde vom angegebenen **dateTimeValue** subtrahiert. Siehe dazu auch die Funktion **dateTime-from-**

epoch.

Beispiele

In den Beispielen unten wird eine lokale Zeitzone von UTC +01:00 angenommen. Daher wird vor Berechnung der Epochenzeit eine Stunde vom angegebenen **dateTime**-Wert subtrahiert.

- **altova:epoch-from-dateTime**(xs:dateTime("1970-01-01T01:00:34+01:00")) gibt 34 zurück.
- **altova:epoch-from-dateTime**(xs:dateTime("1970-01-01T01:00:34")) gibt 34 zurück.
- **altova:epoch-from-dateTime**(xs:dateTime("2021-04-01T11:22:33")) gibt 1617272553 zurück.

[Nach oben 582]

12.2.1.3 XPath/XQuery-Funktionen: Standort

Die folgenden XPath/XQuery-Erweiterungsfunktionen zu Standortdaten werden in der aktuellen Version von StyleVision unterstützt und können in (i) in einem XSLT-Kontext in XPath-Ausdrücken oder (ii) in einem XQuery-Dokument in einem XQuery-Ausdruck verwendet werden.

Anmerkung zur Benennung von Funktionen und zur Anwendbarkeit der Sprache

Altova-Erweiterungsfunktionen können in XPath/XQuery-Ausdrücken verwendet werden. Dadurch stehen neben den Funktionen in der Standardbibliothek der XPath-, XQuery- und XSLT-Funktionen zusätzliche Funktionen zur Verfügung. Die Altova-Erweiterungsfunktionen befinden sich im **Altova-Erweiterungsfunktions-Namespace, http://www.altova.com/xslt-extensions** und sind in diesem Abschnitt mit dem Präfix **altova:**, das als an diesen Namespace gebunden angenommen wird, gekennzeichnet. Beachten Sie, dass manche Funktionen in zukünftigen Versionen Ihres Produkts eventuell nicht mehr unterstützt werden oder dass sich das Verhalten einzelner Funktionen ändern kann. Um zu sehen, welche Altova-Erweiterungsfunktionen unterstützt werden, lesen Sie bitte die Dokumentation zur jeweiligen Release.

XPath-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	XP1 XP2 XP3.1
XSLT-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	XSLT1 XSLT2 XSLT3
XQuery-Funktionen (in XQuery-Ausdrücken in XQuery verwendet):	XQ1 XQ3.1

format-geolocation [altova:]

altova:format-geolocation(Latitude *als xs:decimal*, **Longitude** *als xs:decimal*, **GeolocationOutputStringFormat** *als xs:integer*) als **xs:string** XP3.1 XQ3.1
Erhält als die ersten beiden Argumente die geografische Breite und Länge und gibt den Standort als String zurück. Das dritte Argument, **GeolocationOutputStringFormat**, ist das Format des Ausgabestring für den Standort; darin werden zum Identifizieren des Ausgabestringformats Ganzzahlwerte von 1 bis 4 verwendet (*siehe 'Format des Ausgabestrings für die geografische Position' weiter unten*). Die Werte für die Breite liegen im Bereich von +90 bis -90 (N nach S). Die Werte für die Länge liegen im Bereich von +180 bis -180 (O nach W).

Anmerkung: Zur Bereitstellung der Input-Strings können die Funktion image-exif-data 612 und die

 © 2019-2025 Altova GmbH

Attribute der Exif-Metadaten verwendet werden.

Beispiele

- **altova:format-geolocation**(33.33, -22.22, 4) gibt xs:string "33.33 -22.22" zurück
- **altova:format-geolocation**(33.33, -22.22, 2) gibt xs:string "33.33N 22.22W" zurück
- **altova:format-geolocation**(-33.33, 22.22, 2) gibt xs:string "33.33S 22.22E" zurück
- **altova:format-geolocation**(33.33, -22.22, 1) gibt xs:string "33°19'48.00"S 22° 13'12.00"E" zurück

Ausgabestringformate für die geografische Position:

Die bereitgestellte Breite und Länge ist in einem der unten aufgelisteten Ausgabeformate formatiert. Das gewünschte Format wird anhand seiner Ganzzahl-ID (1 bis 4) identifiziert. Die Breitenwerte liegen im Bereich von +90 bis -90 (N nach S). Die Längenwerte liegen im Bereich von +180 bis -180 (O nach W).

1
Grad, Minuten, Dezimalsekunden, mit nachgestellter Orientierung (N/S, E/W) **D°M'S.SS"N/S D°M'S.SS"E/W** *Beispiel*: **33°55'11.11"N 22°44'66.66"W**

2
Dezimalgrad, mit nachgestellter Orientierung (N/S, E/W) **D.DDN/S D.DDE/W** *Beispiel*: **33.33N 22.22W**

3
Grad, Minuten, Dezimalsekunden mit Vorzeichen (+/-); Plus-Zeichen für (**N/E**) ist optional **+/-D°M'S.SS" +/-D°M'S.SS"** *Beispiel*: **33°55'11.11" -22°44'66.66"**

4
Dezimalgrad, mit Vorzeichen (+/-); Plus-Zeichen für (**N/E**) ist optional **+/-D.DD +/-D.DD** *Beispiel*: **33.33 -22.22**

Altova Exif-Attribut: Geolocation

Der Altova XPath/XQuery-Prozessor generiert anhand der Exif-Standard-Metadaten-Tags das benutzerdefinierte Attribut **Geolocation**. **Geolocation** ist eine Verkettung von vier Exif-Tags: GPSLatitude, GPSLatitudeRef, GPSLongitude, GPSLongitudeRef mit hinzugefügten Werten (*siehe Tabelle unten*).

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation

33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

parse-geolocation [altova:]

altova:parse-geolocation(GeolocationInputString *als xs:string***)** als **xs:decimal+** **XP3.1** **XQ3.1**
Parst das bereitgestellte `GeolocationInputString`-Argument und gibt die geografische Breite und Länge (in dieser Reihenfolge) als Sequenz aus zwei `xs:decimal` Elementen zurück. Die Formate, in denen der Input-String für die geografische Position bereitgestellt werden kann, sind unten aufgelistet.

Anmerkung: Zur Bereitstellung des Input-String für die geografische Position können die Funktion image-exif-data[612] und das @Geolocation[612]-Attribut der Exif-Metadaten verwendet werden (*siehe Beispiel unten*).

Beispiele

- **altova:parse-geolocation**("33.33 -22.22") gibt die Sequenz bestehend aus zwei `xs:decimals (33.33, 22.22)` Elementen zurück
- **altova:parse-geolocation**("48°51'29.6""N 24°17'40.2""") gibt die Sequenz bestehend aus zwei `xs:decimals (48.8582222222222, 24.2945)` Elementen zurück
- **altova:parse-geolocation**('48°51''29.6"N 24°17''40.2"') gibt die Sequenz bestehend aus zwei `xs:decimals (48.8582222222222, 24.2945)` Elementen zurück
- **altova:parse-geolocation**(**image-exif-data**(//MyImages/Image20141130.01)**/@Geolocation**) gibt die Sequenz bestehend aus zwei `xs:decimals` Elementen zurück

Input-String-Formate der Standortdaten:

Der Input-String für die geografische Position muss die Breite und Länge (in dieser Reihenfolge) getrennt durch ein Leerzeichen enthalten. Beide Werte können jedes der folgenden Formate haben. Auch Kombinationen sind zulässig, d.h. die Breite kann in einem anderen Format als die Länge angegeben werden. Die Breitenwerte liegen im Bereich `+90` bis `-90` (`N` nach `S`). Die Längenwerte liegen im Bereich von `+180` bis `-180` (`O` nach `W`).

Anmerkung: Wenn als Trennzeichen für den Input-String einfache oder doppelte Anführungszeichen verwendet werden, kann dies zu einer Fehlinterpretation der einfachen bzw. doppelten Anführungszeichen als Minuten- bzw. Sekundenwerte führen. In solchen Fällen müssen die zur Angabe der Minuten- und Sekundenwerte verwendeten Anführungszeichen durch Verdoppelung mit einem Escape-Zeichen versehen werden. In den Beispielen in diesem Abschnitt sind Anführungszeichen, die als Trennzeichen für den Input-String dienen, gelb markiert (**"**), während Maßeinheitsangaben blau (**""**) markiert sind.

- Grad, Minuten, Dezimalsekunden, mit nachgestellter Orientierung (`N/S, O/W`)
  D°M'S.SS"N/S D°M'S.SS"W/E
  Beispiel: **33°55'11.11"N 22°44'55.25"W**

- Grad, Minuten, Dezimalsekunden mit Vorzeichen (`+/-`); das Plus-Zeichen für (**N**/O) ist optional

© 2019-2025 Altova GmbH

`+/-D°M'S.SS" +/-D°M'S.SS"`
Beispiel: `33°55'11.11" -22°44'55.25"`

- Grad, Dezimalminuten mit nachgestellter Orientierung (`N/S`, `O/W`)
  `D°M.MM'N/S D°M.MM'W/E`
  Beispiel: `33°55.55'N 22°44.44'W`

- Grad, Dezimalminuten mit Vorzeichen (`+/-`); das Plus-Zeichen für (`N/O`) ist optional
  `+/-D°M.MM' +/-D°M.MM'`
  Beispiel: `+33°55.55' -22°44.44'`

- Dezimalgrade, mit nachgestellter Orientierung (`N/S`, `O/W`)
  `D.DDN/S D.DDW/E`
  Beispiel: `33.33N 22.22W`

- Dezimalgrade mit Vorzeichen (`+/-`); das Plus-Zeichen für (`N/S O/W`) ist optional
  `+/-D.DD +/-D.DD`
  Beispiel: `33.33 -22.22`

Beispiele für Formatkombinationen:
`33.33N -22°44'55.25"`
`33.33 22°44'55.25"W`
`33.33 22.45`

Altova Exif-Attribut: Geolocation

Der Altova XPath/XQuery-Prozessor generiert anhand der Exif-Standard-Metadaten-Tags das benutzerdefinierte Attribut **Geolocation**. **Geolocation** ist eine Verkettung von vier Exif-Tags: `GPSLatitude`, `GPSLatitudeRef`, `GPSLongitude`, `GPSLongitudeRef` mit hinzugefügten Werten (*siehe Tabelle unten*).

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

geolocation-distance-km [altova:]

altova:geolocation-distance-km(GeolocationInputString-1 *als xs:string*, **GeolocationInputString-2** *als xs:string*) als **xs:decimal** **XP3.1** **XQ3.1**
Berechnet die Entfernung zwischen zwei geografischen Positionen in Kilometern. Die Formate, in denen der Input-String für die geografischen Position angegeben werden kann, sind unten aufgelistet. Die Breitenwerte liegen im Bereich von `+90` bis `-90` (`N` nach `S`). Die Längenwerte liegen im Bereich von `+180` bis `-180` (`O` nach `W`).

Anmerkung: Zur Bereitstellung des Input-String für die geografische Position können die Funktion `image-exif-data` [612] und das `@Geolocation` [612]-Attribut der Exif-Metadaten verwendet werden.

Beispiele

- `altova:geolocation-distance-km("33.33 -22.22", "48°51'29.6""N 24°17'40.2""")` gibt `xs:decimal 4183.08132372392` zurück

Input-String-Formate der Standortdaten:

Der Input-String für die geografische Position muss die Breite und Länge (in dieser Reihenfolge) getrennt durch ein Leerzeichen enthalten. Beide Werte können jedes der folgenden Formate haben. Auch Kombinationen sind zulässig, d.h. die Breite kann in einem anderen Format als die Länge angegeben werden. Die Breitenwerte liegen im Bereich `+90` bis `-90` (`N` nach `S`). Die Längenwerte liegen im Bereich von `+180` bis `-180` (`O` nach `W`).

Anmerkung: Wenn als Trennzeichen für den Input-String einfache oder doppelte Anführungszeichen verwendet werden, kann dies zu einer Fehlinterpretation der einfachen bzw. doppelten Anführungszeichen als Minuten- bzw. Sekundenwerte führen. In solchen Fällen müssen die zur Angabe der Minuten- und Sekundenwerte verwendeten Anführungszeichen durch Verdoppelung mit einem Escape-Zeichen versehen werden. In den Beispielen in diesem Abschnitt sind Anführungszeichen, die als Trennzeichen für den Input-String dienen, gelb markiert (`"`), während Maßeinheitsangaben blau (`""`) markiert sind.

- Grad, Minuten, Dezimalsekunden, mit nachgestellter Orientierung (`N/S, O/W`)
  `D°M'S.SS"N/S D°M'S.SS"W/E`
  Beispiel: **`33°55'11.11"N 22°44'55.25"W`**

- Grad, Minuten, Dezimalsekunden mit Vorzeichen (`+/-`); das Plus-Zeichen für (**`N/O`**) ist optional
  `+/-D°M'S.SS" +/-D°M'S.SS"`
  Beispiel: **`33°55'11.11" -22°44'55.25"`**

- Grad, Dezimalminuten mit nachgestellter Orientierung (`N/S`, `O/W`)
  `D°M.MM'N/S D°M.MM'W/E`
  Beispiel: **`33°55.55'N 22°44.44'W`**

- Grad, Dezimalminuten mit Vorzeichen (`+/-`); das Plus-Zeichen für (**`N/O`**) ist optional
  `+/-D°M.MM' +/-D°M.MM'`
  Beispiel: **`+33°55.55' -22°44.44'`**

- Dezimalgrade, mit nachgestellter Orientierung (`N/S`, `O/W`)
  `D.DDN/S D.DDW/E`
  Beispiel: **`33.33N 22.22W`**

- Dezimalgrade mit Vorzeichen (`+/-`); das Plus-Zeichen für (**`N/S O/W`**) ist optional
  `+/-D.DD +/-D.DD`
  Beispiel: **`33.33 -22.22`**

Beispiele für Formatkombinationen:

```
33.33N -22°44'55.25"
33.33 22°44'55.25"W
33.33 22.45
```

Altova Exif-Attribut: Geolocation

Der Altova XPath/XQuery-Prozessor generiert anhand der Exif-Standard-Metadaten-Tags das

© 2019-2025 Altova GmbH

benutzerdefinierte Attribut **Geolocation**. **Geolocation** ist eine Verkettung von vier Exif-Tags: GPSLatitude, GPSLatitudeRef, GPSLongitude, GPSLongitudeRef mit hinzugefügten Werten (*siehe Tabelle unten*).

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

▼ geolocation-distance-mi [altova:]

altova:geolocation-distance-mi(GeolocationInputString-1 *als xs:string*, **GeolocationInputString-2** *als xs:string*) als **xs:decimal** **XP3.1** **XQ3.1**
Berechnet die Entfernung zwischen zwei geografischen Positionen in Meilen. Die Formate, in denen der Input-String für die geografischen Position angegeben werden kann, sind unten aufgelistet. Die Breitenwerte liegen im Bereich von +90 bis -90 (N nach S). Die Längenwerte liegen im Bereich von +180 bis -180 (O nach W).

Anmerkung: Zur Bereitstellung des Input-String für die geografische Position können die Funktion image-exif-data[612] und das @Geolocation[612]-Attribut der Exif-Metadaten verwendet werden.

▣ *Beispiele*

- **altova:geolocation-distance-mi**("33.33 -22.22", "48°51'29.6""N 24°17'40.2""") gibt xs:decimal 2599.40652340653 zurück

▣ *Input-String-Formate der Standortdaten:*

Der Input-String für die geografische Position muss die Breite und Länge (in dieser Reihenfolge) getrennt durch ein Leerzeichen enthalten. Beide Werte können jedes der folgenden Formate haben. Auch Kombinationen sind zulässig, d.h. die Breite kann in einem anderen Format als die Länge angegeben werden. Die Breitenwerte liegen im Bereich +90 bis -90 (N nach S). Die Längenwerte liegen im Bereich von +180 bis -180 (O nach W).

Anmerkung: Wenn als Trennzeichen für den Input-String einfache oder doppelte Anführungszeichen verwendet werden, kann dies zu einer Fehlinterpretation der einfachen bzw. doppelten Anführungszeichen als Minuten- bzw. Sekundenwerte führen. In solchen Fällen müssen die zur Angabe der Minuten- und Sekundenwerte verwendeten Anführungszeichen durch Verdoppelung mit einem Escape-Zeichen versehen werden. In den Beispielen in diesem Abschnitt sind Anführungszeichen, die als Trennzeichen für den Input-String dienen, gelb markiert ("), während Maßeinheitsangaben blau ("") markiert sind.

- Grad, Minuten, Dezimalsekunden, mit nachgestellter Orientierung (N/S, O/W)
  D°M'S.SS"N/S D°M'S.SS"W/E
  Beispiel: **33°55'11.11"N 22°44'55.25"W**

- Grad, Minuten, Dezimalsekunden mit Vorzeichen (+/-); das Plus-Zeichen für (**N/O**) ist optional

`+/-D°M'S.SS" +/-D°M'S.SS"`
Beispiel: `33°55'11.11" -22°44'55.25"`

- Grad, Dezimalminuten mit nachgestellter Orientierung (`N/S`, `O/W`)
  `D°M.MM'N/S D°M.MM'W/E`
  Beispiel: `33°55.55'N 22°44.44'W`

- Grad, Dezimalminuten mit Vorzeichen (`+/-`); das Plus-Zeichen für (`N`/`O`) ist optional
  `+/-D°M.MM' +/-D°M.MM'`
  Beispiel: `+33°55.55' -22°44.44'`

- Dezimalgrade, mit nachgestellter Orientierung (`N/S`, `O/W`)
  `D.DDN/S D.DDW/E`
  Beispiel: `33.33N 22.22W`

- Dezimalgrade mit Vorzeichen (`+/-`); das Plus-Zeichen für (`N/S O/W`) ist optional
  `+/-D.DD +/-D.DD`
  Beispiel: `33.33 -22.22`

Beispiele für Formatkombinationen:
```
33.33N -22°44'55.25"
33.33 22°44'55.25"W
33.33 22.45
```

Altova Exif-Attribut: Geolocation

Der Altova XPath/XQuery-Prozessor generiert anhand der Exif-Standard-Metadaten-Tags das benutzerdefinierte Attribut **Geolocation**. **Geolocation** ist eine Verkettung von vier Exif-Tags: `GPSLatitude`, `GPSLatitudeRef`, `GPSLongitude`, `GPSLongitudeRef` mit hinzugefügten Werten (*siehe Tabelle unten*).

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

geolocations-bounding-rectangle [altova:]

altova:geolocations-bounding-rectangle(Geolocations *als xs:sequence*, **GeolocationOutputStringFormat** *als xs:integer***)** als **xs:string** **XP3.1** **XQ3.1**
Erhält als erstes Argument eine Sequenz von Strings, wobei es sich bei jedem String in der Sequenz um eine geografische Position handelt. Die Funktion gibt eine Sequenz von zwei Strings zurück, die die geografischen Positionskoordinaten der linken oberen bzw. rechten unteren Ecke eines Rechtecks bilden, dessen Größe so angepasst ist, dass es alle im ersten Argument angegebenen Positionskoordinaten enthält. Die Formate, in denen der Input-String für die geografischen Position angegeben werden kann, sind unten aufgelistet (*siehe 'Input-String-Formate der Standortdaten'*). Die Breitenwerte liegen im Bereich von `+90` bis `-90` (`N` nach `S`). Die Längenwerte liegen im Bereich von `+180` bis `-180` (`O` nach `W`).

© 2019-2025 Altova GmbH

Im zweiten Argument der Funktion ist das Format der beiden Geolocation-Strings in der Ausgabesequenz angegeben. Das Argument erhält einen Ganzzahlwert von `1` bis `4`, wobei die einzelnen Werte ein jeweils unterschiedliches String-Format definieren (*siehe 'Ausgabestringsformate für die geografische Position' weiter unten).*

Anmerkung: Zur Bereitstellung der Input-Strings können die Funktion image-exif-data[612] und die Attribute der Exif-Metadaten verwendet werden.

Beispiele

- `altova:geolocations-bounding-rectangle(("48.2143531 16.3707266", "51.50939 -0.11832"), 1)` gibt die Sequenz `("51°30'33.804"N 0°7'5.952"W", "48°12'51.67116"N 16°22'14.61576"E")` zurück.
- `altova:geolocations-bounding-rectangle(("48.2143531 16.3707266", "51.50939 -0.11832", "42.5584577 -70.8893334"), 4)` gibt die Sequenz `("51.50939 -70.8893334", "42.5584577 16.3707266")` zurück.

Input-String-Formate der Standortdaten:

Der Input-String für die geografische Position muss die Breite und Länge (in dieser Reihenfolge) getrennt durch ein Leerzeichen enthalten. Beide Werte können jedes der folgenden Formate haben. Auch Kombinationen sind zulässig, d.h. die Breite kann in einem anderen Format als die Länge angegeben werden. Die Breitenwerte liegen im Bereich `+90` bis `-90` (`N` nach `S`). Die Längenwerte liegen im Bereich von `+180` bis `-180` (`O` nach `W`).

Anmerkung: Wenn als Trennzeichen für den Input-String einfache oder doppelte Anführungszeichen verwendet werden, kann dies zu einer Fehlinterpretation der einfachen bzw. doppelten Anführungszeichen als Minuten- bzw. Sekundenwerte führen. In solchen Fällen müssen die zur Angabe der Minuten- und Sekundenwerte verwendeten Anführungszeichen durch Verdoppelung mit einem Escape-Zeichen versehen werden. In den Beispielen in diesem Abschnitt sind Anführungszeichen, die als Trennzeichen für den Input-String dienen, gelb markiert (`"`), während Maßeinheitsangaben blau (`""`) markiert sind.

- Grad, Minuten, Dezimalsekunden, mit nachgestellter Orientierung (`N/S, O/W`)
  `D°M'S.SS"N/S D°M'S.SS"W/E`
  Beispiel: `33°55'11.11"N 22°44'55.25"W`

- Grad, Minuten, Dezimalsekunden mit Vorzeichen (`+/-`); das Plus-Zeichen für (`N/O`) ist optional
  `+/-D°M'S.SS" +/-D°M'S.SS"`
  Beispiel: `33°55'11.11" -22°44'55.25"`

- Grad, Dezimalminuten mit nachgestellter Orientierung (`N/S`, `O/W`)
  `D°M.MM'N/S D°M.MM'W/E`
  Beispiel: `33°55.55'N 22°44.44'W`

- Grad, Dezimalminuten mit Vorzeichen (`+/-`); das Plus-Zeichen für (`N/O`) ist optional
  `+/-D°M.MM' +/-D°M.MM'`
  Beispiel: `+33°55.55' -22°44.44'`

- Dezimalgrade, mit nachgestellter Orientierung (`N/S`, `O/W`)
  `D.DDN/S D.DDW/E`

Beispiel: **33.33N 22.22W**

- Dezimalgrade mit Vorzeichen (+/-); das Plus-Zeichen für (**N/S o/W**) ist optional
  +/-D.DD +/-D.DD
  Beispiel: **33.33 -22.22**

Beispiele für Formatkombinationen:
33.33N -22°44'55.25"
33.33 22°44'55.25"W
33.33 22.45

- *Ausgabestringformate für die geografische Position:*

Die bereitgestellte Breite und Länge ist in einem der unten aufgelisteten Ausgabeformate formatiert. Das gewünschte Format wird anhand seiner Ganzzahl-ID (1 bis 4) identifiziert. Die Breitenwerte liegen im Bereich von +90 bis -90 (N nach S). Die Längenwerte liegen im Bereich von +180 bis -180 (O nach W).

1
Grad, Minuten, Dezimalsekunden, mit nachgestellter Orientierung (N/S, E/W) **D°M'S.SS"N/S D°M'S.SS"E/W** *Beispiel:* **33°55'11.11"N 22°44'66.66"W**

2
Dezimalgrad, mit nachgestellter Orientierung (N/S, E/W) **D.DDN/S D.DDE/W** *Beispiel:* **33.33N 22.22W**

3
Grad, Minuten, Dezimalsekunden mit Vorzeichen (+/-); Plus-Zeichen für (**N/E**) ist optional **+/-D°M'S.SS" +/-D°M'S.SS"** *Beispiel:* **33°55'11.11" -22°44'66.66"**

4
Dezimalgrad, mit Vorzeichen (+/-); Plus-Zeichen für (**N/E**) ist optional **+/-D.DD +/-D.DD** *Beispiel:* **33.33 -22.22**

- *Altova Exif-Attribut: Geolocation*

Der Altova XPath/XQuery-Prozessor generiert anhand der Exif-Standard-Metadaten-Tags das benutzerdefinierte Attribut **Geolocation**. **Geolocation** ist eine Verkettung von vier Exif-Tags: GPSLatitude, GPSLatitudeRef, GPSLongitude, GPSLongitudeRef mit hinzugefügten Werten (*siehe Tabelle unten*).

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation

© 2019-2025 Altova GmbH

33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

▼ geolocation-within-polygon [altova:]

altova:geolocation-within-polygon(Geolocation *als xs:string*, **((PolygonPoint** *als xs:string*)+)) als **xs:boolean** **XP3.1** **XQ3.1**

Ermittelt ob sich **Geolocation** (das erste Argument) innerhalb des durch die **PolygonPoint**-Argumente beschriebenen Polygonbereichs befindet. Wenn die PolygonPoint-Argumente keine geschlossene Form (wenn der erste und der letzte Punkt identisch sind) bilden, so wird der erste Punkt implizit zum letzten Punkt hinzugefügt, um die Form zu schließen. Alle Argumente (Geolocation und PolygonPoint+) werden durch Input-Strings für die geografische Position (*Formatliste siehe unten*) angegeben. Wenn sich das Geolocation Argument innerhalb des Polygons befindet, gibt die Funktion true() zurück; andernfalls gibt sie false() zurück. Die Breitenwerte liegen im Bereich von +90 bis -90 (N nach S). Die Längenwerte liegen im Bereich von +180 bis -180 (O nach W).

Anmerkung: Zur Bereitstellung des Input-String für die geografische Position können die Funktion image-exif-data[612] und das @Geolocation[612]-Attribut der Exif-Metadaten verwendet werden.

▼ *Beispiele*

- **altova:geolocation-within-polygon("33 -22", ("58 -32", "-78 -55", "48 24", "58 -32"))** gibt true() zurück
- **altova:geolocation-within-polygon("33 -22", ("58 -32", "-78 -55", "48 24"))** gibt true() zurück
- **altova:geolocation-within-polygon("33 -22", ("58 -32", "-78 -55", "48°51'29.6""N 24°17'40.2""")))** gibt true() zurück

▼ *Input-String-Formate der Standortdaten:*

Der Input-String für die geografische Position muss die Breite und Länge (in dieser Reihenfolge) getrennt durch ein Leerzeichen enthalten. Beide Werte können jedes der folgenden Formate haben. Auch Kombinationen sind zulässig, d.h. die Breite kann in einem anderen Format als die Länge angegeben werden. Die Breitenwerte liegen im Bereich +90 bis -90 (N nach S). Die Längenwerte liegen im Bereich von +180 bis -180 (O nach W).

Anmerkung: Wenn als Trennzeichen für den Input-String einfache oder doppelte Anführungszeichen verwendet werden, kann dies zu einer Fehlinterpretation der einfachen bzw. doppelten Anführungszeichen als Minuten- bzw. Sekundenwerte führen. In solchen Fällen müssen die zur Angabe der Minuten- und Sekundenwerte verwendeten Anführungszeichen durch Verdoppelung mit einem Escape-Zeichen versehen werden. In den Beispielen in diesem Abschnitt sind Anführungszeichen, die als Trennzeichen für den Input-String dienen, gelb markiert ("), während Maßeinheitsangaben blau ("") markiert sind.

- Grad, Minuten, Dezimalsekunden, mit nachgestellter Orientierung (N/S, O/W)
  D°M'S.SS"N/S D°M'S.SS"W/E
  Beispiel: **33°55'11.11"N 22°44'55.25"W**

- Grad, Minuten, Dezimalsekunden mit Vorzeichen (+/-); das Plus-Zeichen für (N/o) ist optional
  +/-D°M'S.SS" +/-D°M'S.SS"
  Beispiel: **33°55'11.11" -22°44'55.25"**

- Grad, Dezimalminuten mit nachgestellter Orientierung (N/S, O/W)
  D°M.MM'N/S D°M.MM'W/E
  Beispiel: **33°55.55'N 22°44.44'W**

- Grad, Dezimalminuten mit Vorzeichen (+/-); das Plus-Zeichen für (N/o) ist optional
  +/-D°M.MM' +/-D°M.MM'
  Beispiel: **+33°55.55' -22°44.44'**

- Dezimalgrade, mit nachgestellter Orientierung (N/S, O/W)
  D.DDN/S D.DDW/E
  Beispiel: **33.33N 22.22W**

- Dezimalgrade mit Vorzeichen (+/-); das Plus-Zeichen für (N/S O/W) ist optional
  +/-D.DD +/-D.DD
  Beispiel: **33.33 -22.22**

Beispiele für Formatkombinationen:
33.33N -22°44'55.25"
33.33 22°44'55.25"W
33.33 22.45

Altova Exif-Attribut: Geolocation

Der Altova XPath/XQuery-Prozessor generiert anhand der Exif-Standard-Metadaten-Tags das benutzerdefinierte Attribut **Geolocation**. **Geolocation** ist eine Verkettung von vier Exif-Tags: GPSLatitude, GPSLatitudeRef, GPSLongitude, GPSLongitudeRef mit hinzugefügten Werten (*siehe Tabelle unten*).

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

geolocation-within-rectangle [altova:]

altova:geolocation-within-rectangle(Geolocation *als xs:string*, **RectCorner-1** *als xs:string*, **RectCorner-2** *als xs:string*) als **xs:boolean** **XP3.1** **XQ3.1**
Ermittelt, ob sich **Geolocation** (das erste Argument) innerhalb des durch das zweite und dritte Argument, **RectCorner-1** und **RectCorner-2**, definierten Rechtecks befindet. **RectCorner-1** und **RectCorner-2** definieren gegenüberliegende Eckpunkte des Rechtecks. Alle Argumente (Geolocation, **RectCorner-1** und **RectCorner-2**) werden durch Input-Strings für die geografische Position (*Formatliste siehe unten*) angegeben. Wenn sich das Geolocation-Argument innerhalb des Rechtecks befindet, gibt die Funktion true() zurück; andernfalls gibt sie false() zurück. Die Breitenwerte liegen im Bereich von +90 bis -90 (N

© 2019-2025 Altova GmbH

nach S). Die Längenwerte liegen im Bereich von +180 bis -180 (O nach W).

Anmerkung: Zur Bereitstellung des Input-String für die geografische Position können die Funktion image-exif-data[612] und das @Geolocation[612]-Attribut der Exif-Metadaten verwendet werden.

Beispiele

- **altova:geolocation-within-rectangle**("33 -22", "58 -32", "-48 24") gibt true() zurück
- **altova:geolocation-within-rectangle**("33 -22", "58 -32", "48 24") gibt false() zurück
- **altova:geolocation-within-rectangle**("33 -22", "58 -32", "48°51'29.6""S 24° 17'40.2""") gibt true() zurück

Input-String-Formate der Standortdaten:

Der Input-String für die geografische Position muss die Breite und Länge (in dieser Reihenfolge) getrennt durch ein Leerzeichen enthalten. Beide Werte können jedes der folgenden Formate haben. Auch Kombinationen sind zulässig, d.h. die Breite kann in einem anderen Format als die Länge angegeben werden. Die Breitenwerte liegen im Bereich +90 bis -90 (N nach S). Die Längenwerte liegen im Bereich von +180 bis -180 (O nach W).

Anmerkung: Wenn als Trennzeichen für den Input-String einfache oder doppelte Anführungszeichen verwendet werden, kann dies zu einer Fehlinterpretation der einfachen bzw. doppelten Anführungszeichen als Minuten- bzw. Sekundenwerte führen. In solchen Fällen müssen die zur Angabe der Minuten- und Sekundenwerte verwendeten Anführungszeichen durch Verdoppelung mit einem Escape-Zeichen versehen werden. In den Beispielen in diesem Abschnitt sind Anführungszeichen, die als Trennzeichen für den Input-String dienen, gelb markiert ("), während Maßeinheitsangaben blau ("") markiert sind.

- Grad, Minuten, Dezimalsekunden, mit nachgestellter Orientierung (N/S, O/W)
  D°M'S.SS"N/S D°M'S.SS"W/E
  Beispiel: **33°55'11.11"N 22°44'55.25"W**

- Grad, Minuten, Dezimalsekunden mit Vorzeichen (+/-); das Plus-Zeichen für (**N**/O) ist optional
  +/-D°M'S.SS" +/-D°M'S.SS"
  Beispiel: **33°55'11.11" -22°44'55.25"**

- Grad, Dezimalminuten mit nachgestellter Orientierung (N/S, O/W)
  D°M.MM'N/S D°M.MM'W/E
  Beispiel: **33°55.55'N 22°44.44'W**

- Grad, Dezimalminuten mit Vorzeichen (+/-); das Plus-Zeichen für (**N**/O) ist optional
  +/-D°M.MM' +/-D°M.MM'
  Beispiel: **+33°55.55' -22°44.44'**

- Dezimalgrade, mit nachgestellter Orientierung (N/S, O/W)
  D.DDN/S D.DDW/E
  Beispiel: **33.33N 22.22W**

- Dezimalgrade mit Vorzeichen (+/-); das Plus-Zeichen für (**N/S O/W**) ist optional
  +/-D.DD +/-D.DD

Beispiel: `33.33 -22.22`

Beispiele für Formatkombinationen:

```
33.33N  -22°44'55.25"
33.33  22°44'55.25"W
33.33  22.45
```

- *Altova Exif-Attribut: Geolocation*

  Der Altova XPath/XQuery-Prozessor generiert anhand der Exif-Standard-Metadaten-Tags das benutzerdefinierte Attribut **`Geolocation`**. **`Geolocation`** ist eine Verkettung von vier Exif-Tags: `GPSLatitude`, `GPSLatitudeRef`, `GPSLongitude`, `GPSLongitudeRef` mit hinzugefügten Werten (*siehe Tabelle unten*).

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

[Nach oben 600]

12.2.1.4 XPath/XQuery-Funktionen: Bildbezogene

Die folgenden XPath/XQuery-Erweiterungsfunktionen im Zusammenhang mit Bildern werden in der aktuellen Version von StyleVision unterstützt und können in (i) in einem XSLT-Kontext in XPath-Ausdrücken oder (ii) in einem XQuery-Dokument in einem XQuery-Ausdruck verwendet werden.

Anmerkung zur Benennung von Funktionen und zur Anwendbarkeit der Sprache

Altova-Erweiterungsfunktionen können in XPath/XQuery-Ausdrücken verwendet werden. Dadurch stehen neben den Funktionen in der Standardbibliothek der XPath-, XQuery- und XSLT-Funktionen zusätzliche Funktionen zur Verfügung. Die Altova-Erweiterungsfunktionen befinden sich im **Altova-Erweiterungsfunktions-Namespace,** **`http://www.altova.com/xslt-extensions`** und sind in diesem Abschnitt mit dem Präfix **`altova:`**, das als an diesen Namespace gebunden angenommen wird, gekennzeichnet. Beachten Sie, dass manche Funktionen in zukünftigen Versionen Ihres Produkts eventuell nicht mehr unterstützt werden oder dass sich das Verhalten einzelner Funktionen ändern kann. Um zu sehen, welche Altova-Erweiterungsfunktionen unterstützt werden, lesen Sie bitte die Dokumentation zur jeweiligen Release.

XPath-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	XP1 XP2 XP3.1
XSLT-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	XSLT1 XSLT2 XSLT3
XQuery-Funktionen (in XQuery-Ausdrücken in XQuery verwendet):	XQ1 XQ3.1

© 2019-2025 Altova GmbH

▼ suggested-image-file-extension [altova:]

altova:suggested-image-file-extension(Base64String *als string*) als **string?** **XP3.1** **XQ3.1**
Erhält die Base64-Kodierung einer Bilddatei als Argument und gibt die darin enthaltene Dateierweiterung des Bilds zurück. Der Rückgabewert ist ein Vorschlag, basierend auf den in der Kodierung enthaltenen Bilddateitypinformationen. Wenn diese Informationen nicht verfügbar sind, wird ein leerer String zurückgegeben. Diese Funktion ist nützlich, wenn Sie ein Base64-Bild als Datei speichern und die entsprechende Dateierweiterung dynamisch abrufen möchten.

▣ *Beispiele*

- `altova:suggested-image-file-extension(/MyImages/MobilePhone/Image20141130.01)` gibt `'jpg'` zurück
- `altova:suggested-image-file-extension($XML1/Staff/Person/@photo)` gibt `''` zurück

In den Beispielen oben wird von den als Argument der Funktion bereitgestellten Nodes angenommen, dass sie ein Base64-kodiertes Bild enthalten. Im ersten Beispiel wird `jpg` als Dateityp bzw. Dateierweiterung abgerufen. Im zweiten Beispiel enthält die angegebene Base54-Kodierung keine brauchbaren Dateierweiterungsinformationen.

▼ image-exif-data [altova:]

altova:image-exif-data(Base64BinaryString *als string*) als **element?** **XP3.1** **XQ3.1**
Erhält ein Base64-kodiertes JPEG-Bild als Argument und gibt ein Element namens **Exif** zurück, das die Exif-Metadaten des Bilds enthält. Die Exif-Metadaten werden als Attribut-Wert-Paare des `Exif`-Elements erstellt. Bei den Attributnamen handelt es sich um die Exif-Daten-Tags aus der Base64-Kodierung. Weiter unten sehen Sie eine Liste der Exif-Tags. Wenn die Exif-Daten einen anbieterspezifischen Tag enthalten, so wird auch dieser Tag und sein Wert als Attribut-Wert-Paar zurückgegeben. Zusätzlich zu den Standard-Exif-Metadatentags (*siehe Liste unten*) werden auch Altova-spezifische Attribut-Wert-Paare generiert. Diese Altova Exif-Attribute sind unten aufgelistet.

▣ *Beispiele*

- Um ein einziges Attribut abzurufen, verwenden Sie die Funktion folgendermaßen:
  ```
  image-exif-data(//MyImages/Image20141130.01)/@GPSLatitude
  image-exif-data(//MyImages/Image20141130.01)/@Geolocation
  ```
- Um alle Attribute abzurufen, verwenden Sie die Funktion folgendermaßen:
  ```
  image-exif-data(//MyImages/Image20141130.01)/@*
  ```
- Um die Namen aller Attribute abzurufen, verwenden Sie den folgenden Ausdruck:
  ```
  for $i in image-exif-data(//MyImages/Image20141130.01)/@* return name($i)
  ```
  Auf diese Art können Sie die Namen der von der Funktion zurückgegebenen Attribute eruieren.

▣ *Altova Exif-Attribut: Geolocation*

Der Altova XPath/XQuery-Prozessor generiert anhand der Exif-Standard-Metadaten-Tags das benutzerdefinierte Attribut **Geolocation**. **Geolocation** ist eine Verkettung von vier Exif-Tags: `GPSLatitude`, `GPSLatitudeRef`, `GPSLongitude`, `GPSLongitudeRef` mit hinzugefügten Werten (*siehe Tabelle unten*).

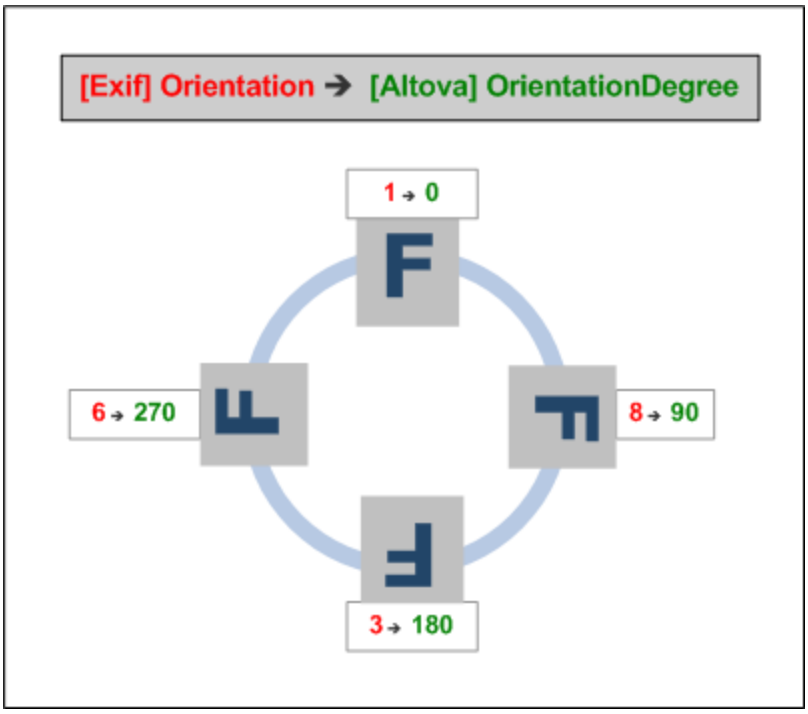
GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation

33 51 21.91	S	151 13 11.73	E	33°51'21.91"S 151° 13'11.73"E

- *Altova Exif-Attribut: OrientationDegree*

  Der Altova XPath/XQuery-Prozessor generiert anhand des Exif-Metadaten-Tags **Orientation** das benutzerdefinierte Attribut **OrientationDegree**.

  OrientationDegree übersetzt den Standard-Exif-Tag Orientation von einem Ganzzahlwert (1, 8, 3 oder 6) in die entsprechenden Gradwerte dafür (0, 90, 180, 270) (*siehe Abbildung unten*). Beachten Sie dass es keine Übersetzung der Orientation-Werte 2, 4, 5, 7 gibt. (Diese Ausrichtungen werden durch Spiegelung des Bilds 1 an seiner senkrechten Mittelachse zur Erzeugung des Bilds mit dem Wert 2 und anschließende Drehung dieses Bilds um jeweils 90 Grad zur Erzeugung der Werte 7 bzw. 4 bzw. 5 erzielt).



- *Liste der Standard-Exif-Metatags*

  - ImageWidth
  - ImageLength
  - BitsPerSample
  - Compression
  - PhotometricInterpretation
  - Orientation
  - SamplesPerPixel
  - PlanarConfiguration
  - YCbCrSubSampling

© 2019-2025 Altova GmbH

- YCbCrPositioning
- XResolution
- YResolution
- ResolutionUnit
- StripOffsets
- RowsPerStrip
- StripByteCounts
- JPEGInterchangeFormat
- JPEGInterchangeFormatLength
- TransferFunction
- WhitePoint
- PrimaryChromaticities
- YCbCrCoefficients
- ReferenceBlackWhite
- DateTime
- ImageDescription
- Make
- Model
- Software
- Artist
- Copyright

- ExifVersion
- FlashpixVersion
- ColorSpace
- ComponentsConfiguration
- CompressedBitsPerPixel
- PixelXDimension
- PixelYDimension
- MakerNote
- UserComment
- RelatedSoundFile
- DateTimeOriginal
- DateTimeDigitized
- SubSecTime
- SubSecTimeOriginal
- SubSecTimeDigitized
- ExposureTime
- FNumber
- ExposureProgram
- SpectralSensitivity
- ISOSpeedRatings
- OECF
- ShutterSpeedValue
- ApertureValue
- BrightnessValue
- ExposureBiasValue
- MaxApertureValue
- SubjectDistance
- MeteringMode
- LightSource
- Flash
- FocalLength
- SubjectArea

- FlashEnergy
- SpatialFrequencyResponse
- FocalPlaneXResolution
- FocalPlaneYResolution
- FocalPlaneResolutionUnit
- SubjectLocation
- ExposureIndex
- SensingMethod
- FileSource
- SceneType
- CFAPattern
- CustomRendered
- ExposureMode
- WhiteBalance
- DigitalZoomRatio
- FocalLengthIn35mmFilm
- SceneCaptureType
- GainControl
- Contrast
- Saturation
- Sharpness
- DeviceSettingDescription
- SubjectDistanceRange
- ImageUniqueID

- GPSVersionID
- GPSLatitudeRef
- GPSLatitude
- GPSLongitudeRef
- GPSLongitude
- GPSAltitudeRef
- GPSAltitude
- GPSTimeStamp
- GPSSatellites
- GPSStatus
- GPSMeasureMode
- GPSDOP
- GPSSpeedRef
- GPSSpeed
- GPSTrackRef
- GPSTrack
- GPSImgDirectionRef
- GPSImgDirection
- GPSMapDatum
- GPSDestLatitudeRef
- GPSDestLatitude
- GPSDestLongitudeRef
- GPSDestLongitude
- GPSDestBearingRef
- GPSDestBearing
- GPSDestDistanceRef
- GPSDestDistance
- GPSProcessingMethod
- GPSAreaInformation

© 2019-2025 Altova GmbH

- GPSDateStamp
- GPSDifferential

[Nach oben 612]

12.2.1.5 XPath/XQuery-Funktionen: Numerische

Die numerischen Erweiterungsfunktionen von Altova können in XPath- und XQuery-Ausdrücken verwendet werden und stellen zusätzliche Funktionen für die Verarbeitung von Daten zur Verfügung. Die Funktionen in diesem Abschnitt können mit dem **XPath 3.0- und XQuery 3.0**-Prozessor von Altova verwendet werden. Sie stehen im Zusammenhang mit XPath/XQuery zur Verfügung.

Anmerkung zur Benennung von Funktionen und zur Anwendbarkeit der Sprache

Altova-Erweiterungsfunktionen können in XPath/XQuery-Ausdrücken verwendet werden. Dadurch stehen neben den Funktionen in der Standardbibliothek der XPath-, XQuery- und XSLT-Funktionen zusätzliche Funktionen zur Verfügung. Die Altova-Erweiterungsfunktionen befinden sich im **Altova-Erweiterungsfunktions-Namespace,** `http://www.altova.com/xslt-extensions` und sind in diesem Abschnitt mit dem Präfix `altova:`, das als an diesen Namespace gebunden angenommen wird, gekennzeichnet. Beachten Sie, dass manche Funktionen in zukünftigen Versionen Ihres Produkts eventuell nicht mehr unterstützt werden oder dass sich das Verhalten einzelner Funktionen ändern kann. Um zu sehen, welche Altova-Erweiterungsfunktionen unterstützt werden, lesen Sie bitte die Dokumentation zur jeweiligen Release.

XPath-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	XP1 XP2 XP3.1
XSLT-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	XSLT1 XSLT2 XSLT3
XQuery-Funktionen (in XQuery-Ausdrücken in XQuery verwendet):	XQ1 XQ3.1

Funktionen zur automatischen Nummerierung

▼ generate-auto-number [altova:]

altova:generate-auto-number(ID *als xs:string*, **StartsWith** *als xs:double*, **Increment** *als xs:double*, **ResetOnChange** *als xs:string*) als `xs:integer` XP1 XP2 XQ1 XP3.1 XQ3.1
Generiert jedes Mal, wenn die Funktion aufgerufen wird, eine Zahl. Die erste Zahl, die beim ersten Aufruf der Funktion generiert wird, wird durch das Argument `StartsWith` definiert. Bei jedem erneuten Aufruf der Funktion wird eine neue Zahl generiert. Diese Zahl wird durch den im Argument `Increment` definierten Wert anhand der zuvor generierten Zahl inkrementiert. Auf diese Art erstellt die Funktion `altova:generate-auto-number` einen Zähler, dessen Name durch das Argument `ID` definiert wird und der jedes Mal, wenn die Funktion aufgerufen wird, inkrementiert wird. Wenn sich der Wert des Arguments `ResetOnChange` seit dem vorherigen Funktionsaufruf geändert hat, so wird der Wert der zu generierenden Zahl auf den Wert `StartsWith` zurückgesetzt. Die Automatische Nummerierung kann auch mit der Funktion `altova:reset-auto-number` zurückgesetzt werden.

▣ *Beispiele*

- `altova:generate-auto-number("ChapterNumber", 1, 1, "SomeString")` gibt bei jedem Aufruf

der Funktion eine einzige Zahl beginnend mit `1` zurück, die bei jedem Aufruf der Funktion um `1` inkrementiert wird. Solange das vierte Argument in jedem anschließenden Aufruf `"SomeString"` bleibt, wird die Inkrementierung fortgesetzt. Wenn sich der Wert des vierten Arguments ändert, wird der Zähler (namens `ChapterNumber`) auf 1 zurückgesetzt. Der Wert von `ChapterNumber` kann auch folgendermaßen durch Aufruf der Funktion `altova:reset-auto-number` zurückgesetzt werden: `altova:reset-auto-number("ChapterNumber")`.

▼ reset-auto-number [altova:]

`altova:reset-auto-number(ID` *`als xs:string`***`)`** **XP1** **XP2** **XQ1** **XP3.1** **XQ3.1**
Diese Funktion setzt die Zahl des im `ID`-Argument angegebenen Zählers zur automatischen Nummerierung zurück. Die Zahl wird auf die Zahl zurückgesetzt, die durch das Argument `StartsWith` der Funktion `altova:generate-auto-number`, die den im `ID`-Argument genannten Zähler erstellt hat, definiert ist

▣ *Beispiele*

- **`altova:reset-auto-number`**`("ChapterNumber")` setzt die Zahl des Zählers zur automatischen Nummerierung (`ChapterNumber`), der durch die Funktion `altova:generate-auto-number` erstellt wurde, zurück. Die Zahl wird auf den Wert des Arguments `StartsWith` der Funktion `altova:generate-auto-number`, die `ChapterNumber` erstellt hat, zurückgesetzt.

[Nach oben 617]

Numerische Funktionen

▼ hex-string-to-integer [altova:]

`altova:hex-string-to-integer(HexString` *`als xs:string`***`)`** als **`xs:integer`** **XP3.1** **XQ3.1**
Verwendet ein String-Argument, das das Base-16-Äquivalent einer Ganzzahl im Dezimalsystem (Base-10) ist, und gibt die dezimale Ganzzahl zurück.

▣ *Beispiele*

- **`altova:hex-string-to-integer`**`('1')` gibt `1` zurück
- **`altova:hex-string-to-integer`**`('9')` gibt `9` zurück
- **`altova:hex-string-to-integer`**`('A')` gibt `10` zurück
- **`altova:hex-string-to-integer`**`('B')` gibt `11` zurück
- **`altova:hex-string-to-integer`**`('F')` gibt `15` zurück
- **`altova:hex-string-to-integer`**`('G')` gibt einen Fehler zurück
- **`altova:hex-string-to-integer`**`('10')` gibt `16` zurück
- **`altova:hex-string-to-integer`**`('01')` gibt `1` zurück
- **`altova:hex-string-to-integer`**`('20')` gibt `32` zurück
- **`altova:hex-string-to-integer`**`('21')` gibt `33` zurück
- **`altova:hex-string-to-integer`**`('5A')` gibt `90` zurück
- **`altova:hex-string-to-integer`**`('USA')` gibt einen Fehler zurück

▼ integer-to-hex-string [altova:]

`altova:integer-to-hex-string(Integer` *`as xs:integer`***`) as xs:string`** **XP3.1** **XQ3.1**
Verwendet ein Ganzzahlargument und gibt das Base-16-Äquivalent als String zurück.

© 2019-2025 Altova GmbH

Beispiele

- `altova:integer-to-hex-string(1)` gibt `1` zurück
- `altova:integer-to-hex-string(9)` gibt `'9'` zurück
- `altova:integer-to-hex-string(10)` gibt `'A'` zurück
- `altova:integer-to-hex-string(11)` gibt `'B'` zurück
- `altova:integer-to-hex-string(15)` gibt `'F'` zurück
- `altova:integer-to-hex-string(16)` gibt `'10'` zurück
- `altova:integer-to-hex-string(32)` gibt `'20'` zurück
- `altova:integer-to-hex-string(33)` gibt `'21'` zurück
- `altova:integer-to-hex-string(90)` gibt `'5A'` zurück

[Nach oben 617]

[nach oben 617]

12.2.1.6 XPath/XQuery-Funktionen: Schema

Die unten aufgelisteten Altova-Erweiterungsfunktionen geben Schemainformationen zurück. Weiter unten finden Sie Beschreibungen der Funktionen zusammen mit (i) Beispielen und (ii) einer Liste von Schemakomponenten und den dazugehörigen Eigenschaften. Diese Funktionen können mit dem **XPath 3.0-** und dem **XQuery 3.0-**Prozessor von Altova verwendet werden und stehen im Zusammenhang mit XPath/XQuery zur Verfügung.

Schemainformationen aus Schema-Dokumenten
Die Funktion `altova:schema` hat zwei Argumente: eines mit null Argumenten und das andere mit zwei Argumenten. Die Funktion mit null Argumenten gibt das gesamte Schema zurück. Sie können anschließend von diesem Ausgangspunkt aus durch das Schema navigieren und zu den gewünschten Schemakomponenten gehen. Die Funktion mit zwei Argumenten gibt eine bestimmte, durch Ihren QName identifizierte Komponentenart zurück. Der Rückgabewert ist in beiden Fällen eine Funktion. Um durch die zurückgegebene Komponente zu navigieren, müssen Sie eine Eigenschaft dieser spezifischen Komponente auswählen. Wenn es sich bei der Eigenschaft um ein nicht atomares Element handelt (d.h. wenn es sich um eine Komponente handelt), können Sie weiter navigieren, indem Sie eine Eigenschaft dieser Komponente auswählen. Wenn es sich bei der ausgewählten Eigenschaft um ein atomares Element handelt, wird der Wert des Elements zurückgegeben und Sie können nicht weiter navigieren.

Anmerkung: In XPath-Ausdrücken muss das Schema in die Verarbeitungsumgebung, z.B. in XSLT mit der Anweisung `xslt:import-schema` importiert werden. In XQuery-Ausdrücken muss das Schema explizit mit Hilfe eines Schemaimports importiert werden.

Schemainformationen aus XML-Nodes
Die Funktion `altova:type` übermittelt den Node eines XML-Dokuments und gibt die Typinformationen des Node aus dem PSVI zurück.

Anmerkung zur Benennung von Funktionen und zur Anwendbarkeit der Sprache

Altova-Erweiterungsfunktionen können in XPath/XQuery-Ausdrücken verwendet werden. Dadurch stehen neben den Funktionen in der Standardbibliothek der XPath-, XQuery- und XSLT-Funktionen zusätzliche Funktionen zur Verfügung. Die Altova-Erweiterungsfunktionen befinden sich im **Altova-**

Erweiterungsfunktions-Namespace, `http://www.altova.com/xslt-extensions` und sind in diesem Abschnitt mit dem Präfix `altova:`, das als an diesen Namespace gebunden angenommen wird, gekennzeichnet. Beachten Sie, dass manche Funktionen in zukünftigen Versionen Ihres Produkts eventuell nicht mehr unterstützt werden oder dass sich das Verhalten einzelner Funktionen ändern kann. Um zu sehen, welche Altova-Erweiterungsfunktionen unterstützt werden, lesen Sie bitte die Dokumentation zur jeweiligen Release.

XPath-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	XP1 XP2 XP3.1
XSLT-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	XSLT1 XSLT2 XSLT3
XQuery-Funktionen (in XQuery-Ausdrücken in XQuery verwendet):	XQ1 XQ3.1

Schema (null Argumente)

altova:schema() als (function(xs:string) als item()*)? XP3.1 XQ3.1
Gibt die **schema**-Komponente als ganzes zurück. Durch Auswahl einer der Eigenschaften der **schema**-Komponente können Sie weiter in die **schema**-Komponente navigieren.

- Wenn es sich bei dieser Eigenschaft um eine Komponente handelt, können Sie durch Auswahl einer dieser Komponenteneigenschaften einen Schritt tiefer navigieren. Dieser Schritt kann wiederholt werden, um weiter in das Schema zu navigieren.
- Wenn es sich bei der Komponente um einen atomaren Wert handelt, wird der atomare Wert zurückgegeben und Sie können nicht tiefer navigieren.

Die Eigenschaften der **schema**-Komponente sind:

```
"type definitions"
"attribute declarations"
"element declarations"
"attribute group definitions"
"model group definitions"
"notation declarations"
"identity-constraint definitions"
```

Die Eigenschaften aller anderen Komponentenarten (neben **schema**) sind unten aufgelistet.

Anmerkung: In XQuery-Ausdrücken muss das Schema explizit importiert werden. In XPath-Ausdrücken muss das Schema in die Verarbeitungsumgebung, z.B. in XSLT mit der Anweisung **xslt:import** importiert worden sein.

Beispiele

- `import schema "" at "C:\Test\ExpReport.xsd"; for $typedef in altova:schema() ("type definitions") return $typedef ("name")` gibt die Namen aller simpleTypes oder complexTypes im Schema zurück.

- `import schema "" at "C:\Test\ExpReport.xsd"; altova:schema() ("type definitions")[1]("name")` gibt den Namen des ersten aller simpleTypes oder complexTyps im Schema zurück.

© 2019-2025 Altova GmbH

Komponenten und ihre Eigenschaften

- Assertion

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Assertion"
test	XPath-Eigenschaftsdatensatz	

- Attribute Declaration

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Attribute Declaration"
name	String	Lokaler Name des Attributs
target namespace	String	Namespace URI des Attributs
type definition	SimpleType oder ComplexType	
scope	Eine Funktion mit Eigenschaften ("class":"Scope", "variety": "global" oder "local", "parent": der enthaltende ComplexTyp bzw. die enthaltende Attributgruppe)	
value constraint	Falls vorhanden, eine Funktion mit Eigenschaften ("class": "Value Constraint", "variety": "fixed" oder "default", "value": atomarer Wert, "lexical form": String. Beachten Sie, dass die Eigenschaft "value" für Namespace-sensitive Typen nicht zur Verfügung steht.	
inheritable	Boolean	

- Attribute Group Declaration

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Attribute Group Definition"
name	String	Lokaler Name der Attributgruppe
target namespace	String	Namespace URI der Attributgruppe
attribute uses	Sequenz von (Attribute Use)	
attribute wildcard	Optionale Attribut-Wildcard	

- Attribute Use

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert

kind	String	"Attribute Use"
required	Boolean	true, wenn das Attribut obligatorisch ist, false, wenn es optional ist.
value constraint	Siehe Attribute Declaration	
inheritable	Boolean	

Attribute Wildcard

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	string	"Wildcard"
namespace constraint	Funktion mit Eigenschaften ("class": "Namespace Constraint", "variety": "any"|"enumeration"|"not", "namespaces": Sequenz von xs:anyURI, "disallowed names": Liste mit QNames und/oder den Strings "defined" und "definedSiblings"	
process contents	String ("strict"|"lax"|"skip")	

Complex Type

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Complex Type"
name	String	Lokaler Name des Typs (leer, wenn anonym)
target namespace	String	Namespace URI des Typs (leer, wenn anonym)
base type definition	Complex Type Definition	
final	String-Sequenz ("restriction"|"extension")	
context	Leere Sequenz (nicht implementiert)	
derivation method	String ("restriction"|"extension")	
abstract	Boolean	
attribute uses	Attribute Use-Sequenz	
attribute wildcard	Optionale Attribute Wildcard	
content type	Funktion mit Eigenschaften: ("class":"Content Type", "variety":string ("element-only"|"empty"|"mixed"|"simple"), particle: optionales Partikel, "open content": Funktion mit Eigenschaften ("class":"Open Content", "mode": string	

© 2019-2025 Altova GmbH

	("interleave"|"suffix"), "wildcard": Wildcard), "simple type definition": Simple Type)	
prohibited substitutions	String-Sequenz ("restriction"|"extension")	
assertions	Assertion-Sequenz	

Element Declaration

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Complex Type"
name	String	Lokaler Name des Typs (leer, wenn anonym)
target namespace	String	Namespace URI des Typs (leer, wenn anonym)
type definition	Simple Type oder Complex Type	
type table	Funktion mit Eigenschaften ("class":"Type Table", "alternatives": Type Alternative-Sequenz, "default type definition": Simple Type oder Complex Type)	
scope	Funktion mit Eigenschaften ("class":"Scope", "variety": ("global"|"local"), "parent": optionaler Complex Type)	
value constraint	siehe Attribute Declaration	
nillable	Boolean	
identity-constraint definitions	Identity Constraint-Sequenz	
substitution group affiliations	Element Declaration-Sequenz	
substitution group exclusions	String-Sequenz ("restriction"|"extension")	
disallowed substitutions	String-Sequenz ("restriction"|"extension"|"substitution")	
abstract	Boolean	

Element Wildcard

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Wildcard"
namespace constraint	Funktion mit Eigenschaften ("class": "Namespace Constraint", "variety":	

	"any"|"enumeration"|"not", "namespaces": xs:anyURI-Sequenz, "disallowed names": Liste mit QNames und/oder den Strings "defined" und "definedSiblings"	
process contents	String ("strict"|"lax"|"skip")	

Facet

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	Der Name des Facet, z.B. "minLength" oder "enumeration"
value	abhängig vom Facet	Der Wert des Facet
fixed	Boolean	
typed-value	Nur für das Enumeration Facet, Array(xs:anyAtomicType*)	Ein Array, das Enumeration-Werte, von denen jeder im Allgemeinen eine Sequenz atomarer Werte sein kann, enthält. (Anmerkung: Die Eigenschaft "value" ist unabhängig vom tatsächlichen Typ für das Enumeration Facet eine String-Sequenz)

Identity Constraint

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Identity-Constraint Definition"
name	String	Lokaler Name des Constraint
target namespace	String	Namespace URI des Constraint
identity-constraint category	String ("key"|"unique"|"keyRef")	
selector	XPath-Eigenschaftsdatensatz	
fields	Sequenz von XPath-Eigenschaftsdatensätzen	
referenced key	(nur für keyRef): Identity Constraint	Der entsprechende Key Constraint

Model Group

Property name	Eigenschaftstyp	Eigenschaftswert
kind	String	"Model Group"
compositor	String ("sequence"|"choice"|"all")	
particles	Partikel-Sequenz	

© 2019-2025 Altova GmbH

Model Group Definition

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Model Group Definition"
name	String	Lokaler Name der Model Group
target namespace	String	Namespace URI der Model Group
model group	Model Group	

Notation

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Notation Declaration"
name	String	Lokaler Name der Notation
target namespace	String	Namespace URI der Notation
system identifier	anyURI	
public identifier	String	

Particle

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Particle"
min occurs	Integer	
max occurs	Integer oder String("unbounded")	
term	Element Declaration, Element Wildcard oder ModelGroup	

Simple Type

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Simple Type Definition"
name	String	Lokaler Name des Typs (leer, wenn anonym)
target namespace	String	Namespace URI des Typs (leer, wenn anonym)
final	String-Sequenz("restriction"|"extension"|"list"|"union")	
context	enthaltende Komponente	
base type definition	Simple Type	
facets	Facet-Sequenz	
fundamental facets	Leere Sequenz (nicht implementiert)	

variety	String ("atomic"|"list"|"union")	
primitive type definition	Simple Type	
item type definition	(nur für Listentypen) Simple Type	
member type definitions	(nur für Union-Typen) Simple Type-Sequenz	

Type Alternative

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Type Alternative"
test	XPath-Eigenschaftsdatensatz	
type definition	Simple Type oder Complex Type	

XPath Property Record

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
namespace bindings	Sequenz von Funktionen mit Eigenschaften ("prefix": string, "namespace": anyURI)	
default namespace	anyURI	
base URI	anyURI	Die statische Basis-UI des XPath-Ausdrucks
expression	String	Der XPath-Ausdruck als String

Schema (zwei Argumente)

altova:schema(ComponentKind *als xs:string*, **Name** *als xs:QName*) **als (function(xs:string) als item()*)? XP3.1 XQ3.1**

Gibt die im ersten Argument angegebene Komponentenart zurück, welche einen Namen hat, der mit dem im zweiten Argument angegebenen Namen übereinstimmt. Durch Auswahl einer der Eigenschaften der Komponente können Sie weiter navigieren.

- Wenn es sich bei dieser Eigenschaft um eine Komponente handelt, können Sie durch Auswahl einer dieser Komponenteneigenschaften einen Schritt tiefer navigieren. Dieser Schritt kann wiederholt werden, um weiter in das Schema zu navigieren.
- Wenn es sich bei der Komponente um einen atomaren Wert handelt, wird der atomare Wert zurückgegeben und Sie können nicht tiefer navigieren.

Anmerkung: In XQuery-Ausdrücken muss das Schema explizit importiert werden. In XPath-Ausdrücken muss das Schema in die Verarbeitungsumgebung, z.B. in XSLT mit der Anweisung **xslt:import** importiert worden sein.

Beispiele

```
import schema "" at "C:\Test\ExpReport.xsd";
```

```
altova:schema("element declaration", xs:QName("OrgChart"))("type definition")
("content type")("particles")[3]!.("term")("kind")
```
gibt die `kind`-Eigenschaft des Terms der dritten `particles`-Komponente zurück. Diese particles-Komponente ist ein Nachfahr der Element-Deklaration mit dem QName `QName OrgChart`.

- ```
  import schema "" at "C:\Test\ExpReport.xsd";
  let $typedef := altova:schema("type definition", xs:QName("emailType"))
  for $facet in $typedef ("facets")
  return [$facet ("kind"), $facet("value")]
  ```
  gibt für jedes **`facet`** jeder **`emailType`**-Komponente ein Array mit der Art und dem Wert des Facet zurück.

Komponenten und ihre Eigenschaften

Assertion

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Assertion"
test	XPath-Eigenschaftsdatensatz	

Attribute Declaration

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Attribute Declaration"
name	String	Lokaler Name des Attributs
target namespace	String	Namespace URI des Attributs
type definition	SimpleType oder ComplexType	
scope	Eine Funktion mit Eigenschaften ("class":"Scope", "variety": "global" oder "local", "parent": der enthaltende ComplexTyp bzw. die enthaltende Attributgruppe)	
value constraint	Falls vorhanden, eine Funktion mit Eigenschaften ("class": "Value Constraint", "variety": "fixed" oder "default", "value": atomarer Wert, "lexical form": String. Beachten Sie, dass die Eigenschaft "value" für Namespace-sensitive Typen nicht zur Verfügung steht.	
inheritable	Boolean	

Attribute Group Declaration

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
  ```

kind	String	"Attribute Group Definition"
name	String	Lokaler Name der Attributgruppe
target namespace	String	Namespace URI der Attributgruppe
attribute uses	Sequenz von (Attribute Use)	
attribute wildcard	Optionale Attribut-Wildcard	

Attribute Use

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Attribute Use"
required	Boolean	true, wenn das Attribut obligatorisch ist, false, wenn es optional ist.
value constraint	Siehe Attribute Declaration	
inheritable	Boolean	

Attribute Wildcard

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	string	"Wildcard"
namespace constraint	Funktion mit Eigenschaften ("class": "Namespace Constraint", "variety": "any"|"enumeration"|"not", "namespaces": Sequenz von xs:anyURI, "disallowed names": Liste mit QNames und/oder den Strings "defined" und "definedSiblings"	
process contents	String ("strict"|"lax"|"skip")	

Complex Type

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Complex Type"
name	String	Lokaler Name des Typs (leer, wenn anonym)
target namespace	String	Namespace URI des Typs (leer, wenn anonym)
base type definition	Complex Type Definition	
final	String-Sequenz ("restriction"|"extension")	
context	Leere Sequenz (nicht implementiert)	

© 2019-2025 Altova GmbH

derivation method	String ("restriction"|"extension")	
abstract	Boolean	
attribute uses	Attribute Use-Sequenz	
attribute wildcard	Optionale Attribute Wildcard	
content type	Funktion mit Eigenschaften: ("class":"Content Type", "variety":string ("element-only"|"empty"|"mixed"|"simple"), particle: optionales Partikel, "open content": Funktion mit Eigenschaften ("class":"Open Content", "mode": string ("interleave"|"suffix"), "wildcard": Wildcard), "simple type definition": Simple Type)	
prohibited substitutions	String-Sequenz ("restriction"|"extension")	
assertions	Assertion-Sequenz	

Element Declaration

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Complex Type"
name	String	Lokaler Name des Typs (leer, wenn anonym)
target namespace	String	Namespace URI des Typs (leer, wenn anonym)
type definition	Simple Type oder Complex Type	
type table	Funktion mit Eigenschaften ("class":"Type Table", "alternatives": Type Alternative-Sequenz, "default type definition": Simple Type oder Complex Type)	
scope	Funktion mit Eigenschaften ("class":"Scope", "variety": ("global"|"local"), "parent": optionaler Complex Type)	
value constraint	siehe Attribute Declaration	
nillable	Boolean	
identity-constraint definitions	Identity Constraint-Sequenz	
substitution group affiliations	Element Declaration-Sequenz	
substitution group	String-Sequenz ("restriction"|"extension")	

exclusions		
disallowed substitutions	String-Sequenz ("restriction"|"extension"|"substitution")	
abstract	Boolean	

Element Wildcard

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Wildcard"
namespace constraint	Funktion mit Eigenschaften ("class": "Namespace Constraint", "variety": "any"|"enumeration"|"not", "namespaces": xs:anyURI-Sequenz, "disallowed names": Liste mit QNames und/oder den Strings "defined" und "definedSiblings"	
process contents	String ("strict"|"lax"|"skip")	

Facet

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	Der Name des Facet, z.B. "minLength" oder "enumeration"
value	abhängig vom Facet	Der Wert des Facet
fixed	Boolean	
typed-value	Nur für das Enumeration Facet, Array(xs:anyAtomicType*)	Ein Array, das Enumeration-Werte, von denen jeder im Allgemeinen eine Sequenz atomarer Werte sein kann, enthält. (Anmerkung: Die Eigenschaft "value" ist unabhängig vom tatsächlichen Typ für das Enumeration Facet eine String-Sequenz)

Identity Constraint

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Identity-Constraint Definition"
name	String	Lokaler Name des Constraint
target namespace	String	Namespace URI des Constraint
identity-constraint category	String ("key"|"unique"|"keyRef")	
selector	XPath-Eigenschaftsdatensatz	

© 2019-2025 Altova GmbH

fields	Sequenz von XPath-Eigenschaftsdatensätzen	
referenced key	(nur für keyRef): Identity Constraint	Der entsprechende Key Constraint

Model Group

Property name	Eigenschaftstyp	Eigenschaftswert
kind	String	"Model Group"
compositor	String ("sequence"|"choice"|"all")	
particles	Partikel-Sequenz	

Model Group Definition

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Model Group Definition"
name	String	Lokaler Name der Model Group
target namespace	String	Namespace URI der Model Group
model group	Model Group	

Notation

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Notation Declaration"
name	String	Lokaler Name der Notation
target namespace	String	Namespace URI der Notation
system identifier	anyURI	
public identifier	String	

Particle

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Particle"
min occurs	Integer	
max occurs	Integer oder String("unbounded")	
term	Element Declaration, Element Wildcard oder ModelGroup	

Simple Type

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Simple Type Definition"

name	String	Lokaler Name des Typs (leer, wenn anonym)
target namespace	String	Namespace URI des Typs (leer, wenn anonym)
final	String-Sequenz("restriction"|"extension"|"list"|"union")	
context	enthaltende Komponente	
base type definition	Simple Type	
facets	Facet-Sequenz	
fundamental facets	Leere Sequenz (nicht implementiert)	
variety	String ("atomic"|"list"|"union")	
primitive type definition	Simple Type	
item type definition	(nur für Listentypen) Simple Type	
member type definitions	(nur für Union-Typen) Simple Type-Sequenz	

Type Alternative

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Type Alternative"
test	XPath-Eigenschaftsdatensatz	
type definition	Simple Type oder Complex Type	

XPath Property Record

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
namespace bindings	Sequenz von Funktionen mit Eigenschaften ("prefix": string, "namespace": anyURI)	
default namespace	anyURI	
base URI	anyURI	Die statische Basis-UI des XPath-Ausdrucks
expression	String	Der XPath-Ausdruck als String

Type

altova:type(Node *als item?***) als (function(xs:string) als item()*)? XP3.1 XQ3.1**

Die Funktion **altova:type** übermittelt einen Element- oder Attribut-Node eines XML-Dokuments und gibt die Typinformationen des Node aus dem PSVI zurück.

© 2019-2025 Altova GmbH

Anmerkung: Das XML-Dokument muss eine Schema-Deklaration haben, damit das Schema referenziert werden kann.

Beispiele

- `for $element in //Email`
  `let $type := altova:type($element)`
  `return $type`
  gibt eine Funktion zurück, die die Typinformationen des Node `Email` enthält.

- `for $element in //Email`
  `let $type := altova:type($element)`
  `return $type ("kind")`
  ermittelt anhand der Typ-Komponente des Node (Simple Type oder Complex Type) den Wert der Eigenschaft **`kind`** der Komponente.

Der Parameter **`"_props"`** gibt die Eigenschaften der ausgewählten Komponente zurück, z.B:

- `for $element in //Email`
  `let $type := altova:type($element)`
  `return ($type ("kind"), $type ("_props"))`
  nimmt die Typkomponente des Node `Email` (Simple Type oder Complex Type) und gibt (i) den Wert der Eigenschaft **`kind`** der Komponente zurück und anschließend (ii) die Eigenschaften dieser Komponente.

Komponenten und ihre Eigenschaften

Assertion

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Assertion"
test	XPath-Eigenschaftsdatensatz	

Attribute Declaration

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Attribute Declaration"
name	String	Lokaler Name des Attributs
target namespace	String	Namespace URI des Attributs
type definition	SimpleType oder ComplexType	
scope	Eine Funktion mit Eigenschaften ("class":"Scope", "variety": "global" oder "local", "parent": der enthaltende ComplexTyp bzw. die enthaltende Attributgruppe)	
value constraint	Falls vorhanden, eine Funktion mit Eigenschaften ("class": "Value	

	Constraint", "variety": "fixed" oder "default", "value": atomarer Wert, "lexical form": String. Beachten Sie, dass die Eigenschaft "value" für Namespace-sensitive Typen nicht zur Verfügung steht.	
inheritable	Boolean	

- Attribute Group Declaration

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Attribute Group Definition"
name	String	Lokaler Name der Attributgruppe
target namespace	String	Namespace URI der Attributgruppe
attribute uses	Sequenz von (Attribute Use)	
attribute wildcard	Optionale Attribut-Wildcard	

- Attribute Use

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Attribute Use"
required	Boolean	true, wenn das Attribut obligatorisch ist, false, wenn es optional ist.
value constraint	Siehe Attribute Declaration	
inheritable	Boolean	

- Attribute Wildcard

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	string	"Wildcard"
namespace constraint	Funktion mit Eigenschaften ("class": "Namespace Constraint", "variety": "any"|"enumeration"|"not", "namespaces": Sequenz von xs:anyURI, "disallowed names": Liste mit QNames und/oder den Strings "defined" und "definedSiblings"	
process contents	String ("strict"|"lax"|"skip")	

- Complex Type

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert

© 2019-2025 Altova GmbH

kind	String	"Complex Type"
name	String	Lokaler Name des Typs (leer, wenn anonym)
target namespace	String	Namespace URI des Typs (leer, wenn anonym)
base type definition	Complex Type Definition	
final	String-Sequenz ("restriction"|"extension")	
context	Leere Sequenz (nicht implementiert)	
derivation method	String ("restriction"|"extension")	
abstract	Boolean	
attribute uses	Attribute Use-Sequenz	
attribute wildcard	Optionale Attribute Wildcard	
content type	Funktion mit Eigenschaften: ("class":"Content Type", "variety":string ("element-only"|"empty"|"mixed"|"simple"), particle: optionales Partikel, "open content": Funktion mit Eigenschaften ("class":"Open Content", "mode": string ("interleave"|"suffix"), "wildcard": Wildcard), "simple type definition": Simple Type)	
prohibited substitutions	String-Sequenz ("restriction"|"extension")	
assertions	Assertion-Sequenz	

Element Declaration

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Complex Type"
name	String	Lokaler Name des Typs (leer, wenn anonym)
target namespace	String	Namespace URI des Typs (leer, wenn anonym)
type definition	Simple Type oder Complex Type	
type table	Funktion mit Eigenschaften ("class":"Type Table", "alternatives": Type Alternative-Sequenz, "default type definition": Simple Type oder Complex Type)	
scope	Funktion mit Eigenschaften ("class":"Scope", "variety":	

	("global"|"local"), "parent": optionaler Complex Type)	
value constraint	siehe Attribute Declaration	
nillable	Boolean	
identity-constraint definitions	Identity Constraint-Sequenz	
substitution group affiliations	Element Declaration-Sequenz	
substitution group exclusions	String-Sequenz ("restriction"|"extension")	
disallowed substitutions	String-Sequenz ("restriction"|"extension"|"substitution")	
abstract	Boolean	

Element Wildcard

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Wildcard"
namespace constraint	Funktion mit Eigenschaften ("class": "Namespace Constraint", "variety": "any"|"enumeration"|"not", "namespaces": xs:anyURI-Sequenz, "disallowed names": Liste mit QNames und/oder den Strings "defined" und "definedSiblings"	
process contents	String ("strict"|"lax"|"skip")	

Facet

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	Der Name des Facet, z.B. "minLength" oder "enumeration"
value	abhängig vom Facet	Der Wert des Facet
fixed	Boolean	
typed-value	Nur für das Enumeration Facet, Array(xs:anyAtomicType*)	Ein Array, das Enumeration-Werte, von denen jeder im Allgemeinen eine Sequenz atomarer Werte sein kann, enthält. (Anmerkung: Die Eigenschaft "value" ist unabhängig vom tatsächlichen Typ für das Enumeration Facet eine String-Sequenz)

Identity Constraint

© 2019-2025 Altova GmbH

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Identity-Constraint Definition"
name	String	Lokaler Name des Constraint
target namespace	String	Namespace URI des Constraint
identity-constraint category	String ("key"|"unique"|"keyRef")	
selector	XPath-Eigenschaftsdatensatz	
fields	Sequenz von XPath-Eigenschaftsdatensätzen	
referenced key	(nur für keyRef): Identity Constraint	Der entsprechende Key Constraint

Model Group

Property name	Eigenschaftstyp	Eigenschaftswert
kind	String	"Model Group"
compositor	String ("sequence"|"choice"|"all")	
particles	Partikel-Sequenz	

Model Group Definition

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Model Group Definition"
name	String	Lokaler Name der Model Group
target namespace	String	Namespace URI der Model Group
model group	Model Group	

Notation

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Notation Declaration"
name	String	Lokaler Name der Notation
target namespace	String	Namespace URI der Notation
system identifier	anyURI	
public identifier	String	

Particle

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Particle"
min occurs	Integer	

max occurs	Integer oder String("unbounded")	
term	Element Declaration, Element Wildcard oder ModelGroup	

Simple Type

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Simple Type Definition"
name	String	Lokaler Name des Typs (leer, wenn anonym)
target namespace	String	Namespace URI des Typs (leer, wenn anonym)
final	String-Sequenz("restriction"|"extension"|"list"|"union")	
context	enthaltende Komponente	
base type definition	Simple Type	
facets	Facet-Sequenz	
fundamental facets	Leere Sequenz (nicht implementiert)	
variety	String ("atomic"|"list"|"union")	
primitive type definition	Simple Type	
item type definition	(nur für Listentypen) Simple Type	
member type definitions	(nur für Union-Typen) Simple Type-Sequenz	

Type Alternative

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
kind	String	"Type Alternative"
test	XPath-Eigenschaftsdatensatz	
type definition	Simple Type oder Complex Type	

XPath Property Record

Eigenschaftsname	Eigenschaftstyp	Eigenschaftswert
namespace bindings	Sequenz von Funktionen mit Eigenschaften ("prefix": string, "namespace": anyURI)	
default namespace	anyURI	
base URI	anyURI	Die statische Basis-UI des XPath-Ausdrucks

© 2019-2025 Altova GmbH

expression	String	Der XPath-Ausdruck als String

12.2.1.7 XPath/XQuery-Funktionen: Sequenz

Die Sequenz-Erweiterungsfunktionen von Altova können in XPath- und XQuery-Ausdrücken verwendet werden und stellen zusätzliche Funktionen für die Verarbeitung von Daten zur Verfügung. Die Funktionen in diesem Abschnitt können mit dem **XPath 3.0-** und **XQuery 3.0**-Prozessor von Altova verwendet werden. Sie stehen im Zusammenhang mit XPath/XQuery zur Verfügung.

Anmerkung zur Benennung von Funktionen und zur Anwendbarkeit der Sprache

Altova-Erweiterungsfunktionen können in XPath/XQuery-Ausdrücken verwendet werden. Dadurch stehen neben den Funktionen in der Standardbibliothek der XPath-, XQuery- und XSLT-Funktionen zusätzliche Funktionen zur Verfügung. Die Altova-Erweiterungsfunktionen befinden sich im **Altova-Erweiterungsfunktions-Namespace, `http://www.altova.com/xslt-extensions`** und sind in diesem Abschnitt mit dem Präfix **`altova:`**, das als an diesen Namespace gebunden angenommen wird, gekennzeichnet. Beachten Sie, dass manche Funktionen in zukünftigen Versionen Ihres Produkts eventuell nicht mehr unterstützt werden oder dass sich das Verhalten einzelner Funktionen ändern kann. Um zu sehen, welche Altova-Erweiterungsfunktionen unterstützt werden, lesen Sie bitte die Dokumentation zur jeweiligen Release.

XPath-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	XP1 XP2 XP3.1
XSLT-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	XSLT1 XSLT2 XSLT3
XQuery-Funktionen (in XQuery-Ausdrücken in XQuery verwendet):	XQ1 XQ3.1

▼ attributes [altova:]

`altova:attributes(AttributeName` *als xs:string*`)` als **`attribute()*`** XP3.1 XQ3.1
Gibt alle Attribute zurück, die einen lokalen Namen haben, der mit dem im Input-Argument `AttributeName` angegebenen Namen identisch ist. Die Groß- und Kleinschreibung wird bei der Suche, die entlang der `attribute::` Achse durchgeführt wird, beachtet. Das bedeutet, dass der Kontext-Node der Parent-Element-Node sein muss.

⊟ *Beispiele*

- **`altova:attributes`**`("MyAttribute")` gibt `MyAttribute()*` zurück

`altova:attributes(AttributeName` *als xs:string*`, SearchOptions` *als xs:string*`)` als **`attribute()*`** XP3.1 XQ3.1
Gibt alle Attribute zurück, die einen lokalen Namen haben, der mit dem im Input-Argument `AttributeName` angegebenen Namen identisch ist. Die Groß- und Kleinschreibung wird bei der Suche, die entlang der `attribute::` Achse durchgeführt wird, beachtet. Der Kontext-Node muss der Parent-Element-Node sein. Das zweite Argument ist ein String, der Options-Flags enthält. Zur Verfügung stehen die folgenden Flags:
`r` = wechselt zu einer Suche mittels Regular Expression; bei `AttributeName` muss es sich in diesem Fall

um einen Regular Expression-Suchstring handeln;
f = Wenn diese Option definiert ist, liefert `AttributeName` eine vollständige Übereinstimmung; andernfalls muss `AttributeName` nur teilweise mit einem Attributnamen übereinstimmen, damit dieses Attribut zurückgegeben wird. Wenn **f** z.B. nicht definiert ist, gibt `MyAtt MyAttribute` zurück;
i = wechselt zu einer Suche ohne Berücksichtigung der Groß- und Kleinschreibung;
p = inkludiert das Namespace-Präfix in die Suche; `AttributeName` sollte in diesem Fall das Namespace-Präfix enthalten, z.B.: `altova:MyAttribute`.
Die Flags können in jeder Reihenfolge angegeben werden. Ungültige Flags erzeugen eine Fehlermeldung. Sie können ein oder mehrere Flags weglassen. Es ist auch der leere String zulässig. Das Resultat ist dasselbe wie bei Verwendung der Funktion mit nur einem Argument (*siehe vorherige Signatur*). Unzulässig ist jedoch die Verwendung einer leeren Sequenz als zweites Argument.

Beispiele

- **altova:attributes**("MyAttribute", "rfip") gibt `MyAttribute()*` zurück
- **altova:attributes**("MyAttribute", "pri") gibt `MyAttribute()*` zurück
- **altova:attributes**("MyAtt", "rip") gibt `MyAttribute()*` zurück
- **altova:attributes**("MyAttributes", "rfip") gibt keine Übereinstimmung zurück
- **altova:attributes**("MyAttribute", "") gibt `MyAttribute()*` zurück
- **altova:attributes**("MyAttribute", "Rip") gibt einen Fehler zurück, dass das Flag unbekannt ist.
- **altova:attributes**("MyAttribute",) gibt den Fehler zurück, dass das zweite Argument fehlt.

elements [altova:]

altova:elements(ElementName *als xs:string***)** als **element()*** **XP3.1** **XQ3.1**
Gibt alle Elemente zurück, die einen lokalen Namen haben, der mit dem im Input-Argument `ElementName` angegebenen Namen identisch ist. Die Groß- und Kleinschreibung wird bei der Suche, die entlang der `child::` Achse durchgeführt wird, beachtet. Der Kontext-Node muss der Parent-Node des gesuchten Elements sein.

Beispiele

- **altova:elements**("MyElement") gibt `MyElement()*` zurück

altova:elements(ElementName *als xs:string*, **SearchOptions** *als xs:string***)** als **element()*** **XP3.1** **XQ3.1**
Gibt alle Elemente zurück, die einen lokalen Namen haben, der mit dem im Input-Argument `ElementName` angegebenen Namen identisch ist. Die Groß- und Kleinschreibung wird bei der Suche, die entlang der `child::` Achse durchgeführt wird, beachtet. Der Kontext-Node muss der Parent-Node des gesuchten Elements sein. Das zweite Argument ist ein String, der Options-Flags enthält. Zur Verfügung stehen die folgenden Flags:
r = wechselt zu einer Suche mittels Regular Expression; bei `ElementName` muss es sich in diesem Fall um einen Regular Expression-Suchstring handeln;
f = Wenn diese Option definiert ist, liefert `ElementName` eine vollständige Übereinstimmung; andernfalls muss `ElementName` nur teilweise mit einem Elementnamen übereinstimmen, damit dieses Element zurückgegeben wird. Wenn **f** z.B. nicht definiert ist, gibt `MyElem MyElement` zurück;
i = wechselt zu einer Suche ohne Berücksichtigung der Groß- und Kleinschreibung;
p = inkludiert das Namespace-Präfix in die Suche; `ElementName` sollte in diesem Fall das Namespace-Präfix enthalten, z.B.: `altova:MyElement`.
Die Flags können in jeder Reihenfolge angegeben werden. Ungültige Flags erzeugen eine Fehlermeldung. Sie können ein oder mehrere Flags weglassen. Es ist auch der leere String zulässig. Das Resultat ist dasselbe wie bei Verwendung der Funktion mit nur einem Argument (siehe vorherige Signatur). Unzulässig

© 2019-2025 Altova GmbH

ist jedoch die Verwendung einer leeren Sequenz.

Beispiele

- **altova:elements**("MyElement", "rip") gibt MyElement()* zurück
- **altova:elements**("MyElement", "pri") gibt MyElement()* zurück
- **altova:elements**("MyElement", "") gibt MyElement()* zurück
- **altova:elements**("MyElem", "rip") gibt MyElement()* zurück
- **altova:elements**("MyElements", "rfip") gibt keine Übereinstimmung zurück
- **altova:elements**("MyElement", "Rip") gibt einen Fehler zurück, dass das Flag unbekannt ist.
- **altova:elements**("MyElement",) gibt den Fehler zurück, dass das zweite Argument fehlt.

find-first [altova:]

altova:find-first((*item()**), **(CheckFunction(item()** *als xs:boolean*)) als **item()? XP3.1 XQ3.1**

Diese Funktion verwendet zwei Argumente. Das erste Argument ist eine Sequenz von einem oder mehreren Elementen eines beliebigen Datentyps. Das zweite Argument, Condition, ist eine Referenz zu einer XPath-Funktion, die ein Argument erhält. (hat einen Stellenwert 1) und einen Booleschen Wert zurückgibt. Jedes Element von **Sequence** wird der Reihe nach der in Condition referenzierten Funktion bereitgestellt. (*Beachten Sie:* Die Funktion hat ein einziges Argument.) Das erste **Sequence** Element, bei dem das Resultat von **Condition true()** ist, wird als das Ergebnis von altova:find-first zurückgegeben. Anschließend wird die Iteration gestoppt.

Beispiele

- **altova:find-first**(5 to 10, function($a) {$a mod 2 = 0}) gibt xs:integer 6 zurück

Das Argument **Condition** referenziert die XPath 3.0 Inline-Funktion, **function()**, welche eine Inline-Funktion **$a** deklariert und diese anschließend definiert. Die einelnen Elemente im Argument **Sequence** von **altova:find-first** werden der Reihe nach an **$a** als sein Input-Wert übergeben. Der Input-Wert wird an der Bedingung in der Funktionsdefinition ($a mod 2 = 0) überprüft. Der erste Input-Wert, der diese Bedingung erfüllt, wird als das Ergebnis von **altova:find-first** (in diese Fall **6**) zurückgegeben.

- **altova:find-first**((1 to 10), (function($a) {$a+3=7})) gibt xs:integer 4 zurück

Weitere Beispiele

Wenn die Datei **C:\Temp\Customers.xml** vorhanden ist:

- **altova:find-first**(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1)) gibt C:\Temp\Customers.xml zurück

Wenn die Datei **C:\Temp\Customers.xml** nicht vorhanden ist und **http://www.altova.com/index.html** vorhanden ist:

- **altova:find-first**(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1)) gibt http://www.altova.com/index.html zurück

Wenn weder die Datei **C:\Temp\Customers.xml** noch **http://www.altova.com/index.html** vorhanden ist:

- **altova:find-first**(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"),

`(doc-available#1) )` gibt kein Ergebnis zurück

Anmerkungen zu den obigen Beispielen

- Die XPath 3.0-Funktion, `doc-available`, erhält ein einziges Argument, das als URI verwendet wird. Sie gibt nur dann `true` zurück, wenn unter der angegebenen URI ein Dokument-Node gefunden wird. Das Dokument unter der angegebenen URI muss daher ein XML-Dokument sein.
- Die Funktion `doc-available` kann für **`Condition`**, das zweite Argument von `altova:find-first` verwendet werden, da sie nur ein Argument erhält (Stelligkeit=1), da sie ein Element `item()` als Input erhält (ein String, der als URI verwendet wird) und einen Booleschen Wert zurückgibt.
- Beachten Sie, dass `doc-available` nur referenziert und nicht direkt aufgerufen wird. Das angehängte Suffix `#1` gibt eine Funktion mit einer Stelligkeit 1 an. Als Ganzes bedeutet `doc-available#1`: *Verwende die Funktion doc-availabe(), welche die Stelligkeit=1 hat und übergib die einzelnen Elemente in der ersten Sequenz der Reihe nach als einziges Argument an die Funktion.* Als Ergebnis wird jeder der beiden Strings an **`doc-available()`** übergeben. Die Funktion verwendet den String als URI und überprüft, ob unter der URI ein Dokument-Node vorhanden ist. Wenn dies der Fall ist, wird **`doc-available()`** zu `true()` ausgewertet und der String wird als Ergebnis der Funktion **`altova:find-first`** zurückgegeben. *Beachten Sie zur Funktion doc-available(), dass relative Pfade relativ zu aktuellen Basis-URI aufgelöst werden. Die Basis-URI ist standardmäßig die URI des XML-Dokuments, von dem aus die Funktion geladen wird.*

find-first-combination [altova:]

`altova:find-first-combination((Seq-01` *`as item()*`**`), (Seq-02`** *`as item()*`**`),`**
`(Condition( Seq-01-Item, Seq-02-Item` *`as xs:boolean`***`)) as item()*`** **XP3.1** **XQ3.1**

Diese Funktion verwendet drei Argumente:

- Die ersten beiden Argumente, **`Seq-01`** und **`Seq-02`**, sind Sequenzen von einem oder mehreren Elementen eines beliebigen Datentyps.
- Das dritte Argument, **`Condition`**, ist eine Referenz auf eine XPath-Funktion, die zwei Argumente erhält (d.h. eine Stelligkeit **2** hat) und einen Booleschen Wert zurückgibt.

Die Elemente von **`Seq-01`** und **`Seq-02`** werden als die Argumente der Funktion **`Condition`** in geordneten Paaren übergeben (je ein Element aus jeder Sequenz bildet ein Paar). Die Paare sind folgendermaßen geordnet.

```
If   Seq-01 = X1, X2, X3 ... Xn
And  Seq-02 = Y1, Y2, Y3 ... Yn
Then (X1 Y1), (X1 Y2), (X1 Y3) ... (X1 Yn), (X2 Y1), (X2 Y2) ... (Xn Yn)
```

Das erste geordnete Paar, bei dem die Funktion **`Condition`** zu **`true()`** ausgewertet wird, wird als Ergebnis von **`altova:find-first-combination`** zurückgegeben. Beachten Sie: (i) Wenn die Funktion **`Condition`** durch die bereitgestellten Argumentpaare iteriert und nicht ein einziges Mal zu **`true()`** ausgewertet wird, so gibt **`altova:find-first-combination`** *Keine Ergebnisse* zurück; (ii) Das Ergebnis von **`altova:find-first-combination`** ist immer ein Elementpaar (eines beliebigen Datentyps) oder gar kein Element.

Beispiele

- `altova:find-first-combination(11 to 20, 21 to 30, function($a, $b) {$a+$b = 32})` gibt die Sequenz `xs:integers (11, 21)` zurück
- `altova:find-first-combination(11 to 20, 21 to 30, function($a, $b) {$a+$b = 33})` gibt die Sequenz `xs:integers (11, 22)` zurück
- `altova:find-first-combination(11 to 20, 21 to 30, function($a, $b) {$a+$b = 34})` gibt die Sequenz `xs:integers (11, 23)` zurück

find-first-pair [altova:]

`altova:find-first-pair((Seq-01` *`als item()*`*`), (Seq-02` *`als item()*`*`), (Condition( Seq-01-Item, Seq-02-Item` *`als xs:boolean`*`))` als **`item()*`** **XP3.1** **XQ3.1**
Diese Funktion erhält drei Argumente:

- Die ersten beiden Argumente, **`Seq-01`** und **`Seq-02`**, sind Sequenzen von einem oder mehreren Elementen eines beliebigen Datentyps.
- Das dritte Argument, **`Condition`**, ist eine Referenz auf eine XPath-Funktion, die zwei Argumente erhält (d.h. eine Stelligkeit **2** hat) und einen Booleschen Wert zurückgibt.

Die Elemente von **`Seq-01`** und **`Seq-02`** werden als die Argumente der Funktion **`Condition`** in geordneten Paaren übergeben. Die Paare sind folgendermaßen geordnet.

```
If   Seq-01 = X1, X2, X3 ... Xn
And  Seq-02 = Y1, Y2, Y3 ... Yn
Then (X1 Y1), (X2 Y2), (X3 Y3) ... (Xn Yn)
```

Das erste geordnete Paar, bei dem die Funktion **`Condition`** zu **`true()`** ausgewertet wird, wird als Ergebnis von **`altova:find-first-pair`** zurückgegeben. Beachten Sie: (i) Wenn die Funktion **`Condition`** durch die bereitgestellten Argumentpaare iteriert und nicht ein einziges Mal zu **`true()`** ausgewertet wird, so gibt **`altova:find-first-pair`** *Keine Ergebnisse* zurück; (ii) Das Ergebnis von **`altova:find-first-pair`** ist immer ein Elementpaar (eines beliebigen Datentyps) oder gar kein Element.

Beispiele

- `altova:find-first-pair(11 to 20, 21 to 30, function($a, $b) {$a+$b = 32})` gibt die Sequenz `xs:integers (11, 21)` zurück
- `altova:find-first-pair(11 to 20, 21 to 30, function($a, $b) {$a+$b = 33})` gibt *Keine Ergebnisse* zurück

Beachten Sie anhand der zwei Beispiel oben, dass die Paare folgendermaßen geordnet sind: `(11, 21) (12, 22) (13, 23)...(20, 30)`. Aus diesem Grund gibt das zweite Beispiel *Keine Ergebnisse* zurück (da keine geordnetes Paar die Summe `33` ergibt).

find-first-pair-pos [altova:]

`altova:find-first-pair-pos((Seq-01` *`as item()*`*`), (Seq-02` *`as item()*`*`), (Condition( Seq-01-Item, Seq-02-Item` *`as xs:boolean`*`)) as xs:integer` **XP3.1** **XQ3.1**
Diese Funktion erhält drei Argumente:

- Die ersten beiden Argumente, **`Seq-01`** und **`Seq-02`**, sind Sequenzen von einem oder mehreren Elementen eines beliebigen Datentyps.

- Das dritte Argument, **Condition**, ist eine Referenz auf eine XPath-Funktion, die zwei Argumente erhält (d.h. eine Stelligkeit 2 hat) und einen Booleschen Wert zurückgibt.

Die Elemente von **Seq-01** und **Seq-02** werden als die Argumente der Funktion **Condition** in geordneten Paaren übergeben. Die Paare sind folgendermaßen geordnet.

```
If   Seq-01 = X1, X2, X3 ... Xn
And  Seq-02 = Y1, Y2, Y3 ... Yn
Then (X1 Y1), (X2 Y2), (X3 Y3) ... (Xn Yn)
```

Als Ergebnis von **altova:find-first-pair-pos** wird die Indexposition des ersten geordneten Paars, bei dem die Funktion **Condition** zu **true()** ausgewertet wird, zurückgegeben. Beachten Sie: Wenn die Funktion **Condition** durch die bereitgestellten Argumentpaare iteriert und kein einziges Mal zu **true()** ausgewertet wird, so gibt **altova:find-first-pair-pos** *Keine Ergebnisse* zurück.

Beispiele

- **altova:find-first-pair-pos**(11 to 20, 21 to 30, function($a, $b) {$a+$b = 32}) gibt 1 zurück
- **altova:find-first-pair-pos**(11 to 20, 21 to 30, function($a, $b) {$a+$b = 33}) gibt *Keine Ergebnisse* zurück

Beachten Sie anhand der zwei Beispiel oben, dass die Paare folgendermaßen geordnet sind: (11, 21) (12, 22) (13, 23)...(20, 30). Im ersten Beispiel gibt die Funktion **Condition** bei Auswertung des ersten Paars **true()** zurück, daher wird dessen Indexposition in der Sequenz, 1, zurückgegeben. Das zweite Beispiel gibt *Keine Ergebnisse* zurück (da keine geordnetes Paar die Summe 33 ergibt).

find-first-pos [altova:]

altova:find-first-pos((*item()), (CheckFunction(item() *als xs:boolean*))** als **xs:integer? XP3.1 XQ3.1**

Diese Funktion verwendet zwei Argumente. Das erste Argument ist eine Sequenz von einem oder mehreren Elementen eines beliebigen Datentyps. Das zweite Argument, Condition, ist eine Referenz zu einer XPath-Funktion, die ein Argument erhält. (hat einen Stellenwert 1) und einen Booleschen Wert zurückgibt. Jedes Element von **Sequence** wird der Reihe nach der in Condition referenzierten Funktion bereitgestellt. (*Beachten Sie:* Die Funktion hat ein einziges Argument.) Das erste **Sequence** Element, bei dem das Resultat von **Condition true()** ist, wird als das Ergebnis von **altova:find-first-pos** zurückgegeben. Anschließend wird die Iteration gestoppt.

Beispiele

- **altova:find-first-pos**(5 to 10, function($a) {$a mod 2 = 0}) gibt xs:integer 2 zurück

Das Argument **Condition** referenziert die XPath 3.0 Inline-Funktion, **function()**, welche eine Inline-Funktion **$a** deklariert und diese anschließend definiert. Die einelnen Elemente im Argument **Sequence** von **altova:find-first-pos** werden der Reihe nach an **$a** als sein Input-Wert übergeben. Der Input-Wert wird an der Bedingung in der Funktionsdefinition ($a mod 2 = 0) überprüft. Die Indexposition in der Sequenz des ersten Input-Werts, die diese Bedingung erfüllt, wird als das Ergebnis von **altova:find-first-pos** zurückgegeben (in diesem Fall 2, da **6**, der erste Wert in der Sequenz, der die Bedingung erfüllt, sich in der Sequenz an der Indexposition 2 befindet).

© 2019-2025 Altova GmbH

Weitere Beispiele
Wenn die Datei **C:\Temp\Customers.xml** vorhanden ist:

- **altova:find-first-pos**(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1)) gibt 1 zurück

Wenn die Datei **C:\Temp\Customers.xml** nicht vorhanden ist und **http://www.altova.com/index.html** vorhanden ist:

- **altova:find-first-pos**(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1)) gibt 2 zurück

Wenn weder die Datei **C:\Temp\Customers.xml** noch **http://www.altova.com/index.html** vorhanden ist:

- **altova:find-first-pos**(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1)) gibt kein Ergebnis zurück

Anmerkungen zu den obigen Beispielen

- Die XPath 3.0-Funktion, doc-available, erhält ein einziges Argument, das als URI verwendet wird. Sie gibt nur dann true zurück, wenn unter der angegebenen URI ein Dokument-Node gefunden wird. (Das Dokument unter der angegebenen URI muss daher ein XML-Dokument sein.)
- Die Funktion doc-available kann für **Condition,** das zweite Argument von altova:find-first-pos verwendet werden, da sie nur ein Argument erhält (Stelligkeit=1), da sie ein Element item() als Input erhält (ein String, der als URI verwendet wird) und einen Booleschen Wert zurückgibt.
- Beachten Sie, dass doc-available nur referenziert und nicht direkt aufgerufen wird. Das angehängte Suffix #1 gibt eine Funktion mit einer Stelligkeit 1 an. Als Ganzes bedeutet doc-available#1: *Verwende die Funktion doc-availabe(), welche die Stelligkeit=1 hat und übergib die einzelnen Elemente in der ersten Sequenz der Reihe nach als einziges Argument an die Funktion.* Als Ergebnis wird jeder der beiden Strings an **doc-available()** übergeben. Die Funktion verwendet den String als URI und überprüft, ob unter der URI ein Dokument-Node vorhanden ist. Wenn dies der Fall ist, wird **doc-available()** zu true() ausgewertet und der String wird als Ergebnis der Funktion **altova:find-first** zurückgegeben. *Beachten Sie zur Funktion doc-available(), dass relative Pfade relativ zu aktuellen Basis-URI aufgelöst werden. Die Basis-URI ist standardmäßig die URI des XML-Dokuments, von dem aus die Funktion geladen wird.*

▼ for-each-attribute-pair [altova:]

altova:for-each-attribute-pair(Seq1 *als element()?*, **Seq2** *als element()?*, **Function** *als function()*) als **item()*** **XP3.1** **XQ3.1**
Die beiden ersten Argumente identifizieren zwei Elemente, anhand deren Attribute Attributpaare gebildet werden, wobei das eine Attribut eines Paars aus dem ersten Element und das andere aus dem zweiten Element stammt. Die Attributpaare werden auf Basis ihres übereinstimmenden Namens ausgewählt und alphabetisch (nach ihren Namen) zu einer Gruppe geordnet. Falls es zu einem Attribut im anderen Element keine Entsprechung gibt, ist das Paar "nicht verbunden", d.h. es besteht nur aus einem Mitglied. Das Funktionselement (das dritte Argument Function) wird auf die einzelnen Paare (verbundene und nicht

verbundene) in der Sequenz der Paare separat angewendet, wodurch als Ausgabe eine Sequenz von Einträgen erzeugt wird.

Beispiele

- **altova:for-each-attribute-pair**(/Example/Test-A, /Example/Test-B, function($a, $b) {$a+b}) gibt zurück ...

  (2, 4, 6) wenn
  <Test-A att1="1" att2="2" att3="3" />
  <Test-B att1="1" att2="2" att3="3" />

  (2, 4, 6) wenn
  <Test-A att2="2" att1="1" att3="3" />
  <Test-B att3="3" att2="2" att1="1" />

  (2, 6) wenn
  <Test-A att4="4" att1="1" att3="3" />
  <Test-B att3="3" att2="2" att1="1" />

  Anmerkung: *Das Ergebnis* (2, 6) *wird mit Hilfe der folgenden Aktion ermittelt:* (1+1, ()+2, 3+3, 4+()). *Wenn einer der Operanden eine leere Sequenz ist, wie dies bei Eintrag 2 und 4 der Fall ist, so ist das Ergebnis der Addition eine leere Sequenz.*

- **altova:for-each-attribute-pair**(/Example/Test-A, /Example/Test-B, concat#2) gibt zurück ...

  (11, 22, 33) wenn
  <Test-A att1="1" att2="2" att3="3" />
  <Test-B att1="1" att2="2" att3="3" />

  (11, 2, 33, 4) wenn
  <Test-A att4="4" att1="1" att3="3" />
  <Test-B att3="3" att2="2" att1="1" />

for-each-combination [altova:]

altova:for-each-combination(FirstSequence *als item()**, **SecondSequence** *als item()**, **Function($i,$j){$i || $j})** als item()* XP3.1 XQ3.1

Die Elemente der zwei Sequenzen in den ersten beiden Argumenten werden miteinander kombiniert, so dass jedes Element in der ersten Sequenz der Reihe nach einmal mit jedem Element in der zweiten Sequenz kombiniert wird. Die als drittes Argument angegebene Funktion wird auf die einzelnen Kombinationen in der erzeugten Sequenz angewendet, wodurch als Ausgabe eine Sequenz von Elementen erzeugt wird (*siehe Beispiel*).

Beispiele

- **altova:for-each-combination**(('a', 'b', 'c'), ('1', '2', '3'), function($i, $j) {$i || $j}) gibt ('a1', 'a2', 'a3', 'b1', 'b2', 'b3', 'c1', 'c2', 'c3') zurück

for-each-matching-attribute-pair [altova:]

altova:for-each-matching-attribute-pair(Seq1 *als element()?*, **Seq2** *als element()?*,

© 2019-2025 Altova GmbH

Function *als function()*) als **item()*** XP3.1 XQ3.1

Die beiden ersten Argumente identifizieren zwei Elemente, anhand deren Attribute Attributpaare gebildet werden, wobei das eine Attribut eines Paars aus dem ersten Element und das andere aus dem zweiten Element stammt. Die Attributpaare werden auf Basis ihres übereinstimmenden Namens ausgewählt und alphabetisch (nach ihren Namen) zu einer Gruppe geordnet. Falls es zu einem Attribut im anderen Element keine Entsprechung gibt, wird kein Paar gebildet. Das Funktionselement (das dritte Argument `Function`) wird auf die einzelnen Paare in der Sequenz der Paare separat angewendet, wodurch als Ausgabe eine Sequenz von Einträgen erzeugt wird.

Beispiele

- **altova:for-each-matching-attribute-pair**(/Example/Test-A, /Example/Test-B, function($a, $b){$a+b}) gibt zurück ...

  (2, 4, 6) wenn
  ```
  <Test-A att1="1" att2="2" att3="3" />
  <Test-B att1="1" att2="2" att3="3" />
  ```

  (2, 4, 6) wenn
  ```
  <Test-A att2="2" att1="1" att3="3" />
  <Test-B att3="3" att2="2" att1="1" />
  ```

  (2, 6) wenn
  ```
  <Test-A att4="4" att1="1" att3="3" />
  <Test-B att3="3" att2="2" att3="1" />
  ```

- **altova:for-each-matching-attribute-pair**(/Example/Test-A, /Example/Test-B, concat#2) gibt zurück ....

  (11, 22, 33) wenn
  ```
  <Test-A att1="1" att2="2" att3="3" />
  <Test-B att1="1" att2="2" att3="3" />
  ```

  (11, 33) wenn
  ```
  <Test-A att4="4" att1="1" att3="3" />
  <Test-B att3="3" att2="2" att1="1" />
  ```

substitute-empty [altova:]

altova:substitute-empty(FirstSequence *als item()**, **SecondSequence** *als item()*) als **item()*** XP3.1 XQ3.1

Wenn `FirstSequence` leer ist, wird `SecondSequence` zurückgegeben. Wenn `FirstSequence` nicht leer ist, wird `FirstSequence` zurückgegeben.

Beispiele

- **altova:substitute-empty**((1,2,3), (4,5,6)) gibt (1,2,3) zurück
- **altova:substitute-empty**((), (4,5,6)) gibt (4,5,6) zurück

12.2.1.8 XPath/XQuery-Funktionen: String

Die folgenden XPath/XQuey-Erweiterungsfunktionen für Strings werden in der aktuellen Version Ihres Altova-Produkts unterstützt und bieten Zusatzfunktionalitäten für die Verarbeitung von Daten. Die Funktionen in diesem Abschnitt können mit dem **XPath 3.0-** und **XQuery 3.0**-Prozessor von Altova verwendet werden. Sie stehen im Zusammenhang mit XPath/XQuery zur Verfügung.

Anmerkung zur Benennung von Funktionen und zur Anwendbarkeit der Sprache

Altova-Erweiterungsfunktionen können in XPath/XQuery-Ausdrücken verwendet werden. Dadurch stehen neben den Funktionen in der Standardbibliothek der XPath-, XQuery- und XSLT-Funktionen zusätzliche Funktionen zur Verfügung. Die Altova-Erweiterungsfunktionen befinden sich im **Altova-Erweiterungsfunktions-Namespace, `http://www.altova.com/xslt-extensions`** und sind in diesem Abschnitt mit dem Präfix **`altova:`**, das als an diesen Namespace gebunden angenommen wird, gekennzeichnet. Beachten Sie, dass manche Funktionen in zukünftigen Versionen Ihres Produkts eventuell nicht mehr unterstützt werden oder dass sich das Verhalten einzelner Funktionen ändern kann. Um zu sehen, welche Altova-Erweiterungsfunktionen unterstützt werden, lesen Sie bitte die Dokumentation zur jeweiligen Release.

XPath-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	XP1 XP2 XP3.1
XSLT-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	XSLT1 XSLT2 XSLT3
XQuery-Funktionen (in XQuery-Ausdrücken in XQuery verwendet):	XQ1 XQ3.1

▼ camel-case [altova:]

`altova:camel-case(InputString` *als xs:string*`)` als **`xs:string`** XP3.1 XQ3.1
Gibt den Input-String **`InputString`** in CamelCase zurück. Der String wird mit Hilfe der Regular Expression **`'\s'`** (welches ein Kürzel für das Leerzeichen ist) analysiert. Das erste Zeichen nach einem Leerzeichen oder einer Sequenz aufeinander folgender Leerzeichen, das kein Leerzeichen ist, wird mit einem Großbuchstaben geschrieben. Das erste Zeichen im Ausgabestring wird mit einem Großbuchstaben geschrieben.

Beispiele

- **`altova:camel-case`**`("max")` gibt `Max` zurück
- **`altova:camel-case`**`("max max")` gibt `Max Max` zurück
- **`altova:camel-case`**`("file01.xml")` gibt `File01.xml` zurück
- **`altova:camel-case`**`("file01.xml file02.xml")` gibt `File01.xml File02.xml` zurück
- **`altova:camel-case`**`("file01.xml  file02.xml")` gibt `File01.xml  File02.xml` zurück
- **`altova:camel-case`**`("file01.xml -file02.xml")` gibt `File01.xml -file02.xml` zurück

`altova:camel-case(InputString` *als xs:string*`,` **`SplitChars`** *als xs:string*`,` **`IsRegex`** *als xs:boolean*`)` als **`xs:string`** XP3.1 XQ3.1
Konvertiert den Input-String **`InputString`** in CamelCase, indem anhand von **`SplitChars`** festgelegt wird, welche(s) Zeichen die nächste Konvertierung in Großbuchstaben auslöst. **`SplitChars`** wird als Regular Expression verwendet, wenn **`IsRegex = true()`** oder als einfache Zeichen, wenn **`IsRegex = false()`**. Das erste Zeichen im Ausgabestring wird mit einem Großbuchstaben geschrieben.

Beispiele

- **`altova:camel-case`**`("setname getname", "set|get", true())` gibt `setName getName` zurück

© 2019-2025 Altova GmbH

- **altova:camel-case**("altova\documents\testcases", "\", false()) gibt Altova\Documents\Testcases zurück

▼ char [altova:]

altova:char(Position *as xs:integer***)** als **xs:string** **XP3.1** **XQ3.1**
Gibt einen String zurück, der das Zeichen an der durch das Argument Position definierten Position enthält. Dieses Zeichen wird durch Konvertierung des Werts des Kontextelements in xs:string ermittelt. Der Ergebnisstring ist leer, wenn an dem durch das Position Argument gelieferten Index kein Zeichen vorhanden ist.

Beispiele

Wenn das Kontextelement **1234ABCD** lautet:

- **altova:char**(2) gibt 2 zurück
- **altova:char**(5) gibt A zurück
- **altova:char**(9) gibt den leeren String zurück.
- **altova:char**(-2) gibt den leeren String zurück.

altova:char(InputString *als xs:string*, **Position** *als xs:integer***)** als **xs:string** **XP3.1** **XQ3.1**
Gibt einen String zurück, der das Zeichen enthält, das sich in dem als InputString Argument gelieferten String an der durch das Argument Position definierten Position befindet. Der Ergebnisstring ist leer, wenn an dem durch das Position Argument gelieferten Index kein Zeichen vorhanden ist.

Beispiele

- **altova:char**("2014-01-15", 5) gibt - zurück
- **altova:char**("USA", 1) gibt U zurück
- **altova:char**("USA", 1) gibt den leeren String zurück.
- **altova:char**("USA", -2) gibt den leeren String zurück.

▼ create-hash-from-string[altova:]

altova:create-hash-from-string(InputString *als xs:string***)** als **xs:string** **XP2** **XQ1** **XP3.1** **XQ3.1**
altova:create-hash-from-string(InputString *als xs:string*, **HashAlgo** *als xs:string***)** als **xs:string** **XP2** **XQ1** **XP3.1** **XQ3.1**
Generiert anhand von InputString mit Hilfe des durch das Argument HashAlgo definierten Hash-Algorithmus einen Hash-String. Es können die folgenden Hash-Algorithmen definiert werden (in Groß- oder Kleinbuchstaben): **MD5**, **SHA-1**, **SHA-224**, **SHA-256**, **SHA-384**, **SHA-512**. Wenn das zweite Argument nicht definiert ist (*siehe erste Signatur oben*), wird der Hash-Algorithmus **SHA-256** verwendet.

Beispiele

- **altova:create-hash-from-string**('abc') gibt einen Hash-String zurück, der mit Hilfe des Hash-Algorithmus **SHA-256** generiert wurde.
- **altova:create-hash-from-string**('abc', 'md5') gibt einen Hash-String zurück, der mit Hilfe des Hash-Algorithmus **MD5** generiert wurde.
- **altova:create-hash-from-string**('abc', 'MD5') gibt einen Hash-String zurück, der mit Hilfe des Hash-Algorithmus **MD5** generiert wurde.

▼ first-chars [altova:]

altova:first-chars(X-Number *as xs:integer*) als **xs:string** **XP3.1** **XQ3.1**
Gibt einen String zurück, der die ersten x Zeichen (bezeichnet durch `X-Number`) des String enthält, der durch Konvertierung des Werts des Kontextelements in `xs:string` erzeugt wird.

Beispiele

Wenn das Kontextelement **1234ABCD** lautet:

- **altova:first-chars**(2) gibt 12 zurück
- **altova:first-chars**(5) gibt 1234A zurück
- **altova:first-chars**(9) gibt 1234ABCD zurück

altova:first-chars(InputString *als xs:string*, **X-Number** *als xs:integer*) als **xs:string** **XP3.1** **XQ3.1**
Gibt einen String zurück, der die ersten x Zeichen (bezeichnet durch `X-Number`) des String enthält, das als das Argument `InputString` angegeben ist.

Beispiele

- **altova:first-chars**("2014-01-15", 5) gibt 2014- zurück
- **altova:first-chars**("USA", 1) gibt U zurück

▼ format-string [altova:]

altova:format-string(InputString *als xs:string*, **FormatSequence** *als item()**) als **xs:string** **XP3.1** **XQ3.1**
Der Input String (erstes Argument) enthält Positionsparameter (`%1`, `%2`, usw.). Jeder Parameter wird durch das String-Element ersetzt, das sich in der (als zweites Argument bereitgestellten) Formatsequenz an der entsprechenden Position befindet. Daher ersetzt des erste Element in der Formatsequenz den Positionsparameter `%1`, das zweite den Positionsparameter `%2`, usw. Die Funktion gibt diesen formatierten String zurück, der die Ersetzungen enthält. Wenn für einen Positionsparameter kein String existiert, wird der Positionsparameter selbst zurückgegeben. Dies kommt vor, wenn der Index eines Positionsparameters größer als die Anzahl der Elemente in der Formatsequenz ist.

Beispiele

- **altova:format-string**('Hello %1, %2, %3', ('Jane','John','Joe')) gibt "Hello Jane, John, Joe" zurück.
- **altova:format-string**('Hello %1, %2, %3', ('Jane','John','Joe', 'Tom')) gibt "Hello Jane, John, Joe" zurück.
- **altova:format-string**('Hello %1, %2, %4', ('Jane','John','Joe', 'Tom')) gibt "Hello Jane, John, Tom" zurück.
- **altova:format-string**('Hello %1, %2, %4', ('Jane','John','Joe')) gibt "Hello Jane, John, %4" zurück.

▼ last-chars [altova:]

altova:last-chars(X-Number *als xs:integer*) als **xs:string** **XP3.1** **XQ3.1**
Gibt einen String zurück, der die letzten x Zeichen (bezeichnet durch `X-Number`) des String enthält, der durch Konvertierung des Werts des Kontextelements in `xs:string` erzeugt wird.

Beispiele

© 2019-2025 Altova GmbH

Wenn das Kontextelement **1234ABCD** lautet:

- **altova:last-chars**(2) gibt CD zurück
- **altova:last-chars**(5) gibt 4ABCD zurück
- **altova:last-chars**(9) gibt 1234ABCD zurück

altova:last-chars(InputString *als xs:string*, **X-Number** *als xs:integer*) als **xs:string** **XP3.1** **XQ3.1**

Gibt einen String zurück, der die letzten x Zeichen (bezeichnet durch X-Number) des String enthält, das als das Argument InputString angegeben ist.

Beispiele

- **altova:last-chars**("2014-01-15", 5) gibt 01-15- zurück
- **altova:last-chars**("USA", 10) gibt USA zurück

pad-string-left [altova:]

altova:pad-string-left(StringToPad *als xs:string*, **Repeats** *als xs:integer*, **PadCharacter** *als xs:string*) als **xs:string** **XP3.1** **XQ3.1**

Das Argument PadCharacter ist ein einzelnes Zeichen. Es wird links vom String als Auffüllzeichen eingefügt, um die Anzahl der Zeichen in StringToPad zu erhöhen, damit diese Anzahl dem Ganzzahlwert des Arguments StringLength entspricht. Das Argument StringLength kann jeden beliebigen (positiven oder negativen) Ganzzahlwert haben, Auffüllzeichen werden aber nur verwendet, wenn der Wert von StringLength größer als die Anzahl der Zeichen in StringToPad ist. Wenn StringToPad mehr Zeichen als der Wert von StringLength hat, bleibt StringToPad unverändert.

Beispiele

- **altova:pad-string-left**('AP', 1, 'Z') gibt 'AP' zurück
- **altova:pad-string-left**('AP', 2, 'Z') gibt 'AP' zurück
- **altova:pad-string-left**('AP', 3, 'Z') gibt 'ZAP' zurück
- **altova:pad-string-left**('AP', 4, 'Z') gibt 'ZZAP' zurück
- **altova:pad-string-left**('AP', -3, 'Z') gibt 'AP' zurück
- **altova:pad-string-left**('AP', 3, 'YZ')gibt einen Fehler zurück, dass das Auffüllzeichen zu lang ist

pad-string-right [altova:]

altova:pad-string-right(StringToPad *als xs:string*, **Repeats** *als xs:integer*, **PadCharacter** *als xs:string*) als **xs:string** **XP3.1** **XQ3.1**

Das Argument PadCharacter ist ein einzelnes Zeichen. Es wird rechts vom String als Auffüllzeichen eingefügt, um die Anzahl der Zeichen in StringToPad zu erhöhen, damit diese Anzahl dem Ganzzahlwert des Arguments StringLength entspricht. Das Argument StringLength kann jeden beliebigen (positiven oder negativen) Ganzzahlwert haben, Auffüllzeichen werden aber nur verwendet, wenn der Wert von StringLength größer als die Anzahl der Zeichen in StringToPad ist. Wenn StringToPad mehr Zeichen als der Wert von StringLength hat, bleibt StringToPad unverändert.

Beispiele

- **altova:pad-string-right**('AP', 1, 'Z') gibt 'AP' zurück
- **altova:pad-string-right**('AP', 2, 'Z') gibt 'AP' zurück

- `altova:pad-string-right('AP', 3, 'Z')` gibt `'ZAP'` zurück
- `altova:pad-string-right('AP', 4, 'Z')` gibt `'ZZAP'` zurück
- `altova:pad-string-right('AP', -3, 'Z')` gibt `'AP'` zurück
- `altova:pad-string-right('AP', 3, 'YZ')` gibt einen Fehler zurück, dass das Auffüllzeichen zu lang ist

repeat-string [altova:]

altova:repeat-string(InputString *als xs:string*, **Repeats** *als xs:integer*) als **xs:string** **XP2** **XQ1** **XP3.1** **XQ3.1**

Generiert einen String, der sich zusammensetzt aus dem ersten `InputString`-Argument, das die Anzahl der `Repeats` wiederholt wird.

Beispiele

- `altova:repeat-string("Altova #", 3)` gibt `"Altova #Altova #Altova #"` zurück

substring-after-last [altova:]

altova:substring-after-last(MainString *als xs:string*, **CheckString** *als xs:string*) als **xs:string** **XP3.1** **XQ3.1**

Falls in `MainString` `CheckString` gefunden wird, so wird der Substring zurückgegeben, der in `MainString` nach `CheckString` steht. Falls `CheckString` in `MainString` nicht gefunden wird, so wird der leere String zurückgegeben. Wenn `CheckString` ein leerer String ist, so wird der gesamte `MainString` zurückgegeben. Falls `CheckString` mehrmals in `MainString`, vorkommt, so wird der Substring nach der letzten Instanz von `CheckString` zurückgegeben.

Beispiele

- `altova:substring-after-last('ABCDEFGH', 'B')` gibt `'CDEFGH'` zurück
- `altova:substring-after-last('ABCDEFGH', 'BC')` gibt `'DEFGH'`zurück
- `altova:substring-after-last('ABCDEFGH', 'BD')` gibt `''` zurück
- `altova:substring-after-last('ABCDEFGH', 'Z')` gibt `''` zurück
- `altova:substring-after-last('ABCDEFGH', '')` gibt `'ABCDEFGH'` zurück
- `altova:substring-after-last('ABCD-ABCD', 'B')` gibt `'CD'` zurück
- `altova:substring-after-last('ABCD-ABCD-ABCD', 'BCD')` gibt `''` zurück

substring-before-last [altova:]

altova:substring-before-last(MainString *as xs:string*, **CheckString** *as xs:string*) **as** **xs:string** **XP3.1** **XQ3.1**

Falls in `MainString` `CheckString` gefunden wird, so wird der Substring zurückgegeben, der in `MainString` vor `CheckString` steht. Falls `CheckString` in `MainString` nicht gefunden wird, so wird der leere String zurückgegeben. Wenn `CheckString` ein leerer String ist, so wird der gesamte `MainString` zurückgegeben. Falls `CheckString` mehrmals in `MainString`, vorkommt, so wird der Substring vor der letzten Instanz von `CheckString` zurückgegeben.

Beispiele

- `altova:substring-before-last('ABCDEFGH', 'B')` gibt `'A'` zurück
- `altova:substring-before-last('ABCDEFGH', 'BC')` gibt `'A'` zurück

© 2019-2025 Altova GmbH

- **altova:substring-before-last**('ABCDEFGH', 'BD') gibt '' zurück
- **altova:substring-before-last**('ABCDEFGH', 'Z') gibt '' zurück
- **altova:substring-before-last**('ABCDEFGH', '') gibt '' zurück
- **altova:substring-before-last**('ABCD-ABCD', 'B') gibt 'ABCD-A' zurück
- **altova:substring-before-last**('ABCD-ABCD-ABCD', 'ABCD') gibt 'ABCD-ABCD-' zurück

substring-pos [altova:]

altova:substring-pos(StringToCheck *als xs:string*, **StringToFind** *als xs:string*) als **xs:integer** **XP3.1** **XQ3.1**

Gibt die Zeichenposition der ersten Instanz von StringToFind im String StringToCheck zurück. Die Zeichenposition wird in Form einer Ganzzahl angegeben. Das erste Zeichen von StringToCheck hat die Position 1. Wenn StringToFind in StringToCheck nicht vorkommt, wird die Ganzzahl 0 zurückgegeben. Um den String auf eine zweite oder eine weiter hinten folgende Instanz von StringToCheck zu überprüfen, verwenden Sie die nächste Signatur dieser Funktion.

Beispiele

- **altova:substring-pos**('Altova', 'to') gibt 3 zurück
- **altova:substring-pos**('Altova', 'tov') gibt 3 zurück
- **altova:substring-pos**('Altova', 'tv') gibt 0 zurück
- **altova:substring-pos**('AltovaAltova', 'to') gibt 3 zurück

altova:substring-pos(StringToCheck *als xs:string*, **StringToFind** *als xs:string*, **Integer** *als xs:integer*) als **xs:integer** **XP3.1** **XQ3.1**

Gibt die Zeichenposition von StringToFind im String StringToCheck zurück. Die Suche nach StringToFind beginnt an der durch das Argument Integer angegebenen Zeichenposition; der Zeichen-Substring vor dieser Position wird nicht durchsucht. Die zurückgegebene Ganzzahl gibt jedoch die Position des gefundenen String innerhalb des *gesamten* String StringToCheck an. Diese Signatur dient dazu, die zweite oder eine weiter hinten folgende Position eines String zu finden, der mehrmals in StringToCheck vorkommt. Wenn StringToFind in StringToCheck nicht vorkommt, wird die Ganzzahl 0 zurückgegeben.

Beispiele

- **altova:substring-pos**('Altova', 'to', 1) gibt 3 zurück
- **altova:substring-pos**('Altova', 'to', 3) gibt 3 zurück
- **altova:substring-pos**('Altova', 'to', 4) gibt 0 zurück
- **altova:substring-pos**('Altova-Altova', 'to', 0) gibt 3 zurück
- **altova:substring-pos**('Altova-Altova', 'to', 4) gibt 10 zurück

trim-string [altova:]

altova:trim-string(InputString *als xs:string*) als **xs:string** **XP3.1** **XQ3.1**

Diese Funktion verwendet ein xs:string Argument, entfernt alle voran- und nachgestellten Leerzeichen und gibt einen "getrimmten" xs:string zurück.

Beispiele

- **altova:trim-string**(" Hello World ") gibt "Hello World" zurück
- **altova:trim-string**("Hello World ") gibt "Hello World" zurück
- **altova:trim-string**(" Hello World") gibt "Hello World" zurück

- **altova:trim-string**("Hello World") gibt "Hello World" zurück
- **altova:trim-string**("Hello World") gibt "Hello World" zurück

▼ trim-string-left [altova:]

altova:trim-string-left(InputString *als xs:string*) als **xs:string** XP3.1 XQ3.1
Diese Funktion verwendet ein xs:string Argument, entfernt alle vorangestellten Leerzeichen und gibt einen "links getrimmten" xs:string zurück.

Beispiele

- **altova:trim-string-left**(" Hello World ") gibt "Hello World " zurück
- **altova:trim-string-left**("Hello World ") gibt "Hello World " zurück
- **altova:trim-string-left**(" Hello World") gibt "Hello World" zurück
- **altova:trim-string-left**("Hello World") gibt "Hello World" zurück
- **altova:trim-string-left**("Hello World") gibt "Hello World" zurück

▼ trim-string-right [altova:]

altova:trim-string-right(InputString *als xs:string*) als **xs:string** XP3.1 XQ3.1
Diese Funktion verwendet ein xs:string Argument, entfernt alle nachgestellten Leerzeichen und gibt einen "rechts getrimmten" xs:string zurück.

Beispiele

- **altova:trim-string-right**(" Hello World ") gibt " Hello World" zurück
- **altova:trim-string-right**("Hello World ") gibt "Hello World" zurück
- **altova:trim-string-right**(" Hello World") gibt " Hello World" zurück
- **altova:trim-string-right**("Hello World") gibt "Hello World" zurück
- **altova:trim-string-right**("Hello World") gibt "Hello World" zurück

12.2.1.9 XPath/XQuery-Funktionen: Diverse Funktionen

Die folgenden XPath/XQuery-Funktionen für allgemeine Zwecke werden in der aktuellen Version von StyleVision unterstützt und können in (i) in einem XSLT-Kontext in XPath-Ausdrücken oder (ii) in einem XQuery-Dokument in XQuery-Ausdrücken verwendet werden.

Anmerkung zur Benennung von Funktionen und zur Anwendbarkeit der Sprache

Altova-Erweiterungsfunktionen können in XPath/XQuery-Ausdrücken verwendet werden. Dadurch stehen neben den Funktionen in der Standardbibliothek der XPath-, XQuery- und XSLT-Funktionen zusätzliche Funktionen zur Verfügung. Die Altova-Erweiterungsfunktionen befinden sich im **Altova-Erweiterungsfunktions-Namespace, http://www.altova.com/xslt-extensions** und sind in diesem Abschnitt mit dem Präfix **altova:**, das als an diesen Namespace gebunden angenommen wird, gekennzeichnet. Beachten Sie, dass manche Funktionen in zukünftigen Versionen Ihres Produkts eventuell nicht mehr unterstützt werden oder dass sich das Verhalten einzelner Funktionen ändern kann. Um zu sehen, welche Altova-Erweiterungsfunktionen unterstützt werden, lesen Sie bitte die Dokumentation zur jeweiligen Release.

© 2019-2025 Altova GmbH

XPath-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	XP1 XP2 XP3.1
XSLT-Funktionen (in XPath-Ausdrücken in XSLT verwendet):	XSLT1 XSLT2 XSLT3
XQuery-Funktionen (in XQuery-Ausdrücken in XQuery verwendet):	XQ1 XQ3.1

▼ decode-string [altova:]

altova:decode-string(Input *als xs:base64Binary*) **als xs:string** XP3.1 XQ3.1
altova:decode-string(Input *als xs:base64Binary*, **Encoding** *als xs:string*) **als xs:string** XP3.1 XQ3.1
Dekodiert den angegebenen base64Binary-Input anhand der definierten Kodierung zu einem String. Wenn keine Kodierung definiert ist, wird die UTF-8-Kodierung verwendet. Die folgenden Kodierungen werden unterstützt: `US-ASCII, ISO-8859-1, UTF-16, UTF-16LE, UTF-16BE, ISO-10646-UCS2, UTF-32, UTF-32LE, UTF-32BE, ISO-10646-UCS4`

▣ *Beispiele*

- **altova:decode-string**(`$XML1/MailData/Meta/b64B`) gibt den base64Binary-Input als UTF-8-kodierten String zurück.
- **altova:decode-string**(`$XML1/MailData/Meta/b64B, "UTF-8"`) gibt den base64Binary-Input als UTF-8-kodierten String zurück.
- **altova:decode-string**(`$XML1/MailData/Meta/b64B, "ISO-8859-1"`) gibt den base64Binary-Input als ISO-8859-1-kodierten String zurück.

▼ encode-string [altova:]

altova:encode-string(InputString *als xs:string*) **als xs:base64Binaryinteger** XP3.1 XQ3.1
altova:encode-string(InputString *als xs:string*, **Encoding** *als xs:string*) **als xs:base64Binaryinteger** XP3.1 XQ3.1
Kodiert den angegebenen String gemäß der definierten Kodierung, falls eine angegeben wird. Wenn keine Kodierung definiert ist, wird die UTF-8-Kodierung verwendet. Der kodierte String wird in base64Binary-Zeichen konvertiert und es wird der konvertierte base64Binary-Wert zurückgegeben. Anfangs wird die UTF-8-Kodierung unterstützt. Die Unterstützung wird auf die folgenden Kodierungen ausgeweitet werden: `US-ASCII, ISO-8859-1, UTF-16, UTF-16LE, UTF-16BE, ISO-10646-UCS2, UTF-32, UTF-32LE, UTF-32BE, ISO-10646-UCS4`

▣ *Beispiele*

- **altova:encode-string**(`"Altova"`) gibt das base64Binary-Äquivalent des UTF-8-kodierten String `"Altova"` zurück.
- **altova:encode-string**(`"Altova", "UTF-8"`) gibt das base64Binary-Äquivalent des UTF-8-kodierten String `"Altova"` zurück.

▼ get-temp-folder [altova:]

altova:get-temp-folder() als `xs:string` XP2 XQ1 XP3.1 XQ3.1
Diese Funktion hat kein Argument. Sie gibt den Pfad zum temporären Ordner des aktuellen Benutzers zurück.

Beispiele

- **altova:get-temp-folder**() würde auf einem Windows-Rechner z.B. den folgenden Pfad als xs:string zurückgeben: C:\Users\<UserName>\AppData\Local\Temp\.

generate-guid [altova:]

altova:generate-guid() als **xs:string** **XP2** **XQ1** **XP3.1** **XQ3.1**
Generiert einen eindeutigen String GUID-String.

Beispiele

- **altova:generate-guid**() gibt (z.B.) 85F971DA-17F3-4E4E-994E-99137873ACCD zurück

high-res-timer [altova:]

altova:high-res-timer() als **xs:double** **XP3.1** **XQ3.1**
Gibt einen hochauflösenden System-Timer-Wert in Sekunden zurück. Wenn in einem System ein hochauflösender Timer zur Verfügung steht, können bei Bedarf (z.B. bei Animationen und zur Ermittlung des exakten Codeausführungszeitpunkts) hochauflösende Zeitmessungen vorgenommen werden. Diese Funktion stellt die Auflösung des Hochauflösungs-Timers des Systems zur Verfügung.

Beispiele

- **altova:high-res-timer**() gibt eine Wert wie '1.16766146154566E6' zurück.

parse-html [altova:]

altova:parse-html(HTMLText *als xs:string***)** als **node()** **XP3.1** **XQ3.1**
Das Argument HTMLText ist ein String, der den Text eines HTML-Dokuments enthält. Die Funktion erstellt anhand des Strings eine HTML-Struktur. Der bereitgestellte String kann das HTML-Element enthalten, muss dies aber nicht tun. In beiden Fällen ist das Root-Element der Struktur ein Element namens **HTML**. Sie sollten sicher stellen, dass der HTML-Code im bereitgestellten String gültiger HTML-Code ist.

Beispiel

- **altova:parse-html**("<html><head/><body><h1>Header</h1></body></html>") erstellt anhand des bereitgestellten Strings eine HTML-Struktur.

sleep[altova:]

altova:sleep(Millisecs *als xs:integer***)** als **empty-sequence()** **XP2** **XQ1** **XP3.1** **XQ3.1**
Unterbricht die Ausführung der aktuellen Operation für die Anzahl der durch das Argument Millisecs angegebenen Millisekunden.

Beispiel

- **altova:sleep**(1000) unterbricht die Ausführung der aktuellen Operation für 1000 Millisekunden.

[Nach oben 654]

© 2019-2025 Altova GmbH

12.2.2 Diverse Erweiterungsfunktionen

Es gibt in Programmiersprachen wie Java und C# eine Reihe von fertigen Funktionen, die nicht als XQuery / XPath 2.0- oder XSLT-Funktionen zur Verfügung stehen. Ein gutes Beispiel dafür sind die mathematischen in Java verfügbaren Funktionen wie z.B. `sin()` und `cos()`. Stünden diese Funktionen für die Erstellung von XSLT Stylesheets und XQuery-Abfragen zur Verfügung, würde sich der Einsatzbereich von Stylesheets und Abfragen erweitern und die Erstellung von Stylesheets wäre viel einfacher. Der in einer Reihe von Altova-Produkten verwendete XSLT- und XQuery-Prozessor von Altova unterstützt die Verwendung von Erweiterungsfunktionen in Java[657] und .NET[667] sowie MSXSL Skripts für XSLT[673]. MSXSL-Skripts für XSLT[673] und die Altova-Erweiterungsfunktionen[577]. In diesem Abschnitt wird beschrieben, wie Sie Erweiterungsfunktionen und MSXSL-Skripts in Ihren XSLT Stylesheets verwenden können. Diese Beschreibungen finden Sie in den folgenden Abschnitten:

- Java-Erweiterungsfunktionen[657]
- .NET-Erweiterungsfunktionen[667]
- MSXSL-Skripts für XSLT[673]

Hauptsächlich werden dabei die folgenden beiden Punkte behandelt: (i) Wie Funktionen in den entsprechenden Bibliotheken aufgerufen werden; und (ii) welche Regeln beim Konvertieren von Argumenten in einem Funktionsaufruf in das erforderliche Format der Funktion befolgt werden und welche Regeln bei der Rückwärtskonvertierung (Funktionsresultat in XSLT/XQuery Datenobjekt) befolgt werden.

Voraussetzungen

Damit die Erweiterungsfunktionen unterstützt werden, muss auf dem Rechner, auf dem die XSLT-Transformation oder die XQuery-Ausführung stattfindet, eine Java Runtime-Umgebung (zum Aufrufen der Java-Funktionen) und ein .NET Framework 2.0 (Mindestvoraussetzung für Zugriff auf .NET-Funktionen) installiert sein oder es muss Zugriff auf eine solche bestehen.

12.2.2.1 Java-Erweiterungsfunktionen

Eine Java-Erweiterungsfunktion kann in einem XPath- oder XQuery-Ausdruck verwendet werden, um einen Java-Konstruktor oder eine Java-Methode (statisch oder Instanz) aufzurufen.

Ein Feld in einer Java-Klasse wird als Methode ohne Argument betrachtet. Bei einem Feld kann es sich um ein statisches Feld oder eine Instanz handeln. Wie man Felder aufruft, wird in den entsprechenden Unterabschnitten zu statischen Feldern und Instanzen beschrieben.

Dieser Abschnitt enthält die folgenden Unterabschnitte:

- Java: Konstruktoren[663]
- Java: Statische Methoden und statische Felder[664]
- Java: Instanzmethoden und Instanzfelder[664]
- Datentypen: XPath/XQuery in Java[665]
- Datentypen: Java in XPath/XQuery[666]

Beachten Sie die folgenden Punkte

- Wenn Sie ein Altova Desktop-Produkt verwenden, versucht die Altova-Applikation, den Pfad zur Java

Virtual Machine automatisch zu ermitteln. Dazu wird zuerst (i) die Windows Registry und dann (ii) die **JAVA_HOME**-Umgebungsvariable gelesen. Sie können im Dialogfeld "Optionen" der Applikation auch einen benutzerdefinierten Pfad hinzufügen. Dieser Eintrag hat Vorrang vor allen anderen automatisch ermittelten Java VM-Pfaden.

- Wenn Sie ein Altova Server-Produkt auf einem Windows-Rechner ausführen, wird der Pfad zur Java Virtual Machine zuerst aus der Windows Registry ausgelesen. Falls dies nicht gelingt, wird die **JAVA_HOME**-Umgebungsvariable verwendet.
- Wenn Sie ein Altova Server-Produkt auf einem Linux- oder macOS-Rechner ausführen, sollten Sie sicherstellen, dass die **JAVA_HOME**-Umgebungsvariable ordnungsgemäß definiert ist und dass sich die Java Virtual Machine-Bibliothek (unter Windows die Datei **jvm.dll**) entweder im Verzeichnis **\bin\server** oder **\bin\client** befindet.

Form der Erweiterungsfunktion

Die Erweiterungsfunktion im XPath/XQuery-Ausdruck muss die folgenden Form haben `präfix:fname()`.

- Der Teil `präfix:` kennzeichnet die Erweiterungsfunktion als Java-Funktion, indem er die Erweiterungsfunktion mit einer in-scope Namespace-Deklaration verknüpft, deren URI mit `java:` beginnen muss (*Beispiele siehe unten*). Die Namespace-Deklaration sollte eine Java-Klasse bezeichnen, z.B:
  `xmlns:myns="java:java.lang.Math"`. Sie könnte aber auch einfach lauten:
  `xmlns:myns="java"` (ohne Doppelpunkt), wobei die Identifizierung der Java-Klasse dem `fname()` Teil der Erweiterungsfunktion überlassen bleibt.
- Der Teil `fname()` identifiziert die aufgerufene Java-Methode und liefert die Argumente für die Methode (*Beispiele siehe unten*). Wenn die durch das `prefix:` Teil identifizierte Namespace URI jedoch keine Java-Klasse bezeichnet (*siehe vorheriger Punkt),* dann sollte die Java-Klasse im `fname()` Teil vor der Klasse identifiziert werden und von der Klasse durch einen Punkt getrennt sein (*siehe zweites XSLT-Beispiel unten*).

Anmerkung: Die aufgerufene Klasse muss sich unter dem Klassenpfad des Rechners befinden.

XSLT-Beispiel

Hier sehen Sie zwei Beispiele dafür, wie eine statische Methode aufgerufen werden kann. Im ersten Beispiel ist der Klassenname (`java.lang.Math`) in der Namespace URI enthalten und darf daher nicht im `fname()` Teil enthalten sein. Im zweiten Beispiel liefert der `prefix:` Teil das Präfix `java:`, während der `fname()` Teil die Klasse sowie die Methode identifiziert.

```
<xsl:value-of xmlns:jMath="java:java.lang.Math"
              select="jMath:cos(3.14)" />

<xsl:value-of xmlns:jmath="java"
              select="jmath:java.lang.Math.cos(3.14)" />
```

Die in der Erweiterungsfunktion (im Beispiel oben `cos()`) angegebene Methode muss mit dem Namen einer öffentlichen statischen Methode in der angegebenen Java-Klasse (im Beispiel oben `java.lang.Math`) übereinstimmen.

© 2019-2025 Altova GmbH

XQuery-Beispiel

Hier sehen Sie ein XQuery-Beispiel, das dem XSLT-Beispiel oben ähnlich ist:

```
<cosine xmlns:jMath="java:java.lang.Math">
   {jMath:cos(3.14)}
</cosine>
```

Benutzerdefinierte Java-Klassen

Wenn Sie Ihre eigenen Java-Klassen erstellt haben, werden die Methoden in diesen Klassen unterschiedlich aufgerufen, je nachdem: (i) ob die Klassen über eine JAR-Datei oder eine Klassendatei aufgerufen werden, und (ii) ob sich diese Dateien (JAR oder Klasse) im aktuellen Verzeichnis befinden (im selben Verzeichnis wie das XSLT- oder XQuery-Dokument) oder nicht. Wie Sie diese Dateien finden, wird in den Abschnitten Benutzerdefinierte Klassendateien 659 und Benutzerdefinierte Jar-Dateien 662 beschrieben. Pfade zu Klassendateien, die sich nicht im aktuellen Verzeichnis befinden, und Pfade zu allen JAR-Dateien müssen jedoch angegeben werden.

Anmerkung: Wenn Sie einen Namespace zu einem XSLT-Stylesheet hinzufügen möchten, das anhand eines in SPS in StyleVision generiert wurde, muss der Namespace zum `schema`-Element der obersten Ebene des XML-Schemas, auf dem das SPS basiert, hinzugefügt werden. Beachten Sie, dass die folgende Namespace Deklaration `xmlns:java="java"` standardmäßig automatisch in jedem SPS erstellt wird, das in StyleVision entworfen wurde.

12.2.2.1.1 Benutzerdefinierte Klassendateien

Wenn der Zugriff über eine Klassendatei erfolgt, gibt es vier Möglichkeiten:

- Die Klassendatei befindet sich in einem Paket. Die XSLT-oder XQuery-Datei befindet sich im selben Ordner wie das Java-Paket. (*Siehe Beispiel unten* 660.)
- Die Klassendatei befindet sich nicht in einem Paket. Die XSLT-oder XQuery-Datei befindet sich im selben Ordner wie die Klassendatei. (*Siehe Beispiel unten* 660.)
- Die Klassendatei befindet sich in einem Paket. Die XSLT-oder XQuery-Datei befindet sich in irgendeinem beliebig gewählten Ordner. (*Siehe Beispiel unten* 661.)
- Die Klassendatei befindet sich nicht in einem Paket. Die XSLT-oder XQuery-Datei befindet sich in irgendeinem beliebig gewählten Ordner. (*Siehe Beispiel unten* 661.)

Gesetzt der Fall, die Klassendatei befindet sich nicht in einem Paket, sondern im selben Ordner wie das XSLT- oder XQuery-Dokument, so muss der Dateipfad nicht angegeben werden, da alle Klassen im Ordner gefunden werden. Die Syntax zum Identifizieren einer Klasse lautet:

```
java:classname
```

wobei

`java`: angibt, dass eine benutzerdefinierte Java-Funktion aufgerufen wird; (Java-Klassen im aktuellen Verzeichnis werden standardmäßig geladen)

`classname` der Name der Klasse der erforderlichen Methode ist

die Klasse in einer Namespace URI identifiziert wird und der Namespace einem Methodenaufruf als Präfix vorangestellt wird.

Klassendatei in einem Paket, XSLT/XQuery-Datei befindet sich im selben Ordner wie das Java-Paket

Im Beispiel unten wird die Methode `getVehicleType()` der Klasse `Car` des Pakets `com.altova.extfunc` aufgerufen. Das Paket `com.altova.extfunc` befindet sich im Ordner `JavaProject`. Die XSLT-Datei befindet sich ebenfalls im Ordner `JavaProject`.

```
<xsl:stylesheet version="2.0"
       xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
       xmlns:xs="http://www.w3.org/2001/XMLSchema"
       xmlns:fn="http://www.w3.org/2005/xpath-functions"
       xmlns:car="java:com.altova.extfunc.Car" >
<xsl:output exclude-result-prefixes="fn car xsl fo xs"/>

<xsl:template match="/">
   <a>
   <xsl:value-of select="car:getVehicleType()"/>
   </a>
</xsl:template>

</xsl:stylesheet>
```

Die Klassendatei wird referenziert, die XSLT/XQuery-Datei befindet sich im selben Ordner wie die Klassendatei

Im Beispiel unten wird die Methode `getVehicleType()` der Klasse `Car` des Pakets `com.altova.extfunc` aufgerufen. Angenommen, (i) die Klassendatei `Car` class befindet sich im folgenden Ordner: `JavaProject/com/altova/extfunc` und (ii) dieser Ordner ist der aktuelle Ordner im Beispiel unten. Die XSLT-Datei befindet sich ebenfalls im Ordner `JavaProject/com/altova/extfunc`.

```
<xsl:stylesheet version="2.0"
       xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
       xmlns:xs="http://www.w3.org/2001/XMLSchema"
       xmlns:fn="http://www.w3.org/2005/xpath-functions"
       xmlns:car="java:Car" >
<xsl:output exclude-result-prefixes="fn car xsl fo xs"/>

<xsl:template match="/">
   <a>
   <xsl:value-of select="car:getVehicleType()"/>
   </a>
</xsl:template>

</xsl:stylesheet>
```


© 2019-2025 Altova GmbH

Die Klassendatei befindet sich in einem Paket, die XSLT/XQuery-Datei befindet sich in einem beliebigen Ordner

Im Beispiel unten wird die Methode `getCarColor()` der Klasse `Car` des Pakets `com.altova.extfunc` aufgerufen. Das Paket `com.altova.extfunc` befindet sich im Ordner `JavaProject`. Die XSLT-Datei befindet sich in einem beliebigen Ordner. In diesem Fall muss der Pfad des Pakets mit der URI als Abfragestring definiert werden. Die Syntax lautet:

```
java:classname[?path=uri-of-classfile]
```

wobei

`java:` angibt, dass eine benutzerdefinierte Java-Funktion aufgerufen wird
`uri-of-classfile` die URI der Klassendatei ist
`classname` der Name der Klasse der benötigten Methode ist

die Klasse in einer Namespace URI identifiziert wird und der Namespace einem Methodenaufruf als Präfix vorangestellt wird. Im Beispiel unten sehen Sie, wie eine Klassendatei aufgerufen wird, die sich in einem anderen als dem aktuellen Verzeichnis befindet.

```
<xsl:stylesheet version="2.0"
       xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
       xmlns:xs="http://www.w3.org/2001/XMLSchema"
       xmlns:fn="http://www.w3.org/2005/xpath-functions"
       xmlns:car="java:com.altova.extfunc.Car?path=file:///C:/JavaProject/" >

<xsl:output exclude-result-prefixes="fn car xsl xs"/>

<xsl:template match="/">
   <xsl:variable name="myCar" select="car:new('red')" />
   <a><xsl:value-of select="car:getCarColor($myCar)"/></a>
</xsl:template>

</xsl:stylesheet>
```

Die Klassendatei wird referenziert, die XSLT/XQuery-Datei befindet sich in einem beliebigen Ordner

Im Beispiel unten wird die Methode `getCarColor()` der Klasse `Car` aufgerufen. Angenommen, die Klassendatei `Car` befindet sich im Ordner `C:/JavaProject/com/altova/extfunc`. Die XSLT-Datei befindet sich in einem beliebigen Ordner. Der Pfad der Klassendatei muss dann in der Namespace-URI als Abfragestring definiert werden. Die Syntax lautet:

```
java:classname[?path=<uri-of-classfile>]
```

wobei

`java:` angibt, dass eine benutzerdefinierte Java-Funktion aufgerufen wird
`uri-of-classfile` die URI der Klassendatei ist
`classname` der Name der Klasse der benötigten Methode ist

die Klasse in einer Namespace URI identifiziert wird und der Namespace einem Methodenaufruf als Präfix vorangestellt wird. Im Beispiel unten sehen Sie, wie eine Klassendatei aufgerufen wird, die sich in einem anderen als dem aktuellen Verzeichnis befindet.

```
<xsl:stylesheet version="2.0"
       xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
       xmlns:xs="http://www.w3.org/2001/XMLSchema"
       xmlns:fn="http://www.w3.org/2005/xpath-functions"
       xmlns:car="java:Car?path=file:///C:/JavaProject/com/altova/extfunc/" >

<xsl:output exclude-result-prefixes="fn car xsl xs"/>

<xsl:template match="/">
   <xsl:variable name="myCar" select="car:new('red')" />
   <a><xsl:value-of select="car:getCarColor($myCar)"/></a>
</xsl:template>

</xsl:stylesheet>
```

Anmerkung: Wenn ein Pfad über eine Erweiterungsfunktion angegeben wird, wird er zum ClassLoader hinzugefügt.

12.2.2.1.2 Benutzerdefinierte Jar-Dateien

JAR-Dateien

Wenn der Zugriff über eine JAR-Datei erfolgt, muss die URI der JAR-Datei mit Hilfe der folgenden Syntax definiert werden:

```
xmlns:classNS="java:classname?path=jar:uri-of-jarfile!/"
```

Die Methode wird anschließend durch Verwendung des Präfix der Namespace URI aufgerufen, der die Klasse bezeichnet: `classNS:method()`

wobei im obigen Beispiel:

`java:` angibt, dass eine Java-Funktion aufgerufen wird
`classname` der Name der Klasse der benutzerdefinierten Klasse ist
`?` das Trennzeichen zwischen dem Klassennamen und dem Pfad ist
`path=jar:` angibt, dass es sich um einen Pfad zu einer JAR-Datei handelt
`uri-of-jarfile` die URI der jar-Datei angibt
`!/` das Trennzeichen am Ende des Pfades ist
`classNS:method()` der Aufruf der Methode ist

Alternativ dazu kann der Klassenname mit dem Methodenaufruf angegeben werden. Hier sehen Sie zwei Beispiele für die Syntax:

```
xmlns:ns1="java:docx.layout.pages?path=jar:file:///c:/projects/docs/docx.jar!/"
ns1:main()

xmlns:ns2="java?path=jar:file:///c:/projects/docs/docx.jar!/"
ns2:docx.layout.pages.main()
```


© 2019-2025 Altova GmbH

Hier sehen Sie ein komplettes XSLT-Beispiel, in dem eine JAR-Datei verwendet wird, um eine Java-Erweiterungsfunktion aufzurufen.:

```
<xsl:stylesheet version="2.0"
       xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
       xmlns:xs="http://www.w3.org/2001/XMLSchema"
       xmlns:fn="http://www.w3.org/2005/xpath-functions"
       xmlns:car="java?path=jar:file:///C:/test/Car1.jar!/" >
<xsl:output exclude-result-prefixes="fn car xsl xs"/>

<xsl:template match="/">
       <xsl:variable name="myCar" select="car:Car1.new('red')" />
       <a><xsl:value-of select="car:Car1.getCarColor($myCar)"/></a>
</xsl:template>

<xsl:template match="car"/>

</xsl:stylesheet>
```

Anmerkung: Wenn ein Pfad über eine Erweiterungsfunktion angegeben wird, wird er zum ClassLoader hinzugefügt.

12.2.2.1.3 Java: Konstruktoren

Eine Erweiterungsfunktion kann dazu verwendet werden, um einen Java-Konstruktor aufzurufen. Alle Konstruktoren werden mit der Pseudofunktion `new()` aufgerufen.

Wenn das Ergebnis eines Java-Konstruktors implizit in XPath/XQuery-Datentypen konvertiert werden kann[666], dann gibt die Java-Erweiterungsfunktion eine Sequenz zurück, bei der es sich um einem XPath/XQuery-Datentyp handelt. Wenn das Ergebnis eines Konstruktoraufrufs nicht in einen passenden XPath/XQuery-Datentyp konvertiert werden kann, dann erstellt der Konstruktor ein wrapped Java-Objekt mit einem Typ, der den Namen der Klasse hat, die dieses Java-Objekt zurückgibt. Wenn z.B. ein Konstruktor für die Klasse `java.util.Date` aufgerufen wird (`java.util.Date.new()`), so wird ein Objekt vom Typ `java.util.Date` zurückgegeben. Das lexikalische Format des zurückgegebenen Objekts stimmt unter Umständen nicht mit dem lexikalischen Format des XPath-Datentyps überein und der Wert müsste daher in das lexikalische Format des erforderlichen XPath-Datentyps und anschließend in den erforderlichen XPath-Datentyp konvertiert werden.

Ein von einem Konstruktor erstelltes Java-Objekt kann für zwei Zwecke verwendet werden:

- Es kann einer Variable zugewiesen werden:
  ```
  <xsl:variable name="currentdate" select="date:new()"
  xmlns:date="java:java.util.Date" />
  ```
- Es kann an eine Erweiterungsfunktion übergeben werden (*siehe Instanzmethode und Instanzfelder*[664]):
  ```
  <xsl:value-of select="date:toString(date:new())" xmlns:date="java:java.util.Date" />
  ```

12.2.2.1.4 Java: Statische Methoden und statische Felder

Eine statische Methode wird direkt über ihren Java-Namen und durch Angabe der Argumente für die Methode aufgerufen. Statische Felder (Methoden, die keine Argumente haben), wie z.B. die Konstantenwertfelder `E` und `PI` werden ohne Angabe eines Arguments aufgerufen.

XSLT-Beispiele

Hier sehen Sie einige Beispiele dafür, wie statische Methoden und Felder aufgerufen werden können:

```
<xsl:value-of xmlns:jMath="java:java.lang.Math"
              select="jMath:cos(3.14)" />

<xsl:value-of xmlns:jMath="java:java.lang.Math"
              select="jMath:cos( jMath:PI() )" />

<xsl:value-of xmlns:jMath="java:java.lang.Math"
              select="jMath:E() * jMath:cos(3.14)" />
```

Beachten Sie, dass die Erweiterungsfunktionen die Form `prefix:fname()` haben. Das Präfix ist in allen drei Fällen `jMath:`. Es ist mit der Namespace URI `java:java.lang.Math` verknüpft. (Die Namespace URI muss mit `java:`. beginnen. In den obigen Beispielen wurde es um den Klassennamen erweitert (`java.lang.Math`).) Der Teil `fname()` der Erweiterungsfunktionen muss mit dem Namen der öffentlichen Klasse (z.B. `java.lang.Math`) gefolgt vom Namen einer öffentlichen statischen Methode mit ihrem/ihren Argument(en) (wie z.B. `(3.14)`) oder einem öffentlichen statischen Feld (z.B. `PI()`) übereinstimmen.

In den obigen Beispielen wurde der Klassenname in die Namespace URI inkludiert. Wäre sie nicht in der Namespace URI enthalten, müsste sie in den `fname()` Teil der Erweiterungsfunktion inkludiert werden. Z.B:

```
<xsl:value-of xmlns:java="java:"
              select="java:java.lang.Math.cos(3.14)" />
```

XQuery-Beispiel

Ein ähnliches Beispiel in XQuery wäre:

```
<cosine xmlns:jMath="java:java.lang.Math">
   {jMath:cos(3.14)}
</cosine>
```

12.2.2.1.5 Java: Instanzmethoden und Instanzfelder

Bei einer Instanzmethode wird als erstes Argument eines Methodenaufrufs ein Java-Objekt an die Methode übergeben. Ein solches Java-Objekt würde normalerweise mit Hilfe einer Erweiterungsfunktion (z.B. eines Konstruktoraufrufs) oder eines Stylesheet-Parameters/einer Stylesheet-Variablen erstellt. Ein XSLT-Beispiel dafür wäre:

© 2019-2025 Altova GmbH

```
<xsl:stylesheet version="1.0" exclude-result-prefixes="date"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:date="java:java.util.Date"
  xmlns:jlang="java:java.lang">
  <xsl:param name="CurrentDate" select="date:new()"/>
  <xsl:template match="/">
      <enrollment institution-id="Altova School"
                date="{date:toString($CurrentDate)}"
                type="{jlang:Object.toString(jlang:Object.getClass( date:new() ))}">
      </enrollment>
  </xsl:template>
</xsl:stylesheet>
```

Im Beispiel oben wird der Wert des Node `enrollment/@type` folgendermaßen erstellt:

1. Es wird ein Objekt mit einem Konstruktor für die Klasse `java.util.Date` (mit dem Konstruktor `date:new()`) erstellt.
2. Dieses Java-Objekt wird als das Argument der Methode `jlang.Object.getClass` übergeben.
3. Das mit der Methode `getClass` abgerufene Objekt wird als das Argument an die Methode `jlang.Object.toString` übergeben.

Das Ergebnis (der Wert von `@type`) ist ein String, der den Wert `java.util.Date` hat.

Ein Instanzfeld unterscheidet sich theoretisch insofern von einer Instanzmethode, als es sich nicht um ein Java-Objekt per se handelt, das als Argument an das Instanzfeld übergeben wird. Stattdessen wird ein Parameter oder eine Variable als Argument übergeben. Der Parameter/die Variable kann allerdings selbst den Wert enthalten, der von einem Java-Objekt zurückgegeben wird. So erhält z.B. der Parameter `CurrentDate` den Wert, der von einem Konstruktor für die Klasse `java.util.Date` zurückgegeben wird. Dieser Wert wird anschließend als Argument an die Instanzmethode `date:toString` übergeben, um den Wert von `/enrollment/@date` bereitzustellen.

12.2.2.1.6 Datentypen: XPath/XQuery in Java

Wenn von einem XPath/XQuery-Ausdruck aus eine Java-Funktion aufgerufen wird, spielt der Datentyp der Argumente der Funktion eine wichtige Rolle, welche von mehreren Java-Klassen desselben Namens aufgerufen wird.

In Java gelten die folgenden Regeln:

- Wenn es mehr als eine Java-Methode mit demselben Namen gibt, jede aber eine andere Anzahl von Argumenten als die andere(n) hat, so wird die Java-Methode ausgewählt, die der Anzahl der Argumente im Funktionsaufruf am ehesten entspricht.
- Die XPath/XQuery-Datentypen "string", "number" und "boolean" (*siehe Liste unten*) werden implizit in einen entsprechenden Java-Datentyp konvertiert. Wenn der bereitgestellte XPath/XQuery-Datentyp in mehr als einen Java-Typ konvertiert werden kann (z.B: `xs:integer`), so wird jener Java-Typ ausgewählt, der für die ausgewählte Methode deklariert wurde. Wenn die aufgerufene Java-Methode z.B. `fx(decimal)` und der bereitgestellte XPath/XQuery-Datentyp `xs:integer` ist, so wird `xs:integer` in den Java-Datentyp `decimal` konvertiert.

In der Tabelle unten sehen Sie eine Liste der impliziten Konvertierungen der XPath/XQuery-Datentypen "string", "number" und "boolean"in Java-Datentypen.

xs:string	java.lang.String
xs:boolean	boolean (primitive), java.lang.Boolean
xs:integer	int, long, short, byte, float, double und die Wrapper-Klassen davon wie z.B. java.lang.Integer
xs:float	float (primitive), java.lang.Float, double (primitive)
xs:double	double (primitive), java.lang.Double
xs:decimal	float (primitive), java.lang.Float, double(primitive), java.lang.Double

Die oben aufgelisteten Subtypen von XML-Schema-Datentypen (die in XPath und XQuery verwendet werden) werden ebenfalls in den/die Java-Typ(en), der/die dem übergeordneten Subtyp entsprechen, konvertiert.

In einigen Fällen ist es nicht möglich, auf Basis der verfügbaren Informationen die richtige Java-Methode auszuwählen. Nehmen Sie als Beispiel den folgenden Fall.

- Das bereitgestellte Argument ist ein xs:untypedAtomic Wert 10 und ist für die Methode mymethod(float) bestimmt.
- Es gibt jedoch eine weitere Methode in der Klasse, die ein Argument eines anderen Datentyps erhält: mymethod(double).
- Da die Methodennamen dieselben sind und der bereitgestellte Typ (xs:untypedAtomic) sowohl in float als auch double korrekt konvertiert werden könnte, kann es geschehen, dass xs:untypedAtomic in double anstelle von float konvertiert wird.
- Infolgedessen handelt es sich dann bei der ausgewählten Methode nicht um die benötigte Methode, sodass nicht das erwartete Ergebnis erzielt wird. Als Umgehungslösung können Sie eine benutzerdefinierte Methode mit einem anderen Namen erstellen und diese Methode verwenden.

Typen, die in der Liste oben nicht enthalten sind (z.B. xs:date), werden nicht konvertiert und generieren einen Fehler. Beachten Sie jedoch, dass es in einigen Fällen unter Umständen möglich ist, den benötigten Java-Typ mittels eines Java-Konstruktors zu erstellen.

12.2.2.1.7 Datentypen: Java in XPath/XQuery

Wenn eine Java-Methode einen Wert zurückgibt und der Datentyp des Werts "string", "numeric" oder "boolean" ist, wird anschließend in den entsprechenden XPath/XQuery-Typ konvertiert. So werden z.B. die Java-Datentypen javaa.lang.Boolean und boolean in xsd:boolean konvertiert.

Von Funktionen zurückgegebene eindimensionale Arrays werden zu einer Sequenz erweitert. Mehrdimensionale Arrays werden nicht konvertiert und sollten daher in einen Wrapper gesetzt werden.

Wenn ein wrapped Java-Objekt oder ein Datentyp zurückgegeben wird, bei dem es sich nicht um den Typ "string", "numeric" oder "boolean" handelt, können Sie sicherstellen, dass die Konvertierung in den benötigten XPath/XQuery-Typ erfolgt, indem Sie zuerst eine Java-Methode (e.g toString) verwenden, um das Java-Objekt in einen String zu konvertieren. In XPath/XQuery kann der String geändert werden, damit er der lexikalischen

 © 2019-2025 Altova GmbH

Darstellung des benötigten Typs entspricht, und anschließend z.B. mit Hilfe des Ausdrucks `cast as` in den benötigten Typ konvertiert werden.

12.2.2.2 .NET-Erweiterungsfunktionen

Wenn Sie auf einem Windows-Rechner mit der .NET-Plattform arbeiten, können Sie Erweiterungsfunktionen verwenden, die in jeder beliebigen der .NET-Sprachen geschrieben wurden (z.B. C#). Eine .NET Erweiterungsfunktion kann in einem XPath- oder XQuery-Ausdruck verwendet werden, um einen Konstruktor, eine Eigenschaft oder Methode (statische oder Instanz) in einer .NET-Klasse aufzurufen.

Eine Eigenschaft einer .NET-Klasse wird mit der Syntax `get_PropertyName()` aufgerufen.

Dieser Abschnitt ist in die folgenden Unterabschnitte gegliedert:

- .NET: Konstruktoren 669
- .NET: Statische Methoden und statische Felder 670
- .NET: Instanzmethoden und Instanzfelder 671
- Datentypen: XPath/XQuery in .NET 672
- Datentypen: .NET in XPath/XQuery 673

Form der Erweiterungsfunktion

Die Erweiterungsfunktion im XPath/XQuery-Ausdruck muss die folgende Form haben `präfix:fname()`.

- Der Teil `präfix:` ist mit einer URI verknüpft, die die benötigte .NET-Klasse definiert.
- Der Teil `fname()` identifiziert den Konstruktor, die Eigenschaft oder die Methode (statisch oder Instanz) innerhalb der .NET-Klasse und liefert alle gegebenenfalls benötigten Argumente.

- Die URI muss mit `clitype:` beginnen (welches die Funktion als .NET-Erweiterungsfunktion kennzeichnet).
- Die Form `prefix:fname()` der Erweiterungsfunktion kann mit Systemklassen und mit Klassen in einer geladenen Assembly verwendet werden. Wenn eine Klasse allerdings geladen werden muss, müssen zusätzliche Parameter mit den benötigten Informationen bereitgestellt werden.

Parameter

Zum Laden einer Assembly werden die folgenden Parameter verwendet:

`asm`	Der Name der zu ladenden Assembly
`ver`	Die Versionsnummer: eine Maximalzahl von vier Ganzzahlen, die durch Punkte getrennt sind
`sn`	Das Key Token des Strong Name der Assembly (16 Hex-Stellen).
`from`	Eine URI gibt den Pfad der zu ladenden Assembly (DLL) an. Wenn die URI relativ ist, ist sie relativ zum XSLT- oder XQuery-Dokument. Wenn dieser Parameter vorhanden ist, werden alle anderen Parameter ignoriert.

`partialname`	Der partielle Name der Assembly. Er wird für `Assembly.LoadWith.PartialName()` bereitgestellt, welches versuchen wird, die Assembly zu laden. Wenn `partialname` vorhanden ist, werden alle anderen Parameter ignoriert.
`loc`	Die Locale, z.B. `en-US`. Die Standardeinstellung ist `neutral`

Wenn die Assembly aus einer DLL geladen werden soll, verwenden Sie den `from` Parameter und lassen Sie den `sn` Parameter weg. Wenn die Assembly aus dem Global Assembly Cache (GAC) geladen werden soll, verwenden Sie den `sn` Parameter und lassen Sie den `from` Parameter weg.

Vor dem ersten Parameter muss ein Fragezeichen eingefügt werden. Parameter müssen durch ein Semikolon getrennt werden. Der Wert des Parameternamens wird durch ein Ist-Gleich-Zeichen angegeben (*siehe Beispiele unten*).

Beispiele für Namespace-Deklarationen

Ein Beispiel für eine Namespace Deklaration in XSLT, die die Systemklasse `System.Environment`: identifiziert.

```
xmlns:myns="clitype:System.Environment"
```

Ein Beispiel für eine Namespace Deklaration in XSLT, die die zu ladende Klasse als `Trade.Forward.Scrip`: identifiziert.

```
xmlns:myns="clitype:Trade.Forward.Scrip?asm=forward;version=10.6.2.1"
```

Ein Beispiel für eine Namespace-Deklaration in XQuery, die die Systemklasse `MyManagedDLL.testClass` identifiziert. Es werden zwei Klassen unterschieden:

1. Wenn die Assembly aus dem GAC geladen wird:
```
declare namespace cs="clitype:MyManagedDLL.testClass?asm=MyManagedDLL;
ver=1.2.3.4;loc=neutral;sn=b9f091b72dccfba8";
```

2. Wenn die Assembly aus der DLL geladen wird (vollständige und partielle Referenzen unten):
```
declare namespace cs="clitype:MyManagedDLL.testClass?from=file:///C:/Altova
Projects/extFunctions/MyManagedDLL.dll;

declare namespace cs="clitype:MyManagedDLL.testClass?from=MyManagedDLL.dll;
```

XSLT-Beispiel

Hier sehen Sie ein vollständiges XSLT-Beispiel, in dem Funktionen in der Systemklasse `System.Math`: aufgerufen werden:

```
<xsl:stylesheet version="2.0"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:fn="http://www.w3.org/2005/xpath-functions">
   <xsl:output method="xml" omit-xml-declaration="yes" />
   <xsl:template match="/">
      <math xmlns:math="clitype:System.Math">
```


© 2019-2025 Altova GmbH

```
        <sqrt><xsl:value-of select="math:Sqrt(9)"/></sqrt>
        <pi><xsl:value-of select="math:PI()"/></pi>
        <e><xsl:value-of select="math:E()"/></e>
        <pow><xsl:value-of select="math:Pow(math:PI(), math:E())"/></pow>
      </math>
   </xsl:template>
</xsl:stylesheet>
```

Die Namespace-Deklaration für das Element `math` verknüpft das Präfix `math:` mit der URI `clitype:System.Math`. Der Beginn der URI `clitype:` gibt an, dass danach entweder eine Systemklasse oder eine geladene Klasse definiert wird. Das Präfix `math:` im XPath-Ausdruck verknüpft die Erweiterungsfunktionen mit der URI (und durch Erweiterung der Klasse) `System.Math`. Die Erweiterungsfunktionen identifizieren Methoden in der Klasse `System.Math` und stellen Argumente bereit, wo dies erforderlich ist.

XQuery-Beispiel

Hier sehen Sie ein XQuery-Beispielfragment ähnlich dem XSLT-Beispiel oben:

```
<math xmlns:math="clitype:System.Math">
   {math:Sqrt(9)}
</math>
```

Wie beim XSLT-Beispiel weiter oben identifiziert die Namespace-Deklaration die .NET-Klasse, in diesem Fall eine Systemklasse. Der XQuery-Ausdruck identifiziert die aufzurufenden Methode und liefert das Argument.

12.2.2.2.1 .NET: Konstruktoren

Eine Erweiterungsfunktion kann verwendet werden, um einen .NET-Konstruktor aufzurufen. Alle Konstruktoren werden mit der Pseudofunktion `new()` aufgerufen. Wenn es mehrere Konstruktoren für eine Klasse gibt, wird der Konstruktor ausgewählt, der der Anzahl der bereitgestellten Argumente am ehesten entspricht. Wenn kein passender Konstruktor gefunden wird, der den bereitgestellten Argumenten entspricht, wird die Fehlermeldung `'No constructor found'` zurückgegeben.

Konstruktoren, die XPath/XQuery-Datentypen zurückgeben

Wenn das Ergebnis eines .NET-Konstruktors implizit in XPath/XQuery-Datentypen konvertiert werden kann 666, gibt die .NET-Erweiterungsfunktion eine Sequenz zurück, bei der es sich um einen XPath/XQuery-Datentyp handelt.

Konstruktoren, die .NET-Objekte zurückgeben

Wenn das Ergebnis eines .NET-Konstruktoraufrufs nicht in einen passenden XPath/XQuery-Datentyp konvertiert werden kann, erstellt der Konstruktor ein wrapped .NET-Objekt mit einem Typ, der der Name der Klasse ist, die dieses Objekt zurückgibt. Wenn z.B. ein Konstruktor für die Klasse `System.DateTime` aufgerufen wird (mit `System.DateTime.new()`), so wird ein Objekt mit dem Typ `System.DateTime` zurückgegeben.

Das lexikalische Format des zurückgegebenen Objekts stimmt unter Umständen nicht mit dem lexikalischen Format eines erforderlichen XPath-Datentyps überein. In solchen Fällen müsste der zurückgegebene Wert: (i)

in das lexikalische Format des benötigten XPath-Datentyps konvertiert werden; und (ii) in den erforderlichen XPath-Datentyp konvertiert werden.

Ein von einem Konstruktor erstelltes .NET-Objekt kann für drei Zwecke verwendet werden:

- Es kann innerhalb einer Variable verwendet werden:
  ```
  <xsl:variable name="currentdate" select="date:new(2008, 4, 29)"
  xmlns:date="clitype:System.DateTime" />
  ```
- Es kann an eine Erweiterungsfunktion übergeben werden (*siehe* *Instanzmethode und Instanzfelder* 664):
  ```
  <xsl:value-of select="date:ToString(date:new(2008, 4, 29))"
  xmlns:date="clitype:System.DateTime" />
  ```
- Es kann in einen String, eine Zahl oder einen Booleschen Ausdruck konvertiert werden:
- ```
  <xsl:value-of select="xs:integer(date:get_Month(date:new(2008, 4, 29)))"
  xmlns:date="clitype:System.DateTime" />
  ```

12.2.2.2.2 .NET: Statische Methoden und statische Felder

Eine statische Methode wird direkt über ihren Namen und durch Angabe der Argumente für die Methode aufgerufen. Der im Aufruf verwendete Name muss exakt mit einer öffentlichen statischen Methode in der angegebenen Klasse übereinstimmen. Wenn der Methodenname und die Anzahl der in der Funktion angegebenen Argumente mit mehr als einer Methode in einer Klasse übereinstimmen, werden die Typen der bereitgestellten Argumente nach der besten Übereinstimmung überprüft. Wenn keine eindeutig passende Methode gefunden werden kann, wird ein Fehler ausgegeben.

Anmerkung: Ein Feld in einer .NET-Klasse wird als Methode ohne Argument betrachtet. Eine Eigenschaft wird mit der Syntax `get_PropertyName()` aufgerufen.

Beispiele

Ein XSLT-Beispiel, in dem Sie einen Methodenaufruf mit einem Argument (`System.Math.Sin(arg)`) sehen:

```
<xsl:value-of select="math:Sin(30)" xmlns:math="clitype:System.Math"/>
```

Ein XSLT-Beispiel, in dem Sie einen Aufruf eines Felds (wird als Methode ohne Argument betrachtet) sehen (`System.Double.MaxValue()`):

```
<xsl:value-of select="double:MaxValue()" xmlns:double="clitype:System.Double"/>
```

Ein XSLT-Beispiel, in dem Sie einen Aufruf einer Eigenschaft (Syntax ist `get_PropertyName()`) (`System.String()`) sehen:

```
<xsl:value-of select="string:get_Length('my string')"
xmlns:string="clitype:System.String"/>
```


© 2019-2025 Altova GmbH
```

Ein XQuery-Beispiel, in dem Sie einen Aufruf einer Methode mit einem Argument (`System.Math.Sin(arg)`) sehen:

```
<sin xmlns:math="clitype:System.Math">
   { math:Sin(30) }
</sin>
```

12.2.2.2.3 .NET: Instanzmethoden und Instanzfelder

Bei einer Instanzmethode wird als erstes Argument des Methodenaufrufs ein .NET-Objekt an die Methode übergeben. Dieses .NET-Objekt wird normalerweise mit Hilfe einer Erweiterungsfunktion (z.B. durch einen Konstruktoraufruf) oder einen Stylesheet-Parameter/eine Stylesheet-Variable erstellt. Ein XSLT-Beispiel dieser Art wäre:

```
<xsl:stylesheet version="2.0"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:fn="http://www.w3.org/2005/xpath-functions">
   <xsl:output method="xml" omit-xml-declaration="yes"/>
   <xsl:template match="/">
      <xsl:variable name="releasedate"
         select="date:new(2008, 4, 29)"
         xmlns:date="clitype:System.DateTime"/>
      <doc>
         <date>
            <xsl:value-of select="date:ToString(date:new(2008, 4, 29))"
               xmlns:date="clitype:System.DateTime"/>
         </date>
         <date>
            <xsl:value-of select="date:ToString($releasedate)"
               xmlns:date="clitype:System.DateTime"/>
         </date>
      </doc>
   </xsl:template>
</xsl:stylesheet>
```

Im Beispiel oben wird ein `System.DateTime` Konstruktor (`new(2008, 4, 29)`) verwendet, um ein .NET-Objekt vom Typ `System.DateTime` zu erstellen. Diese Objekt wird zweimal erstellt, einmal als Wert der Variablen `releasedate`, ein zweites Mal als das erste und einzige Argument der Methode `System.DateTime.ToString()`. Die Instanzmethode `System.DateTime.ToString()` wird zwei Mal aufgerufen, beide Male mit dem `System.DateTime` Konstruktor (`new(2008, 4, 29)`) als erstem und einzigem Argument. In einer dieser Instanzen wird die Variable `releasedate` verwendet, um das .NET-Objekt abzurufen.

Instanzmethoden und Instanzfelder

Der Unterschied zwischen einer Instanzmethode und einem Instanzfeld ist ein theoretischer. In einer Instanzmethode wird ein .NET-Objekt direkt als Argument übergeben; in einem Instanzfeld wird stattdessen ein Parameter oder eine Variable übergeben - auch wenn der Parameter bzw. die Variable selbst ein .NET-Objekt enthalten kann. So enthält z.B. die Variable `releasedate` im Beispiel oben ein .NET-Objekt und es ist diese

Variable, die als das Argument von `ToString()` an den zweiten `date` Elementkonstruktor übergeben wird. Die `ToString()` Instanz im ersten `date` Element ist daher eine Instanzmethode, während die zweite als Instanzfeld betrachtet wird. Das in beiden Instanzen erzeugte Ergebnis ist jedoch dasselbe.

12.2.2.2.4 Datentypen: XPath/XQuery in .NET

Wenn in einem XPath/XQuery-Ausdruck eine .NET-Erweiterungsfunktion verwendet wird, spielen die Datentypen der Argumente der Funktion eine wichtige Rolle bei der Entscheidung, welche der vielen .NET-Methoden mit demselben Namen aufgerufen werden soll.

In .NET gelten die folgenden Regeln:

- Wenn es mehr als eine Methode mit demselben Namen in einer Klasse gibt, so stehen nur die Methoden zur Auswahl, die dieselbe Anzahl von Argumenten wie der Funktionsaufruf haben.
- Die XPath/XQuery-Datentypen "string", "number" und "boolean" (*siehe Liste unten*) werden implizit in einen entsprechenden .NET-Datentyp konvertiert. Wenn der bereitgestellte XPath/XQuery-Datentyp in mehr als einen .NET-Typ konvertiert werden kann (z.B: `xs:integer`), so wird jener .NET-Typ ausgewählt, der für die ausgewählte Methode deklariert wurde. Wenn die aufgerufene .NET-Methode z.B. `fx(double)` und der bereitgestellte XPath/XQuery-Datentyp `xs:integer` ist, so wird `xs:integer` in den .NET-Datentyp `double`

In der Tabelle unten sehen Sie eine Liste der impliziten Konvertierungen der XPath/XQuery-Datentypen "string", "number" und "boolean"in .NET-Datentypen.

`xs:string`	`StringValue, string`
`xs:boolean`	`BooleanValue, bool`
`xs:integer`	`IntegerValue, decimal, long, integer, short, byte, double, float`
`xs:float`	`FloatValue, float, double`
`xs:double`	`DoubleValue, double`
`xs:decimal`	`DecimalValue, decimal, double, float`

Die oben aufgelisteten Subtypen von XML-Schema-Datentypen (die in XPath und XQuery verwendet werden) werden ebenfalls in den/die .NET-Typ(en), der/die dem übergeordneten Subtyp entsprechen, konvertiert.

In einigen Fällen ist es nicht möglich, auf Basis der verfügbaren Informationen die richtige .NET-Methode auszuwählen. Nehmen Sie als Beispiel den folgenden Fall.

- Das bereitgestellte Argument ist ein `xs:untypedAtomic` Wert 10 und ist für die Methode `mymethod(float)` bestimmt.
- Es gibt jedoch eine weitere Methode in der Klasse, die ein Argument eines anderen Datentyps erhält: `mymethod(double)`.
- Da die Methodennamen dieselben sind und der bereitgestellte Typ (`xs:untypedAtomic`) sowohl in `float` als auch `double` korrekt konvertiert werden könnte, kann es geschehen, dass `xs:untypedAtomic` in `double` anstelle von `float` konvertiert wird.
- Infolgedessen handelt es sich dann bei der ausgewählten Methode nicht um die benötigte Methode, sodass nicht das erwartete Ergebnis erzielt wird. Als Umgehungslösung können Sie eine benutzerdefinierte Methode mit einem anderen Namen erstellen und diese Methode verwenden.

© 2019-2025 Altova GmbH

Typen, die in der Liste oben nicht enthalten sind (z.B. `xs:date`), werden nicht konvertiert und generieren einen Fehler.

12.2.2.2.5 Datentypen: .NET in XPath/XQuery

Wenn eine .NET-Methode einen Wert zurückgibt und der Datentyp des Werts "string", "numeric" oder "boolean" ist, wird er anschließend in den entsprechenden XPath/XQuery-Typ konvertiert. So wird z.B. der .NET-Datentyp `decimal` in `xsd:decimal` konvertiert.

Wenn ein .NET-Objekt oder ein Datentyp zurückgegeben wird, bei dem es sich nicht um den Typ "string", "numeric" oder "boolean" handelt, können Sie sicherstellen, dass die Konvertierung in den benötigten XPath/XQuery-Typ erfolgt, indem Sie zuerst eine .NET-Methode (z.B. `System.DateTime.ToString()`) verwenden, um das .NET-Objekt in einen String zu konvertieren. In XPath/XQuery kann der String geändert werden, damit er der lexikalischen Darstellung des benötigten Typs entspricht, und anschließend z.B. mit Hilfe des Ausdrucks `cast as` in den benötigten Typ konvertiert werden.

12.2.2.3 MSXSL-Skripts für XSLT

Das Element `<msxsl:script>` enthält benutzerdefinierte Funktionen und Variablen, die von XPath-Ausdrücken im XSLT-Stylesheet aufgerufen werden können. Das Element `<msxsl:script>` ist ein Element der obersten Ebene, d.h. es muss ein Child-Element von `<xsl:stylesheet>` oder `<xsl:transform>` sein.

Das Element `<msxsl:script>` muss sich im Namespace `urn:schemas-microsoft-com:xslt` (*siehe Beispiel unten*) befinden.

Scripting-Sprache und Namespace

Die im Block verwendete Scripting-Sprache wird im Attribut `language` des Elements `<msxsl:script>` definiert und der für Funktionsaufrufe von XPath-Ausdrücken aus zu verwendende Namespace wird durch das Attribut `implements-prefix` (*siehe unten*) identifiziert.

```
<msxsl:script language="scripting-language implements-prefix="user-namespace-prefix">

   function-1 or variable-1
   ...
   function-n or variable-n

</msxsl:script>
```

Das Element `<msxsl:script>` interagiert mit der Windows Scripting Runtime. Daher können nur Sprachen, die auf Ihrem Rechner installiert sind, im Element `<msxsl:script>` verwendet werden. **Um MXSL Scripts verwenden zu können muss die Plattform .NET Framework 2.0 oder höher installiert sein.** Folglich können die .NET Scripting Sprachen innerhalb des Elements `<msxsl:script>` verwendet werden.

Das Attribut `language` akzeptiert dieselben Werte wie das Attribut `language` des HTML `<script>` Elements. Wenn das Attribut `language` nicht definiert ist, wird als Standardsprache Microsoft JScript verwendet.

Das Attribut `implements-prefix` erhält einen Wert, der ein Präfix eines deklarierten in-scope Namespace ist. Bei diesem Namespace handelt es sich normalerweise um einen Benutzer-Namespace, der für eine Funktionsbibliothek reserviert ist. Alle Funktionen und Variablen, die im Element `<msxsl:script>` definiert sind, werden sich im Namespace befinden, der durch das im Attribut `implements-prefix` definierte Präfixe identifiziert wird. Wenn eine Funktion von einem XPath-Ausdruck aus aufgerufen wird, muss sich der vollständig qualifizierte Funktionsname im selben Namespace wie die Funktionsdefinition befinden.

Beispiel

Hier sehen Sie ein Beispiel für ein vollständiges XSLT Stylesheet, in dem eine Funktion verwendet wird, die in einem `<msxsl:script>` Element definiert ist.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:fn="http://www.w3.org/2005/xpath-functions"
   xmlns:msxsl="urn:schemas-microsoft-com:xslt"
   xmlns:user="http://mycompany.com/mynamespace">

  <msxsl:script language="VBScript" implements-prefix="user">
    <![CDATA[
    ' Input: A currency value: the wholesale price
    ' Returns: The retail price: the input value plus 20% margin,
    ' rounded to the nearest cent
    dim a as integer  = 13
    Function AddMargin(WholesalePrice) as integer
      AddMargin = WholesalePrice * 1.2 + a
    End Function
  ]]>
  </msxsl:script>

  <xsl:template match="/">
    <html>
      <body>
        <p>
          <b>Total Retail Price =
            $<xsl:value-of select="user:AddMargin(50)"/>
          </b>
          <br/>
          <b>Total Wholesale Price =
            $<xsl:value-of select="50"/>
          </b>
        </p>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```


© 2019-2025 Altova GmbH

Datentypen

Die Werte von Parametern, die an und aus dem Script-Block heraus übergeben werden, sind auf XPath-Datentypen beschränkt. Diese Einschränkung gilt nicht für Daten, die zwischen Funktionen und Variablen innerhalb des Script-Blocks übergeben werden.

Assemblies

Eine Assembly kann über das Element **`msxsl:assembly`** in das Script importiert werden. Die Assembly wird über einen Namen oder eine URL identifiziert. Die Assembly wird beim Kompilieren des Stylesheet importiert. Hier sehen Sie ein einfaches Beispiel, wie das Element `msxsl:assembly` zu verwenden ist.

```
<msxsl:script>
   <msxsl:assembly name="myAssembly.assemblyName" />
   <msxsl:assembly href="pathToAssembly" />

   ...

</msxsl:script>
```

Der Assembly-Name kann ein vollständiger Name sein, wie z.B.:

```
"system.Math, Version=3.1.4500.1 Culture=neutral PublicKeyToken=a46b3f648229c514"
```

oder ein Kurzname wie z.B. `"myAssembly.Draw"`.

Namespaces

Namespaces können mit dem Element **`msxsl:using`** deklariert werden. Auf diese Art können Assembly-Klassen ohne ihre Namespaces in das Script geschrieben werden, wodurch Sie sich das mühsame Eintippen ersparen. Hier sehen Sie, wie das Element `msxsl:using` verwendet wird, um Namespaces zu deklarieren.

```
<msxsl:script>
   <msxsl:using namespace="myAssemblyNS.NamespaceName" />

   ...

</msxsl:script>
```

Der Wert des `namespace` Attributs ist der Name des Namespace.

12.3 Datentypen in XML-Schemas, die von DB generiert wurden

Bei Generierung eines XML-Schemas anhand einer Datenbank (DB), werden die Datentypen, die für diese DB spezifisch sind, in XML-Schema-Datentypen konvertiert. In diesem Anhang finden Sie Informationen zum Mapping von DB-Datentypen auf XML-Schema-Datentypen für gebräuchliche Datenbanken. Wählen Sie die gewünschte Datenbank aus der Liste aus.

- ADO 676
- MS Access 676
- MS SQL Server 678
- MySQL 679
- ODBC 679
- Oracle 680
- Sybase 681

12.3.1 ADO

Wenn anhand einer ADO-Datenbank ADO database (DB) ein XML-Schema generiert wird, werden die ADO-DB-Datentypen, wie in der Tabelle unten aufgelistet, in XML-Schema-Datentypen konvertiert.

ADO-Datentyp	XML-Schema-Datentyp
adGUID	xs:ID
adChar	xs:string
adWChar	xs:string
adVarChar	xs:string
adWVarChar	xs:string
adLongVarChar	xs:string
adWLongVarChar	xs:string
adVarWChar	xs:string
adBoolean	xs:boolean
adSingle	xs:float
adDouble	xs:double
adNumeric	xs:decimal
adCurrency	xs:decimal
adDBTimeStamp	xs:dateTime

© 2019-2025 Altova GmbH

adDate	xs:date
adBinary	xs:base64Binary
adVarBinary	xs:base64Binary
adLongVarBinary	xs:base64Binary
adInteger	xs:Integer
adUnsignedInt	xs:unsignedInt
adSmallInt	xs:short
adUnsignedSmallInt	xs:unsignedShort
adBigInt	xs:long
adUnsignedBigInt	xs:unsignedLong
adTinyInt	xs:byte
adUnsignedTinyInt	xs:unsignedByte

12.3.2 MS Access

Wenn anhand einer MS Access-Datenbank (DB) ein XML-Schema generiert wird, werden die MS Access-DB-Datentypen, wie in der Tabelle unten aufgelistet, in XML-Schema-Datentypen konvertiert.

MS Access-Datentyp	XML-Schema-Datentyp
GUID	xs:ID
char	xs:string
varchar	xs:string
memo	xs:string
bit	xs:boolean
Number(single)	xs:float
Number(double)	xs:double
Decimal	xs:decimal
Currency	xs:decimal
Date/Time	xs:dateTime
Number(Long Integer)	xs:integer
Number(Integer)	xs:short
Number(Byte)	xs:byte

OLE Object	xs:base64Binary

12.3.3 MS SQL Server

Wenn anhand einer MS SQL Server-Datenbank (DB) ein XML-Schema generiert wird, werden die MS SQL Server-DB-Datentypen, wie in der Tabelle unten aufgelistet, in XML-Schema-Datentypen konvertiert.

MS SQL Server-Datentyp	XML-Schema-Datentyp
uniqueidentifier	xs:ID
char	xs:string
nchar	xs:string
varchar	xs:string
nvarchar	xs:string
text	xs:string
ntext	xs:string
sysname	xs:string
bit	xs:boolean
real	xs:float
float	xs:double
decimal	xs:decimal
money	xs:decimal
smallmoney	xs:decimal
datetime	xs:dateTime
smalldatetime	xs:dateTime
binary	xs:base64Binary
varbinary	xs:base64Binary
image	xs:base64Binary
integer	xs:integer
smallint	xs:short
bigint	xs:long
tinyint	xs:byte

© 2019-2025 Altova GmbH

12.3.4 MySQL

Wenn anhand einer MySQL-Datenbank (DB) ein XML-Schema generiert wird, werden die MySQL-DB-Datentypen, wie in der Tabelle unten aufgelistet, in XML-Schema-Datentypen konvertiert.

MySQL-Datentyp	XML-Schema-Datentyp
char	xs:string
varchar	xs:string
text	xs:string
tinytext	xs:string
mediumtext	xs:string
longtext	xs:string
tinyint(1)	xs:boolean
float	xs:float
double	xs:double
decimal	xs:decimal
datetime	xs:dateTime
blob	xs:base64Binary
tinyblob	xs:base64Binary
mediumblob	xs:base64Binary
longblob	xs:base64Binary
smallint	xs:short
bigint	xs:long
tinyint	xs:byte

12.3.5 ODBC

Wenn anhand einer ODBC-Datenbank (DB) ein XML-Schema generiert wird, werden die ODBC-DB-Datentypen, wie in der Tabelle unten aufgelistet, in XML-Schema-Datentypen konvertiert.

ODBC-Datentyp	XML-Schema-Datentyp
SQL_GUID	xs:ID
SQL_CHAR	xs:string

SQL_VARCHAR	xs:string
SQL_LONGVARCHAR	xs:string
SQL_BIT	xs:boolean
SQL_REAL	xs:float
SQL_DOUBLE	xs:double
SQL_DECIMAL	xs:decimal
SQL_TIMESTAMP	xs:dateTime
SQL_DATE	xs:date
SQL_BINARY	xs:base64Binary
SQL_VARBINARY	xs:base64Binary
SQL_LONGVARBINARY	xs:base64Binary
SQL_INTEGER	xs:integer
SQL_SMALLINT	xs:short
SQL_BIGINT	xs:long
SQL_TINYINT	xs:byte

12.3.6 Oracle

Bei Generierung eines XML-Schemas anhand einer Oracle-Datenbank (DB) werden die Oracle DB-Datentypen wie unten in der Tabelle gezeigt in XML-Schema-Datentypen konvertiert.

Oracle-Datentyp	XML-Schema-Datentyp
ROWID	xs:ID
CHAR	xs:string
NCHAR	xs:string
VARCHAR2	xs:string
NVARCHAR2	xs:string
CLOB	xs:string
NCLOB	xs:string
NUMBER (mit angewendetem Check Constraint)*	xs:boolean
NUMBER	xs:decimal
FLOAT	xs:double

© 2019-2025 Altova GmbH

DATE	xs: dat eTi me
I NTERVAL YEAR TO MONTH	xs: gYear Mont h
BLOB	xs: base64Bi nar y

* Wenn ein Check Constraint auf eine Spalte vom Datentyp `NUMBER` angewendet wird und der Check Constraint auf die Werte `0` oder `1` überprüft, wird der Datentyp `NUMBER` für diese Spalte in einen XML-Schema-Datentyp `xs:boolean`. konvertiert. Dieser Mechanismus eignet sich zum Generieren eines Datentyps `xs:boolean` im generierten XML-Schema.

12.3.7 Sybase

Bei Generierung eines XML-Schemas anhand einer Sybase-Datenbank (DB) werden die Sybase DB-Datentypen, wie unten in der Tabelle gezeigt, in XML-Schema-Datentypen konvertiert.

Sybase-Datentyp	XML-Schema-Datentyp
char	xs:string
nchar	xs:string
varchar	xs:string
nvarchar	xs:string
text	xs:string
sysname-varchar(30)	xs:string
bit	xs:boolean
real	xs:float
float	xs:float
double	xs:double
decimal	xs:decimal
money	xs:decimal
smallmoney	xs:decimal
datetime	xs:dateTime
smalldatetime	xs:dateTime
timestamp	xs:dateTime
binary<=255	xs:base64Binary
varbinary<=255	xs:base64Binary
image	xs:base64Binary
integer	xs:integer

smallint	xs:short
tinyint	xs:byte

© 2019-2025 Altova GmbH

12.4 Technische Daten

Dieses Kapitel enthält Informationen zu einigen technischen Aspekten Ihrer Software. Es ist in die folgenden Abschnitte gegliedert:

- OS- und Arbeitsspeicheranforderungen 683
- Altova-Prozessoren 683
- Unicode-Unterstützung 684
- Internet-Verwendung 684

12.4.1 OS- und Arbeitsspeicheranforderungen

Betriebssystem

Die Altova-Software-Applikationen stehen für die folgenden Plattformen zur Verfügung:

- Windows 10, Windows 11
- Windows Server 2016 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.

12.4.2 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.

12.4.3 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.

12.4.4 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.*)*

© 2019-2025 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 XMLSpy 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.

12.5 Lizenzinformationen

Dieser Anhang enthält die folgenden Informationen:

- Informationen über den Vertrieb dieses Software-Produkts
- Informationen zur Software-Aktivierung und Lizenzüberwachung
- 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.

12.5.1 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.

© 2019-2025 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.

12.5.2 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.

12.5.3 Altova Endbenutzer-Lizenzvereinbarung

- Die Altova-Endbenutzer-Lizenzvereinbarung kann unter https://www.altova.com/de/legal/eula eingesehen werden.
- Die Altova-Datenschutzbestimmungen finden Sie unter https://www.altova.com/de/privacy.

12.5.4 Verpacken von Lizenzdateien in den StyleVision Installer

Wenn Sie eine stillen Installation von StyleVision durchführen möchten, können Sie gegebenenfalls die MSI-Datenbank ändern, so dass Sie Ihre Lizenzdatei(en) enthält. Dadurch installiert der Installer nicht nur das Produkt, sondern lizenziert es auch. Eine genauere Anleitung dazu finden Sie, wenn Sie diese ZIP-Datei von der Altova-Website herunterladen und das darin enthaltene PDF-Dokument öffnen.

© 2019-2025 Altova GmbH

Index

.

.docx (nur Enterprise Edition), 14, 31
.NET Erweiterungsfunktionen,
Datentypkonvertierungen, .NET in XPath/XQuery, 673
Datentypkonvertierungen, XPath/XQuery in .NET, 672
für XSLT und XQuery, 667
Instanzmethoden, Instanzfelder, 671
Konstruktoren, 669
statische Methoden, statische Felder, 670
.NET-Erweiterungsfunktionen,
in XPath-Ausdrücken, 440, 452
Unterstützung für, in Authentic-Ansicht, 452
Unterstützung in der Authentic-Ansicht, 440

A

Abdocken,
Design-Eingabehilfenfenster, 33
Abkürzungen,
im Benutzerhandbuch, 25
Absatz,
Befehl für das Einfügen im Design, 508
einschließen in, 527
ADO,
Konvertierung von Datentypen in XML-Schema Generierung von DB, 676
Aktive Konfiguration, 546
Aktualisieren eines Node (Enterprise und Professional Edition),
mit dem Ergebnis einer automatischen Berechnung, 257
Alias,
siehe Globale Ressourcen, 362
Allgemeine Arbeitsabläufe, 97
Altova Globale Ressourcen,
siehe Globale Ressourcen, 362
Altova Website, 565
Altova XML Parser,
Info, 683
Altova-Erweiterungen,
Diagrammfunktionen (siehe Diagrammfunktionen), 577
AltovaXML, 424
und FOP, 428
Ändern in (Befehl), 181
Andocken,
Design-Eingabehilfenfenster, 33
Anhänge, 566
Anhängen,
Spalte an Tabelle im SPS anhängen, 533
Zeile in Tabelle im SPS anhängen, 533
Anpassen von StyleVision, 547
Ansichten,
Layout auf der Benutzeroberfläche, 33
Arbeitsspeicher-Anforderungen, 683
Ausblenden,
Funktion der Design-Eingabehilfenfenster, 33
Ausgabedateien,
generieren, 103
Ausgabekodierung, 489
Ausgabevorschauen, 503
Beschreibung, 31
Auswahlliste,
im Schnellstart-Tutorial, 84
Auswahllisten, 161
Authentic Browser, 18
Authentic Desktop, 18
Authentic-Ansicht,
in Altova-Produkten, 18
Automatisch ausblenden,
Funktion der Design-Eingabehilfenfenster, 33
Automatische Berechnung,
Befejhl für das Einfügen im Design, 507
Node mit Wert aus autom. Ber. aktualisieren, 507
Automatische Berechnungen, 257
Beispieldateien, 259
Beispiele, 271
erstellen, bearbeiten, formatieren, 257
Formatieren der Datumsergebnisse, 399
im Schnellstart-Tutorial, 80
Java und .NET-Funktionen (nur in der Enterprise Edition), 257
Symbol in der Design-Ansicht, 435
und Bedingungen, 266
und Output Escaping, 330
verschieben, 257
Verwendung, 257
Automatische Nummerierung, 314
Automatisierte Verarbeitung, 424

© 2019-2025 Altova GmbH

B

Barcodes, 166
Basisjahr,
in der Eingabeformatierung, 333
Batch-Dateien,
und geplante Aufgaben, 433
Bedingte Vorlagen,
Symbol in der Design-Ansicht, 435
Bedingung,
Befehl für das Einfügen in das Design, 518
Bedingungen,
bearbeiten, 265
definieren, 262
einschließen in, 528
im Schnellstart-Tutorial, 84
und automatische Berechnungen, 266
Bedingungsvorlagen, 518
siehe auch Bedingungen, 262
Bedingungszweig "Andernfalls", 262
Befehl "(SPS) ölffnen", 475
Befehl "(SPS) schließen", 475
Befehl "Alles markieren", 493
Befehl "Ausschneiden", 493
Befehl "Beenden", 492
Befehl "Design speichern", 480
Befehl "Drucken", 491
Befehl "Druckvorschau", 491
Befehl "Einfügen", 493
Befehl "Kopieren", 493
Befehl "Löschen", 493
Befehl "Neu", 468
Befehl "Rückgängig", 493
Befehl "Wiederherstellen", 493
Befehle,
anpassen, 547
Befehlszeile, 424
und geplante Aufgaben, 433
und Parameter, 282
Befehlszeilentool, 20
Benutzerdefinierte Elemente, 123, 524, 531
Benutzerdefinierte Schemas, 193
Benutzerdefinierte Textblöcke, 123, 524
Benutzerdefinierte Vorlage,
einfügen, 522
einschließen in, 526
Benutzerdefinierte Vorlagen, 120
Benutzerdefinierte Wörterbücher,
für SPS Rechtschreibprüfung, 542
Benutzerdefinierte XML-Textblöcke, 124
Benutzerdefinierte XPath-Funktionen, 383
Benutzerhandbuch, 552, 561
Benutzeroberfläche,
Dokumentansichten, 28
Hauptfenster, 28
mehrere Dokumente, 28
siehe GUI, 27
Benutzerreferenz, 434
Bestellen der Altova-Software, 561
Betriebssysteme für Altova-Produkte, 683
Bild,
Befehl zum Einfügen in das Design, 509
Bildeinbettung, 489
Bilder,
Aufrufen der Ausgabedarstellung, 152
Definieren von URIs, 152
im Schnellstart-Tutorial, 68
Symbol in der Design-Ansicht, 435
über ungeparste Entities finden, 377
und URIs für ungeparste Entities, 152
unterstützte Typen, 154

C

CALS/HTML-Tabellen, 142, 536
CDATA-Abschnitte, 111
Copyright-Informationen, 686
CoreCatalog.xml, 105
CSS Stylesheets,
externe Stylesheets, 343
Importvorrang von externen, 343
Medien, 343
siehe auch Stile, 343
CSS-Dateien,
in der Seitenleiste Design-Übersicht verwalten, 36
CSS-Stile,
im Schnellstart-Tutorial, 74
in modularen SPSs, 221
siehe auch Stile, 47
CustomCatalog.xml, 105

© 2019-2025 Altova GmbH

D

Darstellung,
siehe auch Formate, Formatierung, 327
Übersicht über Verfahren, 327
Dateien,
zuletzt verwendete öffnen, 492
Datenbank,
Symbolleisten-Schaltflächen für die Bearbeitung, 466
Datenbank (Enterprise und Professional Edition),
siehe DB, 13
Dateneingabeelemente,
Symbol in der Design-Ansicht, 435
Dateneingabe-Steuerelemente, 158
Menübefehle zum Einfügen, 506
Datentypen,
Konvertierung von DB in XML-Schema, 676
siehe auch XML-Schema-Datentypen, 677
Datum,
Formatierung, 333
Datumswerte,
Beispiele für die Bearbeitung von Datumswerten mit1 XPath 2.0, 399
formatieren, 399
verwenden im SPS, 399
DB,
Datentyp-Konvertierung in XML-Schema, 676
DB-Parameter,
erstellen und bearbeiten, 498
Deaktiviert (Befehl), 520, 530
Design Fragment,
einfügen, 523
Design-Ansicht,
Anzeige von Markup, 29
Beschreibung, 29
Symbole im SPS Design, 435
und JavaScript Editor, 29
Design-Eingabehilfen,
ausblenden, 33
automatisch ausblenden, 33
Beschreibung, 33
ein- und ausblenden, 502
Design-Eingabehilfenfenster,
abdocken, 33
andocken, 33
Design-Elemente, 462
Design-Filter,
ein- und ausschalten, 503
Design-Fragmente, 242
Design-Struktur, 184
Eingabehilfenfenster, 41
siehe auch Design-Eingabehilfenfenster, 33
und modulare SPSs, 221
Design-Übersicht,
Seitenleiste, 36
Dezimalzahlen,
Formatierung, 333
Dialogfeld "Anpassen",
zum Anpassen von StyleVision, 501
Dialogfeld "Parameter bearbeiten", 498
disable-output-escaping, 330
Dokumentansicht,
auf der Benutzeroberfläche, 28
Dokumentation,
Überblick, 25
Dokumente,
öffnen und schließen, 28
Dokumenteigenschaften, 255
Dokumentelement,
Definition, 21
Dokumentelemente (siehe Root-Elemente), 186
Dokument-Node,
Definition, 21
Dokumentstile, 255
DPI, 489
DTD,
ungeparste Entities deklarieren, 377
DTDs,
als SPS Quelle, 187
DTDs und Kataloge, 104
Dynamische Listen, 147, 149
Dynamische Tabelle,
Symbolleisten-Schaltflächen zum Bearbeiten, 460
Dynamische Tabellen, 126
geschachtelte dynamische Tabellen, 129
Kopf- und Fußzeilen, 129
siehe auch SPS-Tabellen, 129
siehe auch Tabellen, 136
und globale Vorlagen, 129
Unterschied zwischen angehängten/eingefügten Zeilen, 129
Dynamischer Inhalt,
im Schnellstart-Tutorial, 61

© 2019-2025 Altova GmbH

E

Eigenschaften,
definieren, 49
Seitenleistenfenster, 49
siehe auch Design-Eingabehilfen, 33
und Eigenschaftsgruppen, 49
von SPS Tabelle, 535
von SPS Tabellen, 460
Eigenschaften von Ausgabedokumenten, 255
Einfache globale Vorlagen, 231
Einfügen,
Spalte in SPS Tabelle einfügen, 533
Zeile in Tabelle im SPS einfügen, 533
Einfügen (Menü),
Einfügen | Design Fragment, 523
Einfügen von Design-Elementen über die Symbolleiste, 462
Eingabefelder, 159
Eingabeformatierung, 333
Formatierungsstrings definieren, 538
von Datumswerten, 399
Eingabehilfe "Eigenschaften",
Gruppe "Ereignis", 405
Eingabehilfen in der Design-Ansicht,
ein- und ausblenden, 502
Eingebettete Bilder, 489
Einrichten von StyleVision, 20
Einschließen in (Menü), 525
Elemente,
benutzerdefinierte, 123
Elementvorlagen,
benutzerdefinierrte, 123
Endbenutzer-Lizenzvereinbarung, 686, 688
Entities,
als URI-Platzhalter verwenden, 377
ungeparste, 377
Entity-Referenzen,
und Output Escaping, 330
Ersetzen,
mittels Regular Expressions, 493
Ersetzen (Befehl) (Enterprise und Professional Edition), 493
Erstellen eines neuen SPS Dokuments,
im Schnellstart-Tutorial, 56
Erweiterungsfunktionen für XSLT und XQuery,
Altova-Erweiterungen, 577
Erweiterungsfunktionen in .NET für XSLT und XQuery,
Java-Erweiterungsfunktionen, 667
siehe .NET Erweiterungsfunktionen, 667
Erweiterungsfunktionen in Java für XSLT und XQuery,
Java-Erweiterungsfunktionen, 657
siehe Java-Erweiterungsfunktionen, 657
Erweiterungsfunktionen in MSXSL Scripts, 673
Evaluierungs-Keycode,
für Ihre Altova-Software, 561
Evaluierungszeitraum,
für Altova-Software-Produkte, 686
von Altova Software-Produkten, 686
Event Handler,
Funktionen zuweisen, 405
Excel-Tabelleninhalt,
ins Design einfügen, 115

F

Features,
von StyleVision, 14
Filter,
zum selektiven Anzeigen von Vorlagen, 464
Filter an Node-Vorlagen, 239
Filter für Design-Vorlagen,
ein- und ausschalten, 503
FO-Prozessor (Enterprise Edition),
einrichten, 20
Format,
Befehl für das Einfügen im Design, 508
Formatieren,
von Nodes beim Einfügen, 113
von numerischen Feldern, 333
Formatierung,
für Tabellen, 136
Listen, 459
siehe auch Darstellung, 327
Symbolleisten-Schaltflächen, 459
Textausrichtung, 459
Texteigenschaften, 459
Übersicht über Verfahren, 327
vordefinierte HTML-Formate, 459
Formatierungsstrings,
für Eingabeformatierung definieren, 538

© 2019-2025 Altova GmbH

Formularbasierte Designs, 171, 468
Formular-Steuerelemente,
Menübefehle zum Einfügen, 506
FO-Transformationen, 428
Fragen und Antworten zu StyleVision, 565
Funktioen,
in XPath, benutzerdefiniert, 383
Fußzeilen,
in Tabelle hinzufügen, 533
in Tabellen, 136

G

Ganzzahl,
Formatierung, 333
Generierte Dateien, 103
Generierte Dateien speichern (Befehl), 486
Geplante Aufgaben,
Erstellen eines StyleVisionBatch Befehls, 433
StyleVisionBatch Batch-Dateien, 433
Geteilte Ansichten,
Beschreibung, 31
Global, 684
Globale Ressourcen, 362
definieren, 362
Dialogfeld, 545
Konfiguration über Symbolleiste auswählen, 465
Konfigurationen ändern, 375
Symbolleiste, 465
Typ Datei definieren, 365
Typ Ordner definieren, 370
verwenden, 372, 375
vom Typ Datei und Ordner verwenden, 372
Globale Stile,
Seitenleistenfenster, 45
Globale Typen,
in Vorlagen, 231
Globale Vorlage, 230
Globale Vorlagen, 230, 231
Auswirkung auf restlichen Inhalt, 113
im Schnellstart-Tutorial, 92
Globaler Stil,
siehe Stile, 346
Grafiken,
Beispieldateien, 157
siehe auch Bilder, 152
Übersicht über die Verwendung im SPS, 152
Grafikvorlagen für das Layout, 171
Gruppieren, 267
group-by Beispiel (Persons.sps), 270
group-by Beispiel (Scores.sps), 271
GUI,
Beschreibung, 27

H

Hauptschema, 230
Hauptschema (nur Enterprise Edition), 38
Hauptvorlage, 230
Definition, 21
Hintergrundinformationen, 683
Horizontale Linie,
Befehl für das Einfügen im Design, 511
im Schnellstart-Tutorial, 68
HTM-Import,
neues SPS erstellen, 408
HTML-Ausgabe, 103
und Unterstützung von Bildern, 154
HTML-Dokumenteigenschaften, 359
HTML-Import,
Dateien anhand des SPS generieren, 414
Schemastruktur, 410
SPS Design, 410
von HTML-Listen, 412
von HTML-Tabellen, 412
HTML-Seiteninhalt,
ins Design einfügen, 115
HTML-Tabellen, 142, 536
Hyperlink,
Befehl für das Einfügen im Design, 517
Hyperlinks, 165, 320
einschließen in, 528
entfernen und löschen, 322
erstellen und bearbeiten, 322
mit externen Ressourcen verknüpfen, 322
mit Lesezeichen verknüpfen, 322
Symbol in der Design-Ansicht, 435
über Hyperlinks finden, 377
und ungeparste Entities, 322

© 2019-2025 Altova GmbH

I

IE 9,
siehe Kompatibilität mit Internet Explorer, 101
Import von XSLT-Vorlagen,
ins SPS, 246
In Design ersetzen (Ansicht), 53
In Design suchen-Ansicht, 53
In Katalogen verwendete Umgebungsvariablen, 105
Inhalt,
Befehl für das Einfügen im Design, 505
Inhalt auskommentieren, 520, 530
Inhaltsverz.-Ebenen,
einschließen in, 530
Inhaltsverz.-Markierungen,
einschließen in, 530
Inhaltsverzeichnis,
Beispiel, einfach, 307
Beispiel, hierarchische und sequenzielle, 310
Einträge für das Inhaltsverz. markieren, 293
Menübefehle, 523
Übersicht über die Verwendung, 290
Inhaltsverzeichnisebenen, 293, 295
Inhaltsverzeichniseinträge,
erstellen, 305
formatieren, 305
Inhaltsverzeichnismarkierungen, 293
Assistent für, 298
erstellen, 298
und Ebenen, 298
Inhaltsverzeichnis-Vorlage,
Ebenenreferenzen darin, 303
erstellen und bearbeiten, 301
formatieren, 305
strukturieren, 303
Inhaltsverzref,
siehe Inhaltsverzeichnisreferenzen, 304
Inhatlsverzeichnisreferenzen, 304
Internet Explorer-Kompatibilität, 101
Internet-Verwendung,
in Altova-Produkten, 684
ISO/IEC 10646, 684

J

Java Erweiterungsfunktionen,
Datentypkonvertierungen, Java in Xpath/XQuery, 666
Instanzmethoden, Instanzfelder, 664
Konstruktoren, 663
statische Methoden, statische Felder, 664
Java und .NET-Funktionen (nur in der Enterprise Edition),
in automatischen Berechnungen, 257
Java-Erweiterungsfunktionen,
benutzerdefinierte JAR-Dateien, 662
benutzerdefinierte Klassendateien, 659
Datentyp-Konvertierungen, XPath/XQuery in Java, 665
für XSLT und XQuery, 657
in XPath-Ausdrücken, 440, 452
Unterstützung für, in Authentic-Ansicht, 452
Unterstützung in der Authentic-Ansicht, 440
JavaScript,
siehe Scripts, 403
JavaScript Editor, 403, 404
in der Design-Ansicht, 29

K

Kataloganpassung, 107
Katalogdateien, 104
Kataloge in RaptorXML, 105
Kataloge und Umgebungsvariablen, 108
Keycodes,
für Ihre Altova-Software, 561
Klassenattribute,
im Schnellstart-Tutorial, 74
Kodierung,
für Ausgabedateien, 552
Kodierung (Befehl), 489
Kodierung von Ausgabedokumenten, 489
Kombistile, 356
Kompatibilität mit Internet Explorer, 101
Komplexe globale Vorlagen, 231
Komponenten-Download-Center,
auf der Altova Website, 565
Konfigurationen,
einer globalen Ressource, 362, 546

© 2019-2025 Altova GmbH

Konfigurationen in globalen Ressourcen, 375
Kontext-Node,
im Dialogfeld "XPath", 440
im XPath-Dialogfeld, 452
Kontrollkästchen, 160
Konvertierung von HTML in XML, 408
Kopfzeilen,
in Tabelle hinzufügen, 533
in Tabellen, 136

L

Layout,
von Ansichten der Benutzeroberfläche, 33
Layout Container, 171, 523
Layout Container und Elemente, 462
Layout-Feld, 523
Layout-Felder, 174
Layout-Module,
Schritte bei der Erstellung, 171
Lesezeichen, 165, 320
Befehl für das Einfügen im Design, 515
einschließen in, 528
erstellen und bearbeiten, 320
löschen, 320
Lesezeichen (Anchors),
Symbol in der Design-Ansicht, 435
Linie,
in Layout Containern, 523
Links,
siehe Hyperlinks, 165, 320
Listen, 147
Befehl zum Einfügen in das Design, 513
einschließen in, 527
im Schnellstart-Tutorial, 84
vom HTML-Dokument importierte, 412
Listeneigenschaften, 538
Lizenz, 688
für Ihre Altova-Software, 561
Informationen, 686
Lizenzierung,
Lizenzdateien in den Installer verpacken, 688
Lizenzüberwachung,
in Altova-Produkten, 687
Lokale Stile,
siehe Stile, 348
Lokale Vorlage, 230
Löschen,
Spalte aus Tabelle im SPS, 534
Tabelle im SPS, 532
Zeile aus Tabellen im SPS, 534

M

Markup anzeigen, 29
Markup ausblenden, 29
Markup-Tags in der Design-Ansicht, 29
Mehrere Ausgabedokumente, 248
Dateipfad, 253
und Ausgabevorschauen, 253
verknüpfen, 250
Mehrfach-Dokumentausgabe, 524
Mehrzeilige Eingabefelder, 159
Meldungen (Seitenleiste), 53
Menü,
anpassen, 547
Menü "Ansicht", 501
Menü "Bearbeiten", 493
Menü "Datei", 468
Befehl "Beenden", 492
Datei | Design speichern, 480
Datei | Drucken, 491
Datei | Druckvorschau, 491
Datei | Generierte Dateien speichern, 486
Datei | Kodierung, 489
Datei | Neu, 468
Datei | Öffnen, 475
Datei | Schließen, 475
Datei | Speichern unter, 485
Menü "Eigenschaften", 538
Nummerierung und Aufzählungszeichen, 538
Menü "Einfügen", 505
Einfügen | Absatz, 508
Einfügen | Automatische Berechnung, 507
Einfügen | Bedingung, 518
Einfügen | Bild, 509
Einfügen | Deaktiviert, 520
Einfügen | Format, 508
Einfügen | Hyperlink, 517
Einfügen | Inhalt, 505
Einfügen | Lesezeichen, 515
Einfügen | Nummerierung und Aufzählungszeichen, 513

© 2019-2025 Altova GmbH

Menü "Einfügen", 505
Einfügen | Restlicher Inhalt, 506
Menü "Extras", 541
Menü "Fenster", 560
Menü "Hilfe", 561
Menü "Tabelle", 532
Menü 'Einfügen",
Einfügen | Horizontale Linie, 511
Menüleiste,
verschieben, 27
Metadaten des HTML-Ausgabedokuments, 359
Microsoft Office 2007 (nur Enterprise Edition), 14, 31
MobileTogether Design,
exportieren in, 486
Modulares SPS,
aktivieren und deaktivieren, 221
arbeiten mit, 221
Auswirkung der Reihenfolge, 221
Beispielprojekt, 225
das hinzuzufügende SPS Modul, 221
das SPS Modul hinzufügen, 221
erstellen, 221
Übersicht, 216
und CSS-Stile, 217, 221
und Modulobjekte, 217
und Namespace-Deklarationen, 217
und Schemaquellen, 217
und Scripts, 217
und Shemaquellen, 221
und Warnmeldungen über geänderte Dateien, 221
und XML-Arbeitsdateien, 217
und XML-Vorlagendateien, 217
Module,
in der Seitenleiste Design-Übersicht verwalten, 36
MS Access,
Konvertierung von Datentypen in XML-Schema Generierung von DB, 677
MS SQL Server,
Konvertierung von Datentypen in XML-Schema Generierung von DB, 678
MS Word-Dokumentinhalt,
ins Design einfügen, 115
msxsl:Script, 673
MySQL,
Konvertierung von Datentypen in XML-Schema Generierung von DB, 679

N

Nachfolgende Markup-Tags, 29
Named Templates, 230
Namespaces,
im SPS, 38
Übersicht, 41
zum SPS hinzufügen, 38, 98, 103, 186
Netzwerk-Proxy, 552
Neu von XSLT-Datei, 379
Neue Dokumentvorlagen, 248
Node,
ändern, was erstellt ist als, 181
Nodes automatisch nummerieren, 314
Node-Vorlagen,
benutzerdefinierte, 120
Operationen an, 239
und an Child-Vorlagen binden, 239
und globale Vorlagen, 239
und XPath-Filter, 239
Numerische Felder,
Formatierung, 333
Nummerierung formatieren,
bei der automatischen Nummerierung, 314
Nummerierung und Aufzählungszeichen, 147, 149, 538
Befehl zum Einfügen in das Design, 513
einschließen in, 527

O

ODBC,
Konvertierung von Datentypen in XML-Schema Generierung von DB, 679
Office Open XML (nur Enterprise Edition), 14, 31
Öffnen,
zuletzt verwendete Dateien, 492
Online-Hilfe, 552, 561
OOXML (nur Enterprise Edition), 14, 31
Optionsfelder, 164
Oracle,
Konvertierung von Datentypen in XML-Schema Generierung von DB, 680
Output Escaping, 330

© 2019-2025 Altova GmbH

Querverweise, 318

P

Parameter, 282
allgemeine Beschreibung, 282
erstellen und bearbeiten, 498
für Design-Fragmente, 284
für Schemaquellen, 286
im SPS, 282
in der Seitenleiste Design-Übersicht verwalten, 36
Nodes in mehreren Dokumenten mit Param. identifizieren, 286
Übersicht über benutzerdefinierte Parameter, 41
und Authentic-Ansicht, 282
und Befehlszeile, 282
Parser,
in Altova-Produkte integrierter, 683
Partner-Software, 565
PDF-Ausgabe (Enterprise Edition), 103
und Unterstützung von Bildern, 154
PDF-Hilfe, 552, 561
Pixel,
und Bildschirmauflösung, 489
und Länge in der Druckausgabe, 489
Plattformen für Altova-Produkte, 683
Platzhalter "Contents",
im Schnellstart-Tutorial, 61
Platzhalter "Inhalt",
Node als Inhalt einfügen, 111
Probleme mit der Vorschau, 20
Produktfeatures,
Auflistung, 14
Projektoptionen, 552
Proxy-Einstellungen, 552
Prozessoren,
zum Downloaden, 565
PXF-Dateien, 418
bearbeiten, 421
bereitstellen, 422
erstellen, 418
speichern unter, 485

Q

Quelldateien für das SPS, 98

R

RaptorXML,
und FOP, 428
Rechtliches, 686
Rechtschreibprüfung,
in StyleVision, 541
Rechtschreibprüfungsoptionen,
für SPSs, 542
Registrierung der Altova-Software, 561
Regular Expressions,
suchen und ersetzen, 493
Restlicher Inhalt, 113
Befehl zum Einfügen in ein Design, 506
im Schnellstart-Tutorial, 92
und globale Vorlagen, 231
RootCatalog.xml, 105
Root-Elemente, 38
Root-Elemente (auch Dokumentelemente genannt),
Auswahl für das Schema, 186
und Schemaquellen, 186
RTF-Ausgabe (Enterprise Edition), 103
und Unterstützung von Bildern, 154

S

Schaltflächen, 164
Schema hinzufügen, 468
Schema-Manager,
CLI-Befehl Help, 206
CLI-Befehl Info, 206
CLI-Befehl Initialize, 207
CLI-Befehl Install, 208
CLI-Befehl List, 208
CLI-Befehl Reset, 209
CLI-Befehl Uninstall, 210
CLI-Befehl Update, 211
CLI-Befehl Upgrade, 211
CLI-Übersicht, 205
Patch für Schema installieren, 203
Schema deinstallieren, 204
Schema installieren, 203

© 2019-2025 Altova GmbH

Schema-Manager,
Schemas nach Status auflisten in, 202
starten, 200
Status von Schemas, 202
Übersicht, 197
Upgrade für Schema installieren, 203
zurücksetzen, 204
Schemaquellen, 98, 468
Auswahl für das SPS, 186
in der Seitenleiste Design-Übersicht verwalten, 36
mehrere im SPS (Enterprise Edition), 186
mehrere Quellen und Nodes identifizieren, 286
mehrere Quellen und XPath, 286
Quellen ändern, 286
Seitenleistenfenster, 38
Übersicht, 41
und Root-Elemente (Dokumentelemente), 186
Schemaquellen-Fenster,
siehe auch Design-Eingabehilfenfenster, 33
Schemas,
als SPS Quelle, 187
benutzerdefinierte, 193
über Kataloge finden, 107
Schemas und Kataloge, 104
Schemastrukturoptionen, 552
Schnellstart-Tutorial,
Auswahllisten, 84
Authentic-Ansicht überprüfen (Enterprise und Professional Edition), 96
automatische Berechnungen, 80
Bedingungen, 84
Bilder, 68
CSS-Stile, 74
dynamischer Inhalt, 61
Einführung, 55
Einrichten eines neuen SPS Dokuments, 56
erforderliche Dateien, 55
Erstellen eines neuen SPS Dokuments, 56
globale Vorlagen, 92
horizontale Linien, 68
Klassenattribute, 74
Listen, 84
Platzhalter "Inhalt", 61
restlicher Inhalt, 92
statischer Inhalt, 68
statischer Text, 68
vordefinierte Formate, 74
XSLT Stylesheets generieren, 96
Schnittstelle,
siehe GUI, 27
Scripts,
in der Design-Struktur, 403
in einem SPS verwenden, 403
JavaScript-Funktionen als Event Handler, 405
JavaScript-Funktionen definieren, 404
und JavaScript-Funktionen, 403
Scroll-Schaltflächen,
im Hauptfenster, 28
Seitenleiste "Suchen & Ersetzen", 53
Skripts,
Übersicht, 41
Software-Produktlizenz, 688
Sortieren, 276
Beispieldateien, 278
Funktionsweise, 276
Sortierschlüssel, 276
von Gruppen und innerhalb von Gruppen, 267, 270, 271
Sortierschlüssel, 276
Spalte,
an SPS Tabelle anhängen, 533
aus Tabelle im SPS löschen, 534
in SPS Tabelle einfügen, 533
Spalten (einer Tabelle),
in der HTML-Ausgabe ausblenden, 140
Spezieller Absatz,
einschließen in, 527
SPS,
allgemeine Beschreibung, 19
neu laden, 475
öffnen, 475
schließen, 475
und die Authentic-Ansicht (Enterprise und Professional Edition), 19
und StyleVision, 19
und XSLT Stylesheets, 19
SPS Design-Übersicht, 99
SPS-Dateistruktur, 184
SPS-Tabellen,
siehe auch dynamische Tabellen, 126
siehe auch statische Tabellen, 126
Standard-Benutzerwörterbuch,
für SPS Rechtschreibprüfung, 542
Statische Listen, 147, 527
Statische Tabelle,
einfügen, 532
Statische Tabellen, 126

© 2019-2025 Altova GmbH

Statische Tabellen, 126
in das SPS einfügen, 460
siehe auch SPS-Tabellen, 128
siehe auch Tabellen, 136
Symbolleisten-Schaltflächen zum Bearbeiten, 460
Statischer Inhalt,
im Schnellstart-Tutorial, 68
Statischer Text,
im Schnellstart-Tutorial, 68
und Output Escaping, 330
Statusleiste, 501
Stile,
Arbeiten mit Stilen in StyleVision, 342
aus XML-Daten, 353
definieren, 47
globale Stile im SPS definieren, 346
Kombinieren von CSS-Regeln, 356
lokale Stile definieren, 348
Medien für zugewiesene externe Stylesheets, 343
mehrere kombinieren, 356
mittels XPath-Ausdrücken, 353
Seitenleistenfenster, 47
siehe auch Design-Eingabehilfen, 33
stufenförmige Reihenfolge, 342
Terminologie, 342
und Eigenschaftsgruppen, 47
Vorrang, 45
Vorrangigkeit von Stilen, 346
Zuweisen von CSS Stylesheets zu SPS, 343
Stile von Ausgabedokumenten, 255
Stille Installation,
Lizenzdateien in den Installer verpacken, 688
MSI-Datei ändern, 688
Stil-Repository,
siehe auch Design-Eingabehilfen, 33
und externe CSS Stylesheets, 343
und globale Stile, 346
Struktur des SPS-Designs, 184
Stylesheets,
siehe auch unter CSS Stylesheets, 343
siehe auch unter XSLT Stylesheets, 343
StyleVision,
Benutzerhandbuch, 13
Produktfeatures, 14
StyleVision Power Stylesheet,
siehe SPS, 13
StyleVisionBatch, 20, 424
Suchen,
mittels Regular Expressions, 493
Suchen (Befehl), 493
Support für StyleVision, 565
Support-Optionen, 25
Sybase,
Konvertierung von Datentypen in XML-Schema
Generierung von DB, 681
Symbole in der Design-Ansicht,
von automatischen Berechnungen, 435
von Bedingungsvorlagen, 435
von Bildern, 435
von Dateneingabeelementen, 435
von Hyperlinks, 435
von Lesezeichen (Anchors), 435
von vordefinierten Formaten, 435
von XML-Dokumentinhalt, 435
von XML-Dokument-Nodes, 435
Symbolleisten, 457
"Design-Elemente einfügen" Symbolleiste, 462
anpasssen, 501
Anzeige ein-/ausschalten, 457
auf der Benutzeroberfläche positionieren, 457
ein- und ausblenden, 501
Standard-Symbolleiste, 466
Symbole hinzufügen/entfernen, 457
Symbolleiste Formatierung, 459
Tabellen-Symbolleiste, 460
verschieben, 27
zurücksetzen, 457
Symbolleisten und Fenster wiederherstellen, 552

T

Tabelle,
Eigenschaften bearbeiten, 535
im SPS löschen, 532
Kopf- und Fußzeilen, 532
Kopf- und Fußzeilen hinzufügen, 533
navigieren, 532
Spalte anhängen, 533
Spalte einfügen, 533
Spalte löschen, 534
statische Tabelle einfügen, 532
Umrandung in StyleVision ein oder ausblenden, 535
vertikale Ausrichtung von Zelleninhalt, 537
Zeile anhängen, 533

© 2019-2025 Altova GmbH

Tabelle,
Zeile einfügen, 533
Zeile löschen, 534
Zelleninhalt, 532
Tabellen,
bedingte Verarbeitung, 134
dynamische Tabellen erstellen, 129
erstellen, 512
formatieren, 136
Kopf- und Fußzeilen im PDF, 136
leere Spalten ausblenden, 140
Schließen-Schaltfläche zum Ausblenden von Spalten, 140
statische Tabellen erstellen, 128
Stile für aufeinanderfolgende Zeilen, 353
Übersicht, 126
vom HTML-Dokument importierte, 412
Zeilen erweitern/reduzieren, 140
Zellen verbinden, 534
Tabellen (SPS),
Eigenschaften bearbeiten, 460
Symbolleisten-Schaltflächen zum Bearbeiten, 460
Tabellen in der Design-Ansicht,
Darstellung, 135
Einschließen in Vorlagen und Entfernen der Vorlagen, 135
Tabellenzelle teilen,
horizontal, 535
vertikal, 535
Tabellenzelleninhalt ausrichten,
in SPSs, 537
Tags,
erweitern und reduzieren, 499
Tastaturkürzel,
für Befehle anpassen, 547
Technische Informationen, 683
Technischer Support für StyleVision, 565
Temporäres Ausgabedokument, 20
Terminologie,
in StyleVision, 21
Textreferenzen, 318
Toolbar buttons,
adding and removing, 458
Typbasierte Vorlagen, 231
Typen als Verarbeitungseinheiten,
in globalen Vorlagen, 231

U

Über StyleVision, 565
Übersicht über den Katalogmechanismus, 104
UCS-2, 684
UCS-4, 684
Umgebungsvariablen, 108
Umrandung,
von SPS Tabellen, 535
Ungeparste-Entity-URI-Funktion von XSLT, 377
Unicode,
Konsortium, 684
Standard, 684
URIs,
in ungeparsten Entities enthaltene, 377
UTF-8, 684

V

Validator,
in Altova-Produkten, 683
Variablen, 282, 287
Variablenvorlage, 238
einfügen, 522
einschließen in, 526
Verfahren zur Bearbeitung von Inhalt, 110
Vertikale Ausrichtung von Tabellenzelleninhalt,
in SPSs, 537
Vertikaler Text,
in Layout-Feldern, 174
in Tabellenzellen, 136
Vertrieb,
von Altova Software-Produkten, 686
von Altova-Software-Produkten, 686
Verwendung, 97
Vordefinierte Formate,
Befehl für das Einfügen im Design, 508
beim Einfügen eines Node, 113
Symbol in der Design-Ansicht, 435
Vordefinierte Formate zuweisen,
im Schnellstart-Tutorial, 74
Vordefinierte Formatierungsstrings,
für Eingabeformatierung, 538

© 2019-2025 Altova GmbH

Vorlage,
einfügen, 520
einschließen in, 525
Node-Übereinstimmung ändern für, 181
Vorlagen,
Ansicht ein- und ausschalten, 503
Baumstruktur, 41
Tabellenzeilen und Spalten einschließen in Vorlagen, 135
Vorlagen rund um Tabellenzeilen und -spalten entfernen, 135
Vorlagen für neue Dokumente,
einfügen, 249
und Design-Struktur, 250
URLs, 250
Vorlagen für Nodes,
siehe Node-Vorlagen, 239
Vorlagenauswahl ändern (Befehl), 120
Vorlagenfilter, 464
Vorrang,
von Stilen, 45

W

Warnmeldungen über geänderte Dateien,
in modularen SPSs, 221
Weitersuchen (Befehl), 493
Windows,
Unterstützung für Altova-Produkte, 683
Word 2007 (nur Enterprise Edition), 14, 31
Word-Dokumentinhalt,
ins Design einfügen, 115
WordML (nur Enterprise Edition), 14, 31
Wörterbücher,
für SPS Rechtschreibprüfung, 542

X

XML, 684
ins Design einfügen, 124
XML Parser,
Info, 683
XML-Arbeitsdatei, 38, 98
Definition, 21
drucken, 491
Druckvorschau, 491
und Ausgabeansichten, 31
XML-Datei Global Resources, 362
XML-Daten,
aus mehreren Quellen zusammenführen, 213
in SPS Design einfügen, 111
XML-Dokument-Inhalt,
Symbol in der Design-Ansicht, 435
XML-Dokument-Nodes,
Symbol in der Design-Ansicht, 435
XML-Schema-Datentypen,
bei der Generierung von XML-Schema anhand einer ADO DB, 676
bei der Generierung von XML-Schema anhand einer MS Access DB, 677
bei der Generierung von XML-Schema anhand einer MS SQL Server DB, 678
bei der Generierung von XML-Schema anhand einer MySQL DB, 679
bei der Generierung von XML-Schema anhand einer ODBC DB, 679
bei der Generierung von XML-Schema anhand einer Oracle DB, 680
bei der Generierung von XML-Schema anhand einer Sybase DB, 681
XML-Schemas und DTDs,
als SPS Quelle, 187
XMLSpy, 18
XML-Tabellen (Enterprise und Professional Edition), 126
XML-Vorlagendatei (Enterprise und Professional Edition), 98
Definition, 21
XPath,
Nodes in mehreren Dokumenten identifizieren, 286
XPath (Dialogfeld),
Beschreibung, 440
XPath 1.0,
und Datumswerte, 399
XPath 2.0,
und Datumswerte, 399
XPath-Ausdruck bearbeiten (Dialogfeld),
siehe Dialogfeld "XPath", 440
XPath-Ausdrücke,
im Dialogfeld "XPath-Ausdruck bearbeiten" erstellen, 440, 452
und Stile, 353
XPath-Dialogfeld,
Ausdrücke debuggen im, 445
Ausdrücke testen im, 441
Beschreibung, 452
XPath-Filter,

© 2019-2025 Altova GmbH

XPath-Filter,
an globalen Vorlagen, 231
XPath-Filter an Node-Vorlagen, 239
XPath-Funktionen,
benutzerdefiniert, 383
im Dialogfeld "XPath", 440
im XPath-Dialogfeld, 452
XPath-Operatoren,
im Dialogfeld "XPath", 440
im XPath-Dialogfeld, 452
XPath-Version im SPS, 100
XSLT,
Codefragmente ins Design einfügen, 124
XSLT in SPS, 379
XSLT Stylesheet-Vorschau,
in der Ausgabeansicht, 31
XSLT-Elemente,
als Code ins Design einfügen, 123
XSLT-Import, 379
XSLT-Transformationen, 428
XSLT-Version,
für das SPS konfigurieren, 466
XSLT-Version im SPS, 100
XSLT-Vorlagen, 41
in der Seitenleiste Design-Übersicht verwalten, 36
ins SPS importieren, 246

Z

Zahlen,
Formatierung, 333
Zeichenreferenzen,
und Output Escaping, 330
Zeile,
an SPS Tabelle anhängen, 533
aus Tabelle im SPS löschen, 534
in SPS Tabelle einfügen, 533
Zeilen (einer Tabelle),
in der HTML-Ausgabe erweitern/reduzieren, 140
Zelle (einer Tabelle),
horizontal teilen, 535
vertikal teilen, 535
Zellen,
in SPS Tabellen verbinden, 534
Zellen verbinden,
in SPS Tabellen, 534
Zuletzt verwendete Dateien, 492
Zusätzliche Bearbeitungsverfahren, 361

© 2019-2025 Altova GmbH